Gridgen®

User Manual

Version 15

POINTWISE®

Get the Point.

Pointwise, Inc. reserves the right to make changes in specifications and other information contained in this document without prior notice.

ALTHOUGH DUE CARE HAS BEEN TAKEN TO PRESENT ACCURATE INFORMATION, POINTWISE, INC. DISCLAIMS ALL WARRANTIES WITH RESPECT TO THE CONTENTS OF THIS DOCUMENT (INLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE), EITHER EXPRESSED OR IMPLIED. POINTWISE, INC. SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF, OR IN CONNECTION WITH, THE USE OF THIS DOCUMENT.

Pointwise, Inc.

213 S. Jennings Ave.

Forth Worth, Texas 76104-1107, USA

phone: (817) 377-2807

toll free: (888) GRIDGEN

fax: (817) 377-2799

email: gridgen@pointwise.com

URL: http://www.pointwise.com

RIGHT TO USE LICENSE

1. DEFINITIONS.

(a) "Licensed Software" means any computer program(s) supplied by Pointwise, Inc. (PWI) to Licensee under a valid Purchase Order or Contract, whether in object code, reconfigurable binary, or any other form; video media and training aids; and any backup or other copies, updates, derivative works, modifications, enhancements, and extensions thereof.

(b) "Documentation" means user manuals, documentation binders, release notes, installation notes, written utility programs and other written or graphic materials related to the Licensed Software and all copies thereof.

(c) "Licensed Product(s)" means the Licensed Software and Documentation.

(d) "Maintenance Period" means the first calendar year of a perpetual license or 12 months for an annual license.

2. LICENSE.

PWI grants to Licensee a non-exclusive, non-transferable license to use the Licensed Products in accordance with the terms and conditions set forth herein. As long as this License is in effect, Licensee may transfer its use of the Licensed Products to a replacement computer system on a temporary or permanent basis, provided that Licensee gives PWI written notice of such transfer.

3. TERM and FEES.

The Term of the Right to Use License is annual or perpetual, as set forth in the Purchase Order or Contract, provided that the License has not been terminated as set forth in Section 10.

All applicable fees are payable to PWI within thirty (30) days from Invoice Date. Failure to pay such fees in a timely manner is a material breach of this License. Licensee agrees to pay any legal or collection fees incurred by PWI in collecting any amounts that may be delinquent.

4. COPIES and MODIFICATIONS.

Licensee may copy the Licensed Products in whole or in part, in written or machine readable form for use in understanding the Licensed Software and for archive or backup purposes. Licensee shall reproduce and include PWI's trade secret or copyright notices on and in any copies, in any form. Licensee shall not reverse assemble, reverse compile or otherwise reverse engineer the Licensed Software in whole or in part.

The license includes the right to reproduce the Gridgen Version 14 User Manual exclusively for the use of Licensee and its employees only. All other reproduction and distribution is strictly prohibited.

5. OWNERSHIP.

All Licensed Products and the information they contain, any updates and all copies of them are PWI's or PWI's licensors' property, and title to them remains in PWI or such licensors. All applicable rights in patents, copyrights or trade secrets in the Licensed Products are and will remain in PWI and PWI's licensors. No title to or ownership of the Licensed Software or the information they contain is transferred to Licensee. Licensee agrees that any terms and conditions imposed by PWI's licensor and communicated by PWI to Licensee by PWI, shall apply to the Licensed Products.

6. CONFIDENTIALITY.

(a) Licensee will take all reasonable precautions to maintain the confidentiality of the Licensed Products, and agrees to take all reasonable and necessary steps to protect the patents, trademarks, copyrights, trade secrets and any other forms of intellectual or industrial property of PWI in the Licensed Products.

(b) Licensee will not provide the Licensed Products to any person, other than employees of Licensee, without PWI's prior written consent, except during the period any such person is on Licensee's premises with Licensee's permission for purposes specifically related to Licensee's use of the Licensed Products.

7. COPYRIGHT AND PATENT INDEMNITY.

PWI assures Licensee that, to the best of PWI's knowledge, the Licensed Products do not infringe any patent, copyright, or trade secret. In the event any legal proceedings are brought against Licensee claiming an infringement of a patent, copyright, or trade secret based on Licensee's use of the Licensed Products, PWI agrees to defend at PWI's own expense any such legal proceeding relating to such claim or claims and to hold Licensee harmless from any damage incurred or awarded as the result of settlement or judgment against Licensee, provided Licensee gives PWI prompt written notice within fifteen (15) days of any such claim or the institution of any such claims against Licensee, and further, Licensee cooperates completely with PWI in providing all necessary authority, information, and reasonable assistance to enable PWI, at PWI's option, to settle or defend such claims.

8. LIMITATION OF WARRANTY AND LIABILITY.

PWI warrants that the Licensed Products will perform substantially in accordance with all written specifications furnished to Licensee by PWI if properly used. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, PWI DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH REGARD TO THE LICENSED PRODUCTS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL PWI BE LIABLE TO LICENSEE OR ANY PARTY FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE LICENSED PRODUCTS EVEN IF PWI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF THE FAULT OR NEGLIGENCE OF PWI. PWI's liability to Licensee for damages shall not exceed the amount of the license fee paid by Licensee to PWI.

9. MAINTENANCE AND SUPPORT.

(a) PWI will be responsible for providing corrections for programming errors and periodic software updates only during the Maintenance Period and during any period when the Licensee is covered by a Maintenance Agreement from PWI or an authorized representative of PWI. The specific policy for software updates and enhancement varies on a per product basis. PWI warrants that during the term of this License it will use reasonable efforts to remedy defects in the unaltered Licensed Software made known to it by Licensee. PWI does not warrant that the Licensed Products will meet all requirements of Licensee, or that the operation of the Licensed Software will be uninterrupted or error free, or that all software defects will be corrected.

(b) During the Maintenance Period, PWI agrees to telephone hot-line support available to Licensee. These services may be provided by PWI or a designated third party.

10. TERMINATION.

Licensee may terminate this License upon thirty (30) days written notice to PWI. PWI may terminate this License for any material breach by Licensee of the terms and conditions contained herein upon thirty (30) days written notice to Licensee. Licensee will have thirty (30) days from the date of such notice to cure the breach, and if the breach is cured to the satisfaction of PWI, this License will remain in full force and effect.

Upon termination, Licensee shall promptly:
(i) discontinue use of the Licensed Products;

(ii) remove the Licensed Products from any software in Licensee’s possession or control that incorporates or uses the Licensed Products in whole or in part; (iii) erase or destroy any of the Licensed Products contained in the computer memory or data storage apparatus under the control of Licensee. Licensee’s obligations under Section 6 shall survive any termination of this License.

11. GENERAL TERMS.

The entire agreement between the parties is contained herein and in a valid Purchase Order or Contract, which supersede all proposals, oral or written, and all other communications between the parties relating to this Agreement and it may be executed in any number of counterparts, each of which shall constitute an original, and all of which taken together shall constitute one and the same Agreement. This Agreement is not assignable by Licensee without prior written permission from PWI. The section headings and subheadings herein are for convenience only and shall not affect the interpretation or construction of this Agreement. PWI shall not be liable for any failure or delay in performance due in whole or in part to any cause beyond PWI’s control.

This Agreement and all transactions under it shall be governed by the laws of the State of Texas. All claims arising under or related to this Right to Use License, Purchase Order or Contract shall be settled finally and exclusively by arbitration in accordance with the Commercial Arbitration Rules of the American Arbitration Association (AAA). Arbitration shall take place in Dallas, Texas and be administered by the AAA’s Dallas, Texas office.

Volume 1
Gridgen User Manual
Version 15

TABLE OF CONTENTS

1. Introduction 1-1
1.1 Grid Types 1-1
1.2 Data Hierarchy 1-1
1.3 Command Hierarchy and Menu Structure 1-2
1.4 Planning Your Grid 1-3
1.4.1 Define the Modeling Goals 1-4
1.4.2 Identify Your Grid’s Constraints 1-4
1.4.3 Plan the Grid 1-4
1.5 Outline of the Gridgen Process 1-4
1.6 Gridgen’s History 1-5
1.7 Document Conventions 1-5

2. Graphical User Interface 2-1
2.1 The Gridgen Screen 2-1
2.2 The Mouse 2-2
2.3 Menu Buttons 2-3
2.3.1 Commands 2-3
2.3.2 Hot Keys 2-3
2.3.3 Toggle Buttons 2-4
2.3.4 Radio Buttons 2-4
2.3.5 VCR Buttons 2-4
2.3.6 Quick Keys 2-5
2.3.7 Quick Menu 2-6
2.3.8 Miscellaneous 2-8
2.4 Entity Selection 2-8
2.4.1 Selecting From The Display Window 2-8
2.4.2 Selecting From The Browser 2-9
2.4.3 Pick Masks 2-10
2.4.4 Special Picking Controls 2-10
2.5 Selecting File Names 2-12
2.6 Text Input 2-13
2.7 Image Manipulation 2-14

2.7.1 Axis Systems *2-14*
2.7.2 Glide Toggle *2-14*
2.7.3 Manipulation Device *2-14*
2.7.4 Manipulation Hot Keys *2-15*
2.7.5 Zoom Box *2-15*
2.8 The 3D Cursor *2-16*
2.8.1 3D Cursor Movement *2-16*
2.8.2 Control Point Creation *2-17*
2.8.3 Miscellaneous Controls *2-19*
2.9 Display Commands Menu *2-19*
2.10 Display Editing *2-22*
2.10.1 Display Style *2-23*
2.10.2 Render Mode *2-25*
2.10.3 Render Density *2-26*
2.10.4 Color *2-27*
2.11 Saving Screen Images *2-27*
2.12 On Line Help *2-27*
2.13 Miscellaneous *2-28*

3. Setting Defaults *3-1*
3.1 Connector Defaults *3-1*
3.1.1 Dimensions *3-1*
3.1.2 Average Spacing *3-1*
3.1.3 Maximum Angular Deviation *3-2*
3.1.4 Maximum Chord Height Deviation *3-3*
3.1.5 Beginning Clustering *3-3*
3.1.6 Ending Clustering *3-3*
3.1.7 Distribution Function *3-3*
3.2 Elliptic PDE Solver Control Functions *3-3*
3.2.1 Background Control Function *3-3*
3.2.2 Foreground Control Function *3-3*
3.3 Unstructured Grid Attributes *3-4*
3.3.1 Domains *3-4*
3.3.2 Blocks *3-4*
3.4 Domain Topology Controls *3-5*
3.4.1 Split Angle *3-5*
3.4.2 Join Angle *3-5*
3.5 Database Parameterization *3-6*
3.6 QuickSave File Name *3-6*
3.7 Tolerances *3-6*
3.7.1 Node Tolerance *3-6*
3.7.2 Connector Tolerance *3-6*
3.7.3 Grid Point Tolerance *3-6*
3.7.4 Model Size *3-7*
3.8 DB Pick Mask *3-7*
3.9 Other Information *3-7*

4. Importing and Exporting Files *4-1*
4.1 Import Gridgen File *4-1*
4.1.1 Gridgen File Database Associativity *4-2*
4.1.2 Overwrite vs. Append *4-3*

4.1.3 Gridgen File Backward Compatibility........ 4-3
4.2 Quick Import *4-3*
4.3 Export Gridgen File *4-3*
4.4 Importing Database Files *4-4*
4.4.1 Style........ 4-6
4.4.2 Format 4-6
4.4.3 Precision........ 4-6
4.4.4 File Type........ 4-6
4.4.5 Duplicate Name Detection........ 4-10
4.4.6 Automatic Entity Grouping on Import........ 4-11
4.4.7 Backward Compatibility 4-11
4.5 Export Database File *4-12*
4.6 Import Grid Points *4-13*
4.6.1 Style........ 4-14
4.6.2 Format 4-15
4.6.3 Precision........ 4-15
4.6.4 Importing Structured Grid Points 4-15
4.6.5 Importing Unstructured Grid Points 4-16
4.7 Export Grid Points *4-16*
4.7.1 Style........ 4-17
4.7.2 Format 4-17
4.7.3 Precision........ 4-18
4.7.4 Exporting Structured Grid Points........ 4-18
4.7.5 Exporting Unstructured Grid Points........ 4-19
4.8 Export Glyph Initial File *4-19*
4.9 Export Analysis BCs *4-19*
4.10 Export Analysis Grid *4-19*
4.11 Export Analysis Data *4-19*
4.12 Export FIELDVIEW Boundary Surface File *4-19*
4.13 Incorporate Volume Grid *4-20*
4.14 Block Ell Solver Glyph *4-20*

5. Layer Manager........ *5-1*
5.1 Overview *5-1*
5.2 Database Groups and the Layer Manager *5-1*
5.3 Scripting and the Layer Manager *5-1*
5.4 Layer Assignment *5-1*
5.4.1 Database File Import 5-1
5.4.2 The Assign Layer Command 5-1
5.4.3 The Layer Move, Combine Commands 5-3
5.5 The Layer Manager Panel *5-3*
5.5.1 Opening the Layer Manager Panel........ 5-3
5.5.2 Layer Browser 5-5
5.5.3 List Filters 5-5
5.5.4 Layer Operations........ 5-6
5.5.5 Saved Layer Sets 5-7
5.5.6 Special Features........ 5-8

6. Creating Database Entities *6-1*
6.1 Overview *6-1*
6.2 Point *6-1*

6.3 Curve *6-2*
6.3.1 Line *6-3*
6.3.2 Curve........ *6-3*
6.3.3 Akima Curve........ *6-3*
6.3.4 Conic Section........ *6-3*
6.3.5 Circular Arc *6-3*
6.3.6 Offset........ *6-3*
6.3.7 Extract........ *6-4*
6.3.8 Fit........ *6-5*
6.4 Plane *6-5*
6.4.1 Constant Coordinate Plane........ *6-5*
6.4.2 Point-Normal Plane........ *6-6*
6.4.3 3-Point Plane........ *6-6*
6.4.4 Coefficient Plane........ *6-7*
6.5 Surface *6-7*
6.5.1 Ruled Surface *6-7*
6.5.2 Polyconic Surface *6-9*
6.5.3 Surface of Revolution........ *6-10*
6.5.4 Linear Sweep Surface........ *6-11*
6.5.5 Fit........ *6-12*
6.5.6 Coons........ *6-12*
6.6 Note *6-14*

7. Creating Connectors........ *7-1*
7.1 Overview *7-1*
7.2 Add Segment *7-2*
7.2.1 3D Space Segments........ *7-2*
7.2.2 Database Constrained Segments........ *7-3*
7.2.3 Rational Segments........ *7-4*
7.2.4 Other Segments........ *7-7*
7.2.5 Miscellaneous Segments........ *7-8*
7.2.6 Miscellaneous Commands........ *7-9*
7.3 2 Point Connectors *7-10*
7.4 On DB Entities *7-11*

8. Grid Extrusion Methods........ *8-1*
8.1 Running the Extrusion *8-1*
8.2 Extrusion Attributes *8-4*
8.2.1 Focus *8-4*
8.2.2 Surface Shape *8-4*
8.2.3 Aspect Ratio........ *8-5*
8.2.4 Step Size........ *8-5*
8.2.5 Marching Direction *8-7*
8.2.6 Smoothing Parameters *8-9*
8.2.7 Boundary Conditions........ *8-11*
8.2.8 Stop Criteria *8-12*
8.2.9 Copy Attributes........ *8-13*
8.2.10 Reset Attributes........ *8-14*

9. Creating Domains *9-1*

9.1 Overview *9-1*
9.2 Assembling Structured Domains *9-1*
9.3 Assembling Unstructured Domains *9-4*
9.4 Special Domain Assembly Commands *9-7*
9.5 Convert Structured to Unstructured Domains *9-8*
9.6 On Database Entities *9-9*
9.7 Pole Domains *9-10*
9.8 Bad Domain Types *9-10*
9.8.1 Allowable, But Should Be Avoided *9-10*
9.8.2 Duplicates *9-11*
9.8.3 Overlapping and Adjacent *9-11*
9.8.4 Overlapping *9-12*
9.8.5 Prohibited *9-12*

10. Creating Blocks *10-1*
10.1 Overview *10-1*
10.2 Assembling a Structured Block *10-1*
10.2.1 Adding Faces to a Structured Block *10-1*
10.3 Assembling an Unstructured Block *10-3*
10.3.1 Adding Faces to an Unstructured Block *10-3*
10.4 Assembling a Hybrid Unstructured Block *10-4*
10.5 Face Creation Commands *10-4*
10.6 Linkages *10-6*
10.7 Structured Block Topologies and Orientations *10-7*

11. Grid Methods for Connectors *11-1*
11.1 ReDimension *11-1*
11.1.1 Apply Dimension or Average Spacing *11-1*
11.1.2 From Keyboard or Subconnectors *11-1*
11.1.3 Additional Effects *11-3*
11.2 ReDimension Grids *11-3*
11.3 Distributing Grid Points on Connectors *11-5*
11.3.1 Definitions *11-6*
11.3.2 Distribution Parameters *11-6*
11.3.3 Break Points and Subconnectors *11-10*
11.3.4 Miscellaneous Controls *11-12*
11.4 Set Δs Values *11-13*

12. Grid Methods for Structured Domains and Blocks *12-1*
12.1 Algebraic Methods *12-1*
12.1.1 Standard TFI *12-2*
12.1.2 Linear TFI *12-2*
12.1.3 Polar TFI *12-3*
12.1.4 Ortho TFI *12-3*
12.1.5 Parametric TFI *12-3*
12.1.6 Parametric Fit TFI *12-4*
12.1.7 Pole *12-4*
12.2 Elliptic PDE Methods *12-4*
12.2.1 Running the Solver *12-4*
12.2.2 Setting Attributes *12-6*

12.2.3 Relaxation Parameter *12-7*
12.2.4 Background Control Functions........ *12-8*
12.2.5 Foreground Control Functions *12-9*
12.2.6 Surface Shape *12-11*
12.2.7 Boundary Conditions........ *12-13*
12.3 Structured SubGrids* *12-14
12.3.1 Adding a SubGrid........ *12-14*
12.3.2 Deleting SubGrids........ *12-15*
12.3.3 Using SubGrids........ *12-15*

13. Grid Methods for Unstructured Domains and Blocks........ *13-1*
13.1 Running the Solver* *13-1
13.2 Setting Attributes* *13-2
13.3 Surface Shape Attributes* *13-4
13.3.1 Free........ *13-4*
13.3.2 DB *13-4*
13.4 Relaxation Parameter Attribute* *13-5
13.5 Triangular Mesh Attributes* *13-5
13.5.1 Minimum Triangle Edge *13-6*
13.5.2 Maximum Triangle Edge *13-6*
13.5.3 Maximum Surface Deviation *13-6*
13.5.4 Maximum Angular Deviation *13-6*
13.5.5 Boundary Decay Factor *13-7*
13.6 Tetrahedral Mesh Attributes* *13-7
13.6.1 Tetrahedral Solver Memory Limit........ *13-8*
13.6.2 Tetrahedral Iterations........ *13-8*
13.6.3 Tetrahedral Size Factor........ *13-8*
13.7 Pyramid Mesh Attributes* *13-8
13.7.1 Minimum Pyramid Height........ *13-9*
13.7.2 Maximum Pyramid Height........ *13-9*
13.7.3 Pyramid Aspect Ratio........ *13-9*
13.7.4 Pyramid Display *13-9*
13.8 Grid Point Insertion on Boundaries* *13-9
13.8.1 Domains........ *13-9*
13.8.2 Blocks........ *13-9*

14. Modifying Entities........ *14-1*
14.1 Overview* *14-1
14.2 Linkages* *14-2
14.2.1 Database Linkages *14-2*
14.2.2 Connector Linkages *14-4*
14.2.3 Domain Linkages *14-6*
14.2.4 Block Linkages........ *14-6*
14.3 Transformations* *14-7
14.3.1 Translate *14-7*
14.3.2 Scale *14-8*
14.3.3 Stretch........ *14-9*
14.3.4 Rotate........ *14-10*
14.3.5 Mirror........ *14-12*
14.4 Project* *14-12
14.4.1 Spherical *14-13*

14.4.2 Cylindrical 14-14
14.4.3 Linear 14-14
14.4.4 Closest Point 14-15
14.5 Smooth *14-15*
14.6 Re-extrude *14-15*
14.7 Split *14-16*
14.7.1 Database Entities 14-16
14.7.2 Connectors 14-16
14.7.3 Domains 14-17
14.7.4 Blocks 14-18
14.8 Join *14-19*
14.8.1 Database Entities 14-19
14.8.2 Connectors 14-19
14.8.3 Domains 14-20
14.8.4 Blocks 14-20
14.9 Connector Segment Modification *14-20*
14.9.1 Add 14-20
14.9.2 Insert 14-20
14.9.3 Erase 14-20
14.9.4 Edit 14-21
14.10 Connector Least Squares Fit *14-23*

15. Miscellaneous *15-1*
15.1 Name *15-1*
15.1.1 Database Entities 15-1
15.1.2 Blocks 15-2
15.2 Group *15-2*
15.2.1 Database Entities 15-2
15.2.2 Connectors, Domains, and Blocks 15-2
15.3 Copy *15-3*
15.3.1 Database Entities 15-3
15.3.2 Connectors, Domains, and Blocks 15-3
15.4 Delete *15-3*
15.4.1 Database Entities 15-4
15.4.2 Connectors 15-4
15.4.3 Domains 15-4
15.4.4 Blocks 15-5
15.5 Sort *15-5*
15.6 Enable/Disable *15-7*
15.6.1 Database Entities 15-7
15.6.2 Connectors 15-7
15.6.3 Domains 15-7
15.6.4 Blocks 15-7
15.7 ReSpecify ξ, η and ζ *15-7*
15.8 Align ξ, η, ζ *15-8*
15.9 Periodicity *15-8*
15.10 Merge *15-11*
15.10.1 Nodes 15-11
15.10.2 Connectors 15-12
15.10.3 Nodes and Connectors 15-13
15.10.4 Automatic 15-15

15.10.5 Pick 15-15
15.10.6 Update Doms 15-15
15.10.7 Topological Filter Commands 15-15
15.11 Intersect *15-15*
15.12 Feature Extraction *15-17*

16. Diagnostic Functions *16-1*
16.1 Focus Entity *16-1*
16.2 Diagnostic Function *16-2*
16.2.1 None........ *16-2*
16.2.2 Usage........ *16-2*
16.2.3 Database........ *16-3*
16.2.4 Size *16-3*
16.2.5 Jacobian........ *16-4*
16.2.6 Size Ratio........ *16-4*
16.2.7 Smoothness *16-5*
16.2.8 Aspect Ratio........ *16-5*
16.2.9 Skewness........ *16-6*
16.2.10 Spacing *16-6*
16.2.11 Orthogonality........ *16-7*
16.3 Display Attributes *16-7*
16.3.1 Display Style *16-8*
16.3.2 Color Bar Type *16-8*
16.3.3 Color Bar Values........ *16-9*
16.3.4 Show Cells *16-9*
16.3.5 Histogram........ *16-10*
16.4 Mode *16-10*
16.5 Hiliter *16-10*
16.5.1 Connectors........ *16-11*
16.5.2 Domains........ *16-11*
16.5.3 Blocks........ *16-11*
16.6 Reference Surfaces *16-14*

17. Analysis Software Interfaces........ *17-1*
17.1 Select Analysis Software *17-1*
17.2 Set BCs *17-3*
17.2.1 Structured Grids........ *17-5*
17.2.2 Unstructured Grids........ *17-5*
17.3 Set VCs *17-5*
17.3.1 Applying VCs........ *17-6*
17.4 Examine *17-6*
17.5 Custom BCs, Creating, Editing & Deletion *17-7*
17.6 Custom VCs, Creating, Editing & Deletion *17-7*
17.7 Mirror Blocks for Export *17-7*
17.8 Export Analysis Data, Grid, and BCs *17-7*
17.8.1 generic *17-8*
17.8.2 ADPAC........ *17-8*
17.8.3 CFDSHIP-Iowa........ *17-8*
17.8.4 CFD++........ *17-8*
17.8.5 CFX-4........ *17-9*
17.8.6 CFX-5........ *17-9*

17.8.7 CGNS-Struct *17-9*
17.8.8 CGNS-Unstr *17-9*
17.8.9 CNSFV *17-9*
17.8.10 Cobalt *17-9*
17.8.11 COMO *17-9*
17.8.12 CRUNCH *17-9*
17.8.13 DTNS *17-10*
17.8.14 Exodus II *17-10*
17.8.15 FALCON *17-10*
17.8.16 FANS *17-10*
17.8.17 FDNS/UNIC *17-10*
17.8.18 FLUENT v4 .... *17-10*
17.8.19 FLUENT *17-10*
17.8.20 GASP *17-10*
17.8.21 GUST *17-11*
17.8.22 INCA v2 .... *17-11*
17.8.23 INCA v3 .... *17-11*
17.8.24 NCC *17-12*
17.8.25 NPARC *17-12*
17.8.26 OVERFLOW *17-13*
17.8.27 PHOENICS *17-13*
17.8.28 SCRYU *17-13*
17.8.29 SCRYU/Tetra *17-13*
17.8.30 Splitflow *17-13*
17.8.31 STAR-CD *17-14*
17.8.32 TACOMA *17-14*
17.8.33 TASCflow *17-14*
17.8.34 TEAM *17-14*
17.8.35 TETREX *17-14*
17.8.36 VSAERO *17-15*
17.8.37 VSAEROhybrid *17-15*
17.8.38 WIND *17-15*

18. File Formats *18-1*
18.1 Introduction *18-1*
18.2 Gridgen File *18-1*
18.3 GRIDBLOCK Files *18-1*
18.4 Database Network File *18-2*
18.5 Segment File *18-2*
18.6 Composite Database File *18-3*
18.7 FIELDVIEW Unstructured File *18-3*
18.8 Generic Flow Solver BC File *18-3*
18.9 Surface Grid File *18-5*
18.10 Volume Grid File *18-5*

19. References *19-1*

20. Index *20-1*

1. Introduction

Gridgen is a software system for the generation of three-dimensional (3D) grids and meshes. You may build a grid around any type of object using Gridgen; it is not limited to any particular geometry. The grids you create in Gridgen can be used with any type of analysis software including computational fluid dynamics (CFD) and finite element analysis (FEA). In fact, Gridgen has built-in filters for many analysis codes as well as popular neutral formats.

The following sections describe in detail the use of each Gridgen command. If you are a new user and are interested in learning about how to run Gridgen we recommend that you begin by reading and running the Tutorials in the Tutorial Workbook. The first several chapters of this User Manual are meant to be used as a reference. You will consult them to get detailed information about particular commands you are trying to use.

1.1 Grid Types

Gridgen may be used to generate multi-block grids. A multi-block grid is created by dividing the overall region to be gridded into one or more sub-regions called blocks.

Gridgen may be used to generate both 2D and 3D blocks. A 2D block consists entirely of surface cells while a 3D block contains entirely volumetric cells. All blocks in the same grid must either be 2D or 3D; you cannot combine blocks of differing dimensionality in the same grid.

The grid in each block may either be structured, unstructured, or hybrid. A structured grid consists entirely of hexahedral cells that have been arranged in an ordered `IxJxK` array. An unstructured grid consists of tetrahedral, pyramid, and prism cells having no implicit order. If all blocks contain structured grids, then the overall grid is called structured. When all blocks contain unstructured grids, then the overall grid is called unstructured. If the overall grid consists of both structured and unstructured blocks, the overall grid is called hybrid, a special type of unstructured grid.

1.2 Data Hierarchy

Data hierarchy refers to the four types of data (called entities) you will work with during the construction of your grid. Gridgen organizes the entities into four types:

- database: geometry data that defines the shape of the object being gridded
- connectors: curve grids
- domains: surface grids
- blocks: volume grids

The data hierarchy’s foundation is the *database*: a geometry model of the object on and around which a grid is to be generated. You typically obtain the database from a computer aided design (CAD) system and import it into Gridgen. Gridgen also provides tools for creating database entities. A database may consist of points, curves, surfaces, and other topological data. Keep in mind, however, Gridgen does not require a database; the geometry of the grid entities you create may implicitly define the grid’s shape. Furthermore, Gridgen does not require the database to represent a closed solid model.

The remaining three elements of Gridgen’s hierarchy are the grid entities: connectors (curve grids), domains (surface grids), and blocks (volume grids). Grid entities are ordered according to their computational dimension. Note, however, all of Gridgen's grid entities may have a 3D shape (in other words, all three components of the Cartesian coordinates `(x,y,z)` may vary).

Connectors, the grid entity at the bottom of the hierarchy, are curves on which grid points are distributed. A connector, what a CAD system would call a composite curve, consists of one or more segments. *Segments* are basic curve types, which can be polylines, polycurves, conics and circles, or curves obtained from a database.

The middle grid entity is the *domain*, a surface grid. An initial domain grid is generated automatically using a default method when the domain is first created. A variety of grid methods can then be applied to improve the grid's quality.

Volume grids called *blocks* are at the top of the data hierarchy. As with domains, the initial block grid is created automatically, and a variety of grid methods may then be applied to improve the grid's quality.

This data hierarchy provides the framework by which modifications to any entity are propagated throughout the rest of the hierarchy. For example, modification of a connector's shape will update all domains and blocks that use it. The term ‘‘forward editing’’ refers to grid changes propagated up the hierarchy while ‘‘backward editing’’ refers to downward propagating changes. Gridgen's data hierarchy and forward and backward editing allow you to change one entity of your grid and have Gridgen make the entire grid system consistent with your change.

1.3 Command Hierarchy and Menu Structure

The commands you will execute while running Gridgen are arranged in nested menus. Gridgen's main menu is described below.

Input/Output - commands for reading and writing data files

Database - commands for manipulating the geometry model

Connectors - commands for manipulating curve grids

Domains - commands for manipulating surface grids

Blocks - commands for manipulating volume grids

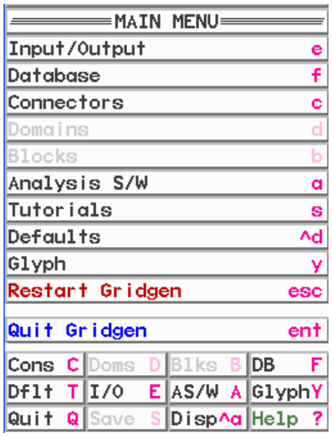
Analysis S/W - commands for preparing your grid for use with your analysis software package

Tutorials - import one of Gridgen's tutorial grids

Defaults - commands for setting various Gridgen switches

Glyph - commands for creating, modifying, and replaying script and journal files

Restart Gridgen - is used to quickly delete all or a portion of the current project. You can choose to delete the grid and/or database. You can also reset tolerances, the selected analysis software and Default menu items back to their original default settings.



Quit Gridgen - exit Gridgen

Display Commands - commands for changing Gridgen’s appearance and the display of your grid and database

Help - invoke Gridgen’s on-line help

One feature that makes Gridgen easy to learn and use is the parallelism of the menu structures for the four data hierarchy entities (database, connectors, domains, blocks). These menus all contain similar commands for each entity type:

Create - make a new entity

Copy - duplicate an existing entity

Delete - remove an entity

Modify - change an existing entity

Examine - obtain detailed information about an entity

Furthermore, the **Modify** menu commands listed below operate similarly for each entity type:

Translate

Scale

Stretch

Mirror

Rotate

Split

Join

Of course, the commands for working with the grid points also follow this parallelism: **Dimension** for assigning a number of grid points to a connector, **Distribute** for clustering grid points along a connector, and the **Solvers** for the domain and block grids.

The object of this command parallelism is obvious; once you know how to copy and translate connectors (for example) you will also know how to copy and translate database entities, domains, and blocks.

1.4 Planning Your Grid

Before you start generating a grid with Gridgen (or by any other means), you should first consider many issues related to the analysis. For example, if your grid is to be used for a CFD analysis, you can save a lot of time and avoid costly false starts and rework by considering all of these items before you begin generating your grid.

1.4.1 Define the Modeling Goals

What specific results are required from the analysis? Are you just looking for general flow patterns or more detailed information about pressure, heat transfer, and separation locations in the flow field?

What degree of accuracy is required? Are you looking for trends or absolutes?

How much time do you have to complete the analysis? Do you have the time (both labor hours and computer) to analyze the complete system or can you obtain meaningful results by analyzing only the component of interest?

1.4.2 Identify Your Grid's Constraints

What types of grid topologies can your analysis software handle?

What is the maximum number of grid points you can use? What is the maximum number of grid blocks you can use?

1.4.3 Plan the Grid

What grid topologies will work best?

Which areas will require a finer grid to accurately resolve a boundary layer, shock wave or other flowfield feature? Which areas can be covered with a coarser grid?

Are there components of the geometry that will be changed for additional analyses? Can they be isolated in the grid for later ease in editing?

1.5 Outline of the Gridgen Process

After you have planned your grid and familiarized yourself with Gridgen's terminology and command structure, you are ready to begin generating your grid. Gridgen is designed to guide you through the grid generation process in a specific order proceeding from grid curves (the one dimensional grid elements) through surface grids (the two dimensional grid elements), to volume grids (the three dimensional grid elements). This process allows Gridgen to automate many of the grid topology and dimension bookkeeping tasks you would have to keep track of yourself otherwise.

The basic steps you will follow are listed below. Some of the steps are optional, as noted.

1. Choose the analysis software you will use and whether the grid will be 2D or 3D.
2. Create or import the geometry model (database). (optional)
3. Create the curves (connectors) that define the boundaries of surface grids.
4. Distribute grid points along the connectors.
5. Pick closed loops of connectors to define surface grids (domains).
6. Apply a grid method to improve the distribution of surface grid points. (optional)
7. Pick closed shells of domains to define volume grids (blocks).
8. Apply a grid method to improve the distribution of volume grid points. (optional)
9. Set analysis software boundary conditions. (optional)
10. Export analysis software data files.

11. Save the Gridgen file.

While the detailed operations performed in each of these steps will vary for each grid you build, you will always follow these basic steps in this general order. We also recommend you create some of your domains (surface grids) as soon as you have enough connectors created. This will allow you to see immediately whether or not the distribution of grid points you applied to your connectors results in a good domain surface grid. By doing this, you will be able to identify and correct potential problems at the earliest possible stage in the grid generation process resulting in a better final grid.

1.6 Gridgen's History

Gridgen has been a product of Pointwise, Inc. since 1994. The software was initiated and has been continually developed by Pointwise's founders since 1984.

Portions of Gridgen Versions 12 and 13 were developed under the sponsorship of the United States Air Force Research Laboratory.

Prior to 1994, Gridgen was developed by Pointwise's founders under contract to various U.S. Government agencies.

GRIDGEN Version 9's development was sponsored by NASA Ames Research Center during the period between October 1992 and April 1994. Additional sponsorship was provided by NASA Lewis Research Center, the U.S. Naval Surface Warfare Center Carderock Division, and the U.S. Army Research Laboratory.

GRIDGEN Version 8's development was sponsored by NASA Langley Research Center during the period between July 1991 and December 1992. Additional sponsorship was provided by the Naval Surface Warfare Center Carderock Division.

GRIDGEN Version 7 was a developmental version that was incorporated into Version 8.

GRIDGEN Version 6's development was sponsored by the Aeromechanics Group, Wright Research and Development Center, Wright-Patterson AFB during the period between October 1987 and September 1990

GRIDGEN originated in mid 1984 and was developed through September 1987 as an Independent Research and Development (IR&D) project at General Dynamics Fort Worth Division

1.7 Document Conventions

Buttons on Gridgen's menus will appear in a bold Helvetica font: for example, **Translate**. Labels for menu buttons will appear in a Helvetica font: for example, Solvers.

Hot keys, keyboard keys invoking a menu button command when pressed, appear in a bold italic font: for example, the ***x*** hot key.

Data you type and messages that Gridgen prints appear in a Courier font: for example, `1,2,3`.

2. Graphical User Interface

2.1 The Gridgen Screen

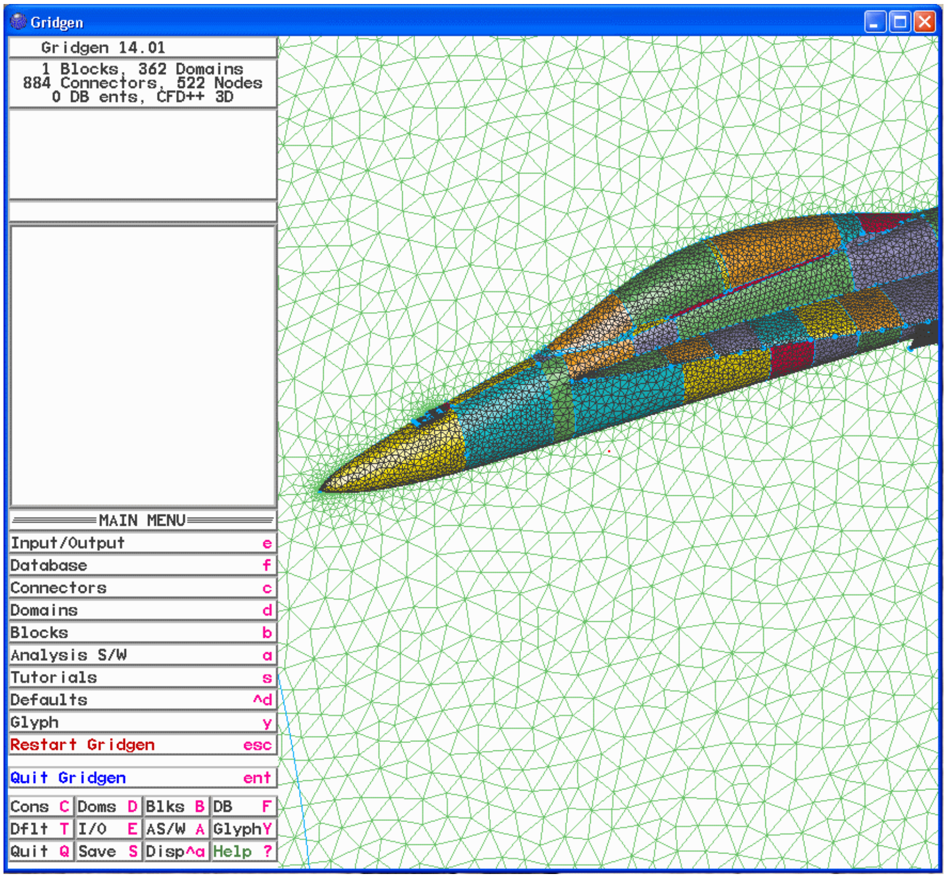
Gridgen's screen is divided into several non-overlapping (tiled) windows, each of which serves a specific function as illustrated in the figure below and described in the following list.

From top to bottom, the windows on the left side of the screen are:

- The *Version window* displays the program's name and its version number. We refer to the number to the left of the decimal point as the major version number. The number to the right of the first decimal point is the minor version number.
- The total number of blocks, domains, connectors, nodes, database entities, and the current analysis software are displayed in the *Status window*.
- The *Message window* is the guidance and prompting window. Messages appear here to provide brief, step-by-step instructions, and to prompt you when input is required from the keyboard.
- The *Text Input window* is where text or numerical data is displayed as you type during input (Section 2.6).
- Various context sensitive diagnostic information is displayed in the *Blackboard window*. This same window also goes by the name *Browser window* when it is used for picking (Section 2.4) items (for example, file names, blocks, database entities, etc.).
- The command menu is displayed in the *Button Menu window*. Most of your interaction with Gridgen will be through this window.
- The main graphics window is called the *Display window* and is used to display the database entities and the grid system.

Obviously, the Display and Button Menu windows are very important. However, the most important window for novice users to monitor is the Message window. During command execution, this window provides concise, step-by-step instructions.

Gridgen's screen is repositionable and resizable. The screen may cover your entire monitor or you can shrink it down to about one quarter of the monitor. During screen resizing the tiled layout of the windows within Gridgen's screen is maintained automatically.



2.2 The Mouse

The mouse and mouse buttons are always used in conjunction with the cursor, the small glyph (usually an arrow pointing up and to the left) that appears on the screen and is moved around the screen by moving the mouse. Horizontal and vertical movements of the mouse result in horizontal and vertical movements of the cursor on the screen, respectively. In Gridgen's default configuration, holding down one of the mouse keys while moving the mouse and while the cursor is in the Display window will have the following effects.

- ***left mouse***: The image in the Display window will be panned across the screen.
- ***middle mouse***: The image in the Display window will be zoomed on the screen.
- ***right mouse***: The 3D cursor will be moved.

The ***right mouse*** button is also used to pick items, including both menu buttons and graphical objects in the Display window. The ***left mouse*** button may also be used to pick menu buttons but not graphical objects. If you prefer to use the ***left mouse*** button for picking all the time, us the **LeftPk** option in the DISPLAY COMMANDS menu.

Gridgen's cursors are context sensitive and will change depending on whether the menu is being probed, the image is being rotated, panned, or zoomed, or whether an error condition has been detected.

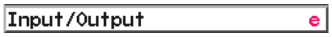
The cursor may also be restricted to move only in directions that you choose (Section 2.8.1).

2.3 Menu Buttons

2.3.1 Commands

To select a command via its menu button follow this procedure:

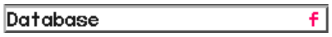
1. Position the cursor over the menu button. The button will be highlighted.
2. Press and release either the ***left*** or ***right mouse*** button.



The Menu window displays menu buttons with text corresponding to the commands that you may execute. The text on each menu button is left-justified and either black in color or "grayed-out". The grayed-out text represents commands that are not available to you in the current state of the grid. As the grid evolves, the grayed-out commands will become available. For example, if no domains have been created, the MAIN MENU button for the BLOCK COMMANDS menu (**Blocks**) will be grayed-out because domains are necessary for block creation. As soon as you create a domain, the menu button for the **Blocks** command will become available, and its text will be black in color.

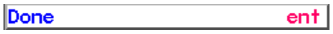
To select a button from the menu (and thereby activate the command described by the button text), position the cursor over the button (using the mouse), thereby highlighting the menu button's display. Then press and release the ***right mouse*** or ***left mouse*** button to activate the command described by the button. In the remainder of this manual, this process of selecting a menu button will be referred to by the phrase "pressing the **X** button" where **X** is the command name. Notice that the menu button is not selected until the ***mouse*** button is released, so the cursor may be moved to a different menu button or even off the menu after the ***mouse*** button is pressed.

2.3.2 Hot Keys

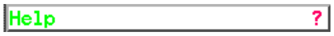


Almost every command can be selected by pressing a specific keyboard key called a hot key. A command's hot key is shown right-justified on its menu button after the command name. For example, the MAIN MENU button **Database f** indicates that the DATABASE COMMAND menu may be invoked by pressing the ***f*** key. A caret (^) preceding a hot key means that the control key must be pressed at the same time as the hot key to invoke the command. For example, the MAIN MENU button **Defaults ^d** indicates that the SET DEFAULT VALUES menu may be invoked by pressing the ***control*** and ***d*** keys simultaneously.

Gridgen has three special hot keys that correspond to commands that appear on most every menu. These keys are named according to the color of the text that appears on the menu button and are described below.



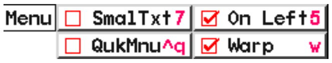
The *blue button* corresponds to commands of the form "Done" or "Continue". The ***enter*** key is always the blue button's hot key.



The *green button* invokes Gridgen's on-line help. The ***?*** (question mark) key is always the green button's hot key.

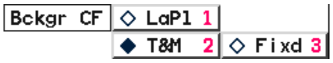
The *red button* corresponds to commands of the form “Abort” or “Exit”. The ***escape*** key is always the red button’s hot key.

2.3.3 Toggle Buttons



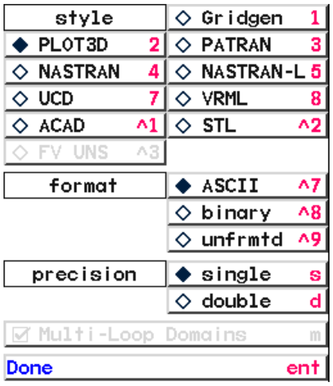
Some of Gridgen’s menu buttons act as a toggle for the setting of a certain switch. For example, in the DISPLAY COMMANDS menu there is a toggle button called Menu **SmalTxt** that determines whether to use a large or small font for menu button text. Toggle buttons can be identified by the red square on the button. If the red square is empty the setting denoted by the button text is inactive. If the red square contains a red check mark, the setting is active. Pressing the toggle button changes the state of the setting.

2.3.4 Radio Buttons



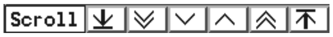
Some of Gridgen’s menu buttons act as a group of choices for the setting of a certain switch. These buttons are called *radio* buttons since they function like the programmed station buttons on a radio. In any group of radio buttons only one button may be active at any one time. Radio buttons can be identified by the blue diamond on the button. The active option in a group of radio buttons is denoted by the filled blue diamond. Pressing one radio button implicitly “unpresses” all other radio buttons in the same group.

In some instances pressing a radio button only sets the switches and another button must be pressed to activate the switch. For example, the FILE ATTRIBUTES menu contains three radio button groups that allow you to specify the style, format, and precision of a file during import. A fourth button, **Done**, also appears on the menu and must be pressed before import proceeds with the switches set according to the radio button settings.



2.3.5 VCR Buttons

When selecting items from the Browser (Section 2.4.2) Gridgen displays in the Menu window a single row of six menu buttons for positioning the Browser’s list of items. These buttons are called *VCR* buttons since they function like the buttons on a video cassette recorder (for example, fast forward, rewind). VCR buttons are easily identified because they use symbols instead of text.

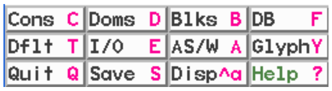


- The button with an arrow pointing downward to a horizontal line is the *End of List* button. When it is pressed the list is positioned so that the last item is at the bottom. Its hot key is →.
- The button with two downward pointing “vees” is the *Page Down* button. When it is pressed the list is moved downward by one page. Its hot key is ^↓.
- The button with one downward pointing “vee” is the *Scroll Down* button. When it is pressed the list is moved downward by one line. Its hot key is ↓.
- The button with one upward pointing “vee” is the *Scroll Up* button. When it is pressed the list is moved upward by one line. Its hot key is ↑.
- The button with two upward pointing “vees” is the *Page Up* button. When it is pressed the list is moved upward by one page. Its hot key is ^↑.
- The button with an arrow pointing upward to a horizontal line is the *Beginning of List* button. When it is pressed the list is positioned so that the first item is at the top. Its hot key is ←.

2.3.6 Quick Keys

To jump to a command menu using the Quick Keys follow this procedure.

1. Position the cursor over the Quick Key.
2. Press and release either the ***left*** or ***right mouse*** button.



Quick Keys are available when the **QukMnu** toggle in the DISPLAY COMMANDS menu (Section 2.9) is turned off. They are used to:

- jump from your current location in Gridgen’s menu structure into another menu
- to export a Gridgen file
- to exit Gridgen

When they are available, the Quick Keys appear as three rows of four buttons each at the bottom of the menu. The Quick Keys’ hot keys are all uppercase. The Quick Keys are:

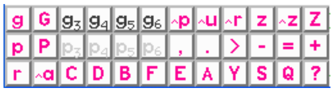
- **Cons** - CONNECTOR COMMANDS menu
- **Doms** - DOMAIN COMMANDS menu
- **Blks** - BLOCK COMMANDS menu
- **DB** - DATABASE COMMANDS menu
- **Dflt** - SET DEFAULT VALUES menu
- **I/O** - INPUT/OUTPUT COMMANDS menu
- **AS/W** - ANALYSIS SOFTWARE COMMANDS menu

- **Glyph** - GLYPH COMMANDS menu
- **Quit** - exit Gridgen
- **Save** - QuickSave, writes a Gridgen file to the file name set in the SET DEFAULT VALUES menu
- **Disp** - DISPLAY COMMANDS menu
- **Help** - start Gridgen's help system

2.3.7 Quick Menu

To use a Quick menu command follow this procedure.

1. Make sure the Quick menu is enabled by the **QukMnu** toggle (Section 2.3.7) button in the DISPLAY COMMANDS menu.
2. Position the cursor over the desired Quick menu button.
3. Identify the command you wish to use by the descriptive text that appears when the cursor is over the button.
4. Press and release either the ***left*** or ***right mouse*** button.


g G g_3 g_4 g_5 g_6 ^p ^u ^r z ^z Z
p P p_3 p_4 p_5 p_6 , . > - = +
r ^a C D B F E A Y S Q ?

The Quick menu is available when the **QukMnu** toggle in the Display Commands menu (Section 2.9) is turned on. Buttons on the Quick menu are used to

- save and restore orientations of the Display window image
- unrotate, untranslate, unzoom, and zoom the Display window image
- control image manipulation sensitivities
- jump from your current location in Gridgen's menu structure into another menu
- export a Gridgen file
- exit Gridgen

Note that the Quick Keys (Section 2.3.6) are a subset of the Quick menu. There are two main reasons why you might choose to enable the Quick menu.

- Lesser used but powerful Gridgen commands are always available on the menu. This allows novice users to find and use these commands by the text description that appears when the mouse is moved over the Quick menu. It also allows advanced users to find these commands without searching through the DISPLAY COMMANDS menu.
- By moving the Quick menu commands to the bottom left corner of the Gridgen screen, the main portion of the Menu window is less cluttered for ease in navigation.

The Quick menu commands are:

g - Save (get) the current orientation as orientation #1.

G - Save (get) the current orientation as orientation #2.

g_3 - Save (get) the current orientation as orientation #3.

g_4 - Save (get) the current orientation as orientation #4.

g_5 - Save (get) the current orientation as orientation #5.

g_6 - Save (get) the current orientation as orientation #6.

^p - (un-pan) Move the Display window image so that the rotation axes are at the Display window center. The zoom depth and rotation of the image are not changed.

^u - (un-zoom) Move the Display window image so that the zoom factor is set to one, relative to the current view limits. The panning and rotation of the image are not changed.

^r - (un-rotate) Move the Display window image so that the body and screen axes systems are aligned. The panning and zooming of the image are not changed.

z - Open the zoom-in box.

^z - Open the zoom-out box.

Z - Pop to the orientation prior to the last issued zoom-in or zoom-out command.

p - Restore (put) orientation #1.

P - Restore (put) orientation #2.

p_3 - Restore (put) orientation #3.

p_4 - Restore (put) orientation #4.

p_5 - Restore (put) orientation #5.

p_6 - Restore (put) orientation #6.

, - (comma) Decrease mouse sensitivity.

. - (period) Increase mouse sensitivity

> - Reset mouse sensitivity.

- - (dash) Decrease keypad sensitivity.

= - Increase keypad sensitivity.

+ - Reset keypad sensitivity.

r - Restore original orientation. Simultaneously un-pans, un-zooms, and un-rotates the image and resets the rotation point.

^a - Go to DISPLAY COMMANDS menu.

C - Go to CONNECTOR COMMANDS menu.

D - Go to DOMAIN COMMANDS menu.

B - Go to BLOCK COMMANDS menu.

F - Go to DATABASE COMMANDS menu.

E - Go to INPUT/OUTPUT COMMANDS menu.

A - Go to ANALYSIS SOFTWARE COMMANDS menu.

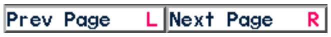
Y - Go to GLYPH COMMANDS menu.

S - Export a QuickSave Gridgen file.

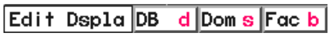
Q - Exit Gridgen.

? - Invoke on-line help.

2.3.8 Miscellaneous



Some menus have too many buttons to fit onto a single page. In this case the menu will be split into two or more pages and buttons **Next Page** and **Previous Page** at the top of the menu may be used to step through all of the menu's pages.



Some menus use *labels* to group similar commands and to provide additional information on command function. Labels are identified by a thin black border in place of the beveled border around buttons. Labels are not pickable.

Gridgen sometimes employs a user interface technique called *mouse warping* in which the cursor is automatically positioned over a particular button, usually the default command for the current menu. Mouse warping may be disabled via the **Warp** command in the DISPLAY COMMANDS menu (Section 2.9).

2.4 Entity Selection

Most of Gridgen's commands require you to select (or pick) an entity or entities to which the command will be applied. For example, a connector must be picked in order to distribute grid points on it, and database entities must be picked in order to translate them. There are two main mechanisms for picking: the Display window and the Browser.

Depending on the command, you may select entities from either the Display window, the Browser, or both. When both mechanisms are available, the choice is yours - both yield equivalent results. Note that when both picking mechanisms are available, pointing to and highlighting an entity's graphical image in the Display window also highlights its text listing in the Browser and vice versa. The cursor looks like a hand with a pointing index finger during selecting. When picking database entities, an additional filter may be applied to limit the type of selectable entities (see Section 2.4.3).

When a command allows multiple entities to be picked, keep in mind that the picked entities are added to and held in a queue until the **Done** button is pressed at which time the command begins operating on the entities. Before the **Done** button is pressed items can be removed from the queue. A command that is limited to working on a single item begins operating as soon as you pick the entity.

2.4.1 Selecting From The Display Window

2.4.1.1 Selecting Entities One At A Time

Entities may be picked one at a time by pointing at the entity and pressing and releasing either the ***left*** or ***right mouse*** button, depending on the setting of the **LeftPk** toggle button in the DISPLAY COMMANDS menu (Section 2.9). When the cursor is positioned over an entity, the entity is highlighted in salmon meaning that it is pickable. After the ***right mouse*** button is pressed the entity is highlighted in white meaning it has been picked.

For historical reasons, this document often refers to entity selection with the phrase "pick with the ***right mouse*** button"; keep in mind that either the ***left*** or ***right mouse*** button may be used.

In some instances several pickable entities will lie beneath the cursor. In these cases Gridgen highlights in salmon one of them (for which one see Section 2.4.4). If this is not the entity you want to select, you can step through all pickable entities beneath the cursor one at a time, by pressing the ***x*** or ^***x*** keys on the keyboard with the cursor held in the same location. The ***x*** key steps through the list of entities beneath the cursor in ascending numerical order while the ^***x*** key moves in descending numerical order. When the desired entity is highlighted in salmon it can be picked using the mouse.

2.4.1.2 Selecting Several Entities At Once

To select several entities at one time from the Display window follow this procedure:

1. Press either the Pick **in box** or Pick **on box** buttons. These buttons allow you to draw a box in the Display window called the pick box. Position the upper left corner of the box using the mouse. Press and hold either the ***left*** or ***right mouse*** (see **LeftPk** in Section 2.9) button and move the mouse to position the lower right corner so that the box encloses the desired entities. Release the ***mouse*** button.
2. If the **in box** is being used, every entity that is fully enclosed by the pick box is picked. If using the **on box,** every entity either enclosed or touched by the pick box is picked.

Repeat the selection process until all desired items are queued and then press the **Done** button.

2.4.2 Selecting From The Browser

Commands that allow picking via the Browser will present a single- or two-column list of entities from which you may pick. The Browser text for each entity depends on the command and the entity type being picked.

When the list of items exceeds the Browser's capacity Gridgen will create a multi-page Browser list. You may scroll through the entire list using VCR menu buttons (Section 2.3.5). When picking from both the Browser and Display windows and when the Browser contains a multi-page list, picking an item's graphical display will cause the Browser list to be repositioned so the item's list entry is visible.

2.4.2.1 Selecting Entities One At A Time

To select entities from the Browser window follow this procedure:

1. Position the cursor over an entity's text in the Browser window. The text will be highlighted.
2. Press and release either the ***left*** or ***right mouse*** button. The text will now be white.
3. If the command can only operate on a single entity, it will begin operating on the picked entity now.
4. If the command allows multiple entities, repeat the selection process until all desired entities are queued and then press the **Done** button.

2.4.2.2 Selecting Several Entities At Once

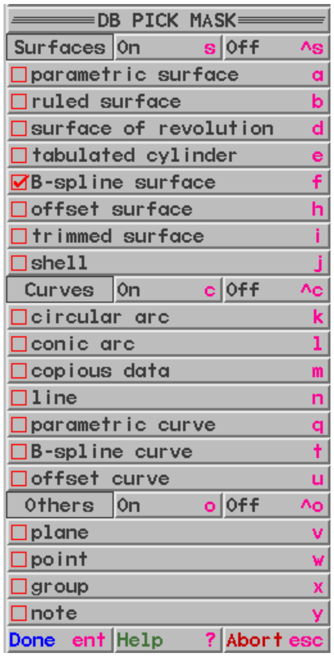
To select several entities from the Browser window follow this procedure:

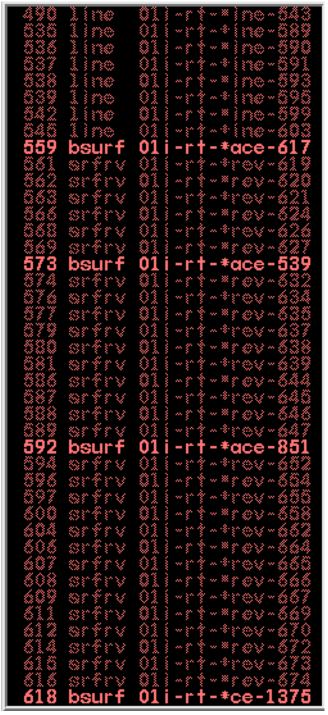
1. Position the cursor over an entity's text in the Browser window. The text will be highlighted.
2. Press and hold either the ***left*** or ***right mouse*** button.
3. Move the cursor in the Browser to sweep over several items and release the ***mouse*** button. The text will be highlighted in white.
4. Repeat the selection process until all desired entities are queued and then press the **Done** button.

Picking multiple items from the Browser is facilitated by the fact that the list automatically scrolls up or down when the ***mouse*** button is held down and the cursor is moved above or below the Browser, respectively.

2.4.3 Pick Masks

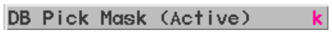
Enitity filtering is currently available only for database entities throught the **DB Pick Mask** command. **DB Pick Mask** can be found in the SET DEFAULT VALUES menu and within the Browser list for database commands where the selection of entities is required. With this command, users can filter the list of entities in the Browser so that only certain database entities types can be selected by the picking functions. Database entities available for filtering are displayed in the DB PICK MASK menu seen below on the left. These entities may be toggled to **On** or **Off** individually or in groups. All entities toggled **Off** will appear darkly shaded in the Browser window list and users will not be able to select them. The figure below on the right shows an example of the Browser window list with only B-spline surfaces toggled to **On.**





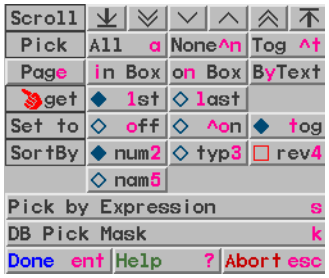
```
490 line  01i-rt-*ine-543
535 line  01i-rt-*ine-589
536 line  01i-rt-*ine-590
537 line  01i-rt-*ine-591
538 line  01i-rt-*ine-593
539 line  01i-rt-*ine-596
542 line  01i-rt-*ine-599
545 line  01i-rt-*ine-603
559 bsurf 01i-rt-*ace-617
561 srfrv 01i-rt-*rev-619
562 srfrv 01i-rt-*rev-620
563 srfrv 01i-rt-*rev-621
566 srfrv 01i-rt-*rev-624
568 srfrv 01i-rt-*rev-626
569 srfrv 01i-rt-*rev-627
573 bsurf 01i-rt-*ace-539
574 srfrv 01i-rt-*rev-632
576 srfrv 01i-rt-*rev-634
577 srfrv 01i-rt-*rev-635
579 srfrv 01i-rt-*rev-637
580 srfrv 01i-rt-*rev-638
581 srfrv 01i-rt-*rev-639
586 srfrv 01i-rt-*rev-644
587 srfrv 01i-rt-*rev-645
588 srfrv 01i-rt-*rev-646
589 srfrv 01i-rt-*rev-647
592 bsurf 01i-rt-*ace-851
594 srfrv 01i-rt-*rev-652
596 srfrv 01i-rt-*rev-654
597 srfrv 01i-rt-*rev-655
600 srfrv 01i-rt-*rev-658
604 srfrv 01i-rt-*rev-662
606 srfrv 01i-rt-*rev-664
607 srfrv 01i-rt-*rev-665
608 srfrv 01i-rt-*rev-666
609 srfrv 01i-rt-*rev-667
611 srfrv 01i-rt-*rev-669
612 srfrv 01i-rt-*rev-670
614 srfrv 01i-rt-*rev-672
615 srfrv 01i-rt-*rev-673
616 srfrv 01i-rt-*rev-674
618 bsurf 01i-rt-*ce-1375
```

If any database entities are being filtered with **DB Pick Mask**, the word **“**Active**”** will appear on the **DB Pick Mask** button as shown below.



2.4.4 Special Picking Controls

Several menu buttons are available for use in selecting multiple entities.



Scroll - The VCR buttons are described in Section 2.3.5.

Pick **All** - All pickable entities are picked.

Pick **None** - All pickable entities are un-picked.

Pick **Tog** - The status (either picked or not picked) of each pickable entity is reversed.

Page - All items on the current page of the Browser are toggled.

in Box - Use of the pick-in box for picking multiple entities at a time from the Display window is described in Section 2.4.1.2.

on Box - Use of the pick-on box for picking multiple entities at a time from the Display window is described in Section 2.4.1.2.

By Text - Pressing this button allows you to type in a text string and all items having your string in their Browser text will be picked.

get **1st** - When multiple pickable entities lie beneath the current cursor location the entity with the lowest entity number (i.e., the first one in numerical order) is highlighted. Pressing the ***x*** key moves to the next (in numerical order) pickable entity at the cursor location (default).

get **last** - When multiple pickable entities lie beneath the current cursor location the entity with the highest entity number (i.e., the last one in numerical order) is highlighted. Pressing the ***x*** key moves to the previous (in numerical order) pickable entity at the cursor location.

Set to **off** - This radio button causes Gridgen to un-pick the entities you select.

Set to **on** - This radio button causes Gridgen to pick the entities you select.

Set to **tog** - This radio button causes Gridgen to toggle the pick status of the entities you select (default).

Sort By - By default, entities are listed in the Browser in numerical order from first to last. The radio buttons labeled Sort By provide additional, context sensitive sort criteria to aid in the selection process by grouping like entities in the Browser. The Sort By **rev** toggle button provides even further control by allowing you to reverse the order of the current list. For example, in the **Set** Δ **Values** command you may choose to sort the break points by the numerical value of the spacing constraint so that all like-constrained connectors are grouped together. Also, you may choose to sort the Browser list by database entity type when using the Database **Delete** command.

Pick by Expression - Pressing this menu button allows you to type in a Glyph expression to select entities. For instance, if you have defined a Glyph variable, MyDomains, that is a list of domains you want to select, you can push this button, and then type `$MyDomains` or pick MyDomains from the list of variables in the Browser.

DB Pick Mask - This button sends you to the DB PICK MASK menu (see Section 2.4.3) which allows you to filter which database entities can be selected from the Browser.

Done - All picked entities will be processed by the command.

Abort - No entities are selected and control is passed back to the command that invoked the Browser.

2.5 Selecting File Names

To select a file name from the Browser follow this procedure:

1. Position the cursor over the file name text in the Browser. The text will be highlighted in salmon.
2. Press and release either the ***left*** or ***right mouse*** button. The text will be highlighted in white.
3. Press the **Open** button.

Gridgen's Input/Output commands require you to select the name of the file from/to which data will be read or written. This is done through the Browser window. When it is time for file name selection, Gridgen lists in the Browser all files within the current directory that match the default extension of the file type to be read or written. The current directory name is displayed in the Message window. List items correspond to either files or directories (denoted by a trailing slash). A list item is picked by moving the cursor over it at which time it will be highlighted (indicating that it's pickable), then pressing either the ***left*** or ***right mouse*** button. If the picked item is a file name it will be queued and highlighted in white. If the picked item is a directory it will become the current directory and a new list of file names will appear in the Browser.

Positioning of the file list in the Browser is controlled using VCR buttons (Section 2.3.5). Other commands of interest during file name selection include:

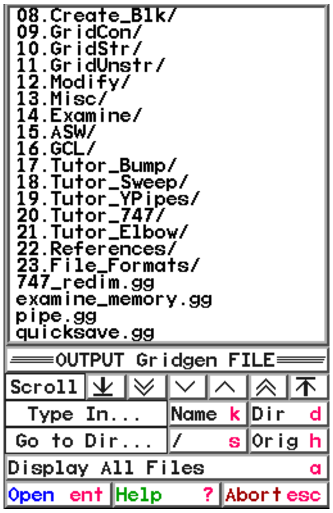
Type In . . . **Name** - This command prompts you to type the file name, including path if necessary.

Type In . . . **Dir** - This command prompts you to type in the name of the directory to make the current directory.

Go to Dir ***/*** - (Unix only) The root directory of the file system (/) will become the current directory when this button is pressed.

Go to Dir . . . **<drv>** - (Windows only) The drive letters of the computer will be displayed in the Blackboard when this button is pressed.

Go to Dir . . . **Orig** - The directory from which Gridgen was started will become the current directory when this button is pressed.

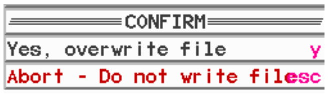


```
08.Create_Blk/
09.GridCon/
10.GridStr/
11.GridUnstr/
12.Modify/
13.Misc/
14.Examine/
15.ASW/
16.GCL/
17.Tutor_Bump/
18.Tutor_Sweep/
19.Tutor_YPipes/
20.Tutor_747/
21.Tutor_Elbow/
22.References/
23.File_Formats/
747_redim.gg
examine_memory.gg
pipe.gg
quicksave.gg
```

Disp Only .ext Files - This command will list in the Browser only the files with extensions that match the string `ext` where `ext` is chosen by Gridgen based on the file type being read or written. For example, the default extension for Gridgen files is `.gg`. When working with database files the Browser will list all files with the default extensions of all database file types.

Disp All Files - This command will list in the Browser all files in the current directory regardless of their extension.

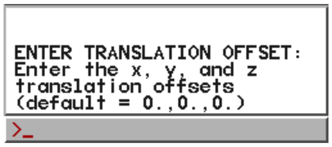
When the correct file name is queued, you press the **Open** button in the menu window. If the file is to be read, reading commences. If the file is to be written, the file name is checked against those in the current directory. If Gridgen determines that you will over-write an existing file, it will present a new menu and ask you to confirm the choice of file name.



2.6 Text Input

You will need to communicate with Gridgen at times by entering data directly through the keyboard. For example, when translating a domain, you may choose to type in the translation offset. In these cases the Text Input window will be highlighted, and the Message window will prompt you for a response. You may type in the desired response, which may be either a character string, real number(s), integer number(s), or a Tcl expression. In each case, you terminate your input by pressing the ***enter*** key.

Gridgen stores previous text entries. They can be accessed and reused or edited using the hot keys described below.



```
ENTER TRANSLATION OFFSET:
Enter the x, y, and z
translation offsets
(default = 0.,0.,0.)
>_
```

Hot keys of interest during text input include:

^***u*** - erase the entire entry

escape - abort from text entry

enter - terminate text entry

→ (right arrow key) - scroll to the right through existing text

← (left arrow key) - scroll to the left through existing text

↑ (up arrow key) - scroll backward through the list of previous text entries

↓ (down arrow key) - scroll forward through the list of previous text entries

Tab - automatically complete partially typed variable name

Text input that consists of multiple "words" (for example, entering a point's `(x,y,z)` coordinates) can be delimited using either spaces or commas. For example, the point `(1,2,3)` may be entered by typing either `1,2,3` or `1 2 3` or `1, 2, 3`. Note that decimal points do not have to be entered when typing real numbers (in other words `1` is the same as `1.0`). Furthermore, note that the default values for a multiple word text input can be used by taking advantage of the delimiters. For example, if the default value for a point being entered is `(2,4,6)` and you wish simply to change `z` to `8`, you could enter `,,8` at the text prompt.

Text input can also be a Glyph expression. When you enter text input mode, the Browser will display the Glyph variables you have already defined. You can pick variables from the browser to use in expressions or type them directly in the Text Input window. You can turn off the display of variables using the **Hide Variables** button Please note that, if variables are defined, they will appear in the Browser window when in text input mode regardless of what menu the text input prompt occurs.

2.7 Image Manipulation

Manipulation of the Display window image is controlled by three factors: the axis system in which you want to manipulate the image (either the body or screen axes), the state of the **Glide** button in the DISPLAY COMMANDS menu, and the device you use to affect the manipulation (either the mouse or keypad keys).

2.7.1 Axis Systems

Gridgen works with two axis systems: the *body axes* and the *screen axes*.

The body axis system is the three dimensional coordinate system in which your database and grid are defined. The body axis system may be visualized by enabling the display of the XYZ Axes which are drawn in blue and located at the origin of the body axis system. The rotation axes, which are parallel to the body axes, are drawn in red in the Display window at the rotation point and are fixed to your grid; as the grid rotates the body axes rotate also.

The screen axes are not drawn in the Display window because, as the name implies, they are always aligned with the screen of your computer's monitor. The x coordinate of the screen axis system (screen-x) always points to the right, screen-y always points up, and screen-z always points out of the screen.

2.7.2 Glide Toggle

The **Glide** toggle button in the DISPLAY COMMANDS menu determines how mouse movement controls image movement according to the following two states.

1. Mouse movement imparts a rate of movement (velocity) to the image.
2. Mouse movement displaces the image.

The default state of this toggle button is 1, mouse movement gives the image a velocity. This implies that as long as the mouse button is pressed the image will continue to move even when the mouse is not moving. In state 2, the image moves only when the mouse moves.

2.7.3 Manipulation Device

You may use either the *mouse* or the *keypad keys* in conjunction with the *control key* to manipulate the Display window image. Note that the effect of the control key is to swap the roles of the mouse and keypad keys with respect to rotation and panning and zooming. By default, the mouse controls panning and zooming and the keypad keys control rotation. With the control key pressed, the mouse controls rotation, and the keypad keys control panning and zooming.

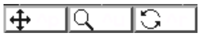
Image Manipulation Devices

device	component (default)	component (with control key)
left mouse	screen-x and screen-y panning	screen-x and screen-y rotation
middle mouse	screen-z zooming	screen-z rotation

Image Manipulation Devices

device	component (default)	component (with control key)
Num Lock	body-x rotation	body-x translation
Pad /	body-y rotation	body-y translation
Pad *	body-z rotation	body-z translation
Pad 7	screen-x rotation	screen-x panning
Pad 8	screen-y rotation	screen-y panning
Pad 9	screen-z rotation	screen-z zooming

Note that during image panning the cursor is changed to four arrows, during zooming the cursor is changed to a magnifying glass, and during rotation it changes to two circular arrows, as shown below.



2.7.4 Manipulation Hot Keys

Several hot keys are available to assist you with image manipulation.

Pad -*, *Pad + - These two hot keys, when pressed in conjunction with other keypad image manipulation keys, will reverse the direction of the rotation or panning.

Page Up - This so-called "turbo key" increases the speed of image manipulation by three times when pressed in conjunction with an image manipulation device.

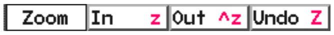
Page Down - When pressed in conjunction with an image rotation key, the image is rotated by 90 degrees.

Alt - When pressed in conjunction with an image manipulation key, the image moves in single step increments.

2.7.5 Zoom Box

Gridgen's zoom box provides an alternate method for enlarging or shrinking the Display window image. It allows you to specify explicitly either a region of the Display window that will be expanded to fill the entire Display window (zoom in) or a region of the Display window into which the entire Display will be compressed (zoom out).

The zoom-in and zoom-out boxes may be activated using the ***z*** and **^*z*** hot keys, respectively. There are also Zoom **In** and Zoom **Out** buttons on the DISPLAY COMMANDS menu (Section 2.9). In either case you specify the extents of the zoom box, first by positioning the upper left corner (marked with a small open square) to the desired screen location, and then by pressing and holding down the ***mouse*** and stretching the lower right corner of the box (marked with a small open square) to the other location. When the ***mouse*** is released the zoom box is complete. When zooming in, the zoom box is labeled Zoom In and the effect is to redraw the screen so that everything inside the zoom box fills the entire Display window. When zooming out, the zoom box is labeled Zoom Out and the effect is to redraw the screen so that everything in the Display window fills the zoom box at the zoom box's location.



Zooming can be aborted by pressing the ***Esc*** key. The zoom box works best when the viewing transformation is orthonormal (use the ***v*** hot key to toggle between orthonormal and perspective viewing transformations).

The Zoom **Undo** button (activated with the ***Z*** hot key) allows you to return the image to its position immediately prior to the last zoom-in or zoom-out. You can think of it as an additional saved orientation.

2.8 The 3D Cursor

Many Gridgen commands require you to select or define points in 3D space. For example, a curve is drawn by placing several control points through which the curve will pass. Whenever control points need to be defined the 3D cursor appears as a small white cross in the Display window. It will be referred to as “the cross” in the discussion to follow (and also in the help messages that appear in the Message window). Do not confuse the white cross 3D cursor with the upper-left pointing arrow mouse cursor.

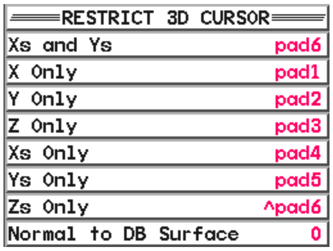
The location of the cross is controlled with the mouse and the ***right mouse*** button. Position the arrow cursor anywhere in the Display window and hold down the ***right mouse*** button. The red arrow cursor will disappear, and mouse movement will control the cross’s movement. The 3D coordinates of the cross will be displayed in the Blackboard.

2.8.1 3D Cursor Movement

Horizontal mouse movement is translated into horizontal (screen-x) 3D cursor movement and vertical mouse movement is translated into vertical (screen-y) 3D cursor movement. Furthermore, the 3D cursor is situated in a plane of constant screen-z called the *drawing plane*. The drawing plane’s depth (screen-z coordinate) is set implicitly by Gridgen as you place control points.

Because control of the 3D cursor actually takes place in the screen coordinate system (by default), the orientation of the body axis system relative to the screen axis system affects how the 3D cursor moves in 3D space. For example, when the body axes are in their default orientation (body-x aligned with screen-x and body-y aligned with screen-y), horizontal mouse movement will move the 3D cursor in the body-x direction and vertical mouse movement will move the 3D cursor in the body-y direction. If you rotate the image so that body-z is aligned with screen-x and body-x with screen-y, then horizontal and vertical mouse movement will move the cross in the body-z and body-x directions, respectively. Finally, if the body axis system is oriented such that it is not aligned at all with the screen axis system, then mouse movement will result in true 3D cross movement, with all three Cartesian components of the cursor position changing.

The cross movement can be restricted using the **Movement Restrictions** button. It will invoke a menu allowing you to select how the movement is to be restricted. The screen-x and screen-y option (**Xs and Ys**) is the same as the default movement. Choosing any of the other options will allow only that coordinate to be changed as the mouse is moved from left to right. Movement restrictions are reset with the start of every segment when creating connectors.



Xs and Ys - 3D cursor is free to move in screen-x and screen-y.

X Only - The 3D cursor moves in body-x only.

Y Only - The 3D cursor moves in body-y only.

Z Only -The 3D cursor moves in body-z only.

Xs Only - The 3D cursor moves in screen-x only.

Ys Only - The 3D cursor moves in screen-y only.

Zs Only - The 3D cursor moves in screen-z only.

Normal to DB Surface - The 3D cursor moves in the direction defined by the normal vector of the database surface on which the previous control point was defined.

The keypad keys listed below may be used to restrict the 3D cursor's movement also.

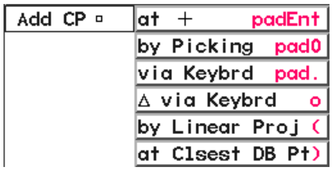
Movement Restriction Hot Keys

hot key	3D coordinate restriction	2D (database) coordinate restriction
Pad 1	body-x	u
Pad 2	body-y	v
Pad 3	body-z	u,v
Pad 4	screen-x	screen-x
Pad 5	screen-y	screen-y
Pad 6	screen-z	screen-x, screen-y
0	normal to database	n/a

If the current Gridgen command involves moving the cursor on a database entity (for example, creating a Line on DB segment), then mouse movement translates into cursor movement in a slightly different way. Horizontal and vertical mouse movement will still cause screen-x and screen-y movement of the cursor. However, the cursor will be further constrained to move along the database shape.

2.8.2 Control Point Creation

The 3D cursor's purpose is to allow you to create control points. There are several places in Gridgen where this takes place: for example, segment creation and database point creation. As the 3D cursor is moved across the screen you will be tasked with specifying control point locations. There are six methods of specifying a control point: adding a point at the current 3D cursor location, picking an existing point, numerically entering the coordinates of the point, numerically entering the offset from the current point, casting a ray from the current 3D cursor location onto the database, and finding the closest point on the database to the current 3D cursor location.



2.8.2.1 At Cross

Select the Add CP □ **at Cross** button to enter a control point at the current cross location. The ***Pad Enter*** key performs the same command. When a point is added, a small open square is drawn at the point. The Blackboard window displays the coordinates of the current cursor location. This method of control point placement is best suited for control points that do not have to be positioned precisely.

2.8.2.2 By Picking

A second method of choosing a control point is by anchoring it to the position of an existing database point or a node. To use this option, position the cross over the node or database point to which the new control point is to be equated and hit the ***Pad 0*** key or select the Add CP □ **by Picking** button. If Gridgen detects a point beneath the cross (it checks for hits within a ten pixel cylinder around the cursor), a control point will be added at that point; otherwise the program will display a message indicating that no point was picked (so no control point was added). Picking searches for hits from (in order): nodes, control points, database entities, and domain points.

When you are creating a database constrained segment and add a point by picking, Gridgen first tries to pick from the current database entity. If Gridgen does not find any points on the current database entity it will try to pick other points, returning the first hit found. If it finds a pickable point that is not on the current database entity, then Gridgen will inform you, project the picked point onto the current entity, and add it to the segment.

Wherever possible, you should add points by picking because this will ensure that components of your grid match end to end.

2.8.2.3 Via Keyboard

A third method of choosing a control point is selected with the Add CP □ **via Keyboard** button (or by hitting the ***Pad Period*** key). A prompt appears in the Message window, and the program waits for the numerical input to be typed in the input window. You will be expected to input `(x,y,z)` coordinates if the current segment is not database constrained and parametric `(u,v)` coordinates if the segment is surface constrained. Default values are the current cursor location, so entering nothing is the same as using the Add CP □ **at Cross** button. It is not necessary to type all components of a coordinate unless it is to be different from the default. For example, to change only the `z` coordinate (say, to `2.2`), simply enter`,,2.2` (note the commas). The `x` and `y` coordinate values will be set to the cross's `x` and `y` values.

2.8.2.4 Via Offset

A fourth method of choosing a control point is selected with the Add CP □ **via Offset** button. A prompt appears in the Message window, and the program waits for the numerical input to be typed in the Text Input window. You will be expected to input `(Δx, Δy, Δz)` values if the current segment is not database constrained and parametric `(Δu, Δv)` values if the segment is surface constrained. The offsets are measured relative to the current cross location.

2.8.2.5 By Linear Projection

You can add a control point on the database where no pickable database point exists using the Add CP □ **by**

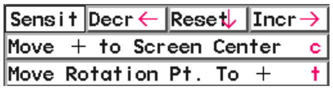
Linear Proj command. A ray is cast in the screen-z direction, through the current cursor location, and into the database. The control point is added if and where the ray intersects the database.

2.8.2.6 By Closest Point Projection

You can also add a control point on the database where no pickable database point exists using the Add CP □ **at Closest DB Pt** command. When this button is pressed, Gridgen adds a control point at the point on the database that is closest to the current cursor location.

2.8.3 Miscellaneous Controls

Various buttons (listed below) allow you to fine tune control of the cross during 3D cursor manipulation.



Sensit **Decr** - The sensitivity of the cross's response to mouse movement is decreased by a factor of ten.

Sensit **Reset** - The sensitivity of the cross's response to mouse movement is re-initialized.

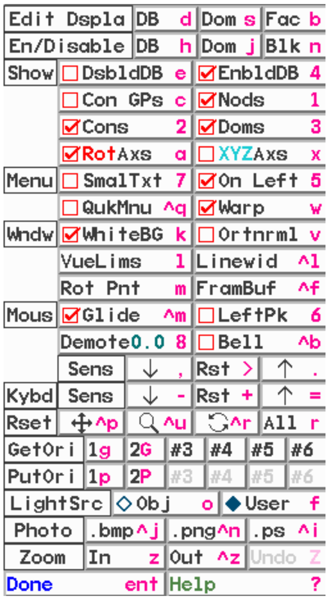
Sensit **Incr** - The sensitivity of the cross's response to mouse movement is increased by a factor of ten.

Move + to Screen Center - The cross will be positioned at the center of the Display window (at the zoom dot).

Move Rotation Pt. to + - The rotation point (the graphical origin of the body-axis system rendered in the Display window) is moved to the current cross cursor location. Thereafter, all image rotations using the keypad take place about this point. The rotation point is reset (to the centroid of the viewport containing the grid and database) whenever the original view orientation (body-x aligned with screen-x, body-y aligned with screen-y) is restored by pressing the ***r*** hot key.

2.9 Display Commands Menu

This menu will be displayed when the **Display Commands** button, which is available from almost every menu in Gridgen, is pressed.



Edit Dspla - These three commands (**DB**, **Dom**, **Fac**) allow you to change the display of database entities, domains, and block faces, respectively (Section 2.10).

En/Disable - These three commands (**DB, Dom, Blk**) allow you to enable and disable database entities, domains, and blocks, respectively (Section 15.6).

Show **DsbldDB** - This toggle button controls whether or not disabled database entities will be rendered in the Display window. Not displaying disabled entities reduces the visual clutter in the Display window. By default this toggle is off; disabled database entities will not be displayed.

Show **EnbldDB** - This toggle button allows you to easily turn off the display of all database entities without having to disable them. By default this toggle is enabled. When it is disabled, enabled database entities are not drawn while in the DISPLAY COMMANDS menu and are drawn in a sparse mode elsewhere.

Show **Con GPs** - This toggle button controls the display of grid points on connectors. By default this toggle is off. When this toggle is on the grid points on all connectors will be rendered as small filled circles in the connector color.

Show **Nods**, **Cons**, **Doms** - These three toggle buttons allow you to turn on and off the display of nodes, connectors, and domains, respectively. By default each of these three toggles is on. By turning off the **Nods** and **Cons** toggles you can make a very nice picture of just the grid. However, **Nods** and **Cons** only affect the display while in the DISPLAY COMMANDS menu. On the other hand, **Doms** remains in effect throughout the entire code allowing you to easily toggle between a nice rendering for visualization of surface grids and a sparse rendering for ease in image manipulation.

Show **Rot Axs** - This toggle button controls the display of the rotation axes in the Display window. By default this toggle is on. The rotation axes are drawn in red with their origin at the center of the grid image; they are not necessarily rendered at the `(0,0,0)` point of the body axis system.

Show **XYZ Axs** - This toggle button controls the display of the xyz axes in the Display window. By default this toggle is off. The xyz axes are drawn in blue and centered at the `(0,0,0)` point of the body axis system.

Menu **SmalTxt** - This toggle button controls the point size of the text used in Gridgen's menus. The normal text is approximately 14 point. The small text available by toggling this switch on is approximately 8 point. By default this toggle is off.

Menu **On Left** - This toggle button allows you to choose the side of the screen on which Gridgen draws the Version, Status, Text Input, Blackboard, and Menu windows. By default this toggle is enabled (the menu is on the left).

Menu **QukMnu** - This toggle button indicates whether or not the Quick menu (Section 2.3.7) is displayed. By default this toggle is off. When the Quick menu is not displayed the Quick Keys (Section 2.3.6) are displayed instead.

Menu **Warp** - This toggle button allows you to disable Gridgen's "mouse warping" whereby the cursor is sometimes moved for you to certain buttons. By default this toggle is enabled.

Wndw **WhiteBG** - This toggle button will change the background color in the Display and Browser windows between white and black. The default background color is black.

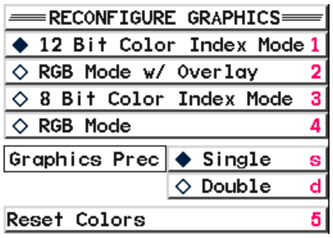
Wndw **Ortnrml** - This toggle button changes the Display window view between perspective and orthonormal viewing transformations. In a perspective transformation, the image will be rendered with a vanishing point so that distant parts of the image (in the screen-z direction) will appear smaller than close parts. An orthonormal transformation does not scale the image based on distance. When using the zoom box (Section 2.7.5) we recommend that you use an orthonormal transformation.

Wndw **VueLims** - This command tells Gridgen to compute the size of the 3D viewport for the objects in the Display window. The current viewport is based on the dimensions of the displayed objects. As new items are added to the display, it may be necessary for you to reset the viewing transformation using this command to take into account the new size of the total image. If you notice that portions of your grid image are being "clipped", as though an invisible plane is slicing through the image, then use this command. The rotation axes will be placed at the center of the 3D viewport after the new view limits are computed.

Wndw **Linewid** - This command controls the thickness with which connectors and database entities are drawn (default is 1). Values between 1 and 5 may be selected.

Wndw **Rot Pnt** - By default, the rotation point is at the center of the 3D viewport. By pressing the **Rot Pnt** button the rotation point can be moved to any visible point in the Display window, including grid points simply by picking it with the mouse. The rotation point is reset whenever the original orientation is restored using the ***r*** hot key. The rotation point may also be set at any time by holding the ***control*** key and then picking a point with the mouse.

Wndw **FramBuf** - This command lets you change Gridgen's frame buffer configuration. The availability of these configurations depends on your computer's graphics hardware which Gridgen determines at start-up. At start-up Gridgen chooses what it considers to be the frame buffer configuration that will perform the best. You may change the frame buffer configuration to one of these four options.



The two buttons labeled Graphics Prec let you specify the numerical precision of the graphics calculations. By default Gridgen's graphics are **Single** precision (4 byte real numbers). However, if the Display window image seems to vibrate when you zoom in very closely on a grid with a large range of length scales (for example, very tight grid point spacing on a very large body) you will benefit from enabling **Double** precision graphics (8 byte real numbers). However, keep in mind that double precision graphics also doubles the size of Gridgen's graphical objects and the memory used by them.

The **Reset Colors** button in this menu will re-initialize the color map to the proper Gridgen settings if it should become corrupted (by another application perhaps).

Mous **Glide** - This toggle button controls whether or not the Display window image will continue to move (glide) even after mouse motion has stopped as long as the mouse button is pressed. By default this toggle is on. See Section 2.7.2.

Mous **LeftPk** - This toggle button allows you to change which mouse button Gridgen uses for Display window item picking. Gridgen uses the ***right mouse*** button for picking by default. Turning on the **LeftPk** toggle causes the ***left mouse*** button to be used for picking - the ***right mouse*** button is then used for image panning.

Mous **Demote** - If the value of **Demote** is non-zero the Display window image is drawn in a sparse mode (demoted) during image manipulation. Sparse mode turns off the display of domain grid points and draws database entities as outlines. The **Demote** value, in the range between `0` and `10`, represents the approximate time in seconds between your release of the image manipulation device (letting go of the mouse button) and the redraw of the Display window. A large value of drawing lag prevents the image from being redrawn unnecessarily between successive manipulations. If you set **Demote** to 10, the Display window image will only be redrawn in full mode when you press the ***space bar***.

Mous **Bell** - This toggle button changes the status of the “bell” between an audible tone and a flash of the Version window in the upper left corner of the screen. The bell rings whenever a button is pressed; it is off (visual bell) by default.

Mous Sens ↓, **Rst**, ↑ - These buttons allow you to decrease, reset, and increase (respectively) the sensitivity of the mouse with regard to image manipulation.

Kybd Sens ↓, **Rst**, ↑ - These buttons allow you to decrease, reset, and increase (respectively) the sensitivity of the keyboard with regard to image manipulation.

Rset - The buttons in this row allow you to un-pan, un-zoom, and un-rotate the Display window image (Section 2.3.7). The **All** button will un-pan, un-zoom, and un-rotate the Display window image at once.

GetOri - Buttons **1** through **6** in this row allow you to save (**Get**) up to six orientations of the Display window image. Note that there are no hot keys for orientations 3-6. These six orientations will be saved in your Gridgen file.

PutOri - Buttons **1** through **6** in this row allow you to recall (**Put**) the six orientations of the Display window image saved by the **Get Ori** command. Note that there are no hot keys for orientations 3-6.

LightSrc -The **Obj** and **User** radio buttons set the type of light source used when rendering shaded surfaces. **User** fixes the light source at your location; in other words, light rays are cast in the screen-z direction. This means that surfaces will appear lighter in color where the surface normal is parallel to screen-`z` and darker in color where the local normal is nearly perpendicular to screen-`z`. On the other hand, **Obj** fixes the position of the light source with the object so that as the image is rotated the relative shading depends on the light source and the viewpoint.

Photo - The **.rgb**, **.png, .bmp** and **.ps** buttons in this row are used to save the Display window image in a file (Section 2.11), either Silicon Graphics RGB (IRIX only), Portable Network Graphics, Window Bitmap (Windows only) or PostScript formats.

Zoom - The **In** and **Out** buttons in this row are used to invoke the zoom-in and zoom-out boxes, respectively (Section 2.7.5). The **Undo** button restores the image’s orientation prior to the last zoom-box.

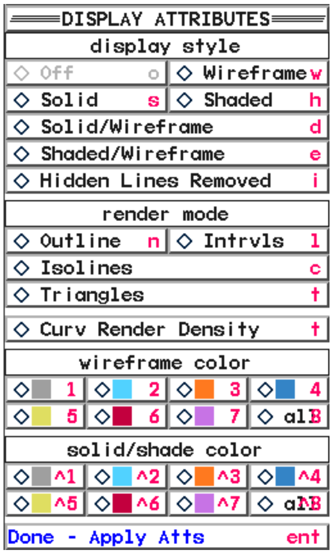
2.10 Display Editing

To change the display of grid or database entities follow this procedure:

1. Press the **Display Commands** button.
2. Choose the command corresponding to entity type you wish to edit via the row of buttons labeled Edit

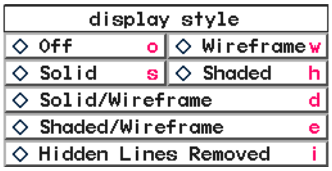
Dspla: **DB** (database entities), **Dom** (domains), or **Face** (block faces).

3. Select the entity or entities to edit. Display style is an option for sorting the Browser list.
4. Select attributes from the DISPLAY ATTRIBUTES menu.
5. Press the **Done** button. The selected attributes will be applied to the selected entities.
6. Repeat as necessary. Press the **Abort** button to return to the DISPLAY COMMANDS menu.



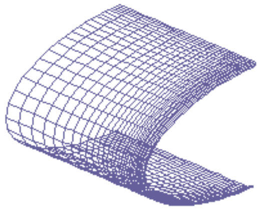
The following sections describe each category of display attribute that may be applied to surface grids and database entities

2.10.1 Display Style

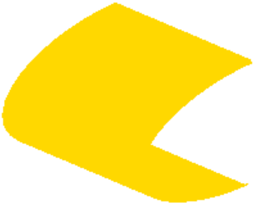


Off - The entity will not be displayed. For domains and block faces this turns off the display of interior grid points but the bounding connectors will continue to be displayed. Off is not a display style for database entities. To eliminate the display of database entities first disable them (Section 15.6) and then disable the **Disabld-DB** toggle (Section 2.9) in the DISPLAY COMMANDS menu.

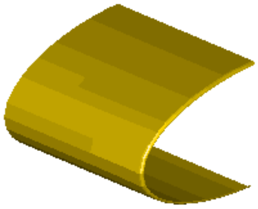
Wireframe - The entity will be rendered as a wireframe (see below).



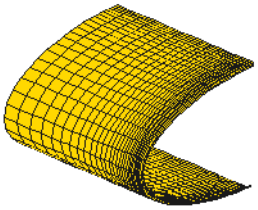
Solid - The entity is rendered as an opaque surface of a single color without light source shading (see below).



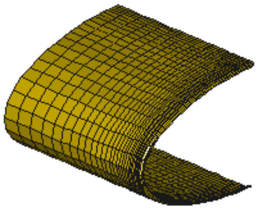
Shaded - The entity is rendered as a single color opaque surface with lightening and darkening based primarily on the angle between the surface normal and the light source (see below).



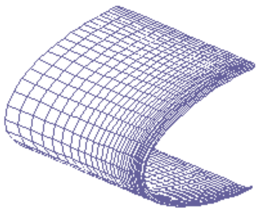
Solid/Wireframe - Solid and Wireframe display styles are superimposed. If the solid/shade color and the wireframe color are the same, the wireframe will be drawn in the background color for improved visibility (see below).



Shaded/Wireframe - Shaded and Wireframe display styles are superimposed. If the solid/shade color and the wireframe color are the same, the wireframe will be drawn in the screen background color for improved visibility (see below).

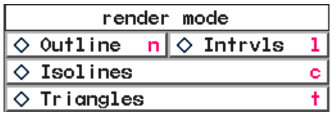


Hidden Lines Removed - The entity surface is rendered as a wireframe model with all hidden wireframe lines removed from display (see below).



2.10.2 Render Mode

Render mode is for use with database entities only and lets you specify how many curves are used to render a database surface.



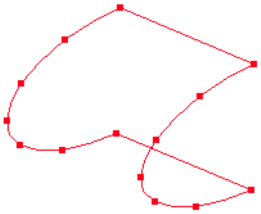
render mode

◇ Outline n ◇ Intrvls l

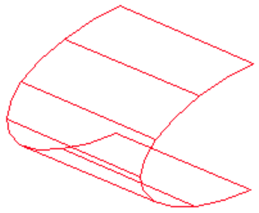
◇ Isolines c

◇ Triangles t

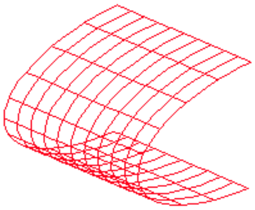
Outline - For database entities, outline mode implicitly sets the number of render curves (Section 2.10.3) in both the u and v directions to 2 and renders the entity’s boundaries. Also, the corner of each interval on the surface is rendered with a small filled square to aid in identification for picking (see below).



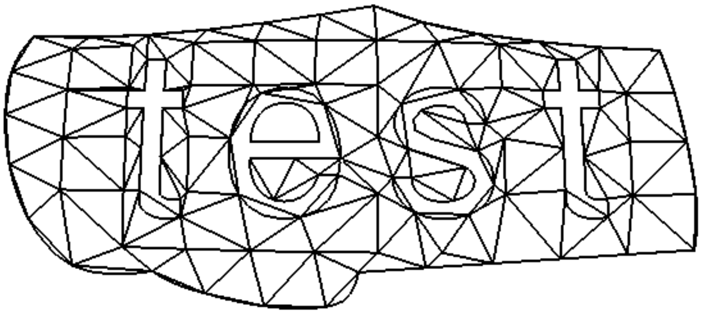
Intrvls - For database entities, Interval mode implicitly sets the number of u and v render curves so that the outline of each of the entity’s intervals is rendered. Interval mode is not available for trimmed surface entities (see below).

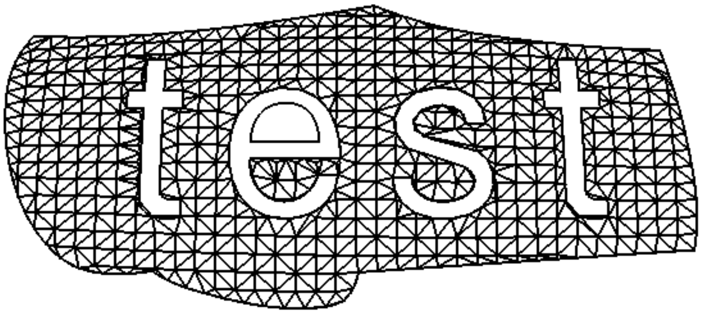


Isolines - For database entities, this mode allows you to specify explicitly the number of u and v render curves. The illustration below shows an `11 x 11` isoline rendering. Isolines may be applied to trimmed surfaces as well as normal database surface entities.



triangles - For trimmed surface database entities only, the graphical object is a triangulated surface. You specify the approximate number of triangular facets to use along the entity's longest coordinate extent. The figures below show a contrived trimmed NURB surface rendered with 15 triangles (the default) above, and 51 triangles below. Entering a value of 1 creates an optimal arrangement of triangles.





2.10.3 Render Density

There are two variations of the render density attributes: one for grids and one for databases.

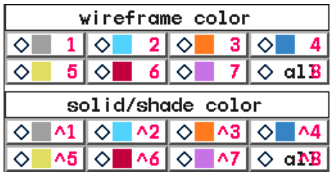
◇ Grid Render Density c

Grid Render Density - For domain and block face entities, you may specify the display increment for `i` and `j` lines in the grid using this button. You may specify a pair of values, each between `0` and `9` for the `i` and `j` increment, respectively. When the grid is rendered every nth `i` or `j` constant grid line will be drawn. Setting the increment to zero will display only the boundary (in other words, the connectors).

◇ Curv Render Density t

Curv Render Density - This command allows you to specify the number of points used to render the u and v constant curves, respectively. By default, 25 points are used on both the u- and v-constant curves. For linear curves and surfaces Gridgen always places render points at the corner of each interval on the surface regardless of the value set here. For other entities Gridgen will adjust the number of render points so that u and v constant render lines cross at a common point. Entering a value of 1 results in an optimal density of points.

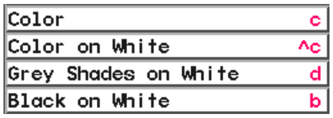
2.10.4 Color



wireframe color, **solid/shade color -** A total of seven colors are available for the display of entities - the default database color and the six default grid colors (one for each face of any potential 3D structured block). These buttons allow you to specify the colors used to render the picked entities. The **All** button is used on database and domain entities to assign an alternating color from a six color sequence to those entities queued for display modification.

2.11 Saving Screen Images

The Display window image may be saved to either a Silicon Graphic's RGB screen dump (on sgi systems only), a Windows BMP (on Windows systems only), a Portable Network Graphics image, or a PostScript line drawing via the Photo buttons in the DISPLAY COMMANDS menu (Section 2.9). Note that the image will consist of the contents of the Display window only. Gridgen's other windows such as the Menu Button window and Blackboard are not included. PostScript files from Gridgen contain vector images and not raster images like the other formats. Therefore they are best for line drawings. Several options are available for customizing the image as described below.



Color - An image of the Display window as you see it is created.

Color on White - Same as **Color** except that the background is converted temporarily to white.

Gray Shades on White - The colors of the current Display window image will be converted to a gray scale and the background color will be converted to white.

Black on White - With this option, the background will be changed to white, and all other colors will be converted to black.

After selecting one of the options, you are placed in the File Browser (Section 2.5) to select the name of the image file to be exported. After the picture file is created, the program will return to the DISPLAY COMMANDS menu.

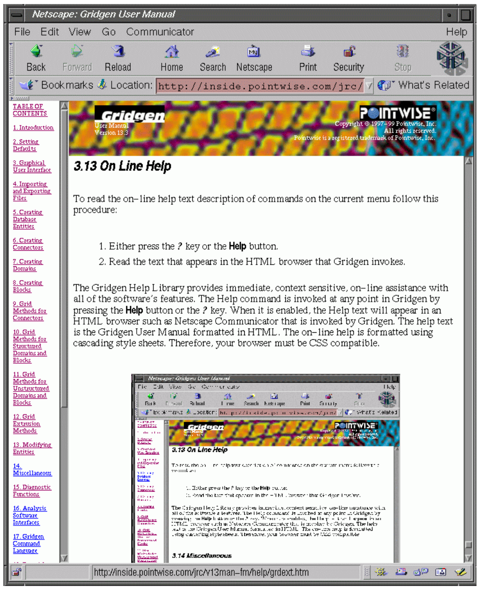
2.12 On Line Help

To read the on-line help text description of commands on the current menu follow this procedure:

1. Either press the ***?*** key or the **Help** button.

2. Read the text that appears in the HTML browser that Gridgen starts.

The Gridgen Help Library provides immediate, context sensitive, on-line assistance with all of the software's features. The Help command is invoked at any point in Gridgen by pressing the **Help** button or the ***?*** key. When it is enabled, the Help text will appear in an HTML browser such as Netscape Communicator that is invoked by Gridgen. The help text is the Gridgen User Manual formatted in HTML. The on-line help is formatted using cascading style sheets. Therefore, your browser must be CSS compatible.


Netscape: Gridgen User Manual
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://inside.pointwise.com/jrc/ What's Related
TABLE OF CONTENTS
1. Introduction
2. Setting Defaults
3. Graphical User Interface
4. Importing and Exporting Files
5. Creating Database Entities
6. Creating Connectors
7. Creating Domains
8. Creating Blocks
9. Grid Methods for Connectors
10. Grid Methods for Structured Domains and Blocks
11. Grid Methods for Unstructured Domains and Blocks
12. Grid Extrusion Methods
13. Modifying Entities
14. Miscellaneous
15. Diagnostic Functions
16. Analysis Software Interfaces
17. Gridgen Command Language
Gridgen
User Manual
Version 13.3
POINTWISE®
Copyright © 1997–99 Pointwise, Inc.
All rights reserved.
Pointwise is a registered trademark of Pointwise, Inc.
http://inside.pointwise.com/jrc/v13man–fm/help/grdext.htm

3.13 On Line Help

To read the on–line help text description of commands on the current menu follow this procedure:

1. Either press the *?* key or the **Help** button.
2. Read the text that appears in the HTML browser that Gridgen invokes.

The Gridgen Help Library provides immediate, context sensitive, on–line assistance with all of the software's features. The Help command is invoked at any point in Gridgen by pressing the **Help** button or the *?* key. When it is enabled, the Help text will appear in an HTML browser such as Netscape Communicator that is invoked by Gridgen. The help text is the Gridgen User Manual formatted in HTML. The on–line help is formatted using cascading style sheets. Therefore, your browser must be CSS compatible.

2.13 Miscellaneous

Pressing the ***control*** key in conjunction with the ***right mouse*** button lets you pick a new rotation point. The coordinates of the new rotation point are displayed in the Message window.

Pressing the ***shift*** key in conjunction with the ***right mouse*** button lets you query a picked point so that its coordinates are displayed in the Message window. Furthermore, the distance between the current point and the previously picked point will also be displayed.

The ***space*** hot key causes a redraw of the Display window.

Gridgen attempts to trap fatal errors before they cause a code failure. You will receive a warning in the Message window that an error occurred. You should then save your work to a file, exit Gridgen, then restart Gridgen with your last saved file and continue.

3. Setting Defaults

The **Defaults** Main menu command allows you to specify your preferences for a variety of grid attributes. Except for the four tolerances, these defaults are not saved in the Gridgen file. We recommend you utilize a Glyph rc file (Section 1 in the Glyph Reference Manual) in order to have a consistent set of defaults each time you run Gridgen. When you enter the Defaults menu the current default settings are displayed in the Blackboard window.

3.1 Connector Defaults

3.1.1 Dimensions

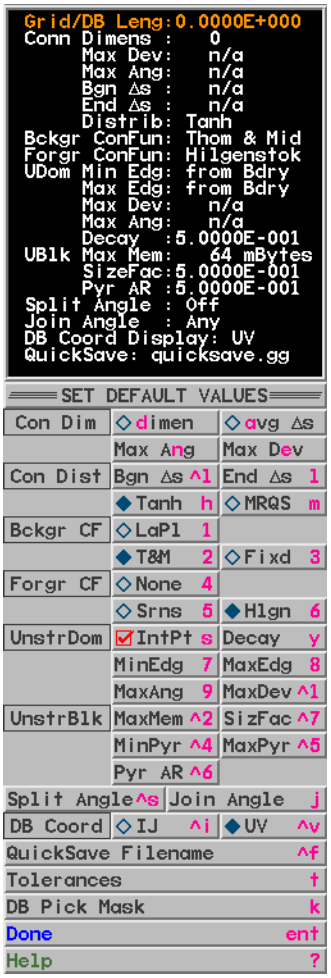
The connector dimension default is the number of grid points that will be applied automatically to new connectors. The dimension and average spacing defaults (Section 3.1.1 and Section 3.1.2) are mutually exclusive. The initial value of this default is `0`; new connectors will be un-dimensioned. To change the value of the default press the Con Dim **dimen** button and type the new value in the Text Input window.

Setting a default dimension is especially useful when used in conjunction with the **On DB Entities** and **2 Point Connector** connector creation commands (Section 7.4 and Section 7.3, respectively) because it allows a large number of connectors to be created and dimensioned at one time. The same holds true for the **On DB Entities** domain creation command (Section 9.6).

3.1.2 Average Spacing

The connector average spacing default is the grid point spacing that will be applied automatically to new connectors. The average spacing and connector dimension defaults (Section 3.1.2 and Section 3.1.1) are mutually exclusive. When the average spacing default is set, all new connectors will be dimensioned so grid points are spaced along them at arc length increments approximately equal to the value of the average spacing default. The initial value of this default is `0.0`, meaning this default will not be applied. To change the value of the default press the Con Dim **avg Δs** button and type the new value in the Text Input window.

Setting the average spacing is very useful when creating unstructured grids because it allows a large number of connectors to be created and dimensioned at one time when used with the **On DB Entities** (Section 7.4) and **2 Point Connector** (Section 7.3) commands. The same holds true for the **On DB Entities** domain creation command (Section 9.6).



```
Grid/DB Leng:0.0000E+000
Conn Dimens :   0
     Max Dev:   n/a
     Max Ang:   n/a
     Bgn Δs :   n/a
     End Δs :   n/a
     Distrib: Tanh
Bckgr ConFun: Thom & Mid
Forgr ConFun: Hilgenstok
UDom Min Edg: from Bdry
     Max Edg: from Bdry
     Max Dev:   n/a
     Max Ang:   n/a
     Decay  :5.0000E-001
UBlk Max Mem:   64 mBytes
     SizeFac:5.0000E-001
     Pyr AR :5.0000E-001
Split Angle : Off
Join Angle  : Any
DB Coord Display: UV
QuickSave: quicksave.gg
```

3.1.3 Maximum Angular Deviation

The maximum angular deviation default is the maximum turning angle in degrees allowed between successive grid points on a connector. If you set a value here it is applied as follows:

1. You create a new connector (Section 7).
2. If neither the Dimension (Section 3.1.1) nor Average Spacing (Section 3.1.2) defaults are set the process is complete and the connector has no grid points.
3. If either the Dimension (Section 3.1.1) or Average Spacing (Section 3.1.2) default is set it is applied to the connector.
4. If the Maximum Angular Deviation default is set grid points are added to the connector until the specified limit is satisfied.

The initial value for this default is `0`, meaning it is not enabled. To set this default press the Con Dim **Max Ang** button and type in the new value in degrees.

3.1.4 Maximum Chord Height Deviation

The maximum chord height deviation default is the maximum distance allowed between a connector's analytical shape (defined by its segments) and its discrete shape (defined by its grid points). The deviation is computed between the segment's analytical definition and the midpoint of the line between two successive grid points. This default is applied as follows.

1. You create a new connector (Section 7).
2. If neither the Dimension (Section 3.1.1) nor Average Spacing (Section 3.1.2) defaults are set, the process is complete and the connector has no grid points.
3. If either the Dimension (Section 3.1.1) or Average Spacing (Section 3.1.2) default is set, it is applied to the connector.
4. If the Maximum Chord Height Deviation default is set grid points are added to the connector until the specified limit is satisfied.

The initial value for this default is `0`; it is not enabled. To set this default press the Con Dim **Max Dev** button and type in the new value.

3.1.5 Beginning Clustering

The beginning clustering default is the spacing constraint (Section 11.3.2.1) that will be applied to the beginning of new connectors. The initial value of this default is `0.0`; the spacing will be unconstrained. To change the value of this default, press the Con Dist **Bgn ∆s** button and type the new value in the Text Input window.

3.1.6 Ending Clustering

The ending clustering default is the spacing constraint (Section 11.3.2.1) that will be applied to the ending of new connectors. The initial value of this default is `0.0`; the spacing will be unconstrained. To change the value of this default press the Con Dist **End ∆s** button and type the new value in the Text Input window.

3.1.7 Distribution Function

The distribution function default is the function (Section 11.3.2.2) that will be applied to distribute grid points along new connectors. The initial value for this default is the hyperbolic tangent function. To change the default value simply press either the Con Dist **Tanh** (hyperbolic tangent) or Con Dist **MRQS** (monotonic quadratic rational spline) buttons.

3.2 Elliptic PDE Solver Control Functions

3.2.1 Background Control Function

The background control function (Section 12.2.4) default will be set on all new structured domains and structured blocks. The initial value for this default is Thomas-Middlecoff. To change the default value simply press either the Bckgr CF **Lapl** (LaPlace), Bckgr CF **T&M** (Thomas-Middlecoff), or Bckgr CF **Fixd** (fixed grid) buttons.

3.2.2 Foreground Control Function

The foreground control function (Section 12.2.5) default will be applied to all boundaries of new structured domains and structured blocks. The initial value for this default is Hilgenstock-White. To change the default value simply press either the Forgr CF **Srns** (von Lavante-Steger-Sorenson) or Forgr CF **Hlgn** (Hilgenstock-White) buttons. If you do not want any foreground applied to newly created domains and blocks, press Forgr CF **None**.

3.3 Unstructured Grid Attributes

3.3.1 Domains

Unstructured grid attributes applied to domains when they are created are set by the buttons labeled UnstrDom in the Defaults menu.

3.3.1.1 Interior Grid Points Toggle

Whether or not grid points are inserted into unstructured domains at the time they are created is controlled with this toggle button. Initially this default is enabled and grid points are inserted into unstructured domains when they are created according to the values of the other unstructured grid attribute defaults. To disable this grid point insertion, thereby reducing the compute time during domain creation, press the **IntPt** button.

3.3.1.2 Decay Factor

The default decay factor qualitatively controls how far into the grid's interior the boundary cell size affects the interior cell size. A value of `0` means there is no influence; whereas, a value of `1` provides full influence. The initial value for this default is `0.5`. To change it press the **Decay** button and type in a value in the range `[0,1]`.

3.3.1.3 Minimum Triangle Area

The area of the smallest allowable triangle in an unstructured domain is specified by pressing the **MinTri** button and typing in the desired minimum area. By default, this value is not explicitly set and is instead set to the area of an equilateral triangle with edge lengths equal to the smallest boundary spacing. This results in interior triangles that are no smaller than the smallest triangles adjacent to the boundary. Setting this attribute establishes a global minimum on the size of triangles rather than computing and using a different value in each domain.

3.3.1.4 Maximum Triangle Area

The area of the largest allowable triangle in an unstructured domain is specified by pressing the **MaxTri** button and typing in the desired maximum area. This value is not explicitly set and is instead set to the area of an equilateral triangle with edge lengths equal to the largest boundary spacing. This results in interior triangles that are no larger than the largest triangles adjacent to the boundary. Setting this attribute establishes a global maximum on the size of triangles rather than computing and using a different value in each domain.

3.3.1.5 Maximum Angular Deviation

The maximum dihedral angle allowed between two triangles sharing an edge may be specified by pressing the **MaxAng** button and typing in the value in degrees. Initially this default is not enabled.

3.3.1.6 Maximum Chord Height Deviation

The maximum distance allowed between a domain's analytical shape (the database on which it is defined) and its discrete shape (the triangular facets) may be specified by pressing the **MaxDev** button and typing in the value. Initially this default is not enabled. This default only applies to domains that are database constrained.

3.3.2 Blocks

Unstructured grid attributes applied to blocks when they are initially created are set by the buttons labeled UnstrBlk in the Defaults menu.

3.3.2.1 Tetrahedral Mesher Memory Limit

The maximum amount of memory to be used by the tetrahedral mesher in the Blocks, Run Solver **Unstructured** command is set by pressing the UnstrBlk **MaxMem** button and entering a value in MBytes that is between 10 and 2000 (the approximate upper limit of memory addressable by a 32-bit address). The default value is 64 MBytes.

3.3.2.2 Tetrahedral Mesh Size Factor

The default size factor qualitatively controls how far into the grid's interior the boundary cell size affects the interior cell size. A value of `0` means there is no influence; whereas, a value of `1` provides full influence. The initial value for this default is `0.5`. To change it press the **SizFac** button and type in a value in the range `[0,1]`.

3.3.2.3 Minimum Pyramid Height

The height of the shortest allowable pyramid in a hybrid unstructured block is specified by pressing the **Min-Pyr** button and typing in the desired minimum height. By default this value is not explicitly set.

3.3.2.4 Maximum Pyramid Height

The height of the tallest allowable pyramid in a hybrid unstructured block is specified by pressing the **Max-Pyr** button and typing in the desired maximum height. By default this value is not explicitly set.

3.3.2.5 Pyramid Aspect Ratio

The aspect ratio from which pyramid height is derived in a hybrid unstructured block is specified by pressing the **Pyr AR** button and typing in the desired value which must be a positive number. By default this value is set to 0.5.

3.4 Domain Topology Controls

3.4.1 Split Angle

The default split angle controls how triangular surface grids will be subdivided when they are imported into Gridgen. This default applies whether they are imported as database entities or domains. The initial value for this default is 0.0; imported surface grids will not be split. To change the default value, press the **Split Angle** menu button and enter an angle in degrees. Triangular surface grids will be split into closed regions on import if the dihedral angles between all the boundary triangles of a region and their neighboring regions are greater than the split angle. If any of the angles are less than the split angle, no action is taken.

The default split angle also is used as:

- the initial value for feature extraction in the Database menu
- the initial value for splitting connectors when creating **On DB Entities** domains

3.4.2 Join Angle

The default join angle provides an initial angle for joining domains when:

- joining unstructured domains in the Modify Domains menu
- creating a face for an unstructured block
- creating a face for extruding a prism block.
- auto-joining domains when creating by **On DB Entities**

Two domains will be joined if at least interval has a turning angle less than the join angle.

3.5 Database Parameterization

You can choose to display database entity parametric coordinates varying from 1 to the number of intervals or normalized coordinates that vary from 0 to 1. To display coordinates corresponding to the intervals in the entity press the DB Coord **IJ** button. To display normalized coordinate press DB Coord **UV**. The initial value for this default is normalized coordinates.

3.6 QuickSave File Name

The QuickSave file name is the name used for the Gridgen file exported by the QuickSave quick key (Section 2.3.6). The initial value for this default is `quicksave.gg`. To change the default value simply press the **QuickSave Filename** button and use the file Browser to choose a new file name.

3.7 Tolerances

Tolerances are an important feature in Gridgen. They are used to control how grid entities are logically connected across gaps in the database and the way many database creation and intersection operations perform. Tolerances are set in a submenu of the DEFAULTS MENU. To set tolerances press **Tolerances**.

3.7.1 Node Tolerance

The node tolerance is the real value within which Gridgen considers two nodes (Section 7) to be the same. The initial value for this default is `0.0001`. To change the value of the default press the **Node** button and type the new value in the Text Input window.

When you enter a new node tolerance the entire grid system is updated to reflect the new value. Any formerly distinct nodes falling within the new tolerance will be combined.

The importance of the node tolerance is it affects whether or not Gridgen’s grid topology logic will consider two connectors to be end-to-end adjacent and, therefore, whether a domain can be built using them. The node tolerance is also used by the **Merge** command (Section 15.10).

3.7.2 Connector Tolerance

The connector tolerance is the real value used by Gridgen to determine whether two connectors (Section 7) with equal, non-zero dimensions are the same. This determination is made by computing the distances between grid points on the two connectors. If the maximum distance between corresponding grid points on the two connectors is less than the connector tolerance, the two connectors are considered the same. One is deleted and replaced with the other. The initial value for this default is `0.0001`. To change the value of the default, press the **Connctr** button and type the new value in the Text Input window.

When you enter a new connector tolerance, the entire grid system is updated to reflect the new value. Any formerly distinct connectors that fall within the new tolerance will be combined.

Note the connector tolerance may be significantly larger than the grid point spacing along the connector allowing for instances of connectors with tight grid point spacing on opposite sides of a large gap in the database.

3.7.3 Grid Point Tolerance

The grid point tolerance is the real value used by Gridgen to determine whether two grid points are the same. The initial value for this default is 10^{-7}. To change the value of the default press the **Grid Point** button and type the new value in the Text Input window.

The grid point tolerance is used to identify singularities (poles) in the grid in order to avoid divide by zero conditions during the computation of various grid control parameters.

3.7.4 Model Size

The model size is the basis for all tolerances used for Gridgen's geometric computations such as intersections, projections, and the triangular surface mesher. The default value for the model size is `1000`. Other tolerances are then computed from the model size as shown in the table below.

Gridgen's Model Size and Geometric Tolerances

type of tolerance	formula	default value
model size	L	1000
same-point	$L/10^7$	0.0001
surface-fit	$L/10^6$	0.001

3.8 DB Pick Mask

DB Pick Mask is a very useful tool in Gridgen for filtering large Browser lists of database entities. The DB PICK MASK MENU provides a toggle list of database entity types that may be toggled **On** or **Off** individually or in groups. Once an entity type is toggled **Off**, all instances of that entity type in the Browser will become darkly shaded and the user will no longer be able to pick that entity type in either the Browser or the Display window.

3.9 Other Information

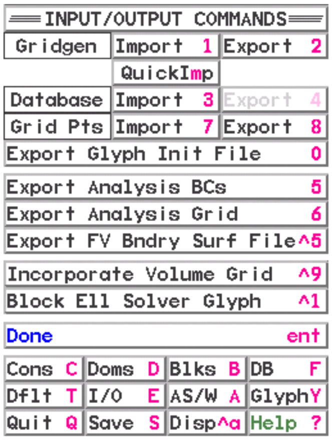
Even though it is not a user-setable default value, the Blackboard in the Defaults menu displays a value called the `Grid/DB Leng`. This value is the approximate size of the largest dimension of the current grid and database and is useful for gauging the size of your grid relative to other lengths such as the model size or average spacing default.

4. Importing and Exporting Files

Gridgen can import and export many different types of files via the INPUT/OUTPUT COMMANDS menu. The file types include:

- Gridgen restart files
- Database files, for example native CAD files, IGES, and STL
- Grid point files, for example PLOT3D, NASTRAN and VRML 1.0
- Glyph initialization and journal files
- Analysis software grid and BC files for many popular CFD and FEA solvers
- FIELDVIEW structured boundary surface files

The INPUT/OUTPUT COMMANDS menu is shown below.



Working with each of these file types is described in the following sections. Additionally, the format of many file types is described in Section 18.

Many files (for example, grid point files, database network files, and some analysis grids) allow you to choose from several attributes for importing and exporting the data. These file attributes are set via a menu consisting of radio buttons for each attribute and the available options depend on the specific type of file being imported or exported.

4.1 Import Gridgen File

The Gridgen file is your grid model restart file and its default extension is `.gg`. It contains everything about the grid including nodes, connectors, domains, blocks, grid points, analysis software specifications, display attributes, database entity name associativity, and database filename associativity. You will save a Gridgen file for each grid; it will serve as the "restart" file or "state" file for continuing the grid generation process over several sessions.

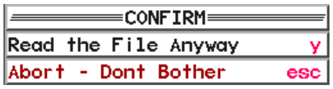
The Gridgen file *does not* contain the database entities created in Gridgen or database entities imported from another file. If you create, delete, or modify database entities, a separate database file must be exported (Section 4.5). If your grid, and therefore the contents of the Gridgen file, was built using a database, you *must* import the database file(s) before you import the Gridgen file (Section 4.1.1).

4.1.1 Gridgen File Database Associativity

If any grid elements in the Gridgen file are defined in terms of a database entity (e.g., Line on DB segments, projection), the database file *must* be imported before the Gridgen file is imported. If the Gridgen file is imported before the database file, a warning message will be posted to the Message Window:

```
WARNING! -N nodes, N connectors and
N domains in the input file contain
references to database entities not in
memory.  Affected connectors and domains
will be PERMANENTLY modified.
```

You will also be presented with the following button menu.



Read the File Anyway - If you choose to import the Gridgen file, despite the warning about missing database references, then the grid data will be loaded, and the grid will appear to be unchanged. However, all references to the database will have been permanently lost. Also, any database constrained segments will have been converted to polylines through the grid points. If you change your mind after choosing **Read the File Anyway**, immediately exit Gridgen without saving the Gridgen file, run Gridgen again, and import the appropriate database files before importing the Gridgen file.

Abort - Don’t Bother - This command aborts the import process and allows you to proceed with importing the appropriate database files.

The importance of Gridgen’s grid to database associativity is its ability to provide parametric grid regeneration. Two factors contribute to this capability: 1) Gridgen associates grid elements to the database via the entity name and 2) the parametric coordinates of the database are stored in the Gridgen file with any entity that references the database.

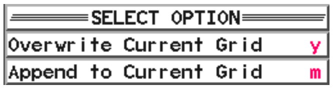
What happens during Gridgen file import is as follows:

1. Gridgen begins reading the grid elements from the Gridgen file.
2. When the Gridgen file indicates that a grid element references a database entity (for example, a Line on DB segment defined on database entity “wing-upper”), it searches the database for an entity with the same name.
3. When Gridgen finds the referenced database entity, it evaluates the parametric coordinates `(u,v)` stored in the Gridgen file in terms of the database entity resulting in Cartesian coordinates `(x,y,z)`. If the referenced database entity is not found, a warning indicating that there are missing database entities appears in the Message window. The name of the file containing the missing database entities is printed to the shell window.
4. Gridgen repeats this process for every node, connector, domain, block, and grid point in the file.

Therefore, if you have imported the same database file as you used when you created the Gridgen file, everything will appear the same as it was before. However, if you have substituted a new “wing-upper” surface for the previous one, you will notice that the grid now conforms to the new “wing-upper”.

4.1.2 Overwrite vs. Append

If at least one connector exists in Gridgen prior to your invocation of the Gridgen **Import** command, you must choose how to treat the existing grid.



SELECT OPTION

Overwrite Current Grid y

Append to Current Grid m

Overwrite Current Grid - All current Gridgen grid information will be deleted and the grid system in the file will be imported. You must confirm your selection of this option via an additional button menu since deleting the current grid cannot be undone.

Append To Current Grid - The contents of the Gridgen file will be added to the current grid system. This capability allows several users to work simultaneously on portions of the same grid and later assemble them into one grid system. When two grid systems are merged using this command, Gridgen culls all duplicate connectors and domains it encounters, allowing the two grids to connect to each other.

4.1.3 Gridgen File Backward Compatibility

Gridgen can import Gridgen restart files from all previous Gridgen releases. Note that in Gridgen Versions 6 and 8 these files were known as GRIDBLOCK files and have a `.blk` extension by default.

Gridgen Version 8 files are fully supported subject to the database associativity rules described in Section 4.1.1

Only a portion of the data in a Gridgen Version 6 `.blk` file is imported by Gridgen. Specifically, only the connector shapes will be loaded. Any Ellipse Segments in the Version 6 file will be automatically converted to Circle Segments, which will modify the connector shape in some cases. Keep in mind that since Gridgen Version 6 files do not contain database parametric data for Line on DB and Curve on DB segments, no connectors in Version 6 files will be constrained to the database.

4.2 Quick Import

The Gridgen **QuickImp** menu button allows you to perform a quick import of your Gridgen restart file. Gridgen assumes that nothing has changed in the grid to database associativity and, therefore, does not recompute the `(x,y,z)` coordinates of any database constrained points. Gridgen also does not check for proper node linkages. **QuickImp** should *only* be used when:

1. The Gridgen file is large and normally takes a long time to import.
2. There is no chance of ambiguity between the database (if any) and the Gridgen file (in other words, the Gridgen file was created on the database file that has been imported).
3. There are no unsatisfied database references in the Gridgen file.

4.3 Export Gridgen File

The Gridgen file is your grid model restart file. It contains everything about the grid including nodes, connectors, domains, blocks, grid points, analysis software specifications, display attributes, and database entity name associativities. You will save a Gridgen file for each grid. The Gridgen file does not contain the database entities.

Upon selecting the Gridgen **Export** button from within the INPUT/OUTPUT COMMANDS menu, you will be placed in the file Browser to determine the name for the file. If the file to be exported already exists, it may be overwritten by selecting the file name from the Browser, assuming that write privileges exist for the selected file. You are encouraged to export Gridgen files often, so recovery from errors may be made by starting from the last saved restart file.

4.4 Importing Database Files

The Database **Import** command, in the INPUT/OUTPUT COMMANDS menu, allows you to read database entities from a file into Gridgen for use in defining the shape of the object being meshed. The **Import** command immediately invokes the Browser from which you may choose the file name. By default, Gridgen lists in the Browser all files with extensions corresponding to database file types. Gridgen determines the file type (Section 4.4.4) based on the selected file name's extension. If the extension cannot be determined, it is assumed to be IGES.

Gridgen can import geometry data for use as a database via the formats in the table below.

Database File Extensions

file type	extension

network	`.net`
	`.grd`
	`.grda`

segment	`.dat`
Gridgen composite	`.dba`
PATRAN	`.pat`

IGES	`.igs`
	`.iges`

Shell	`.stl`
	`.wrl`
	`.nas`

CATIA V4	`.model`

Pro/E	`.asm`
	`.prt`

STEP (AP203, AP214)	`.stp`
	`.step`

For network files, Gridgen can automatically detect whether the file's format is ASCII or binary. However, it cannot determine the file's data precision. For composite files, Gridgen cannot determine the format or precision automatically. Whenever you set attributes using the FILE ATTRIBUTES menu, the attributes are saved for use during any future imports during the current Gridgen session. For example, if you import a Gridgen style, binary, single precision network file once by setting the file attributes using the menu, then the next time you import a network file Gridgen will recall the previous attributes and read the file directly.

The file attribute options used for the various Gridgen database file types are shown in the table below.

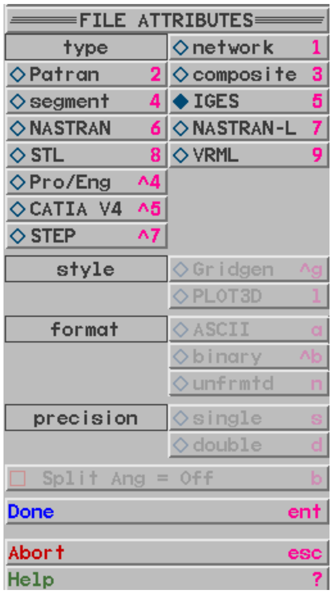
File Attribute Options Per Database File Type

type	style	format	precision
network	PLOT3D (default) Gridgen	ASCII (default) Fortran unformatted binary	single double (default)
segment	n/a	ASCII	double
PATRAN	n/a	ASCII	double
composite	n/a	ASCII (default) binary (import only)	single double (default)
IGES	n/a	ASCII	double
STL	n/a	ASCII (default) binary	n/a

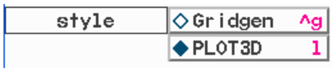
After having determined the file's attributes, Gridgen reads the file, imports the entities, and creates graphical objects that will appear in the Display window. If, for any reason, reading the file produces an error (bad attributes, bad data in the file), Gridgen will display a warning and then invoke the FILE ATTRIBUTES menu under the assumption that the attributes were determined improperly. Use the menu to set the appropriate attributes and press **Done** to try to read the file again or press **Abort** if you think there is a problem with the file itself.

The Database **Import** command may also be invoked via the Database menu.

The FILE ATTRIBUTES menu is shown below.



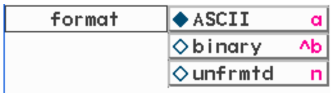
4.4.1 Style



The style attribute pertains to the particular formatting of file types containing discrete data. The style attribute is a subordinate to the type attribute. For example, database files of type network may have either Gridgen or PLOT3D style.

4.4.2 Format

Format refers to the data format in the file.



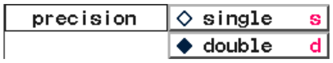
ASCII is a human readable format that is very portable between computers of different types. However, ASCII files will require significantly more disk storage than binary or Fortran unformatted files and will take longer to import and export.

binary is a compact format but suffers from a lack of portability between “big-endian” and “little-endian” computers.

unfrmtd refers to Fortran unformatted, a format similar to binary in that it is very compact. However, binary and Fortran unformatted formats are not interchangeable. Fortran unformatted format is available for importing and exporting Gridgen and PLOT3D style database network files, importing FIELDVIEW unstructured data files, and for exporting some types of analysis software data.

4.4.3 Precision

Precision refers to the word size in the data file.



single precision means that real (floating point) data will be written as 4 byte words.

double precision means that real (floating point) data will be written as 8 byte words

4.4.4 File Type

File type refers to the “brand” of the file to be imported. These files may be obtained from a variety of sources including user written programs and CAD systems. Gridgen supports several file types since some file types do not provide for all the entities. For example, network files may only contain bilinear surfaces. The database file types supported by Gridgen are described below:

4.4.4.1 Network

Network files contain any number of bilinear surfaces. The data in a network file is described in Section 18.4. Each network is similar to a surface grid in that it consists of an ordered `M x N` array of points. Note that Gridgen does not fit a smooth surface through a network's points; the surface will be faceted. If you want to create a smooth surface from the faceted surface obtained from a network file, you should use the **Fit** command (Section 6.5.5) in the DB CREATION menu.

4.4.4.2 Gridgen Segment

A segment file contains any number of linear curves. The data in a segment file is described in Section 18.5. A segmented curve consists of an ordered string of discrete points. Segment files are the same as those used in connector creation under the **Import** segment type. Note that Gridgen does not fit a smooth curve through a segment's points; the curve will be faceted. If you want to create a smooth curve from the faceted curve obtained from a segment file, you should use the **Fit** command (Section 6.3.8) in the DB CREATION menu. Gridgen automatically removes any consecutive coincident points from segment files to make sure that the segment is single valued.

4.4.4.3 Gridgen Composite File

A composite file contains any number of any of the entities supported by Gridgen. It is provided as a means of consolidating all entities within a single database file. It is the only mechanism for saving database entities created in Gridgen. A composite file is also useful if the entities were imported from several different files. Gridgen is able to import composite database files written by all previous Gridgen versions.

4.4.4.4 Native CAD Import

CAD data (geometry models) in their native formats such as CATIA, Pro/E, and STEP can be imported directly into Gridgen, thus providing a more streamlined mesh generation process by eliminating the need for the translation of CAD data into a IGES neutral file format.

Note: The Native CAD reader is an optional feature of Gridgen and is licensed separately. Please contact Pointwise, Inc. for further details on the licensing of this feature.

The following file type attributes are supported for Native CAD file import:

CATIA V4 type refers to the native file format of CATIA Version 4 from Dassault Systems. Gridgen will only read versions 4.1.x and 4.2.x. Note: The CATIA Version 4 native file reader is an optional product.

Pro/E type refers to the native assembly and part file formats of Pro/ENGINEER® design software from Parametric Technology Corporation. Gridgen will read versions up to Pro/E 2000i. Note: The Pro/ENGINEER native assembly file reader is an optional product.

STEP type refers to the STEP AP203 (Configuration Controlled Design) and AP214 (Automotive Design), file formats. Note: The STEP file reader is an optional product.

Gridgen currently supports the Native CAD import on the platforms shown in the table below.

Native CAD Import

Platform	STEP	CATIA	Pro/E
Windows	Y	Y	Y
Linux	Y	Y	Y
IBM AIX			
HP-UX	Y	Y	Y

Native CAD Import

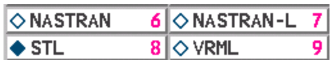
Platform	STEP	CATIA	Pro/E
DEC OSF			
SGI IRIX 6.4	Y	Y	Y
SGI IRIX 6.5	Y	Y	Y
Sun Solaris	Y	Y	Y

4.4.4.5 Shells

Gridgen can import unstructured surface grid files (triangular faceted data) such as STL, VRML, and NASTRAN and use them as database entities. The term “shell” represents the faceted equivalent of a continuous B-Spline surface in Gridgen's database. A shell is a set of point to point, contiguous triangles. All polygons in a unstructured grid file will be triangulated before use as a shell in Gridgen.

Files that have a quadrilateral surface grid are converted to triangles during shell import and are represented as shell database entities in Gridgen.

The following file type attributes are supported for shell import in Gridgen:



◇ NASTRAN 6 ◇ NASTRAN-L 7
◆ STL 8 ◇ VRML 9

NASTRAN type refers to Version 69 of the commercial software product of the same name. For detailed information on this format please consult Reference 16. For surface grid files Gridgen recognizes the following NASTRAN elements:

- CQUAD4
- CTRIA3

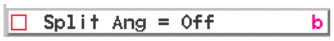
NASTRAN-L type refers to the “large” format for real values in the NASTRAN file. Large format uses 16 characters for real values whereas NASTRAN style uses only 8 characters.

STL type refers to the Stereo Lithography file format described in Reference 42. STL import and export is available for triangular surface grids only. Both ASCII and binary format STL files may be imported and exported, although the binary format is always little-endian.

VRML type refers to the Virtual Reality Modeling Language Version 1.0. For information on this format please consult Reference 41. VRML import and export is available in ASCII only for surface grids. Gridgen recognizes the following VRML keywords:

- Coordinate3
- IndexedFaceSet

VRML’s Translate, Scale, Rotate, and Transform keywords are ignored. Although both triangular and quadrilateral surface grids may be exported to a VRML file, all polygonal cells are triangulated when a VRML file is imported. All mesh data between VRML separators will be converted to at least one Gridgen shell.



□ Split Ang = Off b

During shell import Gridgen allows you to manually set the **Split Ang** tolerance (shown above) used to find geometric features (for example, slope discontinuities) in the shells. By default, the split angle is set to “off” and is disabled but enabled during shell import only.

Split Ang - This allows shells to be divided into smaller shells along edges of the model that meet a turning angle criterion, any triangular surface with a turning angle less than this is split angle. This angle can also be set in the Defaults menu prior to shell import. Refer to Section 15.12 for further details on Feature Extraction.

4.4.4.6 IGES

IGES files conform to the Initial Graphics Exchange Specification Version 5.1 (Reference 15). Gridgen supports only a subset of entities contained in the full IGES standard. Supported IGES entities include those shown in the following table.

IGES Entities Supported By Gridgen

IGES entity number	description
000	null
100	circular arc
102	composite curve
104	conic arc
106 (forms 1,2,3,12)	copious data
108 (form 0)	plane
110	line
112	parametric curve
114	parametric surface
116	point
118 (form 1)	ruled surface
120	surface of revolution
122	tabulated cylinder
124	transformation matrix
126	rational B-Spline curve
128	rational B-Spline surface
130	offset curve
141	boundary
142	curve on surface
143	bounded surface
144	trimmed surface

IGES Entities Supported By Gridgen

IGES entity number	description
186	B-Rep Solid Model Object (trimmed surfaces only)
212	general note
308	subfigure definition
314	color definition
402	associativity instance
406 (form 15)	property
408	singular subfigure instance

The caveats of Gridgen's IGES file support are:

1. A composite curve entity is treated as a group.
2. Points in composite curves are not added to the composite curve group.
3. Vectors are ignored in copious data entity, form=3.
4. Each subfigure instance entity becomes a group. Subfigure members are duplicated.
5. Colors specified in the IGES file are replaced by the closest Gridgen color (closest hue using HSL color scheme).

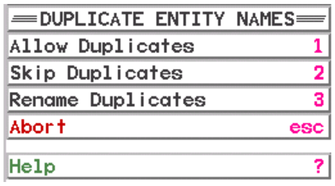
4.4.4.7 PATRAN Neutral Files

The PATRAN Version 2.5 Neutral file contains any number of non-rational, cubic curves and non-rational, bicubic surfaces and is named for the software system that initially defined this file type. The data in a PATRAN Neutral file is described in Reference 13. Gridgen supports only PATRAN's packets 32 and 33, curve and patch data, respectively. All other packets in the PATRAN v2.5 Neutral file are ignored. During PATRAN file import Gridgen attempts to append adjacent curves and surfaces into larger curves and surfaces, respectively.

4.4.5 Duplicate Name Detection

During database file import Gridgen automatically checks to ensure that each database entity has a unique name (Section 15.1). Each database entity must have a unique name in order for Gridgen to correctly associate your grid with your database.

If Gridgen detects duplicate database entity names during file import, the following menu appears:



Allow Duplicates - Pressing this button means that you choose to override Gridgen's requirement for unique entity names. We strongly recommend that you do NOT choose this option.

Skip Duplicates - Pressing this buttons means that only the database entities with unique names will be imported from the file and the other entities (the ones whose names match entity names already in Gridgen) will be skipped.

Rename Duplicates - Pressing this button tells Gridgen to import all entities from the file but rename the duplicates by appending "`-1`", "`-2`", etc. to the name.

Abort - Pressing this button terminates the import process so that you can diagnose why duplicate names were detected.

4.4.6 Automatic Entity Grouping on Import

A group entity containing all entities in a file will be created automatically by Gridgen when importing from Network and PATRAN Neutral files that contain more than one entity.

4.4.7 Backward Compatibility

1. If you are starting Gridgen using a GRIDBLOCK file (`.blk` extension) from GRIDGEN Version 8 and,
2. If that GRIDBLOCK file requires a database and,
3. If that database is imported from more than one Network or PATRAN Neutral file and,
4. If any database file contains more than one entity then,
5. You MUST import the database files first and,
6. You MUST import the database files in the same order as you did when you created the GRIDBLOCK file and,
7. You MUST delete the group entities that Gridgen automatically creates during import.

Gridgen associates grid entities with the database by saving the (`u,v`) coordinates and database entity *name* of any grid point that adheres to the database. GRIDGEN Version 8 associated grid entities with the database by the (`u,v`) coordinates and the entity *number.* Therefore, to ensure that Gridgen establishes the correct grid to database associativity when importing GRIDBLOCK Version 8 files it is imperative that the entity numbering be consistent. The database entity numbering will be made inconsistent because of the group entities that Gridgen creates automatically when importing a Network or PATRAN Neutral file that contains more than one entity.

Consider a GRIDBLOCK file containing a grid that was generated on a database consisting of four entities:

Sample of Gridgen Version 8 Database Import

entity number	description	origin
1	network	`fore.net`
2	network	`fore.net`
3	network	`aft.net`
4	network	`aft.net`

When Gridgen imports this GRIDBLOCK file the only mechanism it has to associate the grid with the database is references to entities numbered 1-4. However, when Gridgen imports these same files the entity numbering is different because of the group entities that Gridgen creates automatically as shown on the following page.

Sample of Gridgen Database Import

entity number	description	origin
1	network	`fore.net`
2	network	`fore.net`
3	group of fore.net	`created automatically`
4	network	`aft.net`
5	network	`aft.net`
6	group of aft.net	`created automatically`

It should be apparent that when Gridgen tries to associate the grid to entity 3 (the first surface in aft.net) it is really looking at the group entity for fore.net, and when it looks for entity 4, it is looking at the first network in aft.net instead of the second.

The solution is to use Gridgen's **UnGroup** command (Section 15.2) in the DATABASE COMMANDS menu to delete entities 4 and 6, the automatically created groups. After the **UnGroup** command is applied, Gridgen will automatically renumber the remaining database entities so that they appear as expected by the GRIDBLOCK file. At this point the Gridgen database is topologically equivalent to the database references in the GRIDBLOCK file, and the GRIDBLOCK file may be imported.

4.5 Export Database File

The **Export** command in the DATABASE COMMANDS menu allows you to write selected database entities from Gridgen into a file. Only Gridgen network, segment, and composite file types may be exported.

Gridgen composite files can be used to save all of the information regarding a database. This includes not only the geometric information but also entity names and graphical display states. Unless the entities are linear, the composite file is the only mechanism for saving any entities created or modified in Gridgen. Linear curves may be exported to segment files, and bilinear surfaces may be exported to network files.

Note that trying to quit Gridgen without saving a modified database will generate a warning in the Message window stating that the current database contains information which has not been saved. This provides a good opportunity to press **Abort - Don't Quit** so you can save your database file.

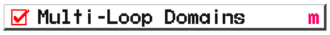
4.6 Import Grid Points

The Grid Pts **Import** command in the INPUT/OUTPUT menu is used to read a file containing only grid points and from the grid points build the appropriate Gridgen elements (connectors, domains, or blocks). Once the grid points have been imported, the entire range of Gridgen commands may be applied to the grid. You are urged strongly to closely inspect the grid after import to ensure that creation of connectors, domains, and blocks was completed properly. Upon completing your work with this grid you may either:

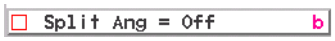
1. Save the modified grid in a Gridgen file (Section 4.3).
2. Export the modified grid as a grid point file (Section 4.7).

Typical uses for importing grid points include applying Gridgen's grid methods to grids created in another application and applying Gridgen's analysis software commands (Section 17) to post-process a grid from another application for one of Gridgen's supported analysis software packages.

Gridgen creates connectors from grid point data by fitting the equivalent of a Line segment through each grid point on an edge. Therefore, even though an edge may be a straight line or a circular arc in the grid file, Gridgen will create a connector that is a Line segment containing as many control points as grid points.



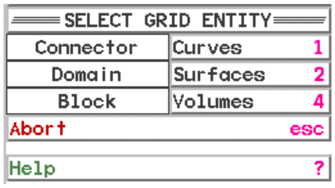
During unstructured surface (domain) grid import you may use the **Multi-Loop Domains** toggle button (shown above) to determine whether triangular surface meshes with multiple loops (internal holes) will be imported into Gridgen intact. By default this button is enabled. If it is disabled, Gridgen will automatically split the surface grid into more than one domain, each domain with a single (outer) loop.



Connections between blocks imported from grid point files are only detected by Gridgen if the connection is full-face to full-face. The implication is that some instances of grid point import may result in Gridgen creating connectors and/or domains that overlap, a situation that is not recommended. You may apply the splitting commands to designate where partial face connections are. Upon completion of the splitting, Gridgen automatically culls duplicate connectors and domains and will then create the inter-block connection.

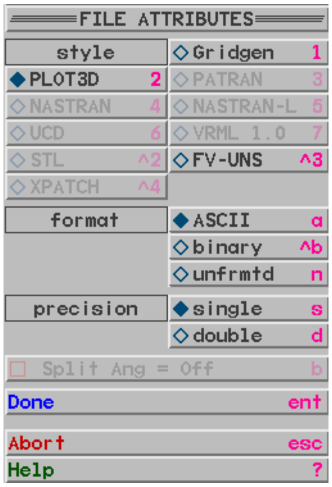
To import grid points from a file and have Gridgen build a grid system (connectors, domains, and blocks) around them, follow the procedure below, starting from the INPUT/OUTPUT menu:

1. Press the Grid Pts **Import** button.
2. Select the type of grid element (connector, domain, or block) for which you will importing grid points, from the SELECT GRID ENTITY menu:



3. Select the name from the Browser. The default extension for the file name is `.grd`

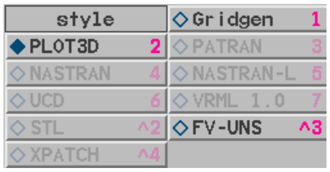
4. For domain and block import, select file attributes from the FILE ATTRIBUTES menu below:



5. Press the **Done** button.

4.6.1 Style

The file style attribute pertains to the particular formatting of certain file types containing discrete data. For grid point import, Gridgen reads the following style attributes:



Gridgen style refers to a format for structured grid data and is described in Section 18.9.

PLOT3D style refers to a format for structured grid data that was popularized by NASA's CFD visualization software of the same name (Reference 38). This format is described in Section 18.4. This toggle is disabled for unstructured grids.

NASTRAN type refers to Version 69 of the commercial software product of the same name. For detailed information on this format please consult Reference 16. For surface grid files Gridgen recognizes the following NASTRAN elements:

- CQUAD4

- CTRIA3

NASTRAN-L type refers to the “large” format for real values in the NASTRAN file. Large format uses 16 characters for real values whereas NASTRAN style uses only 8 characters.

VRML type refers to the Virtual Reality Modeling Language Version 1.0. For information on this format, please consult Reference 41. VRML import is available in ASCII only for surface grids. Gridgen recognizes the following VRML keywords:

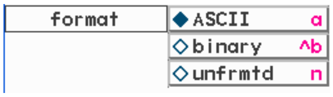
- Coordinate3
- IndexedFaceSet

STL type refers to the Stereo Lithography file format described in Reference 42. STL import and export is available for triangular surface grids only. Both ASCII and binary format STL files may be imported and exported, although the binary format is always little-endian.

FV-UNS style refers to the FIELDVIEW unstructured data file format by the visualization software of the same name and described in Reference 43. FV-UNS import is available for tetrahedral grids only. Both ASCII and binary format FV-UNS files are available, although no wall information is available in the binary format.

4.6.2 Format

Format refers to the data format in the file.



format
◆ ASCII a
◇ binary ^b
◇ unfrmtd n

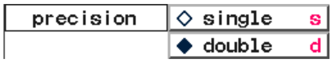
ASCII is a human readable format that is very portable between computers of different types. However, ASCII files will require significantly more disk storage than binary or Fortran unformatted files and will take longer to import and export.

binary is a compact format but suffers from a lack of portability between “big-endian” and “little-endian” computers.

unfrmtd refers to Fortran unformatted, a format similar to binary in that it is very compact. However, binary and Fortran unformatted formats are not interchangeable. Fortran unformatted format is available for importing Gridgen and PLOT3D style structured surface and volume grids, FV-UNS data files, and for exporting some types of analysis software data.

4.6.3 Precision

Precision refers to the word size in the data file.



precision
◇ single s
◆ double d

single precision means that real (floating point) data will be written as 4 byte words.

double precision means that real (floating point) data will be written as 8 byte words.

4.6.4 Importing Structured Grid Points

The Grid Pts **Import** command allows you to import the following structured grid types:

Structured Grid Point Import

style	surface	volume
Gridgen	Y	Y
PLOT3D	Y	Y
PATRAN	N	N
NASTRAN	N	N
UCD	N	N
STL	N	N
VRML 1.0	N	N
FV-UNS	N	N
XPATCH	N	N

4.6.5 Importing Unstructured Grid Points

The Grid Pts **Import** command allows you to import the following unstructured grid types:

Unstructured Grid Point Import

style	surface	volume
Gridgen	N	N
PLOT3D	N	N
PATRAN	N	N
NASTRAN	Y	N
UCD	N	Y
STL	Y	N
VRML 1.0	Y	N
FV-UNS	N	Y
XPATCH	N	N

4.7 Export Grid Points

The Grid Pts **Export** command in the INPUT/OUTPUT menu is used to write a file containing only the grid points from curve, surface (both domains and block faces), and volume grids selected by you. The default extension for the file name is `.grd`. This command is most often used to transfer grid data from Gridgen to another application.

4.7.1 Style

The following style attributes can be exported in Gridgen:

Gridgen style refers to a format for structured grid data and is described in Section 18.9.

PLOT3D style refers to a format for structured grid data that was popularized by NASA's CFD visualization software of the same name (Reference 38). This format is described in Section 18.4. This toggle is disabled for unstructured grids.

PATRAN style refers to Version 2.5 of the commercial software product of the same name. For information on this format please consult Reference 13. PATRAN style is available for domain and block grid point export only.

NASTRAN type refers to Version 69 of the commercial software product of the same name. For detailed information on this format please consult Reference 16. For surface and volume grid files Gridgen recognizes the following NASTRAN elements:

- CQUAD4
- CTRIA3
- CTETRA
- CPENTA
- CTETRA

NASTRAN-L type refers to the "large" format for real values in the NASTRAN file. Large format uses 16 characters for real values whereas NASTRAN style uses only 8 characters.

UCD style refers to the FEA and CFD interface to Advanced Visual Systems' AVS software. For information on this format, please consult Reference 2. UCD style is available for domain and block export only.

VRML type refers to the Virtual Reality Modeling Language Version 1.0. For information on this format, please consult Reference 41. VRML export is available in ASCII only for surface grids. Gridgen recognizes the following VRML keywords:

- Coordinate3
- IndexedFaceSet

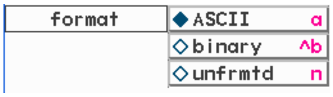
STL type refers to the Stereo Lithography file format described in Reference 42. STL export is available for triangular surface grids only. Both ASCII and binary format STL files may be exported, although the binary format is always little-endian.

FV-UNS style refers to the FIELDVIEW unstructured data file format by the visualization software of the same name and described in Reference 43. FV-UNS export is available for volume grids only and includes tetrahedra, hexahedra, prisms, and pyramids. Both ASCII and binary format FV-UNS files are available, although no wall information is available in the binary format.

XPATCH style refers to the XPATCH unstructured data file format. Gridgen currently supports XPATCH export of unstructured domains.

4.7.2 Format

Format refers to the data format in the file.



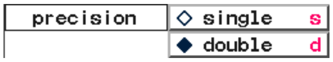
ASCII is a human readable format that is very portable between computers of different types. However, ASCII files will require significantly more disk storage than binary or Fortran unformatted files and will take longer to import and export.

binary is a compact format but suffers from a lack of portability between “big-endian” and “little-endian” computers.

unfrmtd refers to Fortran unformatted, a format similar to binary in that it is very compact. However, binary and Fortran unformatted formats are not interchangeable. Fortran unformatted format is available exporting Gridgen and PLOT3D style structured surface and volume grids and for exporting some types of analysis software data.

4.7.3 Precision

Precision refers to the word size in the data file.



precision ◇ single s
◆ double d

single precision means that real (floating point) data will be written as 4 byte words.

double precision means that real (floating point) data will be written as 8 byte words.

4.7.4 Exporting Structured Grid Points

The Grid Pts **Export** command allows you to export the following structured grid types:

Structured Grid Point Export

style	surface	volume
Gridgen	Y	Y
PLOT3D	Y	Y
PATRAN	Y	Y
NASTRAN	Y	Y
UCD	Y	Y
STL	N	N
VRML 1.0	N	N
FV UNS	N	Y
XPATCH	N	N

4.7.5 Exporting Unstructured Grid Points

The Grid Pts **Export** command allows you to export the following unstructured grid types

Unstructured Grid Point Export

style	surface	volume
Gridgen	N	N
PLOT3D	N	N
PATRAN	Y	Y
NASTRAN	Y	Y
UCD	Y	Y
STL	Y	N
VRML 1.0	Y	N
FV UNS	N	Y
XPATCH	Y	N

Structured surface grids are exported in the unstructured form if the STL, VRML, or NASTRAN style attribute is chosen and their cells are converted to triangles during grid point export.

4.8 Export Glyph Initial File

The **Export Glyph Init File** command is used to export from Gridgen a Glyph file suitable for use as a run commands or rc file (Section 1 of the Glyph Reference Manual) containing the current settings of the **Defaults** command and the display states in the DISPLAY COMMANDS menu.

4.9 Export Analysis BCs

The **Export Analysis BCs** command allows you to write to a file the current analysis software boundary conditions, volume conditions, and other data formatted for the current analysis software package. It is only available if the current analysis software package allows for separate boundary condition and grid files. See Section 17.8 for more information on Analysis Software BCs.

4.10 Export Analysis Grid

The **Export Analysis Grid** command allows you to write to a file the current grid formatted for the current analysis software package. It is only available if the current analysis software package allows for separate boundary condition and grid files. See Section 17.8 for more information.

4.11 Export Analysis Data

The **Export Analysis Data** command allows you to write to a file the current grid, boundary condition, volume condition, and other data formatted for the current analysis software package. See Section 17.8 for more information.

4.12 Export FIELDVIEW Boundary Surface File

For structured grids, the **Export FV Bndry Surf File** command allows you to write a FIELDVIEW boundary surface file that may be imported by the FIELDVIEW visualization software (Reference 43). The contents of the file include the index ranges of all of the surfaces in your grid grouped according to their boundary condition. Once imported into FIELDVIEW along with your grid, this data allows you to easily turn on the display of common surfaces such as walls or symmetry planes.

4.13 Incorporate Volume Grid

The **Incorporate Volume Grid** command in the INPUT/OUTPUT menu is used to merge the structured volume grid results of Gridgen's application of elliptic PDE methods with the rest of the Gridgen system. This command would be used when the entire volume grid must be assembled in Gridgen for subsequent export using the analysis software commands. Otherwise, there is no strict requirement for incorporating the final volume grid with the Gridgen file. You may choose to save the Gridgen file without the volume grids and save a separate volume grid file for use with your analysis software. Incorporating the volume grid into the Gridgen file will substantially increase the size of the Gridgen file and the memory required to run Gridgen.

Gridgen uses two criteria to determine which blocks of the current system the blocks in the volume grid file correspond to:

1. the block dimensions (`IxJxK`) must match, and
2. the distance from the corners of the volume grid to the corners of the block must be less than the node tolerance.

4.14 Block Ell Solver Glyph

To prepare the files necessary to run Gridgen on one or more structured blocks, the following procedure starting from the INPUT/OUTPUT menu:

1. Press the **Block Ell Solver Glyph** button.
2. Select from the button menu whether you will be working from Gridgen or volume grid files, and set the number of iterations to be run (for structured grids).
3. Select the block or blocks to be run in Gridgen.
4. Select the grid file name from the Browser.
5. Set the file attributes using the radio button menu (volume grid files only).
6. Select the Glyph file name from the Browser. The default extension for the file name is `.glf`.

The **Blk Ell Solver Glyph** command is a mechanism for writing a Glyph script to use the elliptical PDE solver on structured blocks.

The first step is to select the blocks to be run. Usually, this selection is made after viewing the volume grid using the **Examine** command in the BLOCK COMMANDS menu (Section 16).

You choose whether Gridgen will import and export grid data from a Gridgen file or a volume grid file. You may also have Gridgen work from your primary Gridgen (`.gg`) file. Should you choose to work from the Gridgen file be advised that your new volume grids from Gridgen will become a permanent part of the Gridgen file thereby increasing its size and the time to import and export. On the other hand, for small and moderately sized grids this is a good way to maintain a single file for your entire grid. If you choose to use volume grid files with Gridgen, you will have to provide for the management of separate Gridgen and volume grid files. If you ever wish to merge the structured volume grid and Gridgen files, you will have to use the **Incorporate Volume Grid** command (Section 4.13). However, the relatively small size of the Gridgen file improves Gridgen's interactive performance. We recommend that you work with volume grid files if you are generating a large grid.

The Glyph file contains the Glyph commands necessary for running Gridgen including, most importantly, how to apply the grid methods. See Section 1 of the Glyph Reference Manual for a detailed discussion of the contents of the Glyph input file.

Please be advised that the Glyph files produced by this command for running Gridgen may have to be hand edited to ensure that the correct values have been written. Therefore, always inspect your Glyph files before running Gridgen.

5. Layer Manager

5.1 Overview

Gridgen's layer manager gives you flexible control over what portions of your database model are displayed. Gridgen's layer manager is implemented in a very similar style to that of common computer aided design (CAD) software packages. Gridgen supports 1024 layers numbered 0 through 1023. An entity must exist in exactly one layer. Layers can be likened to drawing on transparencies that overlay each other. You choose which layers are visible.

Layers are designed for display management only. There are other tools, **En/Disable** (see Section 15.6) and **Group** (see Section 15.2) providing additional control over database entity display and use. This section of the User Manual describes in detail commands associated with and the operation of the layer manager. All layer information is saved in your Composite (.dba) database file (see Section 4.5).

5.2 Database Groups and the Layer Manager

Two points of consideration regarding database groups should be made. First, database groups are a picking mechanism only, and as such, do not belong to a layer. Second, if any child of a group is disabled or is in a layer that is off, the group itself will be unavailable for picking or selection. For more information about database groups, see Section 15.2 of the User Manual.

5.3 Scripting and the Layer Manager

Glyph commands which take actions in any manner related to database entities ignore layer status of the database entities. Layer information is only addressed by layer specific Glyph commands. See Section 5 of the Glyph Reference Manual for more information regarding Glyph commands for the layer manager.

5.4 Layer Assignment

Assignment of database entities to layers takes place in three different functional areas of Gridgen as described in the following subsections.

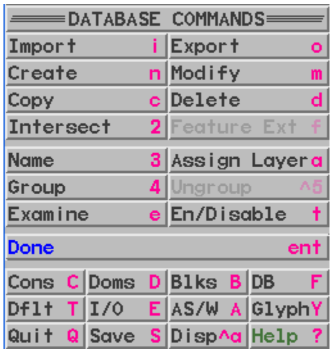
5.4.1 Database File Import

For IGES and native CAD database file import using **Database Import** in the INPUT/OUTPUT COMMANDS menu (see Section 4.4), all level information in the file is transferred to Gridgen's layers. This capability allows users to preserve the layering assignments made in their original CAD models. It is always recommended users save imported database information to a new composite database file. All layer assignments and layer status are preserved in the composite file.

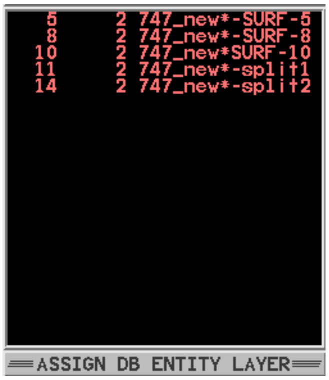
Database models imported from files containing no layer or level information, or imported from file types for which Gridgen does not support layer data, have all entities assigned to layer 0 (zero).

5.4.2 The Assign Layer Command

The **Assign Layer** command in the DATABASE COMMANDS menu is used to manually reassign the layers to which entities belong. The command is shown in the menu example below.

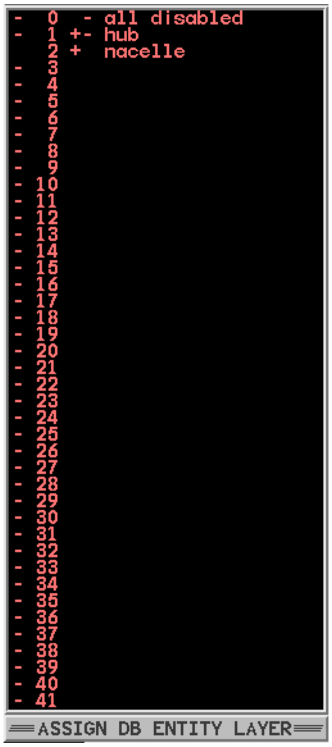


Activation of this command puts you into Browser mode with a list of all available database entities. This list only shows entities that can be assigned to a layer. Entities currently in a layer whose status is off are not available to be reassigned and are not shown in the Browser. Likewise, this list is also subject to the enable state of each entity in the currently displayed layer set; disabled entities are not available to be reassigned. An example of the Browser list is shown below.



```
  5        2 747_new*-SURF-5
  8        2 747_new*-SURF-8
 10        2 747_new*SURF-10
 11        2 747_new*-split1
 14        2 747_new*-split2
```

Each line of the Browser represents a database entity with the entity number, current layer, and entity name. Once you select the entities from this list you wish to reassign and press **Done**, you are put into a second Browser list of all 1024 layers.



```
-  0  - all disabled
-  1 +- hub
   2 +  nacelle
-  3
-  4
-  5
-  6
-  7
-  8
-  9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
```

In this Browser, you see each layer listed by number in the second column. The dash in the first column indicates that particular layer is currently off. The third column contains a plus (+) and/or minus (-) symbol indicating the layer contains either enabled (plus) and/or disabled (minus) database entities. Absence of either symbol indicates the layer is currently empty. The last column contains the layer description, if one has been applied using the layer manager panel (see Section 5.5.4).

Selection of a layer from this list instantly assigns the entities to that layer. If the selected layer’s status is currently off, the newly reassigned entities will no longer be rendered in the Display window.

5.4.3 The Layer Move, Combine Commands

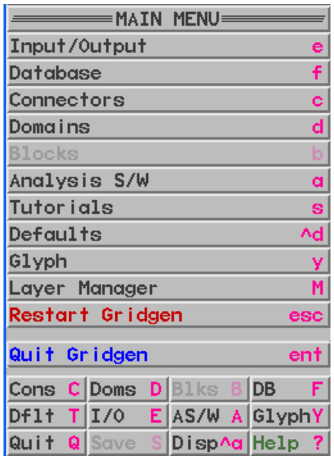
The only other manner in which database entities can be assigned to a new layer is using the Layer **Move** or Layer **Combine** commands in the Layer Manager panel. Please refer to Section 5.5.4 below for more information.

5.5 The Layer Manager Panel

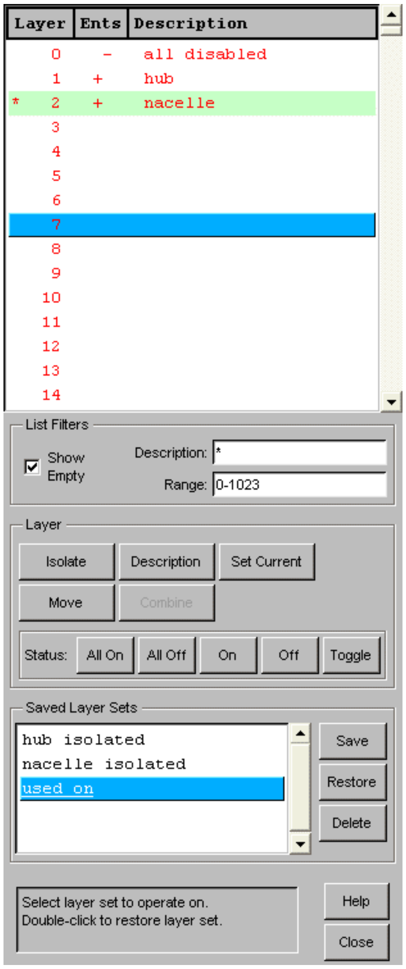
The Layer Manager panel is used to control the status of layers.

5.5.1 Opening the Layer Manager Panel

There are two ways to open the Layer Manager panel: use the **Layer Manager** command in the MAIN MENU shown below or use the associated key board hot key ***Shift+m*** (see Section 2.3.2). Note that this hot key can also be used to pop the Layer Manager panel to the top of your desktop windows even when the panel has already been opened.



An example of the full panel is shown below. It is divided into four logical sections: the layer Browser list, the Filter tools, the various Layer operations, and the Saved Layer Sets tools.



Layer	Ents	Description
0	-	all disabled
1	+	hub
* 2	+	nacelle
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

List Filters

☑ Show Empty

Description: *

Range: 0-1023

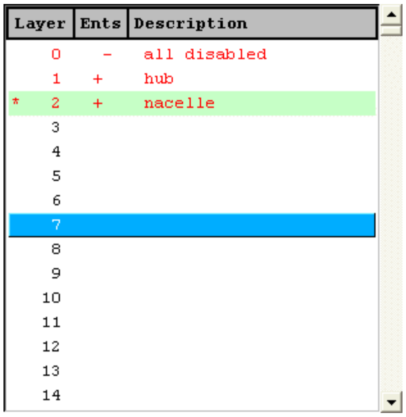
Saved Layer Sets

hub isolated
nacelle isolated
used on

Select layer set to operate on.
Double-click to restore layer set.

5.5.2 Layer Browser

The layer Browser list is used to select individual or sets of layers to operate on using commands in the Layer operations section of the panel (see Section 5.5.4). There is a great deal of information conveyed simply by what is shown in the list and by the colors used. The list is broken into three columns.



Layer	Ents	Description
0	-	all disabled
1	+	hub
* 2	+	nacelle
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

The first column, labeled Layer, contains the layer number starting at layer 0 (zero) and ending at layer 1023. Also, in the first column one layer number is preceded by an asterisk (*). This, as well as the green highlight of that layer, indicates it is the current working layer. This is the layer into which any newly created database entities will be placed.

The second column, labeled Ents, contains a plus (+) and/or minus (-) symbol indicating the layer contains either enabled (plus) and/or disabled (minus) database entities. Absence of either symbol indicates the layer is currently empty.

The third column, labeled Description, contains an alpha-numeric description of that particular layer. Creative and consistent descriptions can facilitate and enhance your use of the list filtering capability described in the next section (see Section 5.5.3).

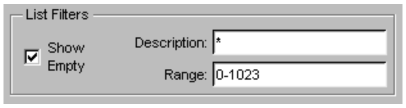
Text color used in the layer Browser indicates the status of the layers. Red text indicates that layer is currently on or its entities are currently rendered (subject to whether or not they are enabled (see Section 15.6)). Black text indicates that layer is currently off (its entities are not currently rendered regardless of whether or not they are enabled). Double clicking of any individual layer will toggle its status between on and off.

Layers highlighted in royal blue are currently selected for use of the Layer operations discussed in (see Section 5.5.4). Not all operations will be available according to whether one or multiple layers are currently selected. Simply move the cursor over a layer entry in the list and press the left mouse button to select it. You can also press and hold the left mouse button while moving the cursor up or down the list to select multiple layers in sequence.

Standard conventions can be used for more elaborate picking operations. For instance, you can select a single layer, move down in the list using the right hand scroll bar, and then use ***Shift+pick*** (left mouse) to select a contiguous list of layers. Use ***Ctrl+pick*** (left mouse) to select multiple non-contiguous layers.

5.5.3 List Filters

The layer Browser does not always have to list all 1024 layers. The list shown in the Browser is subject to the filtering options in the following section of the panel. Keep in mind the current working layer (shown with an asterisk in the Layer column and highlighted in green) will always be shown in the Browser list, regardless of any filtering applied. The three filters are arranged in order of precedence: **Show Empty**, **Description**, and **Range**.

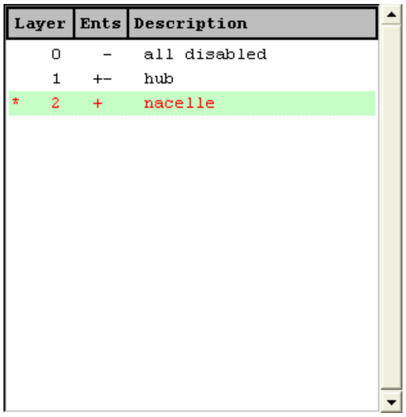

List Filters

☑ Show Empty

Description: *

Range: 0-1023

Show Empty is a toggle command which is on by default. This instructs the Layer Manager to list all of the layers, including those containing no entities. Toggling off this option will remove all empty layers from the Browser list, as shown in the following example.



	Layer	Ents	Description
	0	-	all disabled
	1	+-	hub
*	2	+	nacelle

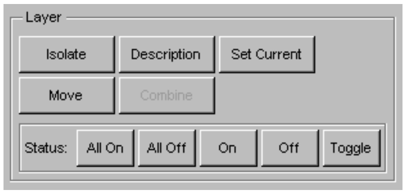
Description: is a text entry field where a text string can be entered, with wildcards (*) to be matched against each layer's description. With each new character added to the string, the list is updated instantly. Non-matches are removed from the Browser list.

Range: is a text entry field where multiple numeric ranges or individual layer numbers can be entered separated by commas with no spaces. You must hit ***Enter*** on your keyboard for the new range(s) to be activated. Valid examples would include:

- 5,10,25-30
- 10-100,200-1023
- 0-10

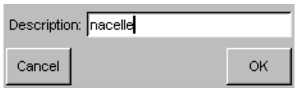
5.5.4 Layer Operations

These operations act on the layer or layers selected in the layer Browser. Some require a single layer to be selected, some multiple and others one or more.



Isolate - only available when a single layer is selected. This command sets the selected layer to the current working layer and turns off all other layers

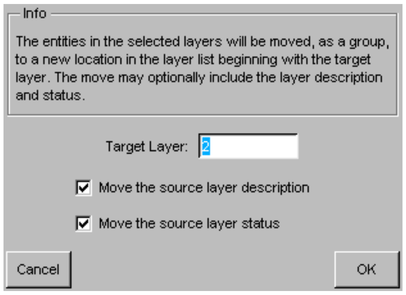
Description - available when one or more layers are selected. This command opens a dialog panel shown below where you can type in a new alpha-numeric description. When multiple layers are selected the same description will be applied to all.



Description: nacelle

Set Current - only available when a single layer is selected. This command sets the selected layer to the current working layer. All newly created database entities will automatically be assigned to the current working layer.

Move - available when one or more layers are selected. This command opens a dialog panel shown below where you can type in a target layer number and choose whether or not the moved layer(s) will take with them their description and/or their on/off status.



Info

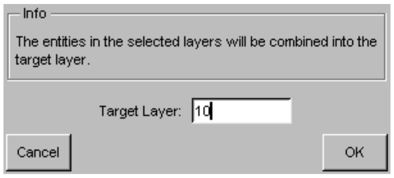
The entities in the selected layers will be moved, as a group, to a new location in the layer list beginning with the target layer. The move may optionally include the layer description and status.

Target Layer: 2

☑ Move the source layer description

☑ Move the source layer status

Combine - available when two or more layers are selected. This command opens a dialog panel shown below where you can type in a target layer into which all entities from those layers selected will be combined.



Info

The entities in the selected layers will be combined into the target layer.

Target Layer: 10

Status commands simply change the on/off state of the selected layer(s):

All On - turns on the display of all layers.

All Off - turns off the display of all layers with the exception of the current working layer, which is always displayed.

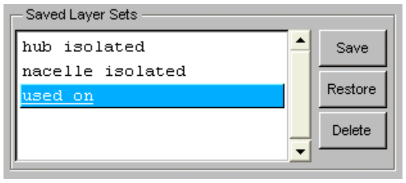
On - turns on the layers currently selected in the Browser list.

Off - turns off the layers currently selected in the Browser list.

Toggle - toggles the on/off state of all layers.

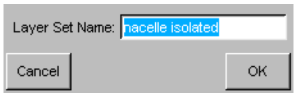
5.5.5 Saved Layer Sets

The Saved Layer Sets tool allows you to store and later recall the on/off state of all layers, as well as the designated current working layer. Saved layer sets do not preserve any filter settings.



```
hub isolated
nacelle isolated
used on
```

Save - opens a dialog panel shown below where you can type in the name of your new saved layer set.



Layer Set Name: nacelle isolated

Restore - restores the layer set that is selected in the Saved Layer Sets Browser. You can also double click a layer set in the Browser to restore it.

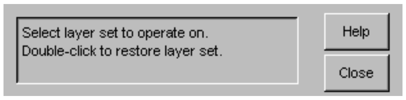
Delete - removes the layer set selected in the Saved Layer Sets Browser.

5.5.6 Special Features

A tool tips field is provided at the bottom of the Layer Manager panel as shown below. As you pass the cursor over any command, Browser entry, or text field, helpful text will be displayed in this field to assist you.

There is a **Close** command to close the Layer Manager panel completely. To reopen, return to the MAIN MENU and use the **Layer Manager** command or use the hot key ***Shift+m*** at any time.

Finally, Gridgen’s html based help feature can be opened from the Layer Manager panel using the **Help** button shown below.

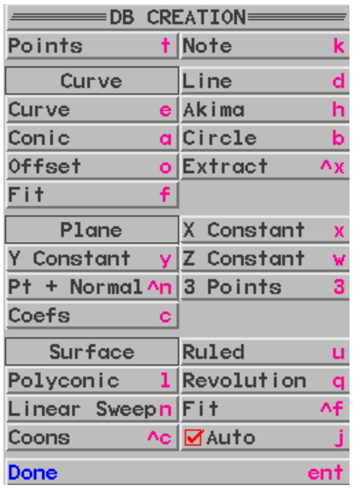


Select layer set to operate on.
Double-click to restore layer set.

6. Creating Database Entities

6.1 Overview

The collection of points, curves, and surfaces defining the shape of the object on and around which you will generate the grid is called the *database*. Each individual point, curve, or surface within the database is called an *entity.* Furthermore, each surface entity consists of a 2D array of elements called *intervals* and each curve entity consists of a 1D array of intervals. The Create command in the Database menu allows you to add entities to the database.



Keep in mind the database has no relationship to the grid other than defining its shape. The number and type of entities, how they are arranged in physical space, and the manner in which they are created have nothing to do with the grid's topology or number of points. The database entities may overlap or may contain gaps. In fact, a database isn't required for running Gridgen.

The database entities that you can create in Gridgen fall into one of five categories as listed below:

- points
- curves
- planes
- surfaces
- notes

6.2 Point

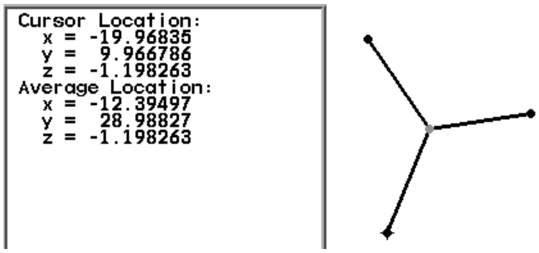
Point entities can be added to the database using the **Points** command in the DB Creation menu. Database points are most often used to:

1. Predefine all known locations that a curve should interpolate, so they may be easily picked during curve

creation (Section 6.3).

2. Create a new point at the average location of other points, such as the midpoint of a line.

The 3D cursor (Section 2.8) is used to identify locations at which point entities are to be created. Each point added is shown in the Display window as a white filled circle. You can create as many points as you need prior to pressing the **Done** button at which time each unique point is saved as a database entity.



```
Cursor Location:
  x = -19.96835
  y =   9.966786
  z = -1.198263
Average Location:
  x = -12.39497
  y =   28.98827
  z = -1.198263
```

You may also choose to create a point that is the average of the 3D locations of the other points you are creating. Once two points have been added the radio button **Keep Avg** becomes available (no longer greyed out). Toggling this button on will cause Gridgen to compute and display an additional point - the 3D average of the other points. The average point is rendered in the Display window as a yellow filled circle. Its coordinates are listed in yellow text in the Blackboard. In addition to the average point's location, the Display window also shows spokes (white lines) emanating from the average point to all of the points used in its computation. When the **Done** button is pressed, the average point is saved along with the others as a database entity.

Other commands involved in database point entity creation are:

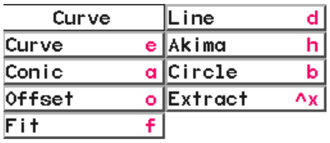
Keep Pts - This toggle button toggles the save status for the points being added; when toggled off, they will not be saved. You might do this when creating a point at the midpoint of two existing database locations.

Keep Avg - This toggle button toggles the save status for the average point; when toggled off, the average will not be computed, displayed, or saved.

Note that if both toggle buttons **Keep Pts** and **Keep Avg** are toggled off, then nothing will be saved when **Done** is pressed.

6.3 Curve

You can create eight types of database curves: Line, Curve, Akima Curve, Circular Arc, Conic Section, Offset, Extract, and Fit. Only three of these curve types, Offset, Extract, and Fit, are unique to the database commands. Creating the other curve types is similar to adding segments (Section 7.2) during connector creation.



6.3.1 Line

Creation of Line database entities is similar to creation of Line connector segments. Please refer to Section 7.2.1 for more information on creating Lines.

6.3.2 Curve

Creation of Curve database entities is similar to creation of Curve connector segments. Please refer to Section 7.2.1 for more information on creating Curves.

6.3.3 Akima Curve

Creation of Akima Curve database entities is similar to creation of Akima Curve connector segments. Please refer to Section 7.2.1 for more information on creating Akima Curves.

6.3.4 Conic Section

Creation of Conic Section database entities is similar to creation of Conic Section connector segments. Please refer to Section 7.2.3 for more information on creating Conic Sections.

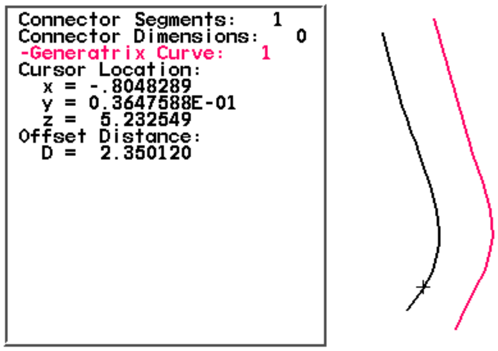
6.3.5 Circular Arc

Creation of Circular Arc database entities is similar to creation of Circular Arc connector segments. Please refer to Section 7.2.3 for more information on creating Circular Arcs.

6.3.6 Offset

Offset curve entities can be added to the database using the **Curve Offset** button in the DB Creation menu. Offset curves are the result of moving a curve (called the *generatrix*) a specified distance normal to itself. You have two controls during offset curve creation: the normal direction and the offset distance.

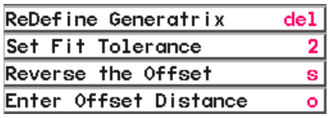
Because, in general, a 3D curve does not have a unique normal direction, Gridgen constrains the normal direction to the plane of the Display window. Therefore, the orientation you choose for the Display window image helps define the offset direction. An additional control is provided by the command **Reverse the Offset** (see below). It allows you to offset the generatrix in either of two directions.



```
Connector Segments:    1
Connector Dimensions:    0
-Generatrix Curve:   1
Cursor Location:
   x = -.8048289
   y = 0.3647588E-01
   z =  5.232549
Offset Distance:
   D =  2.350120
```

You can specify the offset distance directly, or you can specify a point that the curve should pass through. If you use the latter method of specifying the offset, you should be aware it is possible to choose a point through which the offset curve cannot pass. In that case, Gridgen will produce an offset curve that is the same 3D distance from the generatrix as the specified point yet does not pass through the point.

Other commands involved in offset curve creation are:



ReDefine Generatrix - This command lets you re-start the offset curve creation process if you pick the wrong generatrix.

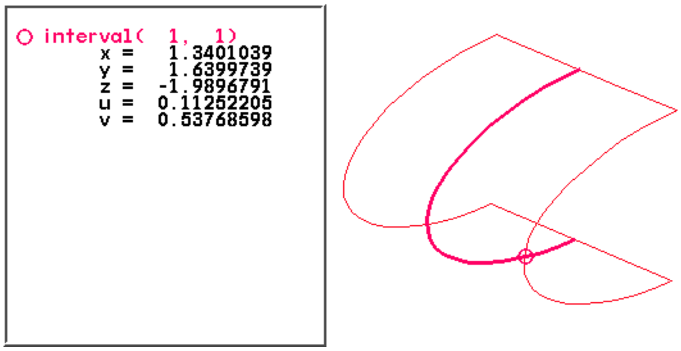
Set Fit Tolerance - This command lets you enter a precision tolerance for the offset curve. This determines how closely the offset curve will match the specified offset distance. The default fit tolerance is 0.001 as described in Section 3.7.1.

Reverse the Offset - This command switches the offset curve to the opposite side of the generatrix from its current position.

Enter Offset Distance - Rather than use the 3D cursor to define a point for the offset curve to pass through, you may use this command to type-in the offset distance.

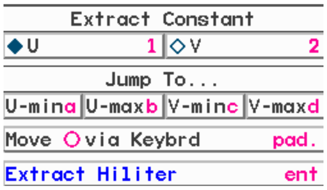
6.3.7 Extract

An extract curve is a curve entity created by copying a constant u or v parameter line from an existing database surface. Shown below is an example of the Blackboard and Display window as the constant parameter curve at the hiliter location (pink line on surface) is being defined for extraction.



```
○ interval(  1,  1)
       x =   1.3401039
       y =   1.6399739
       z =  -1.9896791
       u =  0.11252205
       v =  0.53768598
```

Commands used in Extract curve creation include:



Extract Constant **U** - Convert the hiliter to show lines of constant u parametric coordinate.

Extract Constant **V** - Convert the hiliter to show lines of constant v parametric coordinate.

Jump To **U-min** - Move the hiliter to the surface’s minimum u coordinate line.

Jump To **U-max** - Move the hiliter to the surface’s maximum u coordinate line.

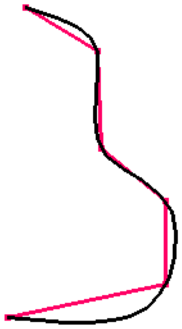
Jump To **V-min** - Move the hiliter to the surface’s minimum v coordinate line.

Jump To **V-max** - Move the hiliter to the surface’s maximum v coordinate line.

Move ❍ via Keybrd - Type-in the u or v coordinate at which to place the hiliter.

Extract Hiliter - Create a new database curve entity by extracting the curve at the hiliter location.

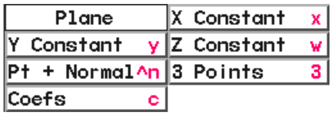
6.3.8 Fit



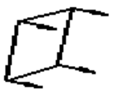
A fit curve is used to create a slope continuous, cubic polynomial curve (dark curve above) through a faceted, linear curve (light curve above). The fit curve will interpolate each of the linear curve's control points. The goal of the fit curve is to create a smooth representation of a faceted curve. Often you are given a geometry definition that is sparsely defined. Rather than use the sparse, faceted curve you may create a fit curve, which is more suitable for grid generation and some of the geometric calculations such as intersections.

6.4 Plane

Planes (infinite planar surfaces) may be added to the database for use in computing intersections of planes with curves and surfaces and for projection of grid entities. Gridgen provides several different plane creation methods as shown below.



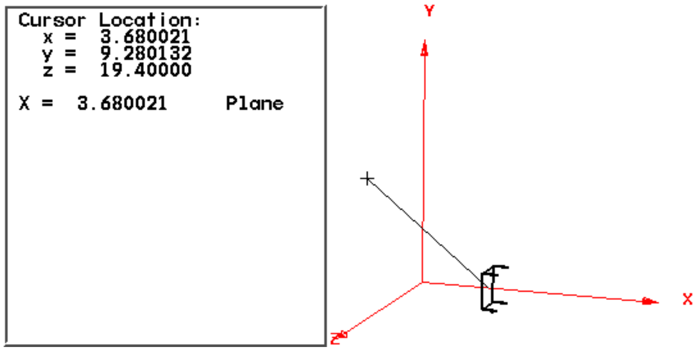
Since a plane by definition is infinite, Gridgen has adopted a finite representation for rendering in the Display window. A plane entity in Gridgen is rendered as a small quadrilateral with a vector emanating from each corner pointing in the plane's normal direction.



6.4.1 Constant Coordinate Plane

Planes at constant x, y, and z locations may be created by pressing the Plane **X Constant**, **Y Constant**, and **Z Constant** buttons, respectively.

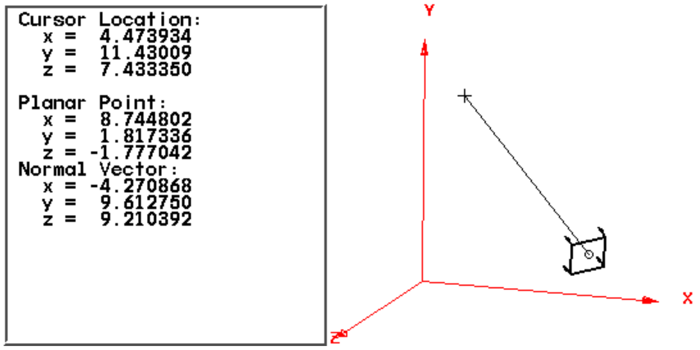
The 3D cursor (Section 2.8) is used to define the plane's location. The cross is moved around the screen by moving the mouse with the ***right mouse*** button pressed. The plane's position can be set using the standard control point creation buttons (Section 2.8.2). Once the plane's location has been set the plane's image will appear on the screen. Even though the plane can be defined at any location of the 3D cursor the plane icon will be rendered along the appropriate body axis. During creation of the plane, Gridgen draws a line between the cross location and the center of the plane icon. Save the plane by pressing the **Done** button.



```
Cursor Location:
  x =  3.680021
  y =  9.280132
  z =  19.40000

X =  3.680021      Plane
```

6.4.2 Point-Normal Plane

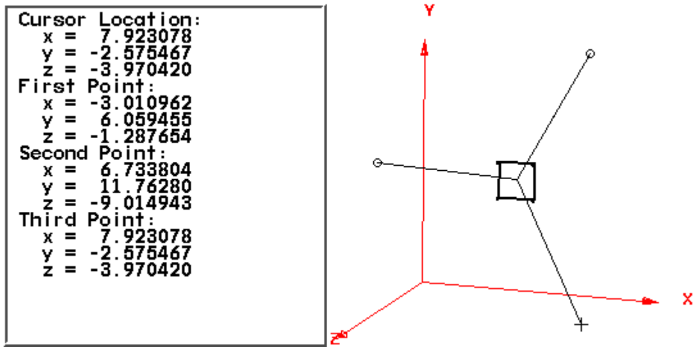
A plane can be created by specifying two points: a point P_1 that lies on the plane and a second point P_2. The vector P_2 - P_1 is the plane's normal vector. The 3D cursor is used to define both points, using the standard control point creation methods (Section 2.8.2). The first point (the point on the plane) is drawn as an open circle after it is added. After the second point's location is added, the plane icon is drawn at the first point's location with a line connecting P_1 and P_2. Save the plane by pressing the **Done** button.



```
Cursor Location:
  x =  4.473934
  y =  11.43009
  z =  7.433350

Planar Point:
  x =  8.744802
  y =  1.817336
  z = -1.777042
Normal Vector:
  x = -4.270868
  y =  9.612750
  z =  9.210392
```

6.4.3 3-Point Plane

A plane may be created by defining three points, P_1, P_2, and $P_{3,}$ that lie on the plane. The plane icon will be located at the average location of the three points. The 3D cursor is used to define all three points, using the standard control point creation commands (Section 2.8.2). Each point is drawn as an open circle after it is added. The triangle P_1 - P_2 - P_3 - P_1 is drawn during construction up until the third point is added. Then the plane icon is rendered at the average location, and spokes are drawn from the average location to each of the three points. The plane entity is saved by pressing the **Done** button.



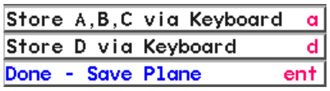
```
Cursor Location:
   x =   7.923078
   y =  -2.575467
   z =  -3.970420
First Point:
   x =  -3.010962
   y =   6.059455
   z =  -1.287654
Second Point:
   x =   6.733804
   y =   11.76280
   z =  -9.014943
Third Point:
   x =   7.923078
   y =  -2.575467
   z =  -3.970420
```

6.4.4 Coefficient Plane

A plane may be created by entering the coefficients of the plane equation

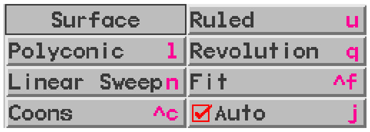
```
A x + B y + C z + D = 0
```

The button **Store A,B,C via Keyboard** allows you to enter the three coefficients defining the normal vector. A second button, **Store D via Keyboard**, allows you to define the location of the plane. The plane icon is rendered once the values for A, B, and C are entered. The plane entity is saved by pressing the **Done** button.



6.5 Surface

Gridgen has tools for creating several different types of surfaces. Except for the Fit surface, each tool uses database curves or existing surface boundary curves to construct database surfaces. Descriptions for each of the surface creation tools can be found in the following sections.



Ruled - fills in linearly between two curves that define opposite edges of the surface.

Polyconic - builds a fillet between two rail curves that define opposite edges of the surface. A third curve, called the shoulder curve, determines the depth of curvature of the surface.

Revolution - makes an axi-symmetric surface by rotating a generatrix curve around an axis.

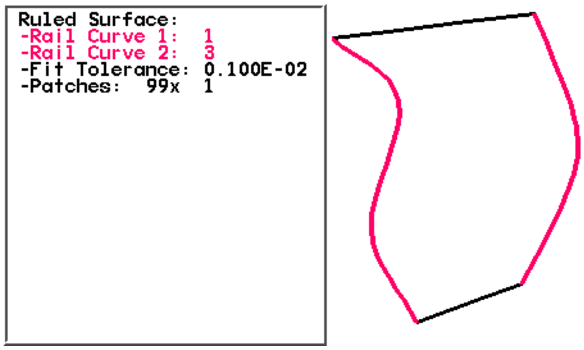
Linear Sweep - builds a surface by sweeping a generatrix curve along a specified linear rail curve.

Fit - creates a slope continuous, cubic polynomial surface through a bilinear surface.

Coons - constructs best possible surface from a group of user selected curves.

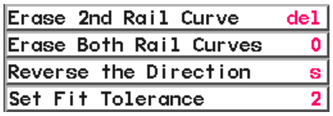
6.5.1 Ruled Surface

A ruled surface is constructed by selecting two curves (called *rails*) that define the surface's opposite boundaries. When the second rail is picked, the ruled surface is created and immediately displayed on the screen. For ruled surfaces, all constant parameter curves between the rails are straight lines.



```
Ruled Surface:
-Rail Curve 1:   1
-Rail Curve 2:   3
-Fit Tolerance: 0.100E-02
-Patches:   99x  1
```

Commands used during construction of a ruled surface are:



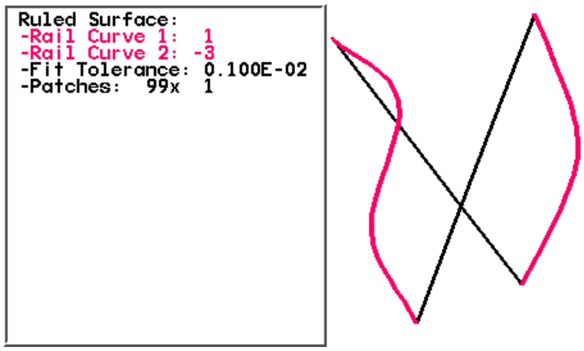
Erase 2nd Rail Curve	del
Erase Both Rail Curves	0
Reverse the Direction	s
Set Fit Tolerance	2

Erase 1st Rail Curve - After the first rail has been selected but before the second rail has been selected, this command allows you to pick another first rail curve.

Erase 2nd Rail Curve - After the second rail has been selected and before the ruled surface is saved, this command allows you to pick a different second rail curve.

Erase Both Rail Curves - This command allows you to start over by picking another set of rail curves.

Reverse the Direction - This command allows you to untwist the surface by implicitly reparameterizing the second rail curve. A twisted ruled surface (such as the one shown below) will result when the two rails are parametrized in opposite directions (the ruled surface is created by linearly interpolating between similar parameter values on the rails).



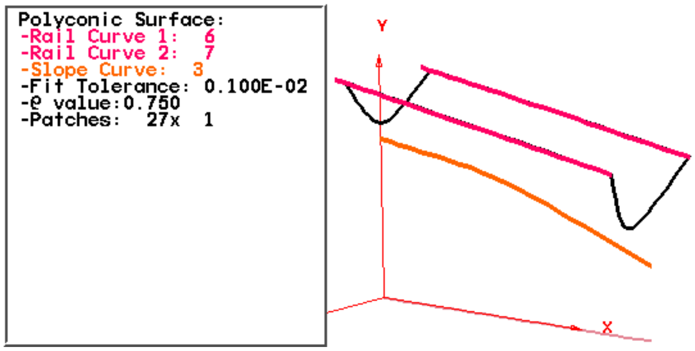
```
Ruled Surface:
-Rail Curve 1:   1
-Rail Curve 2:  -3
-Fit Tolerance: 0.100E-02
-Patches:   99x  1
```

Set Fit Tolerance - The tolerance with which the ruled surface matches the rails may be changed using this command. Decreasing the fit tolerance will result in a ruled surface with a larger number of intervals. The default fit tolerance is 0.001 (Section 3.7.1).

6.5.2 Polyconic Surface

A polyconic surface is the surface equivalent of a conic curve (Section 7.2.3) with control points replaced by control curves. Whereas a conic curve is defined by two end points, a tangent intersection point, and a ρ value, a polyconic surface is defined by two rail curves, a tangent intersection curve, and a ρ value. Polyconics are very useful for creating fillets.

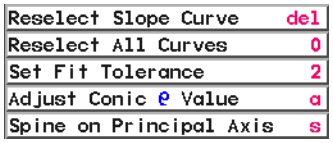
Construction of polyconics begins with graphical selection of the two rail curves. The third curve selected is the tangent intersection curve. By default, the tangent intersection curve serves double duty as the polyconic's spine. The resulting polyconic surface will be constructed such that the intersection of the surface with each plane perpendicular to the spine will represent a conic curve, defined by the rail and intersection curve points and the ρ value. The component curves of a typical polyconic surface are illustrated below.



```
Polyconic Surface:
-Rail Curve 1:  6
-Rail Curve 2:  7
-Slope Curve:  3
-Fit Tolerance: 0.100E-02
-ρ value:0.750
-Patches:  27x  1
```

You may optionally select a fourth curve representing the spine of the surface or you may elect to have the spine lie along any of the three principal axes. If no spine is selected, the tangent intersection curve serves double duty as the spine. The purpose of the spine is analogous to the axis required in the surface of revolution (Section 6.5.3); it allows polyconics to be constructed along general curve paths.

Commands used during construction of a polyconic surface are:



Reselect Slope Curve	del
Reselect All Curves	0
Set Fit Tolerance	2
Adjust Conic ρ Value	a
Spine on Principal Axis	s

Reselect Slope Curve - This command allows you to replace the current tangent intersection curve with a different one.

Reselect All Curves - This command allows you to restart the polyconic surface creation process.

Set Fit Tolerance - This command allows you to set the tolerance with which the polyconic surface will match the rail curves. By default, the fit tolerance is set to 0.001 (Section 3.7.1). Decreasing the fit tolerance improves the match between the surface and its defining curves but at the price of increasing the number of intervals in the surface.

Spine on Principal Axis - This command allows you to choose either the x, y, or z body axes to use for a spine instead of selecting a curve entity. By default the spine is the tangent intersection curve.

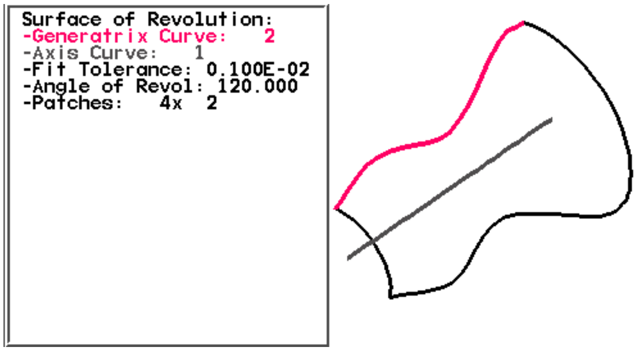
Adjust Conic ρ Value - This command allows you to set the value of ρ in the range $0 < \rho < 1$. The default ρ value is 0.5. The character of the polyconic changes with the ρ value as shown in the following table.

Effect of ρ on conic section type.

ρ	**conic section type**
0 < ρ < 0.5	ellipse
0.5	parabola
0.5 < ρ < 1	hyperbola

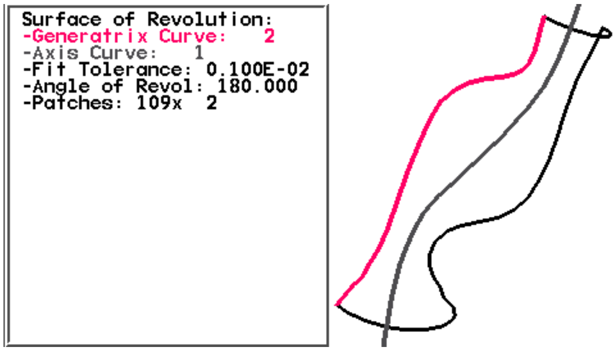
6.5.3 Surface of Revolution

A surface of revolution database entity is created by revolving a curve called the *generatrix* around a second curve called the *axis*. A characteristic of the resulting surface of revolution is that at any plane perpendicular to the axis, the intersection of the revolved surface and the plane is a circular arc. One edge of the surface of revolution will lie on the generatrix (two if the angle of revolution is 360 degrees). You can also specify an angle of revolution, in the range -360 to 360 degrees, excluding 0 degrees. Once the axis is selected the surface is initialized using the default angle of 360 degrees. A typical surface of revolution is depicted below.



```
Surface of Revolution:
-Generatrix Curve:   2
-Axis Curve:    1
-Fit Tolerance: 0.100E-02
-Angle of Revol: 120.000
-Patches:    4x  2
```

There are several interesting features regarding the axis in Gridgen's surface of revolution, which are illustrated in the figure below.

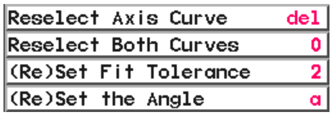


```
Surface of Revolution:
-Generatrix Curve:   2
-Axis Curve:    1
-Fit Tolerance: 0.100E-02
-Angle of Revol: 180.000
-Patches: 109x  2
```

- The axis of revolution does not have to be a line but can be any curve. This allows the creation of very general surfaces of revolution, including pipes with substantial bends. If the generatrix was created as an offset curve (Section 6.3.6) from the axis, then you can create a constant radius pipe.

- If the axis extends beyond the ends of the generatrix, the surface of revolution will be defined only up to the end of the generatrix.
- If the axis is shorter than the generatrix, the axis will be extended based on its end point slopes to the end of the generatrix.

Commands used during construction of a surface of revolution are:



Reselect Axis Curve	del
Reselect Both Curves	0
(Re)Set Fit Tolerance	2
(Re)Set the Angle	a

Reselect Generatrix - After the generatrix has been selected but before the axis has been selected, this command allows you to erase the current generatrix and select another one.

Reselect Axis Curve - After the generatrix and axis have both been selected, this command allows you to erase the current axis and select another one.

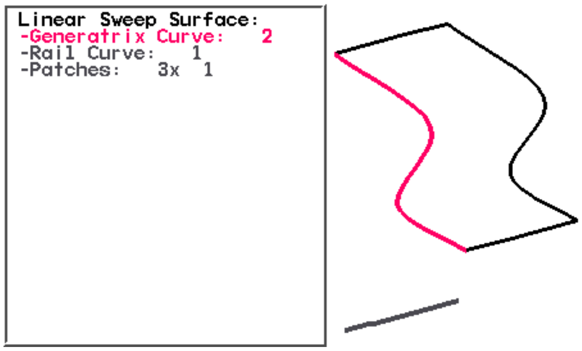
Reselect Both Curves - After the generatrix and axis have both been selected, this command allows you to erase both the current axis and current generatrix and start over.

(Re)Set Fit Tolerance - The tolerance with which the surface matches the generatrix is changed using this command. The default fit tolerance is 0.001 (Section 3.7.1). Decreasing the fit tolerance will result in a surface with a larger number of intervals.

(Re)Set the Angle - This command allows you to set the angle of revolution in the range 360 to 360 degrees. The default is 360 degrees. Setting the angle to 0 will result in a surface that will not be saved.

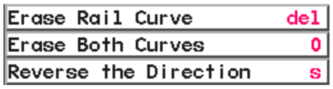
6.5.4 Linear Sweep Surface

A linear sweep surface, also called an extruded surface or a tabulated cylinder, can be created in Gridgen. First, select the curve to be extruded, called the *generatrix*. Then select a line that defines the sweep direction, called the *rail*. The rail must be a line (in other words, not a curve, circle, etc.), but it does not have to share an end point with the generatrix. The resulting surface starts at the generatrix's location and sweeps in the direction of and for the length of the rail.



Linear Sweep Surface:
-Generatrix Curve: 2
-Rail Curve: 1
-Patches: 3x 1

Commands used during construction of a linear sweep surface are:

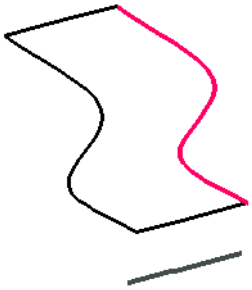


Erase Generatrix - After the generatrix has been selected but before the rail has been selected, you can use this command to erase the current generatrix and select another.

Erase Rail Curve - After the generatrix and rail have both been selected, you can use this command to erase the current rail and select another.

Erase Both Curves - After the generatrix and rail have both been selected, you can use this command to erase both the current rail and generatrix and start over.

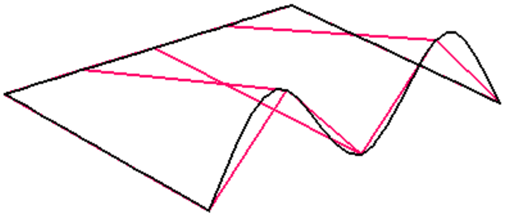
Reverse the Direction - This command allows you to reverse the sweep direction by implicitly reversing the parametrization of the rail. The figure below shows the results of reversing the sweep direction of the sweep surface shown above.



6.5.5 Fit

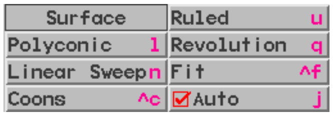
A fit surface is used to create a slope continuous, cubic polynomial surface through a bi-linear surface. The fit surface will interpolate each of the bi-linear surface's control points. The goal of the fit surface is to create a smooth representation of a faceted surface. Often you are given a geometry definition that is sparsely defined. Rather than use the sparse, faceted surface, you may create a fit surface which is more suitable for grid generation and for some of the geometric database calculations such as intersections

.

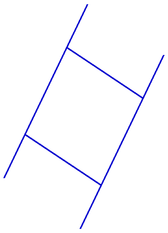


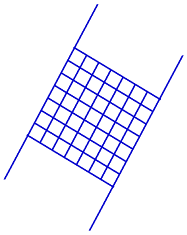
6.5.6 Coons

The Coons surface creation tool constructs a NURB surface based on a set of curves that form a closed perimeter. Coons surfaces can often be very useful in filling the holes and gaps sometimes found in database models. The **Coons** command requires the input of a set of four or less curves which can be existing database curves, surface boundaries, or a combination of both. If the chosen curves form a closed loop, a single Coons surface is created from the loop and the user is prompted to save the surface.

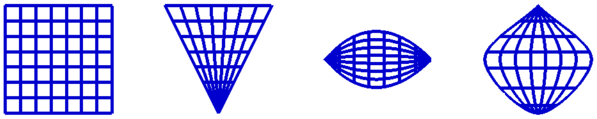


To create a Coons surface, a user will generally have to process potential boundary curves via the join, intersect, and/or split commands first. These curves must be processed so that they form one continous loop. The figure below on the left illustrates a example set of curves that need to be processed. To make these curves form a continuous loop, they must be intersected where they cross each other, then split at the intersection points. The user can then select **Coons** and use the four curves which form a closed perimeter. The figure below on the right shows a network representation of the Coons surface that would be created.



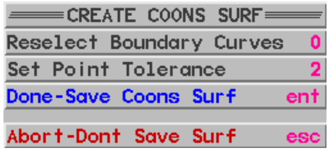


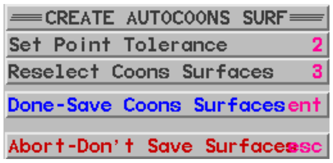
The type of Coons surface constructed will depend on the number of curves selected. If a user selects four curves, a surface similar to the one illustrated above on the right will be created. For three curves, the surface will be created with a singularity at the corner with the smallest interior angle. For two curves, the surface will be created with two singularities. For a case in which only one curve is selected, the curve will be split in half, and a two singularity surface is created from the resulting curves. It is important to note that for a single curve to be selected with Coons, it must be self-connecting. Each of these cases is illustrated sequentially below starting with the four curves case on the left.



Auto - This toggle allows the user to select up to fifty curves that will automatically be processed for the user. The Coons command will then be applied to create as many surfaces as possible from those curves. These potential surfaces will be listed for the user in the Browser list so that desired surfaces can be selected and saved. The **Coons Auto** option is limited to creating only 100 potential surfaces. If the desired surface is not created, the user should start over with a different curve set.

Once the Coons surface is created to the user's satisfaction, one of the following two menus will be presented. The menu on the left represents the Coons surface creation menu with the **Auto** toggle off, while the menu on the right is seen when the **Auto** toggle is on.




CREATE AUTOCOONS SURF
Set Point Tolerance 2
Reselect Coons Surfaces 3
Done-Save Coons Surfaces ent
Abort-Don't Save Surfaces esc

Commands that can be used during Coons surface creation are:

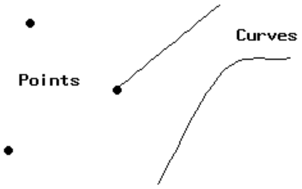
Reselect Boundary Curves - If a Coons surface could not be created on the selected curves or if the wrong curves were chosen by mistake, the user has the option of reselecting the boundary curves.

Reselect Coons Surfaces - This option is available only with the **Auto** option when more than one surface can be created out of the chosen set of curves. This option returns the user to the Browser list of potential Coons surfaces.

Set Point Tolerance - The point tolerance is used to check the selected database curves. The point tolerance checks curve intersections, coincident curves, and curve endpoint matching to determine if curves need to be split, deleted, or joined as a vertex, respectively.

6.6 Note

Notes (single line character strings limited to forty characters) may be added to the database. Notes are useful for providing a descriptive title, creation instructions, or brief reminders. The 3D cursor is used to position the note's text. Pressing the **Done** button saves the note and keeps Gridgen in note creation mode. The **Abort** button may be used to exit from note creation mode.



Other commands used during note creation are:

Enter the Note - This command allows you to enter up to 40 characters of text.

7. Creating Connectors

7.1 Overview

Connectors are Gridgen's lowest level (one dimensional) grid element. They are curves on which grid points will be distributed. The complete definition of a connector includes three attributes:

shape - as defined by one or more segments

dimension - the number of grid points on a connector (Section 11.1)

distribution - the distribution of grid points along a connector (Section 11.3)

As described in the list above, a connector's shape is defined by its segments. A segment is a specific mathematical curve type. Therefore, a connector is what computer aided design (CAD) software would refer to as a composite curve. Some important connector terminology is defined in the following list:

connector - a point-wise (C0) continuous series of segments

segment - a specific type of curve that defines how the curve shape is derived from the control points

control point - a 3D location in (`x,y,z`) space through which a segment will pass

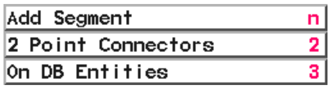
node - the control point at either end of a connector, displayed as a filled circle

Connectors are color coded according to their current state in the grid as listed in the following table. Connector colors are not user-definable.

Connector color scheme.

color	meaning
bright green	un-dimensioned
light green	dimensioned
light blue	used in a domain
dark blue	used in a block

Create a connector by selecting the **Create** command from the Connector menu. There are three methods for creating connectors as shown below.



Add Segment - You create one connector by creating each of its segments individually.

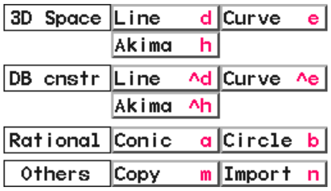
2 Point Connectors - You create connectors rapidly by defining two control points. If the two points lie on the same database entity the connector’s shape is “snapped” to the database. Otherwise, lines are created.

On DB Entities - Many connectors are created at one time by selecting database entities. A connector will be created on each selected curve database entity. Four connectors, one on each boundary, will be created on each selected surface database entity. Connectors will be created on each trimming curve of selected trimmed surface database entities.

Each of these connector creation methods is described in the following sections.

7.2 Add Segment

You may create many different types of segments as shown in the menu below.



Gridgen’s segment types are grouped into four categories:

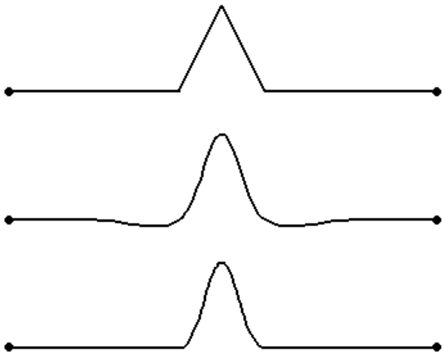
1. 3D Space - These segments are free to be created anywhere in three dimensional space.
2. DB cnstr - These segments are constrained to lie on the database.
3. Rational - This category covers curves that are mathematically rational such as circles and conics.
4. Others - These are segments created by copying other connectors or importing data from a file.

The **Don’t Add Segment** button ends the segment adding process and returns you to the Create menu.

There are other segment types that cannot be created directly. The available segment types are described in the following sections.

7.2.1 3D Space Segments

The 3D Space segments are simple polynomial curves created between successive user-defined control points in three dimensional (x, y, z) space. At least two control points are required to create 3D Space segments. The Line, Curve, and Akima segment types differ in how the polynomial is computed.



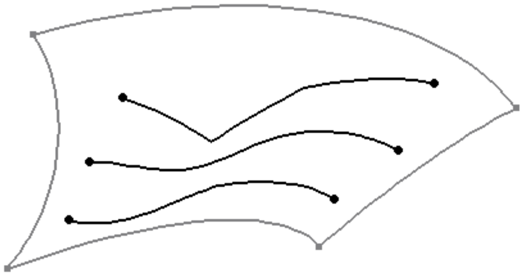
The 3D Space **Line** segment (above top) is a piece-wise linear curve (a polyline).

The 3D Space **Curve** segment (above center) is a cubic polynomial in (x,y,z) space between successive control points, resulting in a curve with continuous slope at interior control points. The cubic polynomial is constructed so the slope at a given control point is parallel to the chord between the two adjacent control points (Catmull-Rom spline). At the Curve segment's end points, control points from adjacent segments are used to compute the slopes. If no adjacent segment exists, the curve slope will be tangent to the end control point and adjacent control point.

The 3D Space **Akima** segment (above bottom) is a cubic polynomial between successive control points, resulting in a curve with continuous slope at interior control points. The cubic polynomial is constructed using a stencil of five adjacent control points to account for possible linear slopes. Akima segments generally have fewer "wiggles" than Curve segments and should probably be used when a smooth curve with minimal overshoot is needed.

7.2.2 Database Constrained Segments

The database constrained segments are simple polynomial curves created between successive user-defined control points in the two dimensional (u,v) space of a database entity. At least two control points are required to create database constrained segments. The Line, Curve, and Akima segment types differ in how the polynomial is computed.



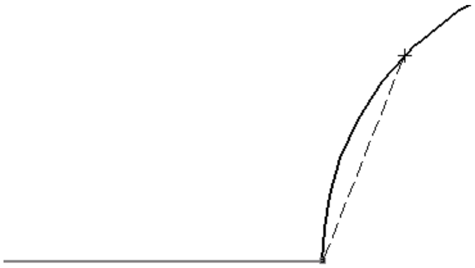
The DB cnstr **Line** segment (above top) is a piece-wise linear curve (a polyline). It is linear in (u,v) space and not necessarily in (x,y,z) space, as you can see from the illustration. This segment type should be used whenever you want to create a connector along a constant parameter line on a database entity (for example, a surface's boundary). You should always use this segment type when creating a segment on a database curve.

The DB cnstr **Curve** segment (above center) is a cubic polynomial in (u,v) space between successive control points. The segment will not necessarily be cubic and slope continuous in (x,y,z) space. The cubic polynomial is constructed so the slope at a given control point is parallel to the chord between the two adjacent control points (Catmull-Rom spline). At the Curve segment's end points, control points from adjacent segments are used to compute the slopes. If no adjacent segment exists, the curve slope will be tangent to the end control point and adjacent control point. Do not use this segment type if your intent is to trace a constant parameter line (for example, a surface's boundary or a curve entity). Use a database constrained Line instead.

The DB cnstr **Akima** segment (above bottom) is a cubic polynomial between successive control points. The segment will not necessarily be cubic and slope continuous in (x,y,z) space. The cubic polynomial is constructed using a stencil of five adjacent control points to account for possible linear slopes. Akima segment generally have fewer "wiggles" than Curve segments and should be used when a smooth curve with minimal overshoot is needed. Do not use this segment type if your intent is to trace a constant parameter line (for example, a surface's boundary or a curve entity). Use a database constrained Line instead.

Any of the database constrained segments may span more than one database entity. However, at least two control points must be defined on each entity. When adjacent control points lie on different entities, the segment between the points is drawn as a dashed line called a *branch segment*. There are two instances when this occurs:

1. If this is not the first segment in the connector, then a branch segment will be inserted between the previous segment end point and the current segment beginning point to maintain the point-wise continuity of the connector.
2. If a single database constrained segment spans two entities, then the portion of the segment that crosses between the two entities will be a branch segment.



Keep in mind that branch segments are straight lines between the two control points on adjacent entities. It is up to you to ensure the shape of the branch segment does not deviate unacceptably from the shape of the database. In the illustration above, a connector is being drawn across two database entities, a horizontal line and a circular arc. Two control points have already been added on the line. The last is at the junction of the line and circle. The cursor has already been moved to the circle entity and the branch segment is being displayed as a dashed line. The deviation of the branch segment's linear shape from the intended circular arc shape is readily apparent. Minimize this deviation by keeping the branch segment as short as possible. However, you should not use coincident control points (putting the first control point on the circular arc at the line-arc junction) since this will cause a lot of editing and grid point distribution problems later.

The database entity on which the segment is being created may be selected using the following commands. As you move from entity to entity, the 3D cursor is positioned at the closest point on the database entity to the last saved control point in the segment.

Next - Move to the next database entity in increasing numerical order.

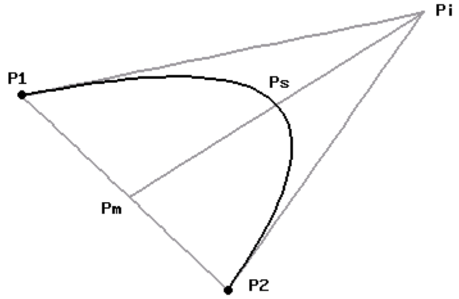
Prev - Move to the previous database entity in decreasing numerical order.

Pick - Graphically select the database entity on which to create the segment.

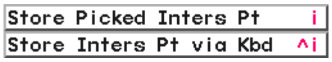
7.2.3 Rational Segments

7.2.3.1 Conic

The Conic Section segment is used to create a general conic section spanning an arc of no more than 180 degrees. Since conics are essentially quadratic polynomials, they are restricted to lie within a plane; however, the plane need not be at any particular orientation. The Conic Section segment is defined by three control points as shown below. The first control point P_1 defines the beginning of the segment, and the second control point P_2 defines the ending of the segment. The third control point defines the tangent intersection point P_i. The initial location of P_i is midway on a line connecting P_1 and P_2 called P_m. As P_i is moved via the cursor, the conic section segment is drawn as both a curve and two tangent slope lines. The segment passes through P_1, P_s, and P_2, where the shoulder point P_s is the point on the conic that intersects the line between P_i and P_m.



Instead of the third point in the conic definition being the shoulder point P_s you may choose to position the tangent intersection point P_i using the menu buttons shown below.



Store Picked Inters Pt i
Store Inters Pt via Kbd ^i

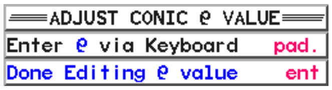
Store Picked Inters Pt - Position P_i at a picked point.

Store Inters Pt via Kbd - Enter the location of P_i via the keyboard.

In addition to positioning point P_i, you can also adjust the ρ value of the conic. ρ will approach 0 as the shoulder point P_s approaches P_m, and ρ will approach 1 as the shoulder point P_s approaches P_i. By default ρ is set to 0.5.

Adjust Conic ρ Value j

Adjust Conic ρ Value - Pressing this button brings up the following menu, allowing you to graphically adjust ρ using the mouse or to enter a ρ value via the keyboard.



ADJUST CONIC ρ VALUE
Enter ρ via Keyboard pad.
Done Editing ρ value ent

To adjust the ρ value move the mouse horizontally while pressing the ***right mouse*** key. Moving the mouse to the right increases the ρ value (upper limit is 1), and moving the mouse to the left decreases the ρ value (lower limit is 0). The mathematical nature of the conic section is defined by the ρ value as shown in the following table.

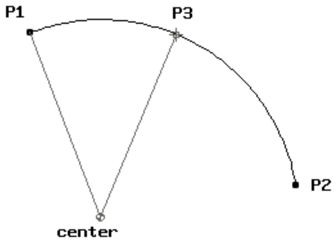
Effect of ρ value on conic section.

ρ value	conic section type
$0 < \rho < 0.5$	elliptic
0.5	parabola
$0.5 < \rho < 1$	hyperbolic

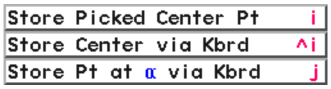
An exact ρ value may be typed in with the **Enter** ρ **via Keyboard** menu button, and control can be passed back to the tangent intersection point with the **Done Editing** ρ **Value** menu button. Further editing of the tangent intersection point will not change the value of ρ input earlier. It will not be possible to edit the ρ value unless the current edit point is the tangent intersection point. If one of the other control points is being edited, use one of the Edit... buttons to edit the tangent intersection point, then select the **Adjust Conic** ρ **Value** menu button.

7.2.3.2 Circular Arc

The circle segment allows you to define a circular arc of not more than 180 degrees by selecting three control points on the circle. The first control point P_1 specifies the beginning of the arc, and the second control point P_2 specifies the end of the arc. When P_2 is defined, a third control point P_3 is initially drawn halfway between P_1 and P_2 and the current center of the circle (as defined by the three control points) is drawn with a "center of gravity" symbol and is listed in the Blackboard. The menu changes to the Edit Circular Arc menu and P_3 can be moved using the mouse to modify the shape of the circular arc. The circular arc is drawn so it passes through points P_1, P_3, and P_2. P_3 can be positioned anywhere on the circular arc. Gridgen will restrict the cursor movement so the circular arc will subtend no more than 180 degrees. Arcs spanning more than a semicircle must be created by more than one segment.



Instead of the third point in the circle definition being P_3 you may choose to position the center point directly.



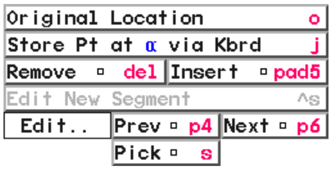
Store Picked Center Pt - Pick a point at which the circular arc's center will be located.

Store Center via Kbrd - Type-in the point at which the circular arc's center will be located.

Store Pt at α **via Kbrd** - Type-in the angle relative to P_1 at which P_3 will be located.

7.2.3.3 Slide Points

Although only three points are used to define rational segments, you can add other control points to the segment's definition. These control points do not alter the shape of the circular arc or conic section but simply "slide" along the shape. For this reason they are referred to as slide points. They provide a control point location that may be used for setting a break point during grid point distribution (Section 11.3.3) and to provide a packable location on the arc.



Original Location - return the slide point to the location it had before it was moved.

Store Pt at α **via Kbrd** - type-in the relative angle from the first control point at which the slide point is to be added.

Remove □ - delete the slide point.

Insert □ - add a slide point to the segment.

Edit **Prev** □ - move the 3D cursor to the control point before the current point.

Edit **Next** □ - move the 3D cursor to the control point after the current point.

Edit **Pick** □ - select the control point to edit.

7.2.4 Other Segments

7.2.4.1 Copy A Connector

The Copy segment type allows previously defined connectors to be copied and translated to a new location. This segment type is often useful for constructing domain or block edges opposite those already created. After selecting this command, position the cursor over the connector to be copied, and press the ***right mouse*** button. Next, pick one of the two highlighted nodes of the selected connector using the cursor and ***right mouse*** button. This node will become the beginning point of the new segment. If other segments have already been defined in the current connector, the Copy segment will automatically be translated to the end of the previous segment. If no previous segments have been defined, the new copied segments can be translated using the picked node as the control point and by positioning the control point in the usual manner. This segment type is the only type that can create more than a single segment at a time (since the copied connector may consist of several segments).

It is important to note that if the connector you choose to copy has any database constrained segments, then that constraint will be lost in the copy. The shape will be maintained but the underlying definition of control points will be altered.

7.2.4.2 Import

The Import segment type provides a means of using discrete geometry data from an external source to define a connector's shape. Using the File Browser (Section 2.5), first enter the name of the file containing the data, and then select one of the two highlighted end points of the input data using the cursor and the ***right mouse*** button. This end point will serve as the beginning of the new segment. If other segments have already been defined in the current connector, the input segment will automatically be translated so the chosen node will be located at the end of the last segment. If no other segments have been defined, the imported segment can be translated and positioned in the usual manner.

The resulting segment will be of the Line type, with a control point located at each data point in the file. The input file format is described in Section 18.5. It is the same file type used to import a database curve (Section 4.4). Consecutive, coincident control points are automatically deleted from this segment at the time of import.

7.2.5 Miscellaneous Segments

7.2.5.1 Pole

A pole segment is a segment of zero length. Poles are not created directly. Instead, you create a Line segment (Section 7.2.1) with two coincident control points (Gridgen will warn you about the coincidence). The pole connector is drawn in the Display window as a node surrounded by a large open circle.

7.2.5.2 Bezier Curve

Once a Line, Curve, or Akima Curve segment is created and saved, it may be edited and converted to a more general mathematical form known as a Bezier Curve. This provides explicit user control of control point locations and slopes. When in the edit mode, control points are adjusted in the usual manner, but now the curve's slope at a control point can be manually adjusted by selecting either of the Edit.. menu buttons, color-coded so that blue refers to the slope preceding the control point, and red refers to the slope following the control point. Upon selection, the cursor will move to the solid filled blue or red slope point, which you can adjust in the same manner as regular control points.

Editing of Bezier curves is described in more detail in Section 14.9.

7.2.5.3 Bezier on DB

This segment type is the surface equivalent of the Bezier segment (Section 7.2.5.2). All control points are constrained to database entities, and the resulting segment shape will lie exactly on the entity(s). Like the Bezier Curve segment, this type of segment can only be formed by editing an existing segment and by converting it to the Bezier form.

Editing of Bezier curves is described in more detail in Section 14.9.

7.2.5.4 General Segment

General segments are never constructed explicitly, but come about whenever a segment is modified in a way that destroys the Bezier properties of the segment. For example, if a connector on a database entity is translated off the entity, the Bezier properties formerly defined in terms of the parametric (u,v) coordinates will no longer hold, since (u,v) are defined only on the entity. In order to preserve the shape of the original segment (after translation, that is), Gridgen stores the original segment shape discretely and uses it to define the new segment.

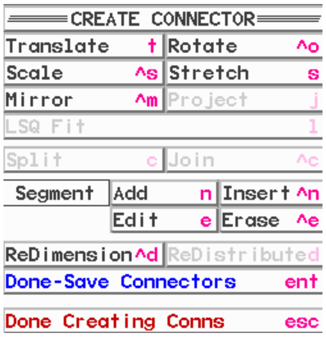
General Segments will arise under these circumstances:

1. a database-constrained segment is translated, stretched, scaled, or rotated from its original position
2. a segment is projected onto database entities
3. a circular arc or conic section segment is split, scaled or stretched
4. a Gridgen file is imported without its database
5. a domain is split

Each scenario is an example of a modified segment that can no longer be represented in a Bezier form while maintaining the same shape. Editing of General Segments is very similar to modifying Bezier Curve segments, except Gridgen attempts to preserve the segment's original shape where possible. Whenever the control point cursor is moved to a control or slope point that has its original shape, you are warned that adjustment of the control/slope point will destroy the local shape of the curve. Once moved, the region of the segment between the control point and its two neighbors will generally be impossible to restore to its original location.

7.2.6 Miscellaneous Commands

As stated previously, a connector may consist of more than one segment. After saving a segment you have several options according to the following menu.



Translate - Translate the connector (Section 14.3.1).

Rotate - Rotate the connector (Section 14.3.4).

Scale - Scale the connector (Section 14.3.2).

Stretch - Stretch the connector (Section 14.3.3).

Mirror - Mirror the connector.

LSQ Fit - Perform a least-squares fit of the connector’s grid points.

Project - Project the connector onto the database (Section 14.4).

Add - Add another segment to the end of the connector (Section 7.2).

Insert - Insert a segment in front of a segment already in the connector.

Edit - Modify the control points of a segment already in the connector (Section 14.9).

Erase - Remove a segment from the connector definition.

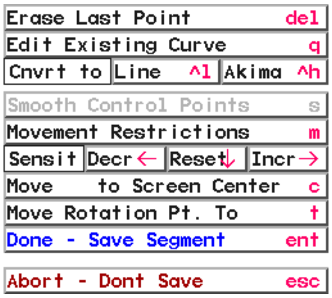
ReDimension - Change the number of grid points on the connector (Section 11.1).

ReDistribute - Change the distribution of grid points along the connector (Section 11.3).

Done - Save Connectors - Save the current connector definition.

Done Creating Conns - Abort the connector creation process without saving the current connector.

Several commands are available when creating and editing segments for adding control points and manipulating Gridgen’s 3D cursor. These commands include:



Erase Last Point - Remove the control point most recently added to the segment definition. The cursor is moved to the deleted control point's location.

Edit Existing Curve - Save the current segment definition and invoke the **Edit** command (Section 14.9).

Cnvrt to... - Change the segment's type to the chosen type while keeping the same set of control points. You can convert between Line, Curve, Akima, Bezier.

Smooth Control Points - Apply a LaPlace filter to all of the connector's control points in order to smooth its shape.

Movement Restrictions - This command invokes a menu that allows you to restrict the motion of the 3D cursor to aid in drawing the segment.

Sensit - These commands allow you to adjust the sensitivity of the 3D cursor to mouse movement.

Move + to Screen Center - Move the 3D cursor to the center of the Display window. For a database constrained 3D cursor, if the location of the screen center is outside the current database entity's boundaries the 3D cursor is moved to the location on the entity that is closest to the screen center.

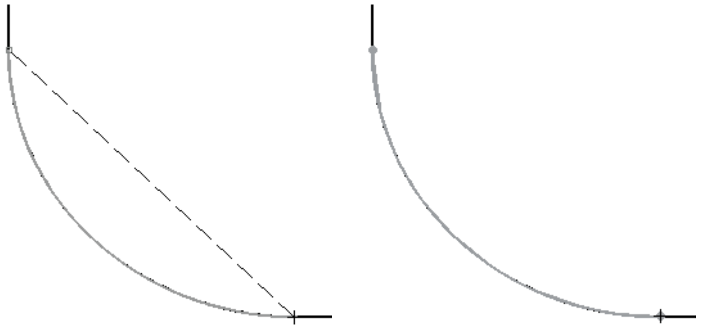
Move Rotation Pt. To + - Move the rotation point (the graphical origin of the body axes) to the current location of the 3D cursor.

Done - Save Segment - Save the current segment definition and add it to the connector definition. Control is passed out of segment creation/insertion.

Abort - Don't Save Segment - Exit segment creation mode without saving the developed segment.

7.3 2 Point Connectors

The 2 Point Connector is a means of very quickly creating many connectors. The resulting connectors will consist of only one segment each. They will be either 3D Space Line or DB Constrained Line segments. As the name implies, a 2 Point Connector consists of only two control points. You define them using the 3D cursor. Gridgen then examines each of the two points and checks whether or not they reference the same database entity. If they do not, Gridgen creates a 3D space line segment between the two points. If both control points are on the same database entity, Gridgen creates a database constrained line between the two points. The image on the left below shows a 2 Point Connector during creation after the first point (upper left) has been saved and the dashed line connects it to the cursor location before the **Store at Picked Pnt** button has been pressed. After the second point is picked from the same database entity the connector shape is snapped to the database curve (as shown on the right below) and the next 2 Point Connector is started with the cursor at the second point of the previous connector.

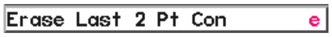


When Gridgen "snaps" the 2 Point Connector to the database it follows this hierarchy.

1. Gridgen determines the shortest distance on the database entity between the two points and creates a database constrained line segment.

2. If the shortest distance is already defined by an existing connector, Gridgen tries to find a second path between the two points.
3. If the shortest path and the alternate path are both already defined by connectors, Gridgen creates a 3D space line segment between the two points.

Other controls are available during creation of 2 Point Connectors.



Erase Last 2 Pt Con - This command allows you to delete the last 2 Point Connector that you created in case the wrong points were selected.

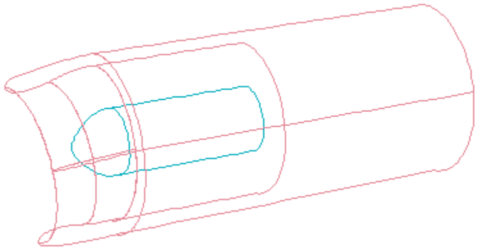
Erase Last Point - This command allows you to delete the first point of a 2 Point Connector, after the first point has been added but before the second point has been added.

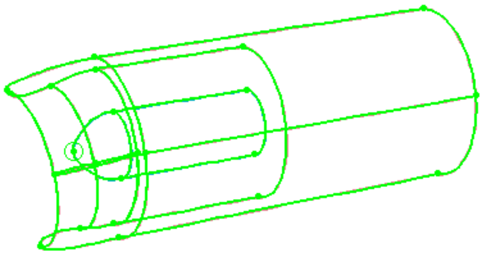
7.4 On DB Entities

The **On DB Entities** command allows you to rapidly create many two point connectors at one time. The connectors created by this command are the same as 2 Point connectors in that they consist of a single segment with two control points. The connectors to be created depend on the database entities picked:

1. One connector is created on each curve entity.
2. Four connectors are created on each surface entity, one connector per boundary.
3. One connector is created on each trimming curve of a trimmed surface entity.

The illustrations below show before (top) and after (bottom) pictures of an engine nacelle database and its connectors that were created using the **On DB Entities** command.





8. Grid Extrusion Methods

Domains and blocks may be created by extruding connectors and domains, respectively, along prescribed paths. Although the extrusion methods are a means of creating domains and blocks (Section 9 and Section 10) they are described in this section because the resulting grids depend strongly on the grid methods. Three types of entities may be created by extrusion:

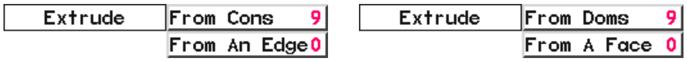
1. Structured domains extruded from connectors
2. Structured blocks extruded from structured domains
3. Prism blocks extruded from unstructured domains

Five different extrusion methods may be applied:

1. Hyperbolic: a partial differential equation (PDE) based method for extruding structured domains and blocks. The hyperbolic PDE extrusion method is the most mathematically elegant of Gridgen's extrusion methods and produces grids of very high quality that are smooth, clustered, and orthogonal.
2. Normal: an algebraic method mimicking the properties of the hyperbolic PDE method but which is also applicable to the extrusion of prism blocks from unstructured domains.
3. Rotation: an algebraic method for rotating grids about a user specified axis.
4. Translation: an algebraic method for linearly extruding grids.
5. Path: an algebraic method for extruding grids along an arbitrary 3D path defined by connectors.

8.1 Running the Extrusion

To create a domain or block by extrusion you press the **Create** button in the DOMAIN COMMANDS or BLOCK COMMANDS menu, respectively. From the CREATE DOMAIN or CREATE BLOCK menu you then press one of the two buttons labeled Extrude as shown below for domains (left) and blocks (right).



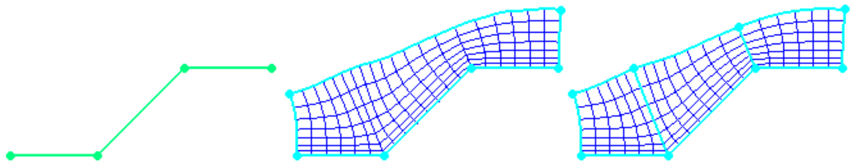
The first decision you must make is the manner in which the extrusion is to take place.

1. For domains:
   a. **From Cons** - A separate extrusion is performed on each individual *connector* that you select. When selected connectors share a node, a floating boundary condition is automatically applied so the extruded grids blend together smoothly. Otherwise, a unique domain is formed by extrusion from each connector.
   b. **From An Edge** - All selected connectors are joined into a single *edge* from which the extrusion takes place forming a single new domain.
2. For blocks:
   a. **From Doms** - A separate extrusion is performed on each individual *domain* that you select. When

selected domains share a connector a floating boundary condition is automatically applied so the extruded grids blend together smoothly.

b. **From A Face** - All selected domains are joined into a single *face* from which the extrusion takes place forming a single new block.

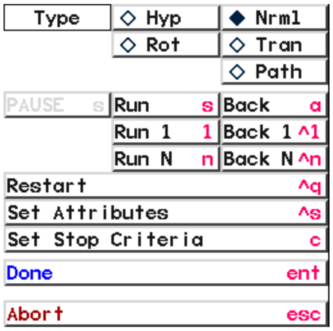
In the example below, the results of extruding three connectors (left) using **From an Edge** (center) and **From Cons** (right) are compared. Notice that extruding **From an Edge** results in a single domain; whereas, extruding **From Cons** results in three domains, one for each of the original connectors.



Whenever possible, we recommend you extrude **From An Edge** or **From A Face**. The grid quality and stability of the extrusion algorithms are both improved when floating boundary conditions (such as you would get automatically when using **From Cons** or **From Doms** and selecting multiple adjacent connectors or domains, respectively) are avoided.

After choosing the extrusion basis, you select the connector(s) or domain(s) that will form the initial grid for the extrusion.

Then the Extrusion menu appears as shown below.



You specify one of five extrusion methods from the radio buttons labeled Type on the Extrusion menu. Note that several extrusion types may be employed in a single extrusion session. For example, you can use a translational extrusion to create a straight section and then switch to a rotational extrusion to add a curved section before saving the final grid with the **Done** button.

Hyp - The hyperbolic extrusion is based on the solution to a set of partial differential equations (Reference 4, Reference 26). The PDEs prescribe a trajectory for the extruded grid lines that is orthogonal to the initial grid, that follow a prescribed marching step size variation in the extrusion direction, and that is very smooth. Because the hyperbolic method is PDE based it is not applicable to the extrusion of prism blocks from triangular (unstructured) domains.

Nrml - The normal extrusion is an algebraic method with results that mimic the hyperbolic PDE method (orthogonality and prescribed marching step sizes), but which is applicable to extrusion of both structured and prismatic grids.

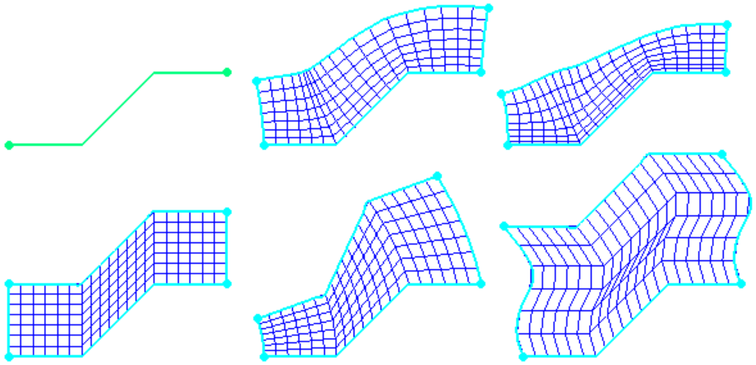
Rot - The rotational extrusion is an algebraic method that sweeps the grid in a circular path around a prescribed axis.

Tran - The translational extrusion is an algebraic method that sweeps the grid in a linear path in a prescribed direction.

Path - The path based extrusion is an algebraic method that allows you to sweep the grid along a path defined by a connector that you select.

The extrusion types are illustrated in the following figure:

- top left - the original connector
- top center - normal
- top right - hyperbolic
- bottom left - translation, in the vertical direction
- bottom center - rotation, 20 degrees about an axis (not shown) to the left of the lower left node in the grid
- bottom right - path, along the curve connector on the right side



Once you have selected an extrusion type, you apply it to the grids you selected by running the required number of marching steps. The extrusion methods are a process by which the initial grids you select are marched outward step by step according to the current attributes. Each step results in a new layer of grid cells, whether they're quadrilaterals, prisms, or hexahedra. The Display window is updated after each marching step so you can monitor the extrusion process. Controls for the extrusion process are described below.

Pause - Pressing this button causes the extrusion process to stop.

Run - Pressing this button begins the extrusion process which will continue until you press the **Pause** button.

Run 1 - Extrudes one step.

Run N - Extrudes `N` steps, where `N` is a number you type in.

Back - Reverses the extrusion process, backing up and removing extruded cells, stopping either when you press **Pause** or all of the extruded cells are removed.

Back 1 - Back up one step.

Back N - Back up `N` steps, where `N` is a number you type in.

Restart - Remove all extruded cells.

Set Attributes - Set a variety of attributes that will control the extrusion.

Set Stop Criteria - Set grid quality parameters which, when exceeded, will pause the extrusion.

Done - Stop the extrusion process, save all extruded grids, and exit the Extrude menu.

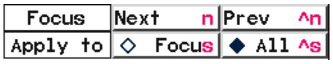
Abort - Stop the extrusion process, remove all extruded grids, and exit the Extrude menu.

8.2 Extrusion Attributes

You can change attributes at any time in the marching solution. For example, one set of attributes can be applied for the first 20 marching steps and another set for the next 40. The various attribute categories, some of which are only available for certain extrusion types, are described in the following subsections.

8.2.1 Focus

The Focus and Apply to buttons are available when you are extruding either **From Cons** or **From Doms** and you select more than one connector or domain for your initial grid, respectively. The focus grid is the one highlighted in the Display window and whose attributes are listed in the Blackboard.



Focus **Next** - Move focus to the next selected grid in numerical order.

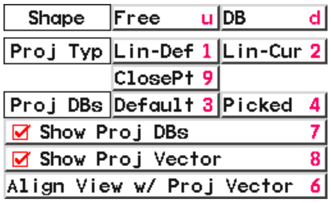
Focus **Prev** - Move focus to the previous selected grid in numerical order.

Apply to **Focus** - Attributes you select will be applied only to the focus grid.

Apply to **All** - Attributes you select will be applied to all grids, regardless of which grid is the focus grid.

8.2.2 Surface Shape

Domains created by extrusion either can obtain their shape purely from the mathematics of the extrusion or can be constrained to the database. Surface shape attributes are not available when creating blocks by extrusion. The surface shape attributes are described below.



Shape **Free** - The shape of the surface grids will result solely from the mathematics of the extrusion method.

Shape **DB** - The surface grids will be constrained to the database while they are being extruded.

Proj Typ **Lin-Def** - This attribute tells Gridgen the projection type is linear and the default projection orientation will be used. The default projection orientation is computed by averaging the grid's normal vector at each grid point. The projection vector direction is shown in the Display window as an arrow at each corner of the grid.

Proj Typ **Lin-Cur** - This attribute tells Gridgen the projection type is linear and that points will be projected onto the database along a fixed vector perpendicular to the Display window (screen-z) when the button is pressed.

Proj Typ **ClosePt** - This attribute tells Gridgen the projection type is closest point and a projection vector is not required.

Proj DBs **Default** - The grids will be projected onto the default set of database entities. The default database entities are computed for each grid based on which entities are currently referenced by the grid points. The entities to be projected onto are highlighted in the Display window.

Proj DBs **Picked** - The grids will be projected onto database entities selected by you. After this button is pressed you may select database entities from either the Browser or Display window. The current set of projection entities is already picked when the Browser is first invoked.

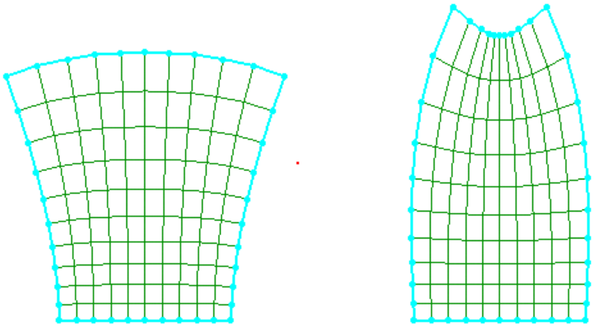
Show Proj DBs - This toggle button controls whether or not the database entities being projected onto will be highlighted in the Display window. Turning off this toggle will un-clutter the display.

Show Proj Vector - This toggle button controls whether or not the projection vector with which grid points are projected onto the database entities, is rendered in the Display window. Turning off this toggle will un-clutter the display.

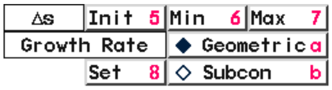
Align View w/ Proj Vector - This command changes the orientation of the Display window image so that the focus grid's projection vector is parallel to screen-`z`.

8.2.3 Aspect Ratio

The **Aspect Ratio Factor** adjusts the local marching step size based on the size of the local cell. At each step, the smallest cell will receive the initial physical step size. All other cells will receive a larger step size based on their size compared to the smallest cell and the value of the aspect ratio factor. The default aspect ratio factor value is `0` (no change to the step size) and it must be specified in the range [`0,1`]. The two grids shown below compare the effects during normal extrusion of the minimum aspect ratio factor (`0.0`, on left) and the maximum aspect ratio factor (`1.0`, on the right).



8.2.4 Step Size



Δs **Init** - This command lets you set the size of the first marching step. By default this value is obtained from the Defaults menu (Section 3.1.5). If the Defaults value is not set, the default first marching step size is `0.01`.

Δs **Min** - This command lets you specify a lower limit for the marching step size. By default this attribute is not enabled.

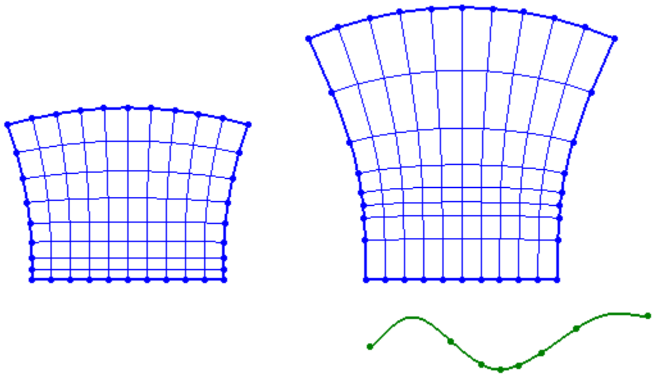
Δs **Max** - This command lets you specify an upper limit for the marching step size. By default this attribute is not enabled.

Growth Rate **Geometric** - Enabling this radio button means that the marching step size will grow from its initial value following a geometric progression law (Section 11.3.2.2). Geometric is the default setting for growth rate. Pressing the **Set** button lets you type in the value of the geometric growth rate.

Growth Rate **Subcon** - Enabling this radio button means the marching step size will grow based on a subconnector string that you select after pressing the **Set** button. When **Subcon** has been selected, the value for Δs **Init** is ignored.

Growth Rate **Set** - If geometric growth rate is enabled, pressing this button lets you type in the factor by which the marching step size will be multiplied to obtain the marching step size for the next step. The default value is `1.01` which means that the step size will grow by 1% each step. If subconnector growth rate is enabled, pressing this button lets you select the subconnector string from which the step sizes will be obtained.

The figure below demonstrates the difference between the **Geometric** and **SubCons** growth rates for a hyperbolic extrusion. The grid on the left used a geometric growth rate of 1.1. The grid on the right obtained its growth rate from the curved connector in the lower right.



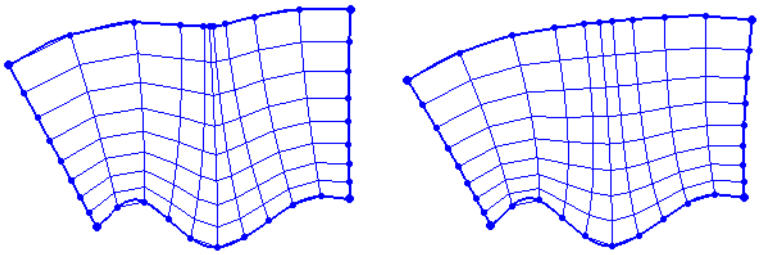
Translational and rotational extrusions also allow you to use equal grid point spacing by enabling the **Equal** radio button shown below.

Spacing ◆ Equal a

Normal extrusions also provide an additional control on the step size that reduces the local step size in convex regions and increases the local step size in concave regions.

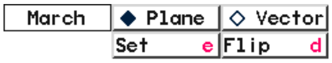
Stretching Factor Set 4

Stretching Factor - (normal extrusion only) Pressing the **Set** button allows you to enter the coefficient of a scale factor that reduces the local marching step size in convex regions and increases the local marching step size in concave regions. The default stretching factor value is `5` and it must be specified in the range `[1,10]`. The two grids shown below compare the effects during a normal extrusion of the minimum allowed stretching factor (`1.0`, left) and the optimal stretching factor (`5.0`, right).



8.2.5 Marching Direction

For hyperbolic and normal extrusion the following attributes control the marching direction. Note that many of these attributes are available for domain extrusion. However for block extrusion the marching direction is uniquely defined (except for reversal) by the normal of the initial surface grid.



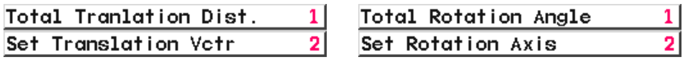
March **Plane** - When enabled this radio button indicates that the marching trajectory is to be parallel to the screen-xy plane (in other words, in the plane of the Display window). Available for domain extrusion only.

March **Vector** - When enabled this radio button indicates that the marching direction is parallel to the screen-z vector (in other words, perpendicular to the Display window). Available for domain extrusion only.

March **Set** - Pressing this button uses the current graphical orientation of the Display window image and the setting of the **Plane**/**Vector** radio button to set the marching direction. Available for domain extrusion only.

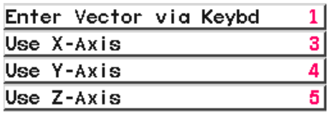
March **Flip** - Reverses the current marching direction by 180 degrees.

Translation and rotation extrusions use special attributes to control the direction of the extrusion as described below.



Total Translation Dist. - Pressing this button lets you type in the total distance to be extruded for a translational extrusion.

Set Translation Vctr - Pressing this button lets you choose the vector along which the translational extrusion will proceed. The translation vector menu is described below.



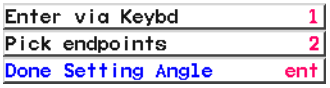
Enter Vector via Keybd - Pressing this button lets you type in the 3D components of the translation vector. It does not have to be a unit vector.

Use X-Axis - Pressing this button uses the body-x axis as the translation vector.

Use Y-Axis - Pressing this button uses the body-y axis as the translation vector.

Use Z-Axis - Pressing this button uses the body-z axis as the translation vector.

Total Rotation Angle - Pressing this button lets you type in the total rotation angle to be extruded for a rotational extrusion. There are two means of setting the rotation angle as described below.

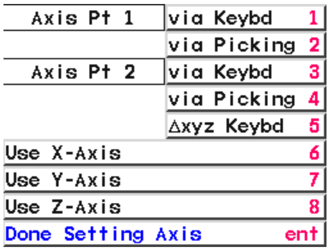


Enter via Keybd - Pressing this button lets you type in the total rotation angle in degrees in the range `[0,360]`.

Pick endpoints - Pressing this button lets you pick from the Display window two points. The angle of rotation from one point to the next around the current rotation axis determines the total angle of rotation.

Done Setting Angle - Pressing this button saves the current rotation angle.

Set Rotation Axis - Pressing this button lets you choose the axis around which the rotational extrusion will proceed. The rotation axis menu is described below.



Axis Pt 1 **via Keybrd** - Pressing this button lets you type in the 3D coordinates of the first axis point.

Axis Pt 1 **via Picking** - Pressing this button lets you pick the first axis point from the Display window.

Axis Pt 2 **via Keybd** - Pressing this button lets you type in the 3D coordinates of the second axis point.

Axis Pt 2 **via Picking** - Pressing this button lets you pick the second axis point from the Display window.

Axis Pt 2 **Δxyz Keybd** - Pressing this button lets you position the second axis point by typing in a 3D offset from the first axis point.

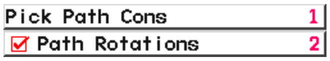
Use X-Axis - Pressing this button uses the body-`x` axis as the rotation axis. Note that this is the true body-`x` axis and doesn't necessarily correspond to the location of the body axes as drawn in the Display window.

Use Y-Axis - Pressing this button uses the body-`y` axis as the rotation axis. Note that this is the true body-`y` axis and doesn't necessarily correspond to the location of the body axes as drawn in the Display window.

Use Z-Axis - Pressing this button uses the body-`z` axis as the rotation axis. Note that this is the true body-`z` axis and doesn't necessarily correspond to the location of the body axes as drawn in the Display window.

Done Setting Axis - Pressing this button saves the current rotation axis.

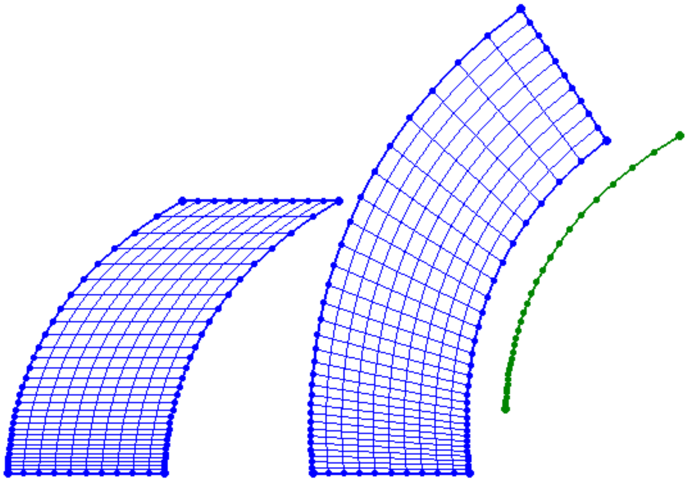
A path based extrusion uses the following attributes to establish the marching direction.



Pick Path Cons - Pressing this button lets you pick from the Display window the subconnector string that will define the path the extrusion is to follow. The marching step size is also obtained from the subconnector string.

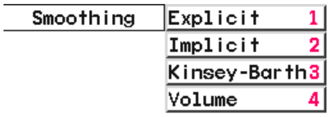
Path Rotations - If this toggle button is enabled, the extruded grid will rotate relative to the tangent vector of the subconnector string defining the path. Otherwise, the grid will remain parallel to the initial grid.

The example below shows the effect of enabling path rotations for a path-based extrusion. The initial grid is the horizontal connector at the bottom of each domain. The path is the curved connector on the right. Extrusion with (right) and without (left) rotation are shown.



8.2.6 Smoothing Parameters

Several parameters are available for adding numerical smoothing to the solution of the PDEs during hyperbolic extrusion. Solutions to systems of hyperbolic PDEs are notoriously unstable so smoothing will always be necessary. The hyperbolic smoothing attributes are shown below.



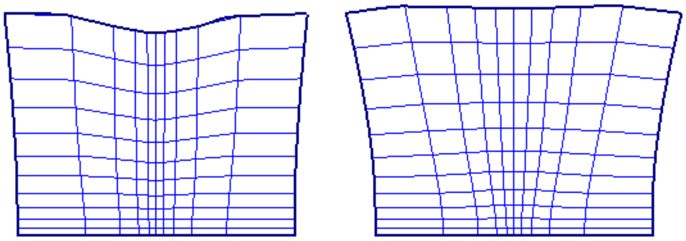
The hyperbolic PDE attributes you can set are divided into two categories: smoothing and spacing. The smoothing terms are used to help prevent grid line crossing and to prevent numerical instabilities. You can control both the explicit and implicit smoothing terms. They are both terms of order one, and the implicit smoothing value has to be at least twice the explicit smoothing value. If you are having problems with grid lines crossing, try increasing the smoothing.

Smoothing **Explicit** - is used to specify the explicit smoothing coefficient for the transverse direction. The default value is `0.1`.

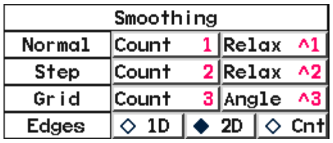
Smoothing **Implicit** - is used to specify the implicit smoothing coefficient for the transverse direction. The default value is `0.5`.

Smoothing **Kinsey-Barth** - provides a means to control the smoothing in the marching direction. Values greater than `1` help to prevent crossing of grid lines in the marching direction. The default value is `0` but you should set this smoothing factor to at least `3` if your initial grid contains any severe concavities.

Smoothing **Volume** - This coefficient in the range `[0,1]` determines how rapidly grid point clustering along the initial grid will be relaxed as the grid is extruded. A value of `0` will retain clustering on the initial grid all the way out to the outer boundary (see left figure below). The default value of `0.5` will rapidly relax the clustering (see right figure below).



Normal extrusions use their own set of smoothing attributes as described below.



Smoothing

Normal	Count 1	Relax ^1	
Step	Count 2	Relax ^2	
Grid	Count 3	Angle ^3	
Edges	◇ 1D	◆ 2D	◇ Cnt

Normal **Count** - The vectors in the marching direction are smoothed locally using a LaPlace filter. The value entered after pressing this button is the number of iterations for this filter. The default value is `5`.

Normal **Relax** - The vectors in the marching direction are smoothed locally using a LaPlace filter. The value entered after pressing this button is the coefficient in the range `[0,1]` for the smoothing. A value of `0` means smoothing is fully enabled while a value of `1` means that smoothing has no effect. The default value is `0.5`.

Step **Count** - The marching step size is smoothed locally using a LaPlace filter. The value entered after pressing this button is the number of iterations for this filter. The default value is `3`.

Step **Relax** - The marching step size is smoothed locally using a LaPlace filter. The value entered after pressing this button is the coefficient in the range `[0,1]` for the smoothing. A value of `0` means smoothing is fully enabled while a value of `1` means that smoothing has no effect. The default value is `0.7`.

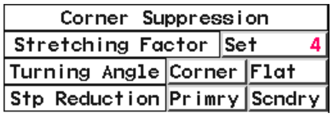
Grid **Count** - The marching direction is smoothed by adding a component to the local trajectory of each grid line. This additional component is obtained by smoothing the grid at the current step, computing the relative movement of each grid point, and adding that movement to the normal vector in the marching direction. The value entered after pressing this button is the number of iterations of the smoothing algorithm. The default value is `0`.

Grid **Angle** - The component added to the marching direction from the grid smoothing sweep will be limited to the angle in degrees entered after this button is pressed. This angle must be in the range `[0,90]`. The default value is `20`.

Edges **1D**, **2D**, and **Cnt** (prism extrusion only) - In **1D** mode, the smoothing of normals, step size, and grid points behaves as follows:

- points extruded from nodes are not smoothed
- points extruded from connectors are smoothed using only other points on the same connector
- points extruded from a domain’s interior are smoothed using all neighboring points

In **2D** mode, the smoothing is performed without regard for where the extrusion originated. The optional parameter **Cnt** is used to transition from **1D** to **2D** smoothing. For the first **Cnt** marching steps, **1D** smoothing is used and all subsequent marching steps use **2D** smoothing.



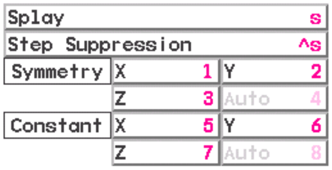
The Turning Angle and Stp Reduction parameters (prism extrusion only) allow you to identify regions along the extrusion front that should be considered to be corners and flats. Identification of these regions is necessary because special actions will be taken in these regions. Specifically, the Turning Angle parameters identify the regions and the Stp Reduction parameters specify the special actions to be taken.

Turning Angle **Corner**, **Flat** - The turning angles entered here are used to identify concave regions in which special actions are to be taken. The **Corner** angle is the angle between adjacent grid cells (the relative turning angle between the two cells' normal vectors) above which the edge is marked as a corner. For example, if the included angle in the corner is 30 degrees, the **Corner** angle would be entered as `150`. The **Flat** angle is the angle between adjacent grid cells below which the edge is marked as a flat region. For example, if the included angle between the two cells is 160 degrees, the **Flat** angle would be entered as `20`.

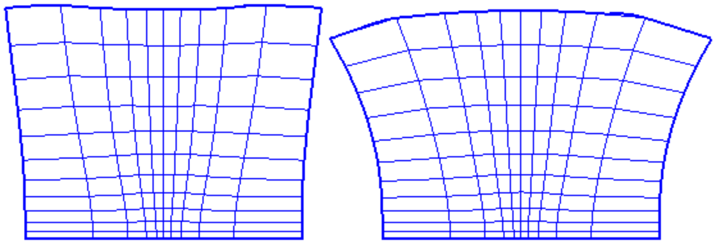
Stp Reduction **Primry**, **Scndry** - These two parameters control the reductions in marching step size to be applied at and near Flat points that are adjacent to Corner points. The **Primry** step reduction is a coefficient (in the range between `0` and `1`, typically `0.3`) to be applied to the step size at Flat points adjacent to Corner points. The **Scndry** step reduction is the coefficient to be applied to the step size at the grid point adjacent to a primary Flat point (typically `0.6`).

8.2.7 Boundary Conditions

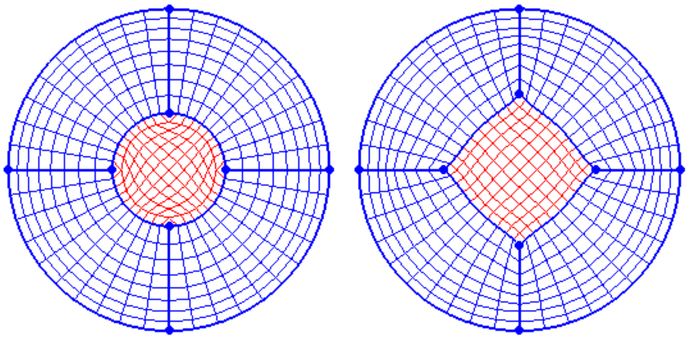
You can apply boundary conditions to hyperbolic and normal extrusions by pressing the **Boundary Conditions** button and then choosing from the menu shown below.



Splay - Pressing this button lets you type in the value of a coefficient in the range `[0,1]` that will cause the boundaries of the grid to move outward as the extrusion proceeds. A value of `0` means no splay while a value of `1` means full splay. The default value is `0.1`. The two grids shown below compare no splay (left) with splay of `0.5` (right).



Step Suppression - This highly specialized boundary condition for normal extrusion scales the marching step size on the boundaries relative to the step size on the grid's interior. The value you enter for the suppression coefficient will be in the range [-1,1]. Positive values will increase the boundary step sizes relative to the interior step sizes. Negative values will have the opposite effect. The default value is 0. The two grids below demonstrate an application of step suppression. The four outer grids in each case were extruded normally into the interior of the circle. The interior grid represents an H grid that was created by assembling edges. The grid on the left has step suppression turned off while the grid on the right has the step suppression coefficient set to -0.5 so that the boundaries (in this case the connectors heading from the circles to the interior) would grow slower than the interior points. The result is that the middle region has more of a square shape resulting in a better quality H grid in the middle.



Symmetry **X** - The grid will be kept symmetric in the x direction.

Symmetry **Y** - The grid will be kept symmetric in the y direction.

Symmetry **Z** - The grid will be kept symmetric in the z direction.

Symmetry **Auto** - Gridgen will examine the grid and automatically determine the direction in which the grid will be symmetric.

Constant **X** - The x coordinate of grid points on the boundary will be held constant during the extrusion.

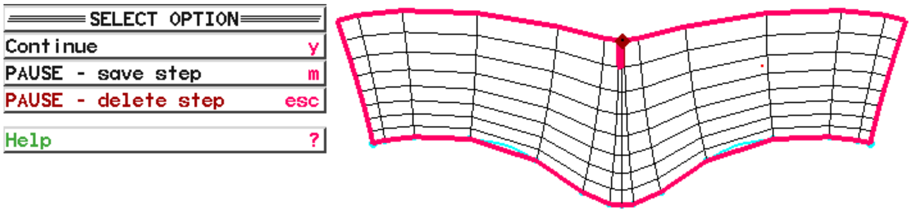
Constant **Y** - The y coordinate of grid points on the boundary will be held constant during the extrusion.

Constant **Z** - The z coordinate of grid points on the boundary will be held constant during the extrusion.

Constant **Auto** - Gridgen will examine the grid and automatically determine which of the three coordinate components x, y, or z should be held constant.

8.2.8 Stop Criteria

The category of attributes accessed by pressing the **Set Stop Criteria** button allow you to specify grid quality constraints, if exceeded, will cause Gridgen to pause the extrusion process so that you can take corrective action. For example, you may choose to limit the total height of the extrusion. Or you may have Gridgen calculate the Jacobian of hexahedra after each marching step and pause if any skewed cells are found. In this later case, pausing gives you the opportunity to back-up the extrusion, change some attributes, and continue extruding. In the figure below, the extrusion has been stopped because the aspect ratio of the highlighted cell exceeds the specified upper limit of 7. You have three choices on how to proceed.

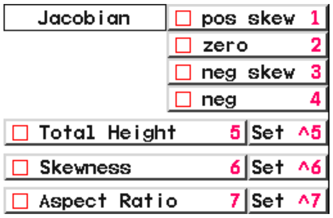


Continue - Pressing this button continues the extrusion process without regard to the highlighted cells.

Pause - Save Step - Pressing this button pauses the extrusion as it is and invokes the main extrusion menu so that you may change extrusion attributes to obtain a grid that meets your criteria or to relax your stop criteria.

Pause - Delete Step - Pressing this button deletes the last marching step, pauses the extrusion, and invokes the main extrusion menu so that you may change extrusion attributes to obtain a grid that meets your criteria or to relax your stop criteria.

The various stop criteria are described below.



Jacobian **pos skew** - Enabling this toggle button will cause Gridgen to pause the extrusion process when a grid cell with a positive skewed Jacobian is computed.

Jacobian **zero** - Enabling this toggle button will cause Gridgen to pause the extrusion process when a grid cell with a zero Jacobian is computed.

Jacobian **neg skew** - Enabling this toggle button will cause Gridgen to pause the extrusion process when a grid cell with a negative skewed Jacobian is computed.

Jacobian **neg** - Enabling this toggle button will cause Gridgen to pause the extrusion process when a grid cell with a negative Jacobian is computed.

Total Height - Enabling this toggle will cause Gridgen to pause the extrusion process when the total distance marched by any grid line exceeds the value you type in by pressing the **Set** button in the same row.

Set - Pressing this button lets you type in the limit on extrusion height.

Aspect Ratio - Enabling this toggle will cause Gridgen to pause the extrusion process when a grid cell exceeds the aspect ratio value you type in by pressing the **Set** button in the same row.

Set - Pressing this button lets you type in the limit on cell aspect ratio.

8.2.9 Copy Attributes

For hyperbolic and normal extrusions, users can copy the extrusion attributes of an already extruded domain or block to use for the extrusion of a new domain or block. Using **Copy Attributes** sets the initial spacing and other extrusion attributes for a new extruded domain or block so that the grid characteristics of the previously extruded domain or block are maintained across their interface.

Copy Attributes is only available for certain criteria:

- Extrusion must be of type **Nrml** or **Hyp**
- Extrusion must be **From Cons** or **From Doms**
- Selected extrusion type must match type of previously extruded domain or block.
- Selected connector or domain must be the last entity created from a previous extrusion.
- Selected domains are the top faces on the same block (prism blocks only).

Please note that Copy Attributes is currently not allowed for creating extruded domains or blocks **From An Edge** or **From A Face**.

8.2.10 Reset Attributes

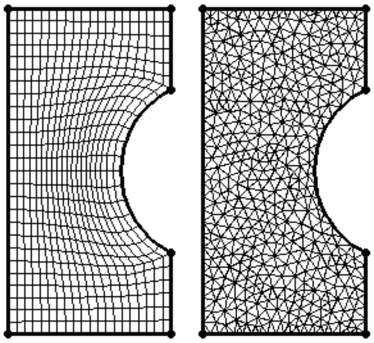
For all types of extrusion, users may reset the extrusion attributes to the default values before they extrude a domain or block.

In cases where an already extruded domain or block exists, users can remove the extruded grid (see Section 14.6), select **Reset Attributes** and restart their extrusion.

9. Creating Domains

9.1 Overview

Domains are Gridgen's mid-level grid element. They are surface grids. Two types of domains are supported in Gridgen: structured and unstructured. A *structured* domain consists of an `IxJ` array of quadrilateral cells and is defined by the four edges on its perimeter. An *unstructured* domain consists of triangular cells and is defined by one perimeter edge and possibly several interior edges enclosing holes in the domain. Examples of structured and unstructured domains are shown below on the left and right, respectively.



This chapter describes three methods for creating domains:

Assemble Edges - The process of assembling edges involves selecting the connectors on a domain's perimeter.

Triangulate Struct Doms - This command creates an unstructured domain by converting an existing structured domain.

On DB Entities - This highly automated command creates a domain on every database surface that you pick and then trims the connectors between adjacent domains so they fit together topologically.

Domain creation by extruding connectors is covered in Section 8.

The chapter concludes with a discussion of special domain types.

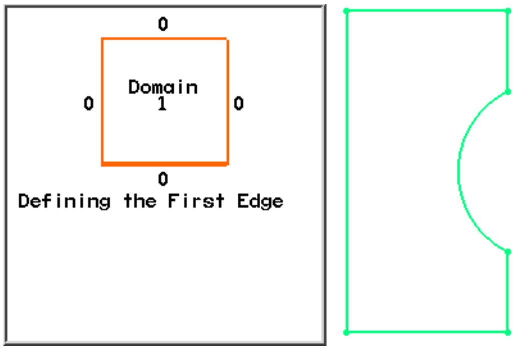
9.2 Assembling Structured Domains

The four edges of a structured domain represent the curves on which the domain's `i` and `j` indices are at their minimum and maximum values. Each edge consists of one or more connectors. Domain creation cannot begin until all connectors that will lie on that domain's edges have been created. Keep in mind each structured domain requires four edges even if an edge or a portion of that edge is collapsed to a point (a point singularity or pole, Section 7.2.5.1). If a domain connects to itself, one connector can be used twice in the same domain. Connectors can be added to the edges in any order during domain assembly without regard for the eventual computational (ξ, η, ζ) axis system (Section 15.7) of any blocks which will use this domain. Gridgen only refers to the computational axes at the block level so you do not have to worry about that while creating connectors or domains.

Domain assembly can be thought of as the selection of the closed loop of connectors that form the domain's perimeter. In addition to defining the perimeter, you also designate where on the perimeter the four corners of the structured domain are located. This permits the connector's grid points to be mapped into a four sided region of `I x J` grid points. There are eight different ways to select the connectors on a structured domain's perimeter: you can start in any of the four corners and proceed either clockwise or counterclockwise. Each of these eight permutations is the same to Gridgen.

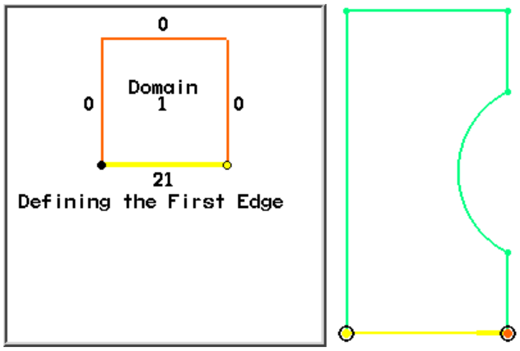
Before beginning to create a domain you must choose:

1. the corner (node) from which the domain's perimeter will begin and end
2. the direction in which the connector loop will be selected



Defining the First Edge

As shown above, after pressing the **Assemble Edges** button, the Blackboard displays a computational schematic (a square) of an empty domain of dimensions 0 x 0. The Blackboard window also displays the message "`Defining the First Edge`". The "first" edge is simply that; the first of the four edges being added to the domain. The edge number has nothing to do with the eventual computational axes.



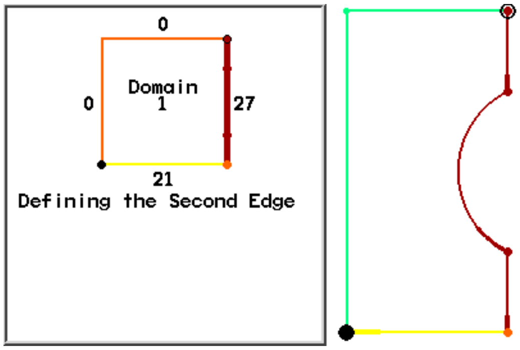
Defining the First Edge

You define the first edge by selecting the connector or connectors on edge 1 starting from the chosen corner node and proceeding in the desired direction. The example above starts the domain from the lower left node and proceeds in a counter-clockwise direction.

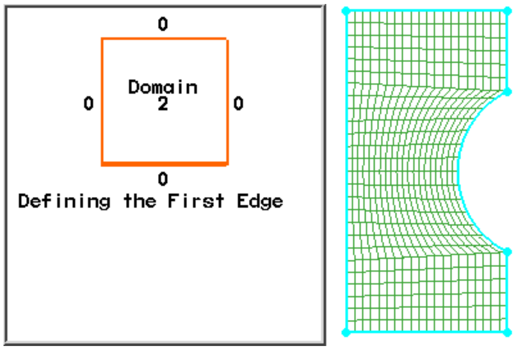
The node to which connectors may be added to the domain's perimeter has an open circle drawn around it. After at least two connectors have been selected or after **Next Edge** has been pressed, the node at the beginning of the domain's perimeter has a filled circle drawn over it. When a connector is added to the edge, the screen is updated. If you select an un-dimensioned connector while creating a domain you will be prompted to enter a dimension via the keyboard. The Display window shows that connector displayed in either yellow (edges 1 and 3) or red (edges 2 and 4). The pairing of edges is meant to reinforce the idea that opposite edges of a structured domain must have the same number of grid points. The schematic of the domain in the

Blackboard follows the same color scheme and also prints the number of grid points on the edge.

After pressing the **Next Edge** button, the three connectors that comprise the right hand side (edge 2) of the domain are picked, proceeding from bottom to top. As shown below, edge 2's connectors are highlighted in red. The Blackboard changes to indicate you are now `Defining the Second Edge`. Note that the node at the end of the first edge will be a corner of the `I x J` surface grid on the domain.



Pressing the **Next Edge** button completes the assembly of the domain. Since **Auto Complete** is on and there is only one connector adjoining edge 2 at its end, Gridgen automatically starts edge 3 and adds the connector. Gridgen knows how many grid points are required on edge 3 (same as edge 1). It automatically applies the **Next Edge** command when enough connectors are added to edge 3 so its dimension matches that of edge 1. Following the same logic, Gridgen completes edge 4 and hence the domain. Gridgen saves the domain, automatically generates its interior grid points, and prepares for assembly of the next domain, as shown below.



The domain's surface grid is automatically initialized using an algebraic method called transfinite interpolation (TFI, Section 12.1). Gridgen chooses the specific TFI method to apply from the following list.

1. Parametric TFI - Gridgen will apply Parametric TFI if the domain's connectors have been defined in terms of one and only one database entity. The connector's may have been created directly on the entity (for example, DB cnstr **Line** segment type) or may have been projected onto the database. Parametric TFI ensures the grid will conform to the database shape.

2. Parametric Fit TFI - If Gridgen cannot apply Parametric TFI, it tries to apply Parametric Fit TFI. In order to apply Parametric Fit TFI, a certain percentage of the domain's connectors must have been defined in terms of any database entity, the connector's grid points must lie within the bounding box of a single database surface entity, and the distance from the connectors to the database must be less than a tolerance. If these conditions are met, the domain's connectors grid points are projected onto the database, and the parametric coordinates obtained from the projection are used to apply Parametric TFI.

This will ensure that the interior grid points will conform to the database shape.

3. Standard TFI - If neither of the two methods above can be applied Gridgen applies Standard TFI. A Standard TFI grid will not adhere to the database. You must modify the domain via projection (Section 14.4) if you want the grid to lie on the database.

Gridgen color codes newly created domains to show you which TFI method was used for the initial domain creation. If Parametric TFI or Parametric Fit TFI were used, the new domain will be purple. If Standard TFI was used, the new domain will be green.

This color coding stays with the domain even if the domain's database relationship changes. For instance, if you project onto the database a domain that was created using Standard TFI it will still be green even though it is now on the database. Similarly, if you translate a domain that was created on the database using Parametric TFI, it will still be displayed in purple even though it is no longer on the database. The color coding only tells you how the domain was originally created.

You can determine the algebraic method last used on a domain using the **Run Structured Solver** command (Section 12.1); the TFI method is listed in the Browser for each domain.

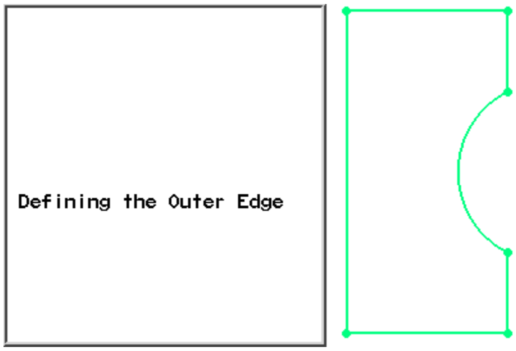
9.3 Assembling Unstructured Domains

An unstructured domain is created by selecting the connector(s) that make up its outer boundary and any inner boundaries. Each boundary is called an edge in Gridgen's terminology. Domain creation cannot begin until all connectors which lie on that domain's edges have been created. Keep in mind each unstructured domain consists of at least one edge for the outer boundary and optional additional edges that define the boundaries of holes inside the outer boundary. You can add connectors to an unstructured domain's outer boundary without regard for the orientation of the loop or its starting point as long as the connectors are selected end-to-end. However, all inner boundary loops must proceed in the direction opposite to that of the outer boundary.

Before beginning to create a domain you must choose:

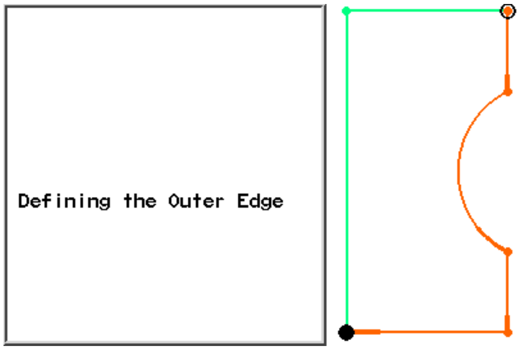
1. the node from which the domain's perimeter will begin and end

2. the direction in which the connector loop will be selected



Defining the Outer Edge

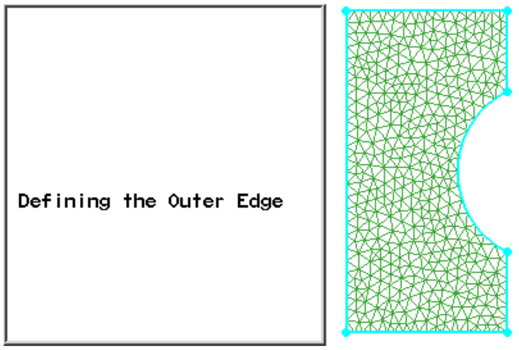
After selecting Cell Type **unstructured** and pressing the **Assemble Edges** button, the Blackboard prints the message "`Defining the Outer Edge`". The outer boundary is always the first edge in an unstructured domain.



Defining the Outer Edge

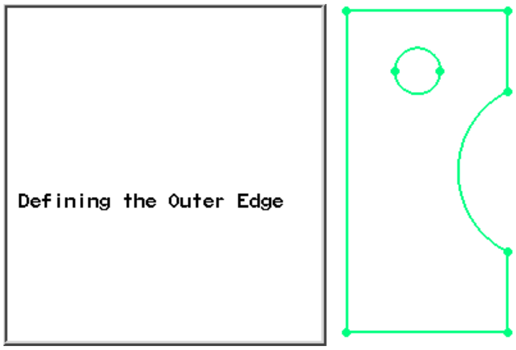
Define the edge by selecting the connector or connectors starting from the chosen corner node and proceeding in the desired direction. The example above starts the domain from the lower left node and proceeds in a counter-clockwise direction. The node to which connectors can be added to the domain's perimeter has an open circle drawn around it. After at least two connectors have been selected, the node at the beginning of the domain's perimeter has a filled circle drawn over it. When a connector is added to the edge, the screen is updated.

Continue the connector selection until all connectors on the perimeter of the domain have been added. If you select an un-dimensioned connector while creating a domain, you will be prompted to enter a dimension via the keyboard. When the domain's perimeter is closed, press the **Save Domain** button. Gridgen saves the domain and automatically generates its interior grid points, unless explicitly disabled as described in Defaults (Section 3.3.1).



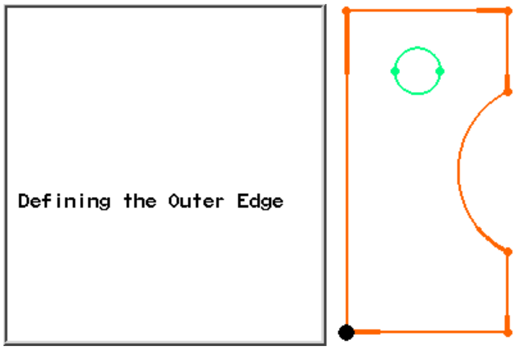
Defining the Outer Edge

Assembly of a domain with multiple edges (internal holes) proceeds in much the same manner. The first edge must always be the outer boundary of the domain. Make sure that **Auto Save** is toggled off. Otherwise, Gridgen will save the domain and put triangles on the interior as soon as the outer edge is complete and before you have had a chance to define the interior edges. As shown below, the process begins with the selection of the initial edge of the outer boundary.



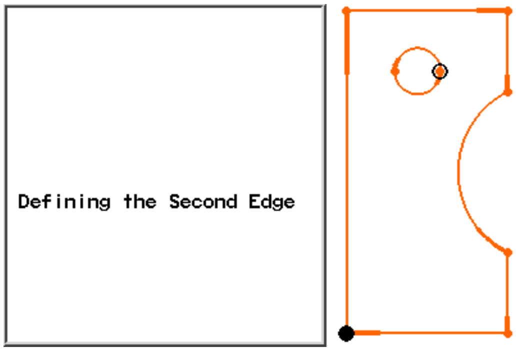
Defining the Outer Edge

Connector selection continues around the entire perimeter of the outer boundary.



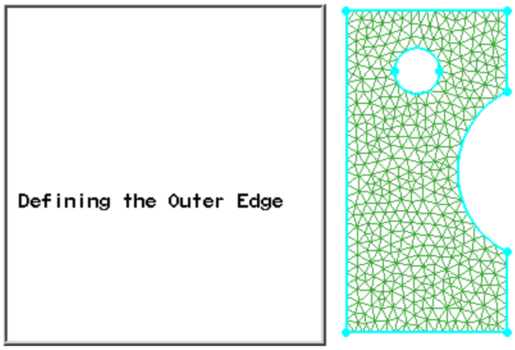
Defining the Outer Edge

When the outer boundary closes on itself, press the **Next Edge** button to begin defining the inner boundary. Be sure to select the inner loop's connectors in an order such that the inner loop direction is opposite of the outer loop. You may have to use the **Reorient Edge** command to do this.



Defining the Second Edge

Finally, press the **Save Domain** button and the domain grid will be triangulated automatically in the region between the outer and inner boundaries.



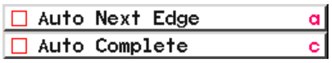
The domain's interior grid is automatically initialized using a triangulation method when the domain is created . Gridgen chooses how to apply the triangulation method from the following list.

1. **Parametric Triangulation** - Gridgen will apply the triangulation method in the parameter space of a database surface if the domain's connectors have been defined in terms of one and only one database entity. The connector's may have been created directly on the entity (for example, DB cnstr **Line** segment type) or may have been projected onto the database. Parametric Triangulation ensures the grid will conform to the database shape.
2. **Parametric Fit Triangulation** - If Gridgen cannot apply Parametric Triangulation, it tries to apply Parametric Fit Triangulation. In order to apply Parametric Fit Triangulation, a certain percentage of the domain's connectors must have been defined in terms of any database entity, the connector's grid points must lie within the bounding box of a single database surface entity, and the distance from the connectors to the database must be less than a tolerance. If these conditions are met, the domain's connector's grid points are projected onto the database entity, and the parametric coordinates obtained from the projection are used to apply Parametric Triangulation. This will ensure that the grid will conform to the database shape.
3. **Standard Triangulation** - If neither of the two methods above can be applied, Gridgen applies Standard Triangulation. A Standard Triangulation grid will not adhere to the database. You must modify the domain via projection (Section 14.4) if you want the grid to lie on the database.

As with structured domains, Gridgen color codes the newly created domains to show how they were created: purple for database constrained and green for free. You can determine the method last used on a domain using the **Run Unstructured Solver** command (Section 13.1); the method is listed in the Browser for each domain.

9.4 Special Domain Assembly Commands

The two commands shown below help automate the assembly of structured domains.



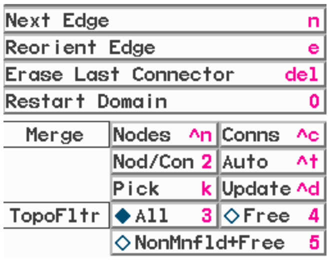
Auto Next Edge - Enabling this toggle button automatically executes the **Next Edge** command after each connector is picked. The implication is that at least two adjacent edges of the domain consist of a single connector. If you pick those two first, Gridgen can determine when the third and fourth edges are complete by comparing the number of grid points on edges one and two.

Auto Complete - When this toggle is enabled (default), Gridgen will attempt to automatically complete the assembly of structured domains after you define the first two edges. In instances where two distinct domains can be created by the automatic completion process, Gridgen does nothing. Rather you proceed with the connector selection process for the third and fourth edges. After each connector is picked, Gridgen again attempts to automatically complete the domain.

Another command is available for automating the assembly of unstructured domains.

Auto Save - This command causes the **Save Domain** command to be executed automatically when the domain's first edge (outer boundary) closes on itself. The implication is that the domain has no inner boundaries.

Several other commands are available during domain assembly.



Next Edge - Saves the picked connectors as the current edge and begins creation of the next edge. Since an edge in an unstructured domain has to form a closed loop, Gridgen will not let you save the edge in an unstructured domain until the beginning and end are at the same node

Previous Edge - Returns control to the previous edge (structured domains only).

Reorient Edge - Changes the direction of an edge (unstructured domains only). This is used to ensure the inner loops of an unstructured domain are oriented in the opposite direction from the outer loop.

Erase Last Connector - Removes the last connector you added.

Restart Domain - Remove all connectors from all edges of the domain and start over.

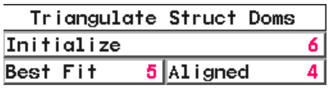
Save Domain - Save the domain as defined by the selected edges. You must use this command for unstructured domains if you have not enabled **Auto Save**.

Abort - Same as **Restart Domain** except the domain **Create** command is exited.

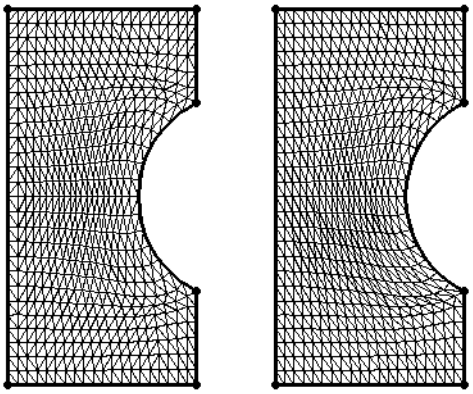
The Merge and TopoFltr commands are described in Section 15.10.

9.5 Convert Structured to Unstructured Domains

To convert structured domains to unstructured domains, use one of the buttons below the **Triangulate Struct Doms** label after pressing **Create**.



The **Initialize** command will create new unstructured domains by taking each selected structured domain's connectors, forming a new domain, and applying the unstructured solver in the same manner as if the domain had been created as unstructured originally. The original structured domain is not deleted. An **Aligned** triangulation (below, right) triangulates each quadrilateral cell along the diagonal from its `(i,j)`$_{min}$ corner to the `(i,j)`$_{max}$ corner. A **Best Fit** triangulation (below, left) triangulates each quadrilateral cell along the diagonal maximizing the minimum included angle of the resulting triangles.



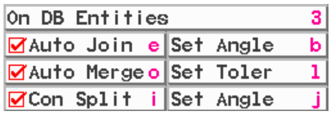
9.6 On Database Entities

The **On DB Ents** command for creating domains is the surface grid analogy of the same named command for creating connectors (Section 7.4). It allows you to very rapidly create surface grids for each database surface entity you select. This command is not available unless you first set a default connector dimension in the Defaults Menu.

Gridgen performs the following steps.

1. Executes the **On DB Ents** command creating connectors, resulting in dimensioned and distributed connectors joined in a domain around the perimeter of each trimmed and untrimmed database surface you select.
2. The **Merge** command is applied with the default tolerances to remove coincident and overlapping connectors between adjacent domains.
3. The default Algebraic or Unstructured initialization methods are applied to the connectors around each database surface to create the initial domain grid.

Structured domains will only be created by this command for instances where four connectors can be used around the database surface's perimeter and the default connector dimensioning has resulted in equal numbers of grid points on opposite edges. The connectors are still created even in cases where structured domains cannot be created.



Options you can set before On DB Entity domain creation are:

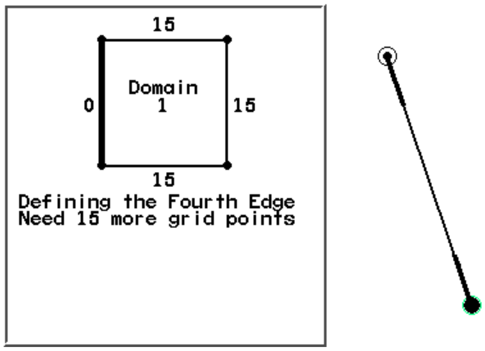
Auto Join Set Angle is used to merge multiple domains into a single domain. This is useful if you have many small database surfaces and do not need their edge topology maintained in the final grid. If **Auto Join** is checked on, you can enter a join angle by pressing the **Set Angle** button. Two domains will be joined if the bending angle between every pair of adjacent triangles along their common edge is less than the specified angle.

Auto Merge Set Toler is used to automatically merge and split connectors to try to form a water tight geometry as domains are created. If Auto Merge is checked on, you can enter a merge tolerance by pressing the **Set Toler** button. Gridgen applies the **Merge Nodes**, **Merge Conns**, and **Merge Nod/Con** commands (Section 15.10) based on this tolerance.

Con Split Set Angle is used to split connectors at break lines in the geometry. If **Con Split** is checked on, you can enter a break angle by pressing the **Set Angle** button. Connectors created after this will be split at locations where the angle between any three adjacent points along the connector is greater than the specified angle.

9.7 Pole Domains

A pole domain is a structured surface grid collapsed to a line. Pole domains are necessary for certain structured grid topologies requiring singularities. For example, if a block was to be constructed for a prism, one of the block's faces would correspond to the peak vertex of the prism and the entire surface grid is collapsed to a line.



Defining the Fourth Edge
Need 15 more grid points

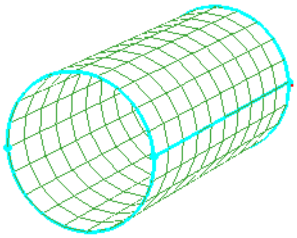
A pole domain is created in Gridgen from three connectors: two pole connectors (Section 7.2.5.1) and another curve. The two poles will be situated at either end of the third curve. Domain creation for the pole proceeds in exactly the same manner as any other domain except, the curve is picked twice; in other words, the domain's edges will be pole 1, curve, pole 2, curve.

Gridgen automatically detects that a domain is a pole and applies a grid method other than TFI to ensure a mathematical singularity.

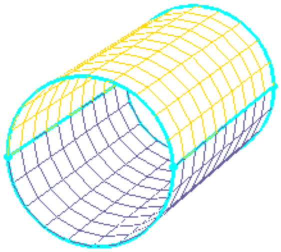
9.8 Bad Domain Types

9.8.1 Allowable, But Should Be Avoided

In general, domains can be created to represent any surface. For structured domains, that requirement is extended further to require that domains must be mapped into a computational rectangle. This includes surfaces that connect to themselves and surfaces with poles. The straight line connector in the cylindrical domain shown below represents two edges of the domain. This is called "self connecting" and should be avoided because it introduces ambiguities when the domain is used in a block. Splitting this domain into two 180 degree half cylinder domains by adding another straight line connector to the front would be a better approach.

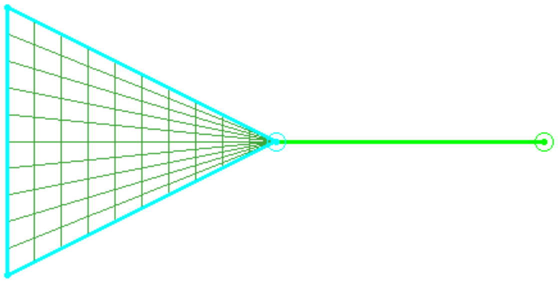


Bad: A single self-connecting domain.



Better: Two domains.

It is best if domains are not partially singular. For example, the domain shown below should be split into two domains: a pole and a triangle.



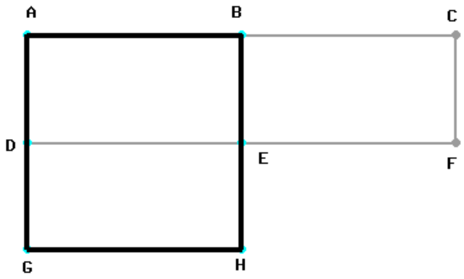
9.8.2 Duplicates

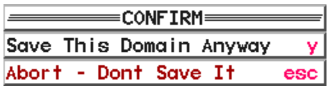
Sometimes you inadvertently will create the same domain twice. Gridgen automatically compares each new domain with all other domains in the system. If the new domain already exists in the grid, you are notified and the new domain is not saved. This comparison takes into account the different ways in which a domain can be constructed.

9.8.3 Overlapping and Adjacent

When a domain is created, Gridgen checks whether it overlaps and is adjacent to any other domain. Overlap means that the new domain is not identical to any other domain, but is adjacent to another domain on at least part of two edges and one corner (node). Overlapping domains usually signal an inconsistency in the grid system, but may be part of a perfectly legal grid. Therefore, Gridgen does not automatically save overlapping and adjacent domains.

In the example below, the square domain ABHG already exists and the rectangular domain ACFD has just been created. Gridgen has identified the two as overlapping and adjacent since they share connectors AB and AD and node A. You must then choose one of the two menu buttons shown below.





Save This Domain Anyway - The new domain (ACFD) will be saved anyway despite the warning. There are rarely instances when you will choose this option.

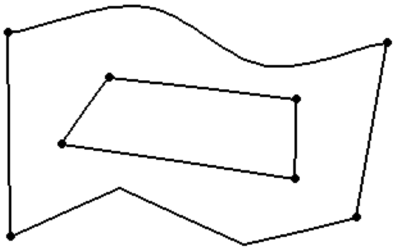
Abort - Don't Save It - Do not save domain ACFD. You will be returned to the beginning of domain creation.

There are three ways you could create domains for the above topology without getting overlapping and adjacent warnings.

1. Create two domains: ABHG and BCFE.
2. Create two domains: ACFD and DEHG
3. Create three domains: ABED, BCFE, and DEHG.

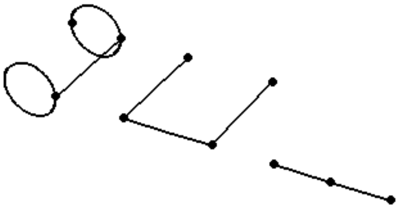
9.8.4 Overlapping

Sometimes Gridgen cannot detect domain overlap as illustrated below. Here the smaller domain lies entirely within the boundaries of the larger domain, but since there are no shared connectors in the two domains, no overlap is detected. In general, it is impossible for Gridgen to detect this type of overlap since the domain is defined solely by its connectors, and not by the surface on which the domain is intended to lie. Therefore, you should take great care in constructing domains to insure new domains do not overlap.



9.8.5 Prohibited

Despite the general nature of domains that can be created in Gridgen, there are a few types that still should not be constructed. These domain types are illustrated below.



The domain in the left of the figure contains a circular arc connector that begins and ends at the same node - a self-connecting connector. Domain's containing self-connecting connectors should be avoided. This topology can be corrected by dividing the self-connecting connector into two or more connectors.

The domain in the center of the figure illustrates a domain that does not trace a loop in physical space but represents a surface of finite area. The topological problem here is that a rectangular domain is "folded" in half. The correction would simply require splitting the domain at the fold.

The domain in the right of the figure illustrates a zero-area domain that doesn't contain any poles. This topological problem involves a domain that is sheared until it collapses onto itself leaving only two unique edges.

It is possible to arrive at other domain topologies (possibly combinations of the three examples above) that are not possible to generate, or more likely, ones that are of no use in block construction. In most of these cases, the solution will be to further divide the surface into smaller domains, so the topology conforms to Gridgen's logic.

10. Creating Blocks

10.1 Overview

Blocks are Gridgen's highest level grid element. They are volume grids in 3D and surface grids in 2D. A structured block consists entirely of hexahedral cells arranged in an `IxJxK` array. An unstructured block consists of either all tetrahedral cells or a combination of tetrahedra and pyramids (the latter is often referred to as a hybrid unstructured block). A prism block consists entirely of prismatic cells that are generated by extruding an unstructured (triangular cell) domain.

This chapter describes the procedure for assembling structured and unstructured blocks by selecting the domains on each of the block's faces. Creating structured blocks and prism blocks by extrusion is covered in Section 8.

10.2 Assembling a Structured Block

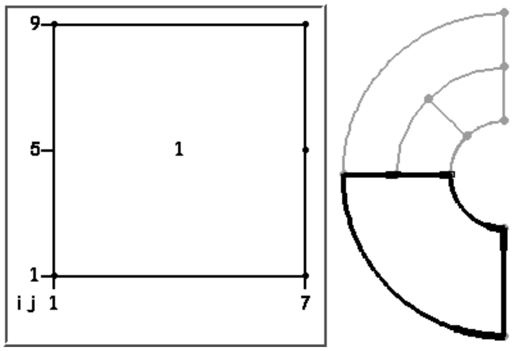
The faces of a structured block are the surfaces on which the block's ξ, η, and ζ computational coordinates are at their minimum and maximum values. It is assembled from structured domains. A block cannot be assembled until all domains on that block's faces have been created (Section 9). Each structured 3D block requires six faces even if a face or a portion of that face is collapsed to a line (a line singularity or pole, Section 9.7). Therefore, you must have created at least six domains before a 3D block can be assembled. Because a 2D block has only one face, at least one domain must have been created in order to assemble a 2D block. On the other hand, if a block connects to itself, then one domain can be used twice in the same block (3D case only). Domains can be added to a face and faces can be added to a block without regard for the structured block's eventual computational coordinate system (Section 15.7).

Begin block creation by going to the Block Commands menu and pressing the **Create** button. Then begin assembling the block face by face by first pressing the **Assemble Faces** button and then the **Add 1st Face** button. Repeat the face assembly process described in the following section for each of the block's faces.

10.2.1 Adding Faces to a Structured Block

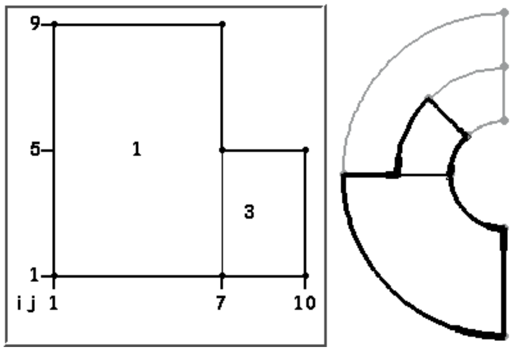
A block face consists of one or more domains. After the **Add 1st Face** button is pressed you select the first domain to be added to the face by positioning the cursor near one of the edges of the domain until it highlights and pressing the ***right mouse*** button. The pickable part of a domain during block face assembly is the first row of cells along its boundary. If the wrong domain highlights, try moving the cursor to lie only over the desired domain, or use the ***x*** hot key to cycle through the available choices until the one you want is highlighted.

The edges of the developing face are highlighted in the Display window. A schematic representation of the face in computational space is drawn in the Blackboard with the composing domain numbers drawn in the center of each domain. The Display window and Blackboard images are color coded: connectors are colored red if they are `i` directed, and orange if they are `j` directed. The indices `i` and `j` in this discussion correspond only to a local set of coordinates used to set up the face; these local coordinates have no relationship to the final ξ, η, and ζ computational coordinates of the block. The Blackboard also shows the dimensions of the face and each node (shown as a filled circle).

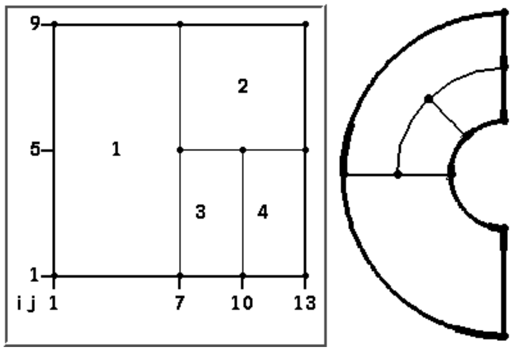


If the entire face consists of a single domain, then nothing more is required, and the face can be loaded into the block by pressing the **Save the Face** button.

In the case of a multi-domain face, such as the example above, other domains need to be added to the face. Select each additional domain to be added to the face using the procedure described above with one exception: each domain added to the face must be adjacent to domains already in the face. The Blackboard and Display windows are updated to reflect the new face structure. Notice that only the connectors on the perimeter are rendered in red or orange; these are the only possible regions to which another domain may be adjacent.



If the selected domain does not connect to the existing face structure, you are warned and can select another domain. It is not possible to save the face until it is a rectangular shape in computational space (as displayed in the Blackboard window below).



10.3 Assembling an Unstructured Block

Each face in a 3D unstructured block represents a closed surface on the boundary of the volume grid. The boundaries of an unstructured block can be both structured and unstructured domains. Wherever a structured (quadrilateral) domain is included in an unstructured face, pyramid cells are used to transition from the structured portion of the face to the unstructured (tetrahedral) volume grid. Only triangular domains can be added to a 2D unstructured block.

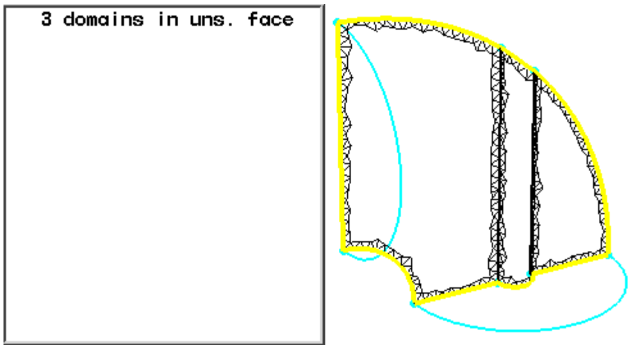
The first face of a 3D unstructured block is the closed, outer boundary of the volume grid. Additional faces, if any, are the closed boundaries of voids inside the volume. Block assembly cannot be started until all domains on that block's faces have been created (Section 9.3). Each 3D unstructured block requires at least one face. Since a single domain cannot be created so that it closes on itself, at least two domains must be added to each face. Therefore, you must have created at least two domains before a 3D block can be assembled. Since a 2D block only consists of one face, at least one domain must be created prior to assembling the block.

After pressing the **Create** button, begin assembling a face by pressing the **Assemble Faces** button and **Add 1st Face** button, as described in the following section. Repeat the process until all of the block's faces have been added.

10.3.1 Adding Faces to an Unstructured Block

An unstructured block face consists of two or more domains. After the **Add 1st Face** button is pressed, select the first domain to be added to the face by positioning the cursor over one of the domain's edges until it highlights and pressing the ***right mouse*** button. The pickable part of the domain is the row of cells adjacent to its boundary. If the wrong domain highlights, try moving the cursor to lie only over the desired domain, or use the ***x*** hot key to cycle through the available choices until the one you want is highlighted.

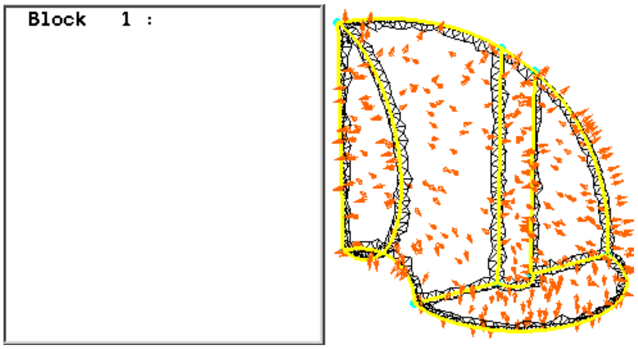
The picked domain is highlighted in the Display window, and the cumulative count of domains in the face is listed in the Blackboard. "Free" edges, those on the perimeter of the face, are drawn in yellow. Since each face of a 3D unstructured block must be a closed surface, the appearance of yellow connectors indicates that the face is not yet complete. In the illustration below, only 3 (two on the symmetry plane and one on the valve stem) of the 6 domains in the block's face have been added: note the yellow connectors around the perimeter.



3 domains in uns. face

Each additional domain to be added to the face is selected using the procedure described above. Each domain added to the face must be adjacent to the domains already in the face. The Blackboard and Display windows are updated to reflect the new face structure.

If the selected domain does not connect to the existing face structure, you are warned and can select another domain. For 3D blocks, it is not possible to save the face until it fully encloses a volumetric region. However, once the face is fully closed (and no yellow "free" edges appear in the Display window) the face may be saved and the Display window shows in orange the face's normal vector as illustrated below.



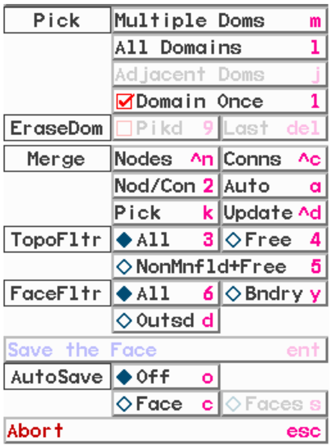
Note that this normal vector should point into the volume to be gridded. If Gridgen does not choose the correct normal vector, you may flip the normal by using the **Reorient Face** command (Section 10.5).

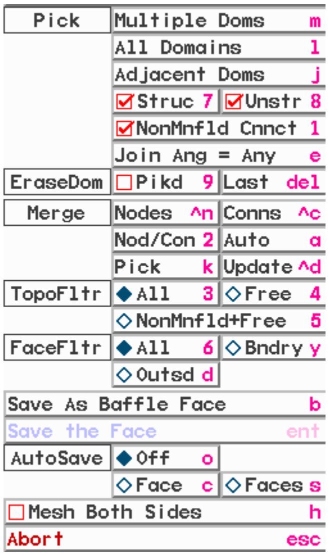
10.4 Assembling a Hybrid Unstructured Block

A hybrid unstructured block, consisting of pyramid cells near the quadrilateral boundaries and tetrahedral cells elsewhere, is assembled in the same manner as an unstructured block (Section 10.3) with one exception: both structured (quadrilateral) and unstructured (triangular) domains may be added to the faces. Pyramid cells are created in the volume grid adjacent to every structured domain on the block's faces. Tetrahedral cells are created elsewhere in the volume grid. Control of the pyramid cells is covered in Section 13.

10.5 Face Creation Commands

Commands used during face creation include the following:





Pick **Multiple Doms** - When this button is pressed, you are able to select more than one domain to be added to the face using the Browser and Display window. For unstructured blocks, this makes it very easy to select all of the domains in a face. Each domain selected in this mode will be successively added to the developing face using the first linkage found. For structured blocks, this method should only be used on faces that are not self-connecting (i.e., without multiple linkages).

Pick **All Domains** - This command picks all domains and adds them to the face.

Pick **Adjacent Doms** - This command adds to the current face all domains adjacent to the “free” (yellow) edges. Using this command allows you to assemble the face in discrete “layers”.

Pick **Struc** - Enabling this toggle button allows structured (quadrilateral) domains to be picked during face assembly. By default this toggle is enabled. This button is available when assembling unstructured blocks only.

Pick **Unstr** - Enabling this toggle button allows unstructured (triangular) domains to be picked during face assembly. By default this toggle is enabled. This button is available when assembling unstructured blocks only.

Pick **NonMnfld Cnnct** - Enables non-manifold connections to be picked during face asembly. Non-manifold domain connections are defined as three or more domains sharing the same edge. By default this toggle is enabled. This button is available only for unstructured grids.

Pick **Join Angle = Any -** This command lets you specify a maximum join angle between adjacent domains. After you enter an angle, Gridgen checks any new domains you try to add to the developing face to make sure the dihedral angle between each cell on their boundaries and the adjoining cells already in the face are less than the specified angle. If any of them are greater, the domain will not be added to the face.

EraseDom **Pikd** - This command lets you pick, using the normal picking mechanisms, the domains to remove from the current face. As long as it is toggled on, picking domains removes them from the face.

EraseDom **Last** - This command removes from the current face the last domain you picked.

The Merge and TopoFltr commands are described in Section 15.10.

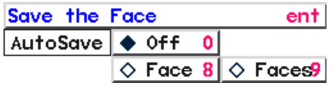
FaceFltr **All** - This radio button lets all connectors be candidates for merging when using the Merge commands.

FaceFltr **Bndry** - This radio button lets only connectors on the boundary of the developing face be candidates for merging with their neighbors..

FaceFltr **Outsd** - This radio button filters lets only connectors outside the current face boundary be candidates for merging.

Save As Baffle Face - This command is only available for unstructured block faces. For structured grids, baffles must be integrated into the block face topology. For structured and unstructured grids, baffles should be assigned as walls within the **Set BCs** command.

Once the free edges have been eliminated from an unstructured face or when a structured face has a rectangular computational mapping, the following command appears:



Save the Face - Save the current domain(s) as one face of the block. When a face is saved, its outline is rendered in white and its grid points are rendered in blue in the Display window.

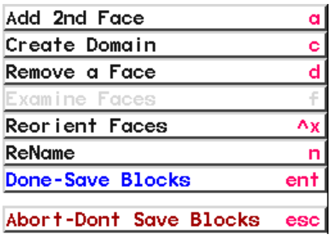
AutoSave **Off** - This radio button is the default setting which requires manual saves of all faces.

AutoSave **Face** - This radio button instructs Gridgen to automatically save each selected domain as a face.

AutoSave **Faces** - This radio button instructs Gridgen to save all possible faces and baffles in conjunction with the **All** command. This command is available for unstructured grids only.

Mesh Both Sides - For nested faces, enable this toggle to have Gridgen created tetrahedral mesh throughout the entire volume. Otherwise Gridgen will mesh between the nested faces. By default this is disabled. This command is available for unstructured blocks only.

Abort - Clear the current collection of domains from the face, pause the face creation process, and invoke the following menu.



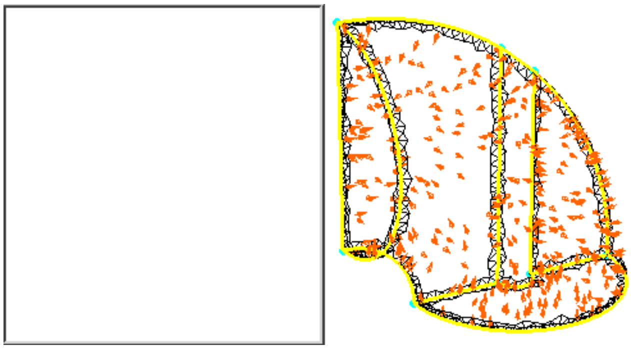
Add nth Face - Continue the block creation process by creating the nth face where n is one more than the number of faces currently in the block.

Create Domain - This command is used to create a domain during face creation in case it is discovered a domain necessary for a block has not yet been created.

Remove a Face - Delete one face in its entirety from the block. You select the face to be removed from the Display window. This does not delete the face's domains,but only the topological face entity.

Examine Faces - You may step through a graphical display of each face currently in the block using this command. This command is available for structured blocks only.

Reorient Faces - Since any face has two possible normal directions, you may have to flip the face orientation automatically chosen by Gridgen to ensure that each face's normal vector points into the volume grid. When you invoke this command the face's normal vectors are drawn as shown below. Selecting a face causes the normals to be flipped to the opposite direction. This command is available for unstructured blocks only.

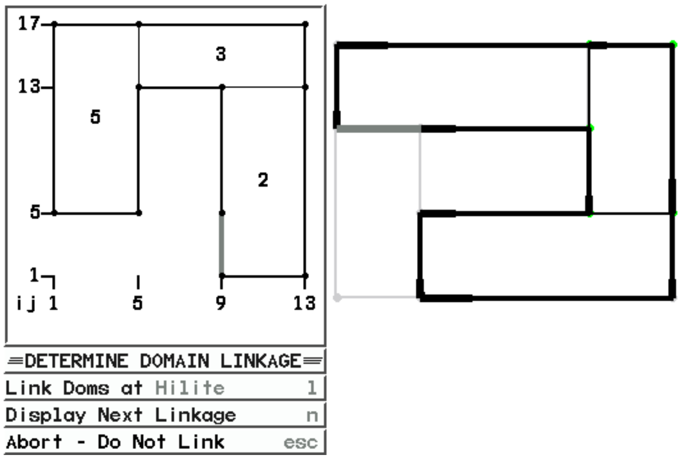


10.6 Linkages

The manner in which a domain attaches to a face during face creation is called the linkage. Sometimes a domain being added to a face has more than one linkage. In the example below, suppose the vertical domain on the left of the Display window is being added to the face. It can connect to the face in either of two places. Gridgen determines all of the possible linkages and highlights in pink the first of them in both the

Display and Blackboard windows. Three menu buttons are available for you to choose how to proceed.

Selecting the proper linkage is generally only an issue when you have a self-connecting face in a structured block, such as a face consisting of an upper and lower half-cylinder. Here, you should select the proper linkage so the face's computational extremes lie at the intended physical locations.



Link Doms at Hilite - Press this button if the highlighted linkage is the one you want.

Display Next Linkage - Press this button if you want to see the next possible linkage.

Abort - Do Not Link - Press this button if the linkage problem was caused by picking the wrong domain.

You may encounter situations when the domain to be added clearly attaches to the face, but Gridgen will not allow the connection to take place. These situations usually arise when none of the linkages maintains the rectangular alignment of the computational domain.

10.7 Structured Block Topologies and Orientations

When you save the last face of the block, Gridgen automatically attempts to find all distinct topologies in which the faces can be arranged onto the boundaries of a computational block.

1. If there are not any valid topologies, you are warned, the last face is removed from the block, and you are returned to block creation mode. Invalid blocking topologies are most often caused by having coincident connectors along the intended block edges. Any edges having coincident connectors will be highlighted in yellow when Gridgen warns of the topology problem.
2. If there is only one topology, the block has been constructed properly. The block is saved, and you may proceed with other block commands such as **Name** or **ReSpecify** ξ,η,ζ.
3. If there is more than one possible block topology, you will be warned and Gridgen's first valid topology will be used for the block. Multiple topologies usually occur when two faces in the block are defined by identical sets of domains. In this case the warning raises no concern, since the multiple valid topologies are in reality identical.

After Gridgen identifies a valid block topology, a computational coordinate system is assigned. The default (ξ, η, ζ) axes are chosen based on the first face added to the block. The block image in the Display window is color-coded to indicate the index orientations according to the following table.

Computational Axis Colors

computational coordinate	color
ξ	red
η	orange
ζ	yellow

The Display window also shows the direction of increasing computational coordinate along each block edge using color-coded arrows. The block's dimensions (number of grid points) are listed in the Blackboard using the same color coding. The orientation of the computational axis system may be changed at any time using the **ReSpecify** ξ**,** η**,** ζ button (Section 15.7).

A structured block's initial volume grid is automatically created using Standard TFI (Section 12.1).

A unique name is assigned automatically to each block when it is created. The default block name consists of combinations of the uppercase alphabetic characters. For example, the first 26 blocks created will be named A, B... Z. You can assign a more descriptive name to a block using the **Name** command (Section 15.1).

11. Grid Methods for Connectors

11.1 ReDimension

The **ReDimension** command is used to assign a number of grid points to one or more connectors. The initial dimension for new connectors is zero (undimensioned). You may override this value by using the **Defaults** command (Section 3).

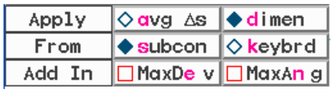
Connectors are the only entity to which you can assign a number of grid points. The dimensions of domains and blocks are defined implicitly by the connectors used in their creation. The dimension of unstructured domains and blocks is also determined by point insertion during application of the unstructured grid methods (Section 13). Only connectors that are not used in structured domains may be redimensioned using this command. Connectors belonging to structured domains must be redimensioned using the **ReDimension Grids** command (Section 11.2) The change in number of grid points is propagated throughout the multi-block grid topology.

After a connector is dimensioned, its Display window image changes from a bright green color (meaning undimensioned) to a light green color (meaning dimensioned). If the connector being redimensioned already has a dimension and a distribution of grid points (Section 11.3), the distribution will be updated automatically to match the new dimension using the current distribution constraints.

Three interrelated controls allow you to specify a connector’s dimension.

1. Specify a number of grid points or an average spacing along the connector.
2. Type in a number of grid points or select a string of connectors with the desired dimension from the Display window. You can also copy the distribution of grid points.
3. Add additional grid points to further resolve the connector’s shape. Using this option changes the connector’s dimension and distribution.

Each of these three controls is illustrated in the figure below and described in the following sections.



11.1.1 Apply Dimension or Average Spacing

You can explicitly dimension a connector by pressing the Apply **dimen** radio button. The source of the total number of points is specified by the setting of the radio buttons labeled From.

You can implicitly dimension a connector by pressing the Apply **avg Δs** radio button. The value of the average spacing is specified by the setting of the radio buttons labeled From. This only sets the number of grid points. Any existing spacing constraints will still be applied.

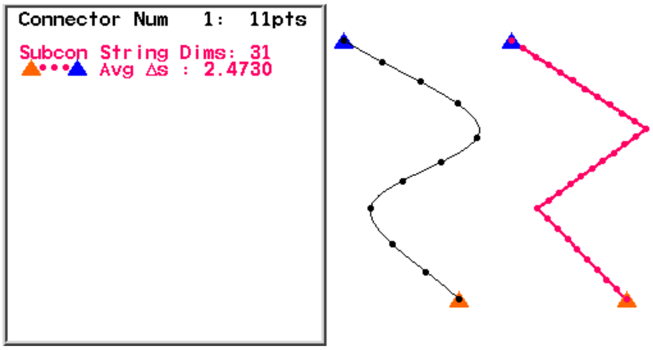
11.1.2 From Keyboard or Subconnectors

Pressing the From **keybrd** button allows you to type in either a number of grid points or a grid point spacing depending on whether the Apply radio button is set to **dimen** or **avg Δs**, respectively.

If at least one other dimensioned connector exists in your grid, you can press the From **subcon** radio button and dimension a connector by copying the dimension of one or more other subconnectors. The selected subconnectors are referred to as the *subconnector string*.

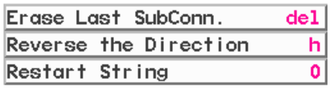
If the Apply **dimen** radio button is set, the sum of the dimensions of the subconnector string will be applied to the current connectors. If the Apply **avg Δs** radio button is set, the average spacing along the subconnector string will be applied to the current connectors.

The Blackboard window will display the dimension of each subconnector to be picked (when the cursor is positioned over it) and the cumulative dimension and average Δs to be applied to the current connectors. The subconnector string (the subconnectors must be selected in an end to end order) is highlighted in the Display window with an orange triangle at the beginning of the string and a blue triangle at the end of the string. As each subconnector is picked, it must attach to the blue triangle. In the illustration below, the three straight line connectors on the right have been selected as a subconnector string to dimension the curve on the left. Note that the Blackboard displays the current connector, the dimension of the subconnector string, and the average Δs computed from the subconnector string (total string length divided by total string number of points).



```
Connector Num   1:  11pts

Subcon String Dims: 31
▲••▲  Avg Δs : 2.4730
```

Several commands are available to assist you in picking the subconnector string.



Erase Last SubConn - Remove the last subconnector from the string.

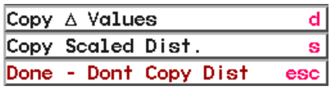
Reverse the Direction - Change the direction of the subconnector string, reversing the order of the blue and orange triangles. New subconnectors may only be added to the blue end of the string. If you choose to copy any of the string’s distribution parameters to the current connector, the colored ends of the string and the current connector must correspond.

Restart String - Remove all subconnectors from the string and start over.

Done - Apply the cumulative dimension of the subconnector string to the current connector.

Abort - End the subconnector selection process without changing the dimension of the current connector.

If the connector being dimensioned consists of a single subconnector (which will be the case if the connector is being dimensioned for the first time), you will have the opportunity to copy the distribution (Section 11.3) of grid points from the selected string of subconnectors to the current connector.



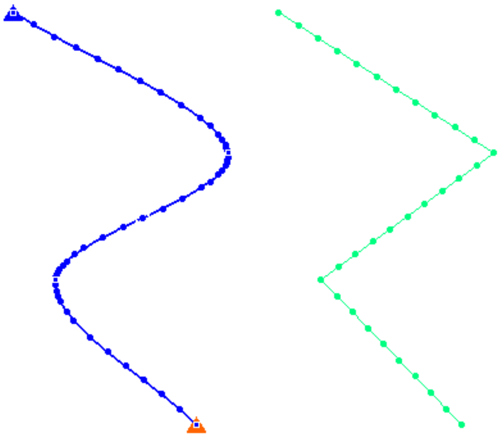
Copy Δ Values - If the string consists of a single subconnector then this command allows you to copy the distribution parameters (distribution function, spacing constraints, etc.) from the subconnector to the current connector.

Copy Scaled Dist. - Create a general distribution function (Section 11.3.2.2) on the current connector that has the same features as the distribution of grid points along the selected string, but scale the distribution from the selected string to the current connector based on the relative arc length of the string and connector.

Done - Don’t Copy Dist - Do not copy any distribution parameters from the string to the current connector; just copy the dimension.

11.1.3 Additional Effects

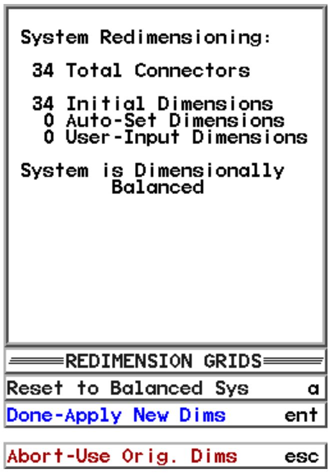
In addition to dimensioning the connector by the means described in the previous subsections, you can add to the dimension by enabling either or both of the **MaxDev** and **MaxAng** toggle buttons. These buttons allow you to enter the maximum distance allowed between the connector shape and the discrete shape defined by the grid points, and the maximum turning angle between successive grid points, respectively. In the illustration below, the curve on the left has been dimensioned by copying the dimension of the 3-line connector string on the right with points added to limit the maximum turning angle between successive grid points to 5 degrees.



11.2 ReDimension Grids

The **ReDimension Grids** command is available in the Connector, Domain, and Block menus. This command can be used to redimension a grid system containing multiple blocks and/or domains and to assign dimensions quickly to previously undimensioned connectors. The **ReDimension Grids** command is the only way to assign new dimensions to connectors belonging to structured domains, since the connector **ReDimension** command in the Modify menu (Section 11.1) is grayed-out when modifying a connector used by a structured domain.

Redimensioning proceeds by selecting connectors, assigning new dimensions to them, and for structured grids letting Gridgen propagate the new dimension to other related connectors automatically. For a structured grid, when you position the cursor over a connector, the connector and all connectors in the entire grid that must have the same dimension (the connector group) will be highlighted. When the new dimension is typed-in, it will be applied not only to the picked connector but to all other highlighted connectors.



```
System Redimensioning:
 34 Total Connectors

 34 Initial Dimensions
  0 Auto-Set Dimensions
  0 User-Input Dimensions

System is Dimensionally
        Balanced
```

When the Blackboard shows the `System is Dimensionally Balanced` message you may:

Reset to Balanced Sys - Undo your most recent redimensioning changes and return the system to its last balanced state, which Gridgen saves as the process proceeds.

Done - Apply New Dims - Gridgen will perform the following steps automatically.

- Apply the new dimensions to each redimensioned connector in the entire grid.
- Distribute the new number of grid points using the existing distribution function and spacing constraints.
- Initialize surface grids of each touched domain using each domain's last applied transfinite interpolation grid method.
- Initialize volume grids of each touched block using each block's last applied transfinite interpolation grid method.

Abort - Use Orig. Dims: Terminate the re-dimensioning process without making any changes.

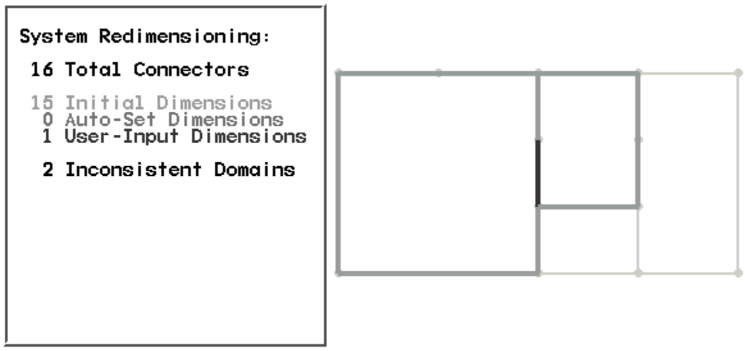
Any surface or volume structured grids touched by the redimensioning process previously run in one of the grid solvers or projected onto the database will have to be rerun or reprojected.

Other than propagating dimensional changes from one full connector to another full connector, Gridgen will not make decisions for you. Because of this, redimensioning one connector may unbalance the dimensions of other domains in a structured grid, requiring you to redimension a second connector before the entire grid is once again dimensionally balanced. Inside Gridgen, after a connector's new dimension is typed, it is assigned to all members of the connector group, and each domain's dimensions are checked. If a particular domain becomes unbalanced by virtue of the new dimension, Gridgen attempts to correct the imbalance by redimensioning connectors on its own, and the process is repeated for the next domain. All domains are checked again until repeated application of the domain balancer does not change the connector dimensions. At this point, a summary of the current system dimensional status is provided in the Blackboard Window, any unbalanced domains are highlighted in the Display window, and all connectors are assigned one of three color-coded classes according to the following table.

ReDimension Grids Colors

color	connector dimension status
yellow	original
orange	automatically assigned
pink	user set

Note that the Blackboard keeps a tally of the number of connectors of each class and the number of unbalanced domains as shown below.



```
System Redimensioning:
 16 Total Connectors

 15 Initial Dimensions
  0 Auto-Set Dimensions
  1 User-Input Dimensions

  2 Inconsistent Domains
```

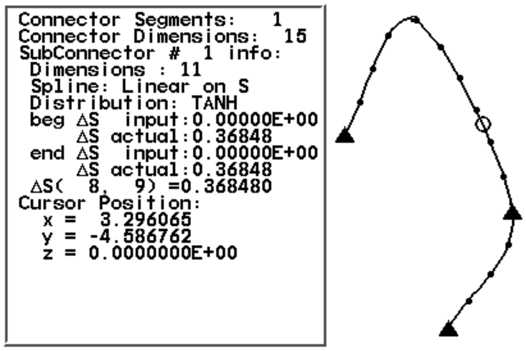
Note that the inconsistently dimensioned domains are highlighted. It is up to you to select and redimension another connector to help Gridgen propagate your changes throughout the system.

11.3 Distributing Grid Points on Connectors

The **Redistribute** command allows you to change the distribution of grid points along a connector. This command is not available for undimensioned connectors. Distribution of grid points involves two basic tasks:

- the distribution of points within each subconnector
- the division of the connector into subconnectors (optional)

11.3.1 Definitions



```
Connector Segments:   1
Connector Dimensions:  15
SubConnector #  1 info:
 Dimensions : 11
 Spline: Linear on S
 Distribution: TANH
 beg ΔS  input:0.00000E+00
     ΔS actual:0.36848
 end ΔS  input:0.00000E+00
     ΔS actual:0.36848
 ΔS(  8,  9) =0.368480
Cursor Position:
  x =  3.296065
  y = -4.586762
  z = 0.0000000E+00
```

grid points - refer to discrete locations along a connector’s shape that will become the grid points on the boundaries of surface and volume grids. The number of grid points assigned to the connector is the connector’s dimension set via the **ReDimension** command (Section 11.1). During grid point distribution, Gridgen draws the grid points (small filled circles) on the connector shape.

control points - are used to define the shape of the connector during connector creation (Section 7.2) and are rendered using small open squares. The connector interpolates the control points.

break points - are special control points that you designate at which a grid point will lie and spacing constraints can be specified. They are drawn on the connector as large filled triangles. The beginning and ending of each connector will always be break points.

cursor - is a large open circle drawn on the connector. The cursor location and the grid point spacing at the cursor’s location are displayed in the Blackboard. Horizontal mouse movement with the ***right mouse*** button pressed will move the cursor along the connector.

subconnector - is the region of a connector between two consecutive break points. The subconnector in which the cursor lies is known as the current subconnector and is always drawn in yellow. Menu buttons drawn in yellow are commands that apply only to the current subconnector. Break points at the beginning and ending of the current subconnector are color-coded in orange and blue, respectively. Break points lying on noncurrent subconnectors are drawn in red.

beginning and ending - refer to the direction in which the connector was drawn. Keep in mind that Gridgen does not restrict the manner in which connectors are drawn—drawing a line from left to right is equivalent to drawing it from right to left. Also, the connector can be drawn without regard to its eventual orientation in a structured block’s computational coordinate system. However, at some point you must be able to identify the relation of subconnectors and other components relative to each other. Gridgen’s convention is to use the terms beginning and ending in reference to the direction in which the connector was drawn. In addition, the break points are color coded (orange for beginning and blue for ending) to aid in identifying the connector’s direction.

11.3.2 Distribution Parameters

11.3.2.1 Spacing Constraint

Spacing constraint in the discussion to follow refers to the distance between the two grid points at the end of the subconnector. By default, this distance will be the total arc length `s` between the two points. However, the spacing constraint may also be set to refer to the distance in a single Cartesian coordinate direction `x`, `y` or `z` (Section 11.3.2.3). Initially, the spacing constraints are set to the type specified by the **Defaults** command (Section 3.), which is originally set to zero (unconstrained).

There are three valid entries for the spacing constraint.

`-1` - Entering a value of -1 will insure grid point spacing continuity with the adjacent connector or subconnector. For an adjacent subconnector (an internal break point), the spacing constraint stored at the common break point will be copied into the current subconnector. The result will be a continuous distribution of grid point spacing across the break point. For an adjacent connector (a break point at either end of a connector), Gridgen first checks to see whether there are any other connectors that share the same break point with the current connector. If there are, all adjacent connectors are highlighted, and you can pick one from which to copy the spacing constraint. Positioning the cursor over a highlighted connector will cause the connector to be redrawn with its grid points turned on. Also, information will be displayed about the highlighted connector in the Blackboard, including its spacing constraint at the adjacency. In this manner you can browse through all highlighted connectors until the desired spacing is found.

`0` - Entering a value of zero removes the spacing constraint from (unconstrains) the break point. If both of a subconnector's break points are unconstrained, the grid points will be equally spaced along the connector. If one of a subconnector's two break points is unconstrained, the distribution function switches to a one-sided form. If neither break point is un-constrained, the distribution function uses a two-sided form.

`>0` - A positive value will constrain the distance between the break point and the first grid point from the break point to be the typed-in value. The maximum allowable spacing constraint is 40% of the total distance between the two subconnector end points. An input value above the limit will result in a request for a more reasonable number.

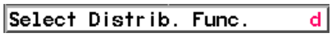
Three menu buttons allow you to assign spacing constraints:

Begin. Δ - The spacing constraint you enter will be applied to the break point at the beginning of the subconnector. The beginning break point is orange.

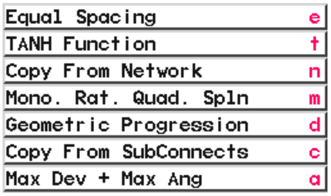
Ending Δ - The spacing constraint you enter will be applied to the break point at the end of the subconnector. The end break point is blue.

Enter Δs at Picked Δ - The spacing constraint you enter will be applied to both sides of the break point on which the cursor is currently positioned. This command will guarantee grid point spacing continuity at the break point, but may not be used at the beginning or ending of the connector since only one spacing constraint exists at each of those points. This command will always enforce a constraint on the arc length, even if the spline variable (Section 11.3.2.3) is selected to be `x`, `y`, or `z`. To specify a Δ`x`, Δ`y` or Δ`z` constraint on both sides of a break point, set the spline variable to the appropriate value and then select the **Begin.** Δ and **Ending** Δ menu buttons successively on the two subconnectors touching the break point.

11.3.2.2 Distribution Function



The spacing constraints are the primary control in determining the distribution of grid points along the subconnector, but it is the distribution function (a mathematical equation) that determines how the points are distributed between the constraints. Gridgen's distribution functions can be selected by pressing the **Select Distrib. Func.** button.



Equal Spacing - This is not a true distribution function. This is a shortcut command to convert the distribution function to hyperbolic tangent and unconstrain both the beginning and ending spacing constraints.

TANH Function - Gridgen's default distribution function is the hyperbolic tangent (Reference 36). If the spacing is unconstrained at both ends, grid points will be distributed uniformly using a simple uniform interpolation scheme. If the spacing is unconstrained at only one end, an alternative one-sided hyperbolic tangent distribution function is used. Hyperbolic tangent works very well for the majority of cases and for that reason it is the default distribution function.

Copy From Network - This distribution function positions the grid points so they lie exactly on the intervals of a database bilinear surface entity (network). This distribution will not be available for selection unless four criteria are met:

- All segments in the connector are Line on DB segments, and all segments lie on the same database bilinear surface (network).
- Each control point lies exactly on a network interval's corner.
- All control points share either a u or a v parametric database coordinate.
- The dimension of the connector is equal to one more than the total number of intervals traversed on the network by the connector.

Mono. Rat. Quad. Spln - The Monotonic Rational Quadratic Spline (MRQS) function (Reference 1) can be applied to several subconnectors simultaneously. The MRQS function will force a grid point to be fixed to a break point and automatically provide spacing continuity and spacing variation continuity across the break point. Gridgen links two MRQS subconnectors if their spline variables and spline types are the same and if the spacing constraints across their common break point are equal.

Geometric Progression - This is a one-sided distribution function. It is available only if a spacing constraint is specified at one end of the subconnector. If no spacing constraints are specified, the grid points will be distributed uniformly. Geometric progression works by distributing the grid points from the constrained end so the ratio of spacing between adjacent grid points is constant. Therefore, if tight clustering is specified at one end, the distance between successive points will increase monotonically towards the other end of the subconnector. A warning will be issued if both ends of the subconnector are constrained, and the subconnector will be converted to a hyperbolic tangent distribution. Therefore, you must unconstrain at least one of a subconnector's break points before you can apply the geometric distribution function.

Copy From SubConnects - This option allows you to copy the distribution of grid points from any subconnectors in the grid and apply it to the current subconnector. After pressing the menu button, position the cursor over the subconnector to copy from and press the ***right mouse*** button. More than one subconnector can be selected as long as all picked subconnectors are pointwise continuous. The selected subconnectors are rendered in the Display window as a highlighted string with an orange triangle at the beginning of the string and a blue triangle at the end of the string. Each successive subconnector you pick may only be attached at the blue triangle. To attach a new subconnector to the opposite end, the string must first be reversed in direction using the **Reverse the Direction** menu button. During selection of the subconnector string, the last loaded subconnector can be erased with the **Erase Last SubConn** menu button. All loaded subconnectors can be erased with the **Restart String** menu button. When the desired string of subconnectors to be copied is picked, select the **Done Picking Subconns.** menu button. When copying the distribution function you are also given the choice of copying either the exact spacing constraints (**Copy ∆ Values**) or the relative distribution of points (**Copy Scaled Dist**). Alternately, the **Abort-Don't Copy Dist.** menu button will abort from the command, retaining the previous distribution. If the **Copy Scaled Dist.** command is selected, the resulting distribution function will be a General distribution, indicating that it does not follow any of Gridgen's mathematical forms, but was formed by copying from some other subconnector.

Max Dev + Max Ang - This command allows you to enter values for:

- the maximum deviation of the discrete connector from the segment shape
- the maximum turning angle allowed between adjacent grid points along the connector
- a parameter designating whether or not the current beginning and ending spacing constraints should be maintained or released

Note that unlike implementations of this feature in the Defaults menu (Section 3) and during connector dimensioning (Section 11.1), the connector dimension is NOT changed when this feature is applied as a distribution function. This means that the deviation and turning angle limits you specify here will not be enforced exactly, but the general trend in grid point clustering will be applied. Furthermore, if you choose to maintain the beginning and ending spacing constraints when applying this distribution function be aware that the level to which the limits will be enforced is seriously degraded.

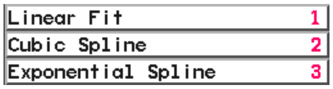
General - This type is not a distribution function that you can explicitly apply to a connector. It is applied to connectors by Gridgen whenever the distribution of points along a connector is created by something other than one of the mathematical distribution functions. This includes:

- **Copy from SubConnects** distribution function
- Ortho BC in the elliptic PDE solver
- Grid point import
- Projecting a connector onto the database
- Copying a connector off the database

11.3.2.3 Spline

Gridgen uses the distribution function and the spacing constraints to establish an array of grid point coordinates in terms of the splining variable. It then couples the grid point spline variable array with the connector shape to create Cartesian coordinates `(x,y,z)` using a spline function.

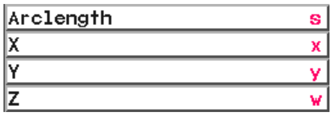
Three spline types are available in Gridgen and are selected with the **Spline Type** menu button.



Linear Fit - This is Gridgen's default spline type. The connector definition is treated as a series of connected line segments, and grid points are placed on these segments. Since Gridgen uses the same dense collection of points used to render the connectors as the basis for the spline routines, the result of using a linear spline fit will be grid points that almost always appear to stay on the connector definition.

Cubic Spline - This spline type will fit a cubic polynomial through successive discrete data points that define the connector shape. Cubic splines have the benefit of having slope continuity at both sides of a control point, instead of mere point continuity as exhibited in the linear spline. The cubic spline's most attractive feature is that it can pass a smooth curve through relatively coarse data; its drawback is that it can produce false inflection points. The only instance where this type of spline might be beneficial is when a large number of grid points are to be distributed on a smooth shape represented by a relatively coarse collection of straight line segments (as might be obtained by an Import segment type). Even then the cubic spline shape should be inspected carefully.

Exponential Spline - This spline type has features of both the linear and cubic spline. It should be used when a cubic spline is needed but cannot be used because of unwanted wiggles in the grid point representation. The exponential spline will eliminate those wiggles.



The spline variable can be changed with the **Spline Var.** menu button. By default, grid points are distributed based on arc length along the connector. Hence, a spacing constraint of 0.4 will make the distance between adjacent grid points at the end be 0.4. However, you may want to distribute grid points on the basis of only one of the coordinate components such as x, y or z instead of arc length. If the spline variable is changed to x, for example, a spacing constraint of 0.4 will result in a grid point distribution such that the difference in x values of grid points at the break point will be 0.4. The arc length spacing at the break point will, in general, not be equal to the Δx value.

In order to use a spline variable other than arc length, the subconnector must be monotonic (either increasing or decreasing) in the spline variable. If the subconnector is not monotonic, Gridgen prints a warning and the spline variable is set to arc length. Furthermore, if the connector shape is later modified so that the subconnector is no longer monotonic in the selected spline variable, the spline variable will be reset automatically to arc length.

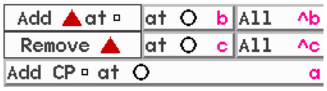
A spacing constraint prescribed on a non-arc length splined subconnector is stored in terms of the arc length spacing that would provide the specified spacing constraint in the splining variable. In this way, if the spacing on the end of a subconnector is copied from the adjacent subconnector, the spacing will be continuous in terms of arc length at the common break point, even if the splining variable is not arc length on either side of the break point. This will also keep the spacing identical at the ends of a subconnector if the splining variable is changed from its original value.

11.3.3 Break Points and Subconnectors

To gain further control of grid point distribution on a connector it can be split into two or more subconnectors. By splitting the connector into subconnectors at user-specified break points, you can apply distribution functions and spacing constraints to each individual subconnector. There are two main reasons you would need to add a break point to a connector:

- To force a grid point to lie at a specific location since, by definition, a grid point will lie at every break point. By placing a grid point (via a break point) at a slope discontinuity you would ensure that the discrete (grid point) representation of the connector was true to the original shape.
- To have the ability to specify a spacing constraint on a connector’s interior rather than just at its ends.

11.3.3.1 Setting, Clearing, and Editing Break Points



A connector is divided into subconnectors by identifying a control point as a break point. This will divide the current subconnector into two subconnectors at the new break point. To do this, position the cursor ❍ over the control point and select the Add Δ at □ **at** ❍ menu button. If the cursor is not close enough to the control point, Gridgen will give a warning. Once picked, the control point will have a filled triangle drawn on it, meaning that the connector has been divided into two subconnectors. The current subconnector will be indicated by a yellow color, and information pertinent to the current subconnector will be displayed in the Blackboard window. Moving the cursor (using the mouse or menu button controls) across the new break point will change the current subconnector.

Instead of adding one break point at one control point you can add a break point at every control point by pressing the Add Δ at □ **All** button.

Break points (and therefore subconnectors) can be deleted by positioning the cursor over the break point and selecting the Remove Δ **at** ❍ menu button. This will turn the break point into a control point and will combine the subconnectors on each side of the break point into a single subconnector.

To delete all of the break points and get only one subconnector you should press the Remove Δ **All** button.

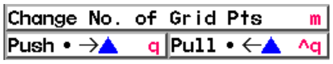
When a break point is needed where no control point exists, the **Add CP** □ **At** ❍ menu button may be used. This will add a control point to the connector at the current cursor position. This new control point can then be picked to be a break point as described in the previous paragraph. The **Add CP** □ **At** ❍ menu button is the only command available under the Distribution commands menu that will alter the segment definition (though not the shape) of the connector. Any control point added using this technique can only be deleted using the **Edit** command in the Connector Modify menu. The **Add CP** □ **At** ❍ command should not be used indiscriminately. For example, if the cursor is on the branch segment between two database constrained segments, a control point may not be added and you will be warned.

Two additional buttons are available for quick editing of the subconnector setup. The Δ **to Next** □ is used to move a break point one control point toward the end of the connector. The Δ **to Prev** □ button will move a break point one control point toward the beginning of the connector. The buttons are used by first positioning the cursor over the break point to be moved. If the cursor is not positioned over a break point, a notice is given and nothing changes. If the control point to the right or left (depending on the button selected) is already a break point, you are informed that the break point can not be shifted in the desired direction.

11.3.3.2 Dimensioning Subconnectors

When a connector has been split into multiple subconnectors, you can control the number of grid points in each subconnector, subject to the connector's total dimension. Specifically, the sum of the number of grid points in each subconnector should equal the connector's dimension minus the number of interior break points. For example, three subconnectors dimensioned 11, 24, and 18 will have an overall connector dimension of (11+24+18)-(2)=51. Here, two points are subtracted for the two interior break points. When a subconnector is split into two subconnectors by setting a break point on a control point, the dimensions of the current subconnector are automatically divided between the two new subconnectors on the basis of the subconnectors' arc lengths. If the current subconnector contains only two grid points, it will not be possible to select a break point on the subconnector interior since each subconnector needs at least two grid points.

Three commands allow you to change a subconnector's dimension.



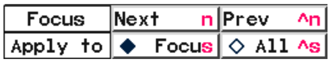
Change No. of Grid Pts - This command allows you to type-in a new dimension.

Push • → Δ - This command reduces the current subconnector's dimension by one and adds one to the dimension of the following subconnector.

Pull • ← Δ - This command increases the current subconnector’s dimension by one by taking one grid point from the following subconnector.

In order to keep the number of grid points on the connector constant when re-dimensioning a subconnector, Gridgen confines its dimension changes to the current subconnector and the subconnector immediately following it. Notice that this limits the range of new dimensions for the current subconnector to the range 2 to n_1+n_2-2, where n_1 is the number of points in the current subconnector, and n_2 is the number of points in the following subconnector. Notice also that it is not possible to change the number of grid points in a connector’s last subconnector. The dimension of the last subconnector can only be modified implicitly by changing the dimensions of any or all of the subconnectors preceding it.

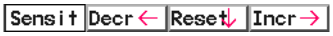
11.3.4 Miscellaneous Controls



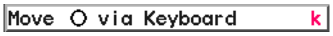
When more than one connector has been selected for the **ReDistribute** command the Focus **Next** and Focus **Prev** commands move focus to the next and previous selected connectors in numerical order.

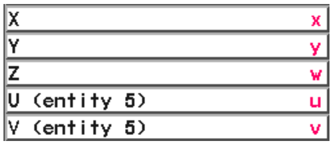
The Apply to **Focus** and **All** radio buttons establish whether the grid point distribution attributes you select should be applied to the focus connector only or all selected connectors.

The cursor may be moved along the connector by moving the mouse to the left and right with the ***right mouse*** button pressed. Other controls are available for moving the cursor.

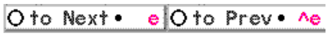


Decr, **Reset**, and **Incr** allow you to decrease, reset, and increase the sensitivity of the cursor movement with respect to mouse movement, respectively.



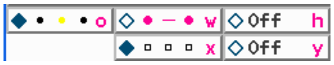


Move ❍ via Keyboard - Pressing this button allows you to move the cursor to a specific x, y, or z value on the connector. You can also choose to move to a specific u or v value if the connector shape is constrained to a database entity. Choose the coordinate via the menu and type-in the specific value. If there are several locations on the connector that have the value of the coordinate that you specify, Gridgen will move the cursor to the value that is closest to the original cursor location.



❍ to Next • - Pressing this button moves the cursor to the next grid point on the connector.

❍ to Prev • - Pressing this button moves the cursor to the previous grid point on the connector.



The first row of radio buttons provides options for the display of the focus connector's grid points.

• • • - This radio button displays the connector's grid points as small, filled, yellow and white circles depending on whether or not the grid point is in the current subconnector, respectively. This is the default setting.

• - • - This radio button displays the connector's grid points as slightly larger, filled, pink circles and connects the grid points with a dashed pink line. This display option is useful for visualizing the faceting effect of the grid point distribution on the connector shape.

Off - This radio button turns off the display of grid points.

The second row of radio buttons provides two options for the display of the focus connector's control points.

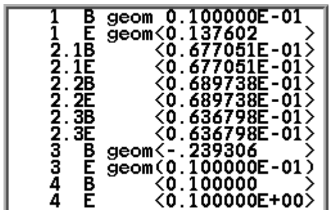
☐ ☐ ☐ - This radio button displays the connector's control points as small, open, white squares. This is the default setting.

Off - This radio button turns off the display of the focus connectors control points. Individual control points are temporarily displayed when the connector hiliter moves near them.

11.4 Set Δs Values

Rather than use the **Redistribute** command in the Connector Modify menu for each connector in your grid, one connector at a time, you can use the **Set Δs Vals** command to specify spacing constraints for many break points at once. This command also allows you to change the distribution function of these connectors. And, like the **ReDimension Grids** command, the changes are propagated throughout the grid causing domain and block grids to be updated. The **ImmdRegen** toggle button determines whether or not your spacing constraint changes cause immediate regeneration of surface and volume grids (enabled by default) or whether the surface and volume grids are only regenerated when you exit the command.

Spacing constraints may be picked from either the Browser or Display window. When picking from the Display window the first and last 10% of each subconnector's length is pickable, corresponding to the region close to the break point. All picked spacing constraints are shown in the Display window as the connector's 10% of length with an arrow head pointing toward the constraint location. The Browser lists each spacing constraint as shown below.



```
  1   B geom 0.100000E-01
  1   E geom<0.137602     >
  2.1B      <0.677051E-01>
  2.1E      <0.677051E-01>
  2.2B      <0.689738E-01>
  2.2E      <0.689738E-01>
  2.3B      <0.636798E-01>
  2.3E      <0.636798E-01>
  3   B geom<-.239306     >
  3   E geom(0.100000E-01)
  4   B     <0.100000     >
  4   E     <0.100000E+00>
```

The first number listed in each entry is the connector number. If subconnectors are present, then they are listed to the right of the decimal point. Therefore, `2.2` in the example above refers to the second subconnector of connector number 2.

The B and E refer to the beginning and ending break point of the subconnector, respectively.

The next field lists the distribution function if it is *not* hyperbolic tangent (TANH). The possible values of this field are

- **blank** - hyperbolic tangent

- **mrqs** - monotonic rational quadratic spline
- **geom** - geometric progression
- **gnrl** - general

The final field lists the current value of the spacing constraint in terms of arc length along the connector. If the spacing constraint is surrounded by angle brackets (< and >), the break point is actually unconstrained, and the current spacing is being listed. If the spacing constraint is surrounded by parentheses, the subconnector has a spline variable (Section 11.3.2.3) other than arc length. Constraints without angle brackets are ones specified by you.

Once you have selected the spacing constraint locations (break points) you wish to set, you are prompted to enter the new constraint value. Valid entries are:

-3 - convert the distribution function of all of the subconnectors referenced by the picked spacing constraints to MRQS (monotonic rational quadratic spline)

-2 - convert the distribution function of all of the subconnectors referenced by the picked spacing constraints to TANH (hyperbolic tangent)

0 - unconstrain all selected constraints

>0 - enter a new spacing constraint. The average spacing constraint of all of the picked spacing constraints is listed in the message window.

When a new constraint value is entered three things happen:

- Gridgen redistributes grid points on connectors subject to the new constraints.
- Domain grid points are updated to reflect the new connector grid point positions.
- Block grid points are updated to reflect the new domain grid point positions.

Note that structured domains and blocks that have been run in one of the solvers (Section 12.) will be regridded in a manner that takes into account the grid's current qualities. In other words, TFI is only applied to those grids that have only been TFI'd. Elliptically smoothed grids will be modified in a manner that takes into account the smoothness, clustering, and orthogonality the grid currently has. This is true even for structured domains constrained to the database. This regridding technique works well for small perturbations of the connector's spacing constraints. Large perturbations may yield unacceptable results, in which case the grid methods should be reapplied manually.

There is no regridding technique for unstructured grids. Unstructured domains are regridded using the last applied unstructured solver attributes. Unstructured blocks must be regridded manually.

12. Grid Methods for Structured Domains and Blocks

There are two types of grid methods that you can apply to structured surface (domain) and volume (block) grids: algebraic, and elliptic PDE.

Algebraic methods in Gridgen are based on transfinite interpolation (TFI). Domain and block grids are initialized using TFI. Gridgen automatically chooses the specific TFI method for you in order to make the best possible grid. You may re-apply TFI to any grid at any time to “start over” with a new grid. TFI methods have the advantage that they can be computed relatively quickly.

Elliptic PDE methods in Gridgen are based on iterative solution of the elliptic partial differential Poisson’s equation. You may apply the elliptic PDE methods to any grid at any time in order to improve the smoothness, clustering, and orthogonality of the grid. The elliptic PDE method’s default controls have been established to give you the best possible grid. Additionally, a wide variety of controls are available for fine control over the grid characteristics.

The algebraic and elliptic PDE methods are both available via the Run Solver **Structured** command in both the Domains and Blocks menus. The following sections describe in detail how to apply each grid method.

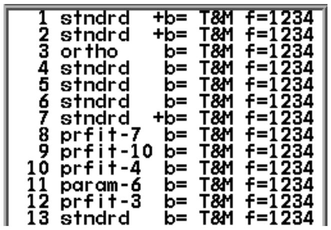
If you select the **List by BC** toggle in the Domain Commands menu, Gridgen will display the boundary condition set on each domain when you are selecting domains to run in the solver. This may make it easier to select the domains you want to elliptically refine.

12.1 Algebraic Methods

The **Initialize via TFI** command in the Run Structured Solver menu for both domains and blocks is used to modify grid point locations on the interior of selected grids through application of one of several algebraic transfinite interpolation (TFI) (Reference 9, Reference 17, Reference 19) methods. TFI is automatically applied to a grid when it is first generated but can be applied manually any time thereafter as well.

Running the algebraic solver proceeds in three steps.

1. Select the grids to which TFI is to be applied from either the Browser or Display window. The TFI method last applied to each grid is listed in the Browser window’s second column after the grid number. The appearance of an plus sign or an asterisk after the algebraic method type indicates the following:
   a. A plus sign (+) indicates a grid that has been “locked” by application of the elliptic PDE solver.
   b. An asterisk (*) indicates a grid that has been “locked” by application of the hyperbolic solver, by importing the grid from a file, or by projection, scaling, or stretching.

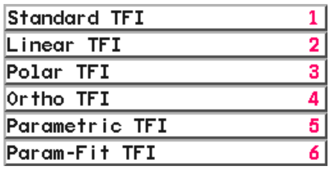


```
 1 stndrd  +b= T&M f=1234
 2 stndrd  +b= T&M f=1234
 3 ortho    b= T&M f=1234
 4 stndrd   b= T&M f=1234
 5 stndrd   b= T&M f=1234
 6 stndrd   b= T&M f=1234
 7 stndrd  +b= T&M f=1234
 8 prfit-7  b= T&M f=1234
 9 prfit-10 b= T&M f=1234
10 prfit-4  b= T&M f=1234
11 param-6  b= T&M f=1234
12 prfit-3  b= T&M f=1234
13 stndrd   b= T&M f=1234
```

2. To re-apply the last TFI method used for each selected grid…
   a. Press the **Initialize via TFI** button.
3. To apply a new TFI method…
   a. Press the **Set Solver Attributes** button.
   b. Press **TFI Methods** button.
   c. Press the button for the specific TFI method to be applied.
   d. Press the **Done Setting Attributes** button.
   e. Press the **Initialize via TFI** button.
4. Select other grids to run or press the **Done** button.

All TFI methods in Gridgen are based on a set of closed form algebraic equations that use the grid points on the boundaries of the surface to calculate grid points on the interior. Only Parametric TFI and Parametric Fit (see below) constrain grid points to the database shape. All other methods calculate `x`, `y`, and `z` solely from the distribution and location of points on the boundaries. The resulting grid shape depends on the particular TFI method used.

The TFI methods are described in the following sections.



12.1.1 Standard TFI

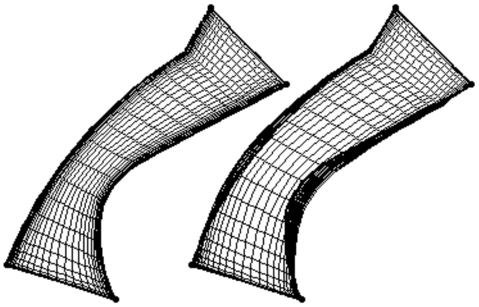
Standard TFI applies blending functions based on the relative spacing of the grid points on the boundaries (Reference 20). The TFI algorithm is applied independently to the `(x,y,z)` coordinates of a grid. Because the blending functions are arc length based, the resultant grids usually have satisfactory clustering.

Standard TFI is the third method after Parametric and Parametric Fit TFI that Gridgen tries to apply by default to create the initial surface grid for domains. Standard TFI is the default method for new blocks.

12.1.2 Linear TFI

Linear TFI applies the TFI algorithm to the `(x,y,z)` coordinates of the grid, but the blending functions are no longer related to the distribution of grid points on the boundaries. This method is rarely better than Standard TFI but is provided for comparison and historical purposes.

Standard TFI (left) is compared to Linear TFI (right) below.

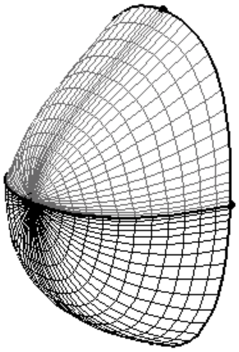


12.1.3 Polar TFI

Polar TFI method is best suited for grids intended to be shaped like a body of revolution. Polar TFI requires you to either enter or pick (via the Display window) two non-coincident points lying on the intended axis of revolution. Gridgen then transforms the `(x,y,z)` grid point coordinates on the boundaries to polar coordinates `(r,α,z')` relative to the axis of revolution. The TFI algorithm is applied to each of the polar coordinates. Finally, the polar coordinates are transformed back to Cartesian coordinates.

Polar TFI is not available for blocks.

Polar TFI (bottom) is compared to Standard TFI (top) in the figure below.



12.1.4 Ortho TFI

Ortho TFI uses a generalized form of the TFI equations allowing the specification of first derivative values at the grid boundaries. First derivatives are computed on the boundaries so grid lines intersect the boundaries orthogonally with spacing interpolated from the adjacent boundaries.

Ortho TFI is not available for blocks.

12.1.5 Parametric TFI

Parametric TFI can be used when all of the grid points on the boundaries of a grid are on the same database entity. In this method, the basic TFI algorithm is applied to the parametric database coordinates `(u,v)`. Once the parametric coordinates`(u,v)` are computed at each grid point, the physical coordinates `(x,y,z)` are calculated from direct evaluation of the database entity. The result is a grid that lies exactly on the database entity. If you apply Parametric TFI to grids that do not reference one and only one database entity, you will be warned and the grid points will not be updated.

Parametric TFI is the first method that Gridgen tries to apply by default to create the initial surface grid for domains.

Parametric TFI is not available for blocks.

12.1.6 Parametric Fit TFI

Param-Fit TFI is similar to Parametric TFI in that it will produce grids that lie on the database shape. As is the case for Parametric TFI, the grid points will be on a single database surface entity. However, it is much more general than Parametric TFI because the boundary points do not have to lie on a single database entity. Two less stringent conditions must be met before Parametric Fit TFI may be applied.

1. A certain percentage of the grid's boundary points must reference database entities of any kind.
2. All of the grid's boundary points must lie within the extent box of at least one database surface entity.

If the grid passes the two tests above, then the grid's boundary points are projected (using closest point) onto each of the candidate (according to 2. above) database surfaces. The projection that results in the smallest cumulative movement of the boundary points determines the database entity to use for Parametric TFI. The grid's boundary points are projected onto the entity, and the resulting `(u,v)` coordinates are used to drive Parametric TFI. Note that the projection of boundary points is only a temporary operation; the original coordinates of the boundary points are restored.

Parametric Fit TFI is the second method after Parametric TFI that Gridgen tries to apply by default to create the initial surface grid for domains. For Parametric Fit TFI to be applied automatically to a new domain, a third condition must be satisfied:

3. The total distance projected from the grid's boundary points to the candidate surface must be less than a tolerance.

Parametric Fit TFI is not available for blocks.

12.1.7 Pole

You cannot explicitly apply Pole TFI. However, no matter what TFI method you choose, if Gridgen determines the domain is a pole (a surface collapsed to a line), it uses a special method for creating the interior domain grid points. Specifically, the grid is made by copying the edge points onto all other grid lines in the domain. This approach prevents round-off error that sometimes results from applying a TFI method to a singularity. Domains constructed in this manner will be tagged with the label "pole" in the Blackboard window.

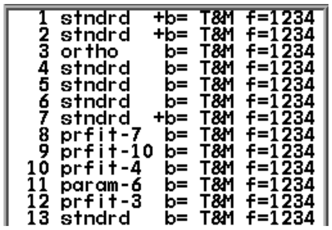
12.2 Elliptic PDE Methods

Gridgen's elliptic PDE methods let you improve the smoothness of structured domains and blocks, while at the same time controlling orthogonality and clustering. For domains, the elliptic methods can be constrained to a database so that grid quality is improved and the desired shape is maintained.

12.2.1 Running the Solver

The Elliptic Slvr **Run** menu button is used to apply an elliptic partial differential equation (PDE) method (Reference 40, Reference 35) to one or more grids. The elliptic solver iteratively solves a set of PDEs to improve the quality of the grid. In the following discussion, the term "refine" refers to improving the grid's quality, not improving its resolution through the addition of grid points.

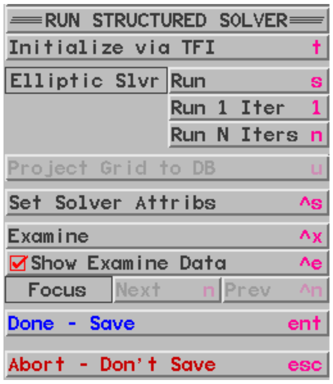
The first step in running the elliptic solver is to select, from either the Browser or Display window, the grids to be run. During selection of the grids, a terse description of the current elliptic PDE attributes (described in the next section) will be displayed for each grid in the Browser.



```
 1 stndrd  +b= T&M f=1234
 2 stndrd  +b= T&M f=1234
 3 ortho    b= T&M f=1234
 4 stndrd   b= T&M f=1234
 5 stndrd   b= T&M f=1234
 6 stndrd   b= T&M f=1234
 7 stndrd  +b= T&M f=1234
 8 prfit-7  b= T&M f=1234
 9 prfit-10 b= T&M f=1234
10 prfit-4  b= T&M f=1234
11 param-6  b= T&M f=1234
12 prfit-3  b= T&M f=1234
13 stndrd   b= T&M f=1234
```

For example, in the sample shown above, grid number `2` has Thomas & Middlecoff background control functions (`b= T&M`) and with foreground control functions turned on for boundaries 1, 2, 3, and 4 (`f=1234`).

When you have selected any number of grids, the elliptic solver is invoked and immediately enters the “paused” state with the menu shown below.



Initialize via TFI - Apply the last used algebraic method to each picked grid (Section 12.1).

Run - Begin execution of the elliptic solver and continue indefinitely, until you select the **Pause** menu button.

Run 1 Iter - Run 1 iteration of the elliptic solver and return the solver to the paused state.

Run N Iters - You type in the number of iterations to run the solver. Execution begins immediately after the number is entered and will continue through the input number of iterations, unless you select the **Pause** button.

Pause - Stops the solver (not shown on above menu).

Project Grid to DB - Will project any grids with database constrained shapes to the database using the current shape attributes. This command is not available for blocks.

Set Solver Attribs - Set solver attributes, such as control functions, on the grids currently being run in the solver.

Examine - Go to the Examine menu to inspect grid quality and select the grid quality function to be displayed.

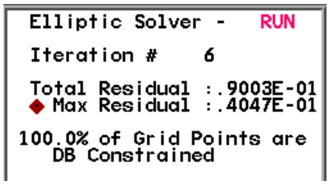
Show Examine Data - Display grid quality while elliptic solver is paused using the functions available in the Examine menu.

Focus **Next** - Examine the next grid entity in the list of grids being run in the elliptic solver. Only applies if Mode **Local** has been selected in the Examine menu.

Focus **Prev** - Examine the previous grid entity in the list of grids being run in the elliptic solver. Only applies if Mode **Local** has been selected in the Examine menu.

Done - Save - Exit the solver and save the current grids.

Abort - Don't Save - Exit the solver and revert the selected grids, including the solver attributes, back to the state that existed at the time the **Done** button was pressed.

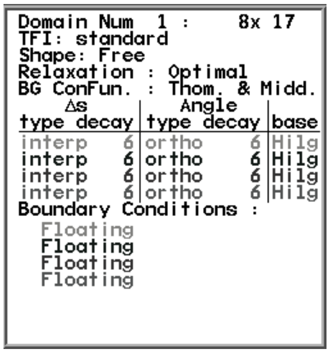


```
Elliptic Solver -   RUN

Iteration #    6

Total Residual :.9003E-01
◆ Max Residual :.4047E-01

100.0% of Grid Points are
  DB Constrained
```

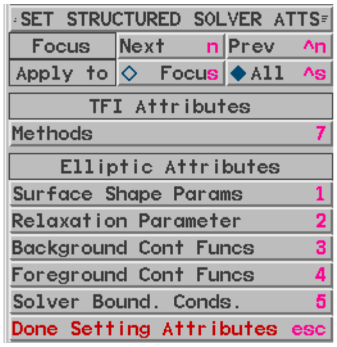
While the elliptic solver is running, the current iteration number, the total residual, and the maximum residual are displayed in the Blackboard window. The total residual is the root mean square (RMS) of the grid point movement per iteration summed over all points in the selected grids. The maximum residual refers to the largest single RMS term. The grid point where it occurs is highlighted in the Display window using the same small glyph (red diamond) shown in the Blackboard. The percentage of the grid points that are currently constrained to lie on database entities is also displayed.

12.2.2 Setting Attributes

When the **Set Solver Attribs** command is selected the Blackboard will display the current attributes of the focus grid as shown on the left below. The various attributes can be set using the menu shown on the right.



```
Domain Num  1 :    8x 17
TFI: standard
Shape: Free
Relaxation : Optimal
BG ConFun. : Thom. & Midd.
     Δs     |    Angle    |
type  decay |type   decay |base
interp    6 |ortho      6 |Hilg
interp    6 |ortho      6 |Hilg
interp    6 |ortho      6 |Hilg
interp    6 |ortho      6 |Hilg
Boundary Conditions :
   Floating
   Floating
   Floating
   Floating
```



Focus **Next** - Moves focus to the next grid in numerical order of the grids selected for the solver. The focus grid is the grid whose attributes are listed in the Browser and whose image is highlighted in the Display window.

Focus **Prev** - Moves focus to the previous grid in numerical order of the grids selected for the solver. The focus grid is the grid whose attributes are listed in the Browser and whose image is highlighted in the Display window.

Apply to **Focus** - This toggle button specifies that the selected attributes are to be applied to the focus grid only. When enabled, this toggle gives you access to the individual boundaries of the focus grid for setting foreground control functions and boundary conditions.

Apply to **All** - This toggle button specifies that the selected attributes are to be applied to all grids currently selected for the solver. When enabled, this toggle applies foreground control function and boundary condition attributes to all boundaries of all selected grids.

TFI Methods - The TFI methods are used to establish an initial grid from the boundary point information only (Section 12.1).

Surface Shape Params - These attributes define how the shape of the selected surface grids is computed. These settings are not available for blocks.

Relaxation Parameter - This attribute controls the convergence rate of the iterative solution of the PDE.

Background Cont Funcs - The background control function influences the distribution of grid points on the interior of the grid.

Foreground Cont Funcs - The foreground control function influences the distribution of grid points near the boundaries of the grid.

Solver Bound. Conds. - Movement of the grid points on the grid boundaries during elliptic refinement is controlled with this attribute.

Two other commands come into play when you are setting attributes. The following two buttons appear within each of the specific attribute menus.

Set Other Attributes - Pressing this button will invoke the main solver attributes menu so that you may select a new attribute type to set.

Done Setting Attributes - Pressing this button will invoke the solver and enter "pause" mode. You may then press the **Run** button to continue running the solver with the new attributes.

12.2.3 Relaxation Parameter

Gridgen calculates the elliptic PDE solution on a grid using a successive over-relaxation (SOR) numerical algorithm. This is an iterative method to converge upon the grid point locations that satisfy the elliptic PDEs. After each iteration, the grid point solution is "relaxed" according to the following equation

$$\mathbf{r}_{new} = \omega\ \mathbf{r}_{PDE} + (1-\ \omega)\ \mathbf{r}_{old}$$

in which **r** is the grid point vector and ω is the relaxation factor. With a relaxation factor of 0, the grid will not move from iteration to iteration, since the new value will be exactly equal to the old value. Notice also that with ω = 1, the new value of the grid point will be exactly that calculated from the PDE. It is possible to use values of ω greater than one, hence the term over-relaxation.

The focus grid's relaxation factor is shown in the Blackboard with the label `Relaxation`.

Use Optimal Relaxation - This will calculate a unique optimal relaxation factor at each grid point based on an eigenvalue analysis of the finite difference stencil used to solve the PDE (Reference 7). It works very well for most cases, but is not always stable on grids with pronounced surface curvature (because ω locally reaches values of 1.8 or more), possibly shooting grid points off of the surface or causing convergence problems.

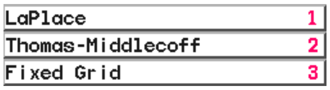
Use Nominal Relax 0.7 - This button assigns an under-relaxed value of ω = 0.7 to all grid points. This value almost always results in a stable iterative solution to the elliptic PDE and is best for database constrained grids.

Enter Relax via Keyboard - You can type in a value for ω with this option. Input values must be in the range $0 < \omega < 2.0$.

12.2.4 Background Control Functions

Control functions control the distribution of grid points during iterative solution of the elliptic PDE. These control functions are calculated and updated automatically between elliptic PDE iterations and are formed as a blended combination of two independent function sets known as background and foreground control functions.

Background control functions are used to influence the distribution of grid points on the interior of the grid. Three different background control functions, described below, are available in Gridgen. The current background control function applied to the focus grid is displayed in the Blackboard window with the label `BG ConFun`.

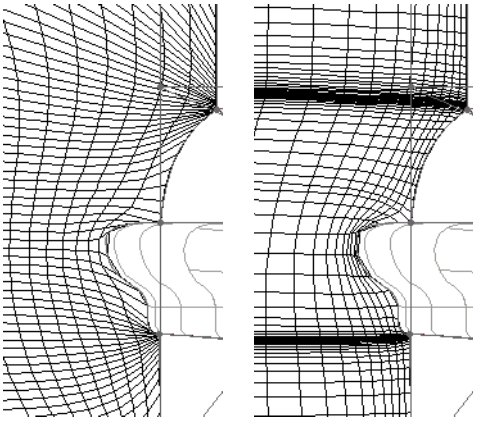


12.2.4.1 LaPlace

LaPlace background control functions provide a very smooth distribution of grid points in the grid interior, but provide no degree of orthogonality or clustering.

12.2.4.2 Thomas - Middlecoff

Thomas - Middlecoff (Reference 34) background control functions will cluster grid points on the grid's interior based on how the grid points are clustered on the boundaries. This method is very reliable and stable for a wide range of applications and is, therefore, Gridgen's default background control function. Thomas-Middlecoff (right) is compared to LaPlace (left) below.



12.2.4.3 Fixed Grid

Fixed Grid background control functions are used to eliminate grid line slope discontinuities from the grid's interior while preserving the rest of the grid. Its effects are very subtle. The final grid will exhibit the same general features as the starting grid, including regions of clustering and orthogonality, but any slope discontinuities will be smoothed. This method should not be used unless the starting grid is of adequate quality except for the discontinuities.

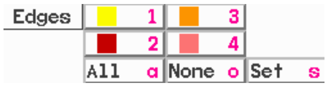
12.2.5 Foreground Control Functions

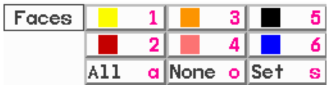
Foreground control functions are used to influence the distribution of grid points near the grid boundaries. They are calculated by adjusting the control functions locally so the grid points one layer inward from the boundaries will satisfy a specified spacing and angle constraint, with all other interior points blended smoothly. Spacing constraint refers to the distance between a boundary grid point and the grid point one layer inward, and angle refers to the angle that the grid line between these two points makes with the boundary itself.

The current foreground control functions for each boundary of the focus grid are displayed in the table in the center of the Blackboard. The left column lists how the spacing constraint is calculated, the center column lists how the angle constraint is calculated, and the right column describes the type of foreground control function. Within both the spacing and angle constraint columns, the left entry tells how the constraint is computed and the right entry tells how the constraint is blended into the grid's interior.

12.2.5.1 Selecting Boundaries

When you are setting foreground control function attributes with the Apply to **All** toggle button enabled, the attributes are applied to all boundaries of all grids selected for the solver. However, if the Apply to **Focus** toggle button is enabled, you must designate the boundaries to which you intend to set the attributes. This is done using either of the two menus below depending on whether you are running the solver on domains and setting attributes on edges (left) or blocks and setting attributes on faces (right).





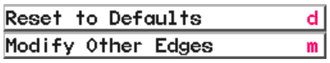
The buttons with the squares indicate the grid boundaries: 4 edges for domains and 6 faces for blocks. They are color coded to match the Blackboard listing and the highlights in the Display window. These buttons act like toggle buttons; when pressed they change state from enabled (filled square) to disabled (empty square). The state of these buttons is maintained even after you exit the solver.

All - Enable all boundaries.

None - Disable all boundaries.

Set - Proceed with attribute selection for the enabled boundaries.

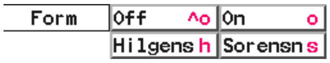
The foreground control function attributes menu has two other commands related to the boundaries.



Reset to Defaults - Change all foreground control function attributes on the boundaries to their default values.

Modify Other Edges - Return to the boundary selection menu.

12.2.5.2 Form



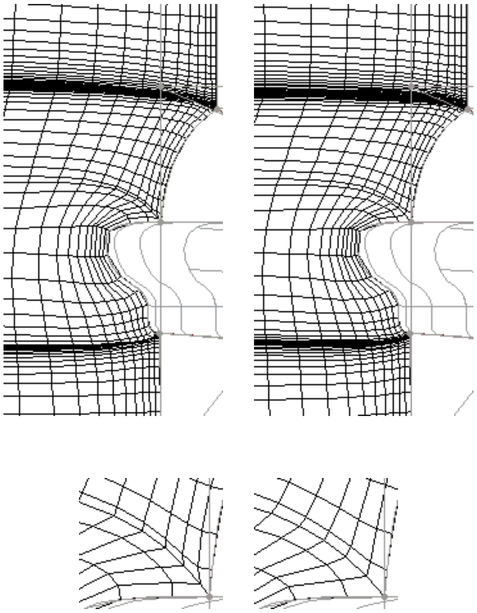
Off - Foreground controls are turned off on the selected boundaries.

On - Foreground controls with the specified attributes including form are applied to the selected boundaries. This is the default for all non-singular boundaries.

Sorensn - Apply the Steger-Sorenson (Reference 21) foreground control function formulation to the selected boundaries. The resulting grid is smooth, clustered, and orthogonal at the boundaries. However, the constraints are only maintained in an approximate sense, especially in concave regions.

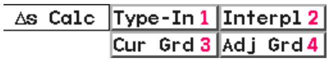
Hilgens - Apply the von Lavante-Hilgenstock-White (Reference 10, Reference 37, Reference 39) foreground control function formulation to the selected boundaries. This method yields virtually exact enforcement of the constraints but the grid may lack some of the smoothness provided by Steger-Sorenson. This method is the default.

von Lavante-Hilgenstock-White (left) is compared to Steger-Sorenson (right) below.



12.2.5.3 Spacing Constraints

The spacing constraint refers to the distance between the boundary (edge for domains and face for blocks) grid points and the first layer of grid points inward from the boundary. You can choose from four methods for computing the spacing constraint.



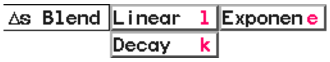
Type-In - This command sets the spacing constraint equal to the value you enter via the keyboard.

Interpl - The spacing constraint is computed by interpolating linearly from the perimeter of the boundary. This gives a smooth variation of spacing that matches the spacing at the perimeter of the boundary. The boundary’s arc length is used as the basis for the interpolation. This is the default method.

Cur Grd - This command maintains the spacing currently in place in the grid.

Adj Grd - The spacing constraint is copied from another grid selected for the elliptic solver sharing the selected boundary with the current grid. If the adjacent grid cannot be determined (either if none exists or if more than one exists), the constraint is changed to **Cur Grd**. This constraint provides spacing continuity across grid boundaries.

There are two methods for blending the spacing constraint portion of the control function from the boundary into the grid’s interior.



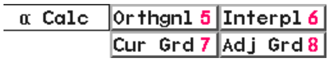
Linear - The spacing constraint is blended linearly from the boundary into the interior.

Exponen - An exponential function is used to rapidly decay the spacing constraint into the grid’s interior. This is the default blending function type.

Decay - This command allows you to specify the number of grid points away from the boundary at which the spacing constraint portion of the foreground control function will be equal to 10% of its boundary value. The default value is 6. Smaller values are more stable, but may dissipate the spacing constraint too quickly. Negative decay values are not allowed.

12.2.5.4 Angle Constraints

The angle constraint refers to the angle that transverse grid lines make with the boundary. You can choose from four methods for computing the angle constraint.



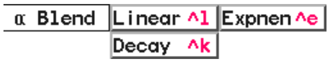
Orthgnl - This command sets the angle constraint to 90 degrees. This is the default method.

Interpl - The angle constraint is computed by interpolating linearly from the perimeter of the boundary onto the interior. A simple linear basis is used for the interpolation.

Cur Grd - This command maintains the angle currently in place in the grid.

Adj Grd - The angle constraint is copied from another grid selected for the elliptic solver sharing the selected boundary with the current grid. If the adjacent grid cannot be determined (either if none exists or if more than one exists), the constraint is changed to **Cur Grd**. This constraint provides angle continuity across grid boundaries.

There are two methods for blending the angle constraint portion of the control function from the boundary into the grid’s interior.



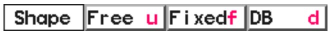
Linear - The angle constraint is blended linearly from the boundary into the interior.

Exponen - An exponential function is used to rapidly decay the angle constraint into the grid’s interior. This is the default blending function type.

Decay - This command allows you to specify the number of grid points away from the boundary at which the angle constraint portion of the foreground control function will be equal to 10% of its boundary value. The default value is 6. Smaller values are more stable, but may dissipate the angle constraint too quickly. Negative decay values are not allowed.

12.2.6 Surface Shape

Gridgen’s elliptic PDE solver is a true “surface” solver, in that grid points may be redistributed directly on a specified geometrical shape. This command is used to specify how the shape is maintained during elliptic solution. The following three techniques are available:



Surface shape attributes are only available for domains.

12.2.6.1 Free

The shape is computed implicitly as a result of the elliptic PDE solution for the Cartesian coordinates (`x,y,z`). If the grid is constrained to the database and you run the elliptic solver with the free shape attribute, the grid will no longer conform to the database. This is the default shape attribute. When this shape attribute is applied with LaPlace background control functions and no foreground control functions, the resulting surface shape will be of minimum area, similar to a "soap film" across a general boundary.

12.2.6.2 Fixed

The grid will maintain the shape it had when the **Fixed** button was pressed. The solution of the elliptic PDEs is performed in a parametric space corresponding to the grid's original (`u,v`) coordinates. If the grid is constrained to the database, this method will not destroy the grid - database relationship. This method is generally used when the existing grid points lie on the intended shape but are not distributed in the desired manner and no database is available.

12.2.6.3 DB

This method is used to have the grid conform to one or more database entities. For best results, the starting grid should already lie on the database entities. This can be accomplished in two different ways:

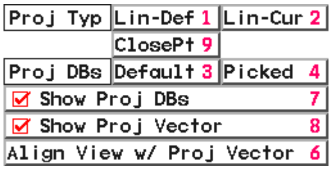
1. Parametric TFI (Section 12.1.5) or Parametric Fit TFI (Section 12.1.6).
2. Projection, either via the Modify menu (Section 14.4) or via the Run Structured Solver menu (Section 12.2).

Assuming the grid points have been initially constrained to the database, Gridgen attempts to make grid points conform to the database using either of two methods (Gridgen chooses automatically on a point by point basis).

1. If a grid point is surrounded by grid points that all lie on the same database entity, the grid generation equations are solved in the parametric (`u,v`) coordinates of the database entity.
2. Otherwise, the elliptic PDEs are solved using special terms accounting for surface curvature. This method does not keep the point exactly on the DB entity, but will be fairly close, particularly if grid point movement is minimal. PDE solution is followed by projection of the grid point onto the database, ensuring that it conforms exactly to the desired shape.

Though two different equation sets are used to move grid points, the equations are equivalent through a transformation of coordinate variables. Therefore, there is no improvement in grid quality of one method over the other. The parametric method is considerably faster to apply, but can be used only in special circumstances.

Several attributes can be set to control how Gridgen manages database shape control as shown in the figure below and described in the following sections.



Proj Typ **Lin-Def** - This attribute tells Gridgen that the projection type is linear and the default projection orientation will be used. The default projection orientation is computed by averaging the grid's normal vector at each edge grid point. The projection vector direction is shown in the Display window as an arrow at each corner of the grid.

Proj Typ **Lin-Cur** - This attribute tells Gridgen that the projection type is linear. The projection is along a fixed vector perpendicular to the Display window (screen-z) when the button is pressed.

Proj Typ **ClosePt** - This attribute tells Gridgen that the projection type is closest point and a projection vector is not required.

Proj DBs **Default** - The grids will be projected onto the default set of database entities. The default database entities are computed for each grid based on which entities are currently referenced by the grid points. The entities to be projected onto are highlighted in the Display window.

Proj DBs **Picked** - The grids will be projected onto database entities selected by you. After this button is pressed you can select database entities from either the Browser or Display window. The current set of projection entities is already picked when the Browser is first invoked.

Show Proj DBs - This toggle button controls whether or not the database entities being projected onto will be highlighted in the Display window. Turning off this toggle will un-clutter the display.

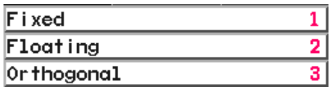
Show Proj Vector - This toggle button controls whether or not the projection vector with which grid points are projected onto the database entities is rendered in the Display window. Turning off this toggle will un-clutter the display.

Align View w/ Proj Vector - This command changes the orientation of the Display window image so that the focus grid's projection vector is parallel to screen-z.

12.2.7 Boundary Conditions

By default, boundary grid points are not moved by the elliptic solver. There are instances, however, when moving the boundary points subject to the PDE solution may be advantageous. Gridgen's three types of boundary conditions are described below. Note that these boundary conditions are not related in any way whatsoever to the analysis software boundary conditions (Section 17.2).

Before selecting the boundary condition, you must first select the boundaries to which it applies as described in Section 2.4.



Fixed - Boundary points are not affected by the elliptic PDE solution. This is the default boundary condition.

Floating - Boundary points will "float" as though they were interior grid points. Floating boundary conditions are only applicable if Gridgen is able to find a unique neighbor to the boundary among the grids selected for the elliptic solver. If a unique neighbor cannot be found for a particular grid point, then the boundary condition is converted to fixed for that grid point. Although the grid points on floating boundaries are computed as though they are interior grid points, the control functions are computed in each grid individually. This may result in control function discontinuities across the boundary leading to unexpected results.

Orthogonal - Boundary points will move along the boundary shape in order to maintain orthogonality with the interior points.

12.3 Structured SubGrids

For structured surface and volume grids, you can define a subset of the grid known as a subgrid for the application of the grid methods. You may consider a subgrid to be a template that isolates a subregion of the overall grid that allows you to precisely control where the grid methods are to be applied. In fact, only the structured grid solver commands can utilize subgrids. You would use a subgrid to apply a grid method only to the portion of a grid that needed improvement rather than the entire grid. There are three aspects to working with subgrids: adding them, deleting them, and using them.

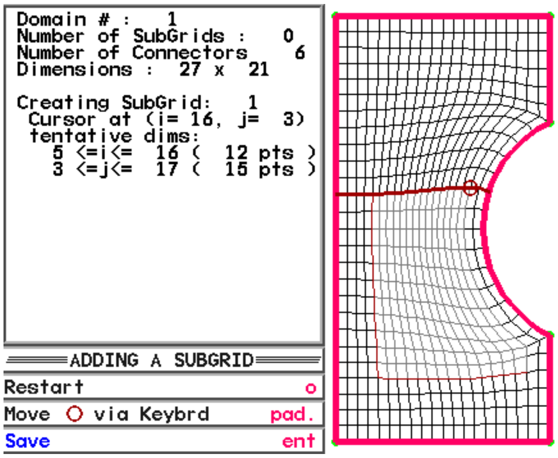
12.3.1 Adding a SubGrid

The SubGrids **Add** command is used to specify a subset of a structured domain's or block's grid points (called a subgrid) for later use with the algebraic (Section 12.1) and elliptic PDE (Section 12.2) grid methods. Subgrids are used only in conjunction with the algebraic and elliptic grid methods. The subgrid lets you apply grid methods to the precise regions where control is necessary without having to alter the grid's topology by splitting domains and/or blocks.

Each subgrid must have at least three grid points in each computational direction. SubGrids can overlap or abut one another, or they may be separate. The set of all subgrids on a given grid need not encompass all of the grid's grid points.

12.3.1.1 Adding a SubGrid to a Domain

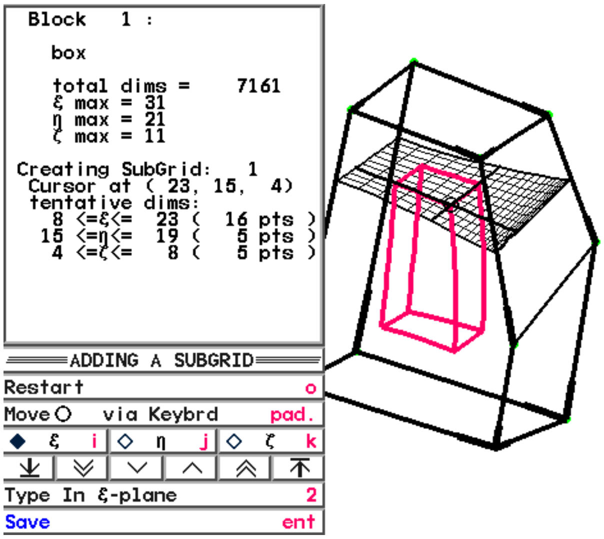
When the SubGrids **Add** button is pressed for creating a subgrid the highlighter appears on the domain's grid. The Blackboard shows the number of the subgrid being added and the index location of the highlighter. The highlighter is moved around the domain by moving the mouse with the ***right mouse*** button pressed. When the open circle at the intersection of the highlighter lines is positioned over one of the corners of the desired subgrid, press **Add Corner at ❍** button. Then move the highlighter to the grid point at the opposite corner of the subgrid. During this process the subgrid is rendered in the Display window with a yellow wireframe. The Blackboard shows the subgrid's dimensions. Save the subgrid by pressing the **Save** button. The saved subgrid is rendered in the Display window with a pink outline and white grid lines and Gridgen remains in subgrid mode awaiting the definition of the next subgrid. SubGrid creation is completed by pressing the **Done** button.



```
Domain # :    1
Number of SubGrids :    0
Number of Connectors    6
Dimensions :  27 x  21

Creating SubGrid:   1
 Cursor at (i= 16, j=  3)
 tentative dims:
    5 <=i<=  16 (  12 pts )
    3 <=j<=  17 (  15 pts )
```

12.3.1.2 Adding a SubGrid to a Block

When the SubGrids **Add** button is pressed for creating a subgrid, the highlighter appears in the block's grid. The Blackboard shows the number of the subgrid being added and the index location of the highlighter. The constant index plane being highlighted may be moved using the menu's VCR buttons, the ***left arrow*** and ***right arrow*** key, or by entering the plane index using the **Type In ξ-plane** button. The highlighter is moved around the constant index plane by moving the mouse with the ***right mouse*** button pressed. When the open circle at the intersection of the highlighter lines is positioned over one of the corners of the desired subgrid press **Add Corner at ❍** button. Then move the highlighter to the grid point at the opposite corner of the subgrid. During this process the subgrid's outline is rendered in the Display window in pink. The Blackboard shows the subgrid's dimensions. Save the subgrid by pressing the **Save** button. The saved subgrid is rendered in the Display window with a pink outline and Gridgen remains in subgrid mode awaiting the definition of the next subgrid. SubGrid creation is completed by pressing the **Done** button.



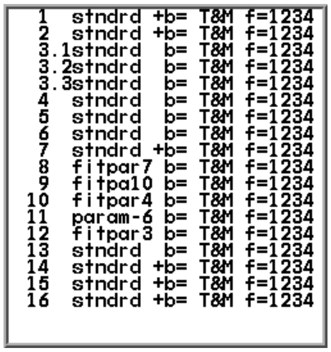
```
 Block   1 :

   box

   total dims =    7161
   ξ max = 31
   η max = 21
   ζ max = 11

Creating SubGrid:    1
 Cursor at ( 23, 15,  4)
 tentative dims:
   8 <=ξ<=  23 (  16 pts )
  15 <=η<=  19 (   5 pts )
   4 <=ζ<=   8 (   5 pts )
```

12.3.2 Deleting SubGrids

The SubGrids **Del** command is used to remove subgrids from a structured domain or block. Upon invoking this command the grid's subgrids are rendered with pink outlines. A subgrid is selected for deletion by moving the scissors cursor over the subgrid's pink outline and pressing the ***right mouse*** button. Keep in mind that deleting a subgrid does not delete any grid points.

12.3.3 Using SubGrids

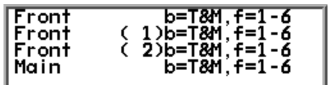
Subgrids are used with the structured grid solvers. When the solver is run, the Browser will list any subgrids as pickable entities in the Browser as shown below for domains.



```
  1   stndrd +b= T&M f=1234
  2   stndrd +b= T&M f=1234
  3.1stndrd  b= T&M f=1234
  3.2stndrd  b= T&M f=1234
  3.3stndrd  b= T&M f=1234
  4   stndrd  b= T&M f=1234
  5   stndrd  b= T&M f=1234
  6   stndrd  b= T&M f=1234
  7   stndrd +b= T&M f=1234
  8   fitpar7 b= T&M f=1234
  9   fitpa10 b= T&M f=1234
 10   fitpar4 b= T&M f=1234
 11   param-6 b= T&M f=1234
 12   fitpar3 b= T&M f=1234
 13   stndrd  b= T&M f=1234
 14   stndrd +b= T&M f=1234
 15   stndrd +b= T&M f=1234
 16   stndrd +b= T&M f=1234
```

As you can see from the illustration, whenever a domain has subgrids, each subgrid is listed in the Browser. The subgrid number is shown to the right of the decimal point after the domain number.

The Browser appears differently for blocks as shown below. The block name is listed first and any subgrids are listed in parenthesis next to the block name.

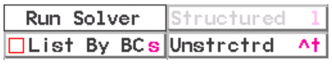


```
Front        b=T&M,f=1-6
Front     ( 1)b=T&M,f=1-6
Front     ( 2)b=T&M,f=1-6
Main         b=T&M,f=1-6
```

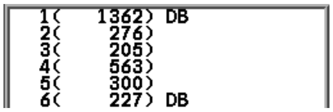
13. Grid Methods for Unstructured Domains and Blocks

Gridgen's unstructured grid methods are based on a Delaunay technique by which points are inserted into a triangular (domain) or tetrahedral (block) grid in order that the individual grid cells meet one or more criteria that you specify. The surface grid method is a "modified" Delaunay method since LaPlace smoothing is applied to the grid, whereas the volume grid method is "strict" Delaunay.

13.1 Running the Solver



The Run Solver **Unstrctrd** menu button is used to apply a triangulation method to one or more unstructured grids. The first step in running the unstructured solver is to select, from either the Browser or the Display window, the grids to be run. During selection of the grids a terse description of the current unstructured solver attributes (Section 13.2) will be displayed for each grid in the Browser.

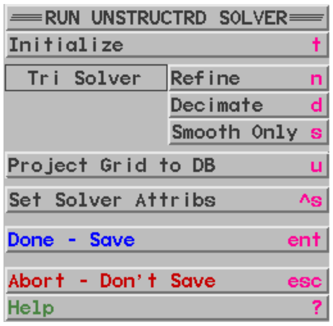


```
1(   1362) DB
2(    276)
3(    205)
4(    563)
5(    300)
6(    227) DB
```

For example, in the sample shown above, grid number 6 has 227 points and is constrained to the database (DB). The **List By BC** menu button instructs Gridgen to list domains in the browser by their boundary condition defined using the ANALYSIS S/W COMMANDS menu (Section 16.2).

If a grid's Browser text contains a plus sign (+), this indicates that the mesh has been run in the solver and is 'locked'. An asterisk (*) in the Browser text indicates a mesh locked by another action, such as copying and translation off the database. Also, running the tetrahedral solver on a block locks all domains used on the block's faces.

When you have selected any number of grids, the unstructured solver is invoked and immediately enters the "paused" state with the menu shown below for domains.



Initialize - Clear any existing interior grid points and create a new mesh by starting from the boundary points and applying the solver with the current attributes.

Tri Solver **Refine** - Insert points into an existing mesh with the current attributes.

Tri Solver **Decimate** - Remove points from an existing mesh with current attributes.

Tri Solver **Smooth Only** - Apply smoothing to an existing mesh (no point insertion).

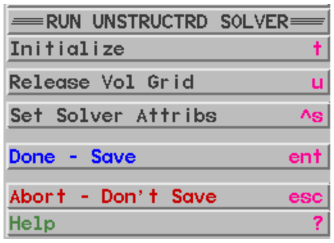
Project Grid to DB - Project the interior points of all selected grids with database constrained shapes using the current shape attributes.

Set Solver Attribs - Set unstructured solver attributes such as grid parameters and shape function on the grids currently being run in the solver.

Done - Save - Exit the solver and save the current grids.

Abort - Don’t Save - Exit the solver and revert the selected grids, including the solver attributes, back to the state that existed at the time the **Done** button was pressed.

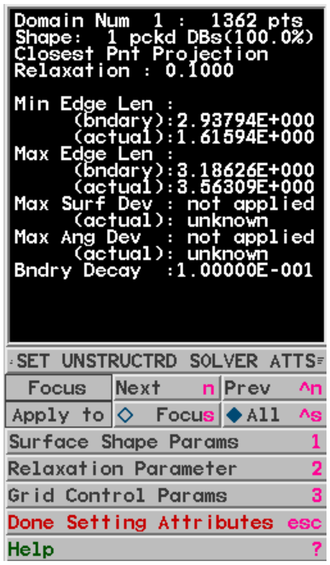
The unstructured solver’s Refine menu is slightly different for blocks as shown below. Each time the unstructured solver is applied to blocks by pressing the **Initialize** button, the grid generation procedure discards any current volume points and starts over from the boundary points and the current attributes. Also note that since the unstructured solver for blocks is a strict Delaunay method, the button for **Smooth Only** does not appear.



Release Vol Grid - This command is an explicit means of purging the unstructured volume grid from memory. This will minimize Gridgen’s memory usage and the size of the Gridgen (.gg) file at the cost of having to regenerate the volume mesh when it is needed. The equivalent command for structured blocks is to apply the algebraic solver.

13.2 Setting Attributes

When the **Set Solver Attribs** command is invoked the Browser window will display the current attributes of the focus grid as shown below. The various attributes may be set using the menu buttons.



```
Domain Num  1 :  1362 pts
Shape:  1 pckd DBs(100.0%)
Closest Pnt Projection
Relaxation : 0.1000

Min Edge Len :
     (bndary):2.93794E+000
     (actual):1.61594E+000
Max Edge Len :
     (bndary):3.18626E+000
     (actual):3.56309E+000
Max Surf Dev : not applied
     (actual): unknown
Max Ang Dev  : not applied
     (actual): unknown
Bndry Decay  :1.00000E-001
```

Focus **Next** - Moves focus to the next grid in numerical order of the grids selected for the solver. The focus grid is the grid whose attributes are listed in the Blackboard and whose image is highlighted in the Display window.

Focus **Prev** - Moves focus to the previous grid in numerical order of the grids selected for the solver. The focus grid is the grid whose attributes are listed in the Blackboard and whose image is highlighted in the Display window.

Apply to **Focus** - This radio button specifies that the selected attributes are to be applied to the focus grid only.

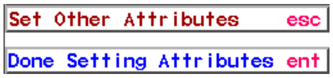
Apply to **All** - This radio button specifies the selected attributes are to be applied to all grids currently selected for the solver.

Surface Shape Params - These attributes define how the shape of the selected surface grids is to be maintained by the solver. These settings are not available for blocks.

Relaxation Parameter - This attribute controls the coefficient for the smoothing to be applied to the grid. Smoothing is not available for blocks.

Grid Control Params - These parameters allow you to control the criteria for point insertion into the grid.

Two other commands come into play when you are setting attributes. The following two buttons appear within each of the specific attributes menus.

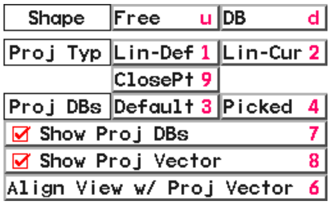


Set Other Attributes - Pressing this button will invoke the main unstructured attributes menu so that you may select a new attribute type to set.

Done Setting Attributes - Pressing this button will invoke the solver and enter “pause” mode. You may then apply the solver with the new attributes using the buttons on that menu.

13.3 Surface Shape Attributes

Gridgen's unstructured solver for domains is a true "surface" solver, in that grid points may be inserted directly on a specified geometrical shape (the database). This attribute is used to specify the manner in which that shape is maintained during solution. The following two techniques are available:



Surface shape attributes are only available for domains.

13.3.1 Free

The shape is computed implicitly as a result of the triangulation method for the Cartesian coordinates (`x,y,z`). If the grid is constrained to the database and you run the solver with the free shape attribute, then the grid will no longer conform to the database. Free is the default shape attribute unless Gridgen has automatically applied either Parametric or Parametric Fit Triangulation when the domain was created (Section 9.3). In this case the default shape attribute is database.

13.3.2 DB

This method is used to have the grid conform to one or more database entities. For best results, the starting grid should already lie on the database entities which can be accomplished in two different ways:

1. Parametric Triangulation or Parametric Fit Triangulation (Section 13)
2. Projection, either via the Modify menu (Section 14.4) or via the Run Solver menu (Section 13.3)

Proj Typ **Lin-Def** - This attribute tells Gridgen that the projection type is linear and the default projection orientation will be used. The default projection orientation is computed by averaging the grid's normal vector at each grid point. The projection vector direction is shown in the Display window as an arrow at each corner of the grid.

Proj Typ **Lin-Cur** - This attribute tells Gridgen that the projection type is linear and that points will be projected onto the database along a fixed vector perpendicular to the Display window (screen-z) when the button is pressed.

Proj Typ **ClosePt** - This attribute tells Gridgen that the projection type is closest point and a projection vector is not required.

Proj DBs **Default** - The grids will be projected onto the default set of database entities. The default database entities are computed for each grid based on which entities are currently referenced by the grid points. The entities to be projected onto are highlighted in the Display window.

Proj DBs **Picked** - The grids will be projected onto database entities selected by you. After this button is pressed you may select database entities from either the Browser or Display window. The current set of projection entities is already picked when the Browser is first invoked.

Show Proj DBs - This toggle button controls whether or not the database entities being projected onto will be highlighted in the Display window. Turning off this toggle will un-clutter the display.

Show Proj Vector - This toggle button controls whether or not the projection vector with which grid points are projected onto the database entities is rendered in the Display window. Turning off this toggle will un-clutter the display.

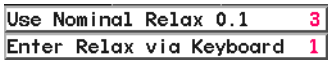
Align View w/ Proj Vector - This command changes the orientation of the Display window image so that the focus grid’s projection vector is parallel to screen-z.

13.4 Relaxation Parameter Attribute

The unstructured solver’s relaxation parameter is the coefficient to be applied to a smoothing sweep through the grid points which is performed after all grid point insertion is completed. Relaxation is only available for domains. The smoothing algorithm works as shown below

$\mathbf{r}_{\text{smooth}} = (1 - \omega)\ \mathbf{r}_{\text{original}} + \omega\ \mathbf{r}_{\text{average}}$

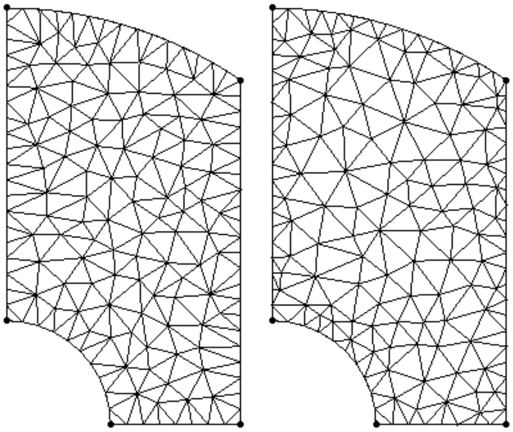
where $\mathbf{r}_{\text{average}}$ is the average value of all of the grid point’s neighbors. The value of the relaxation parameter ω is limited to values between 0 (no smoothing) and 2.0 since values greater that 2.0 can introduce numerical instability.



Use Nominal Relax - This button sets the value of the relaxation factor to Gridgen’s default value which has been deemed “nominal”.

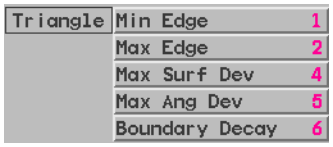
Enter Relax via Keyboard - This button allows you to type-in any value for the relaxation parameter in the range $0 < \omega < 0.5$.

The figure below illustrates the difference between an un-smoothed (ω=0) grid on the left and a “fully” smoothed grid (ω=0.5) on the right.



13.5 Triangular Mesh Attributes

The unstructured solver’s grid control parameters allow you to establish the criteria by which the solver decides whether a point needs to be inserted into the grid. As long a cell is larger than the minimum cell size, a point will be inserted into it if any of the other grid control parameter values are exceeded.



13.5.1 Minimum Triangle Edge

The Triangle **Min Edge** attribute allows you to specify a lower limit on triangle edges. If a cell's edge is smaller than the specified minimum edge, a point will not be inserted into the cell.

This attribute is used to compute triangle areas where each triangle is assumed to an equilateral triangle that has a spacing equal to that of the minimum edge size on each of its three sides.

The minimum edge parameter is used primarily as a safeguard, so combinations of the other parameters will not allow the grid to be refined to infinitesimal sizes.

13.5.2 Maximum Triangle Edge

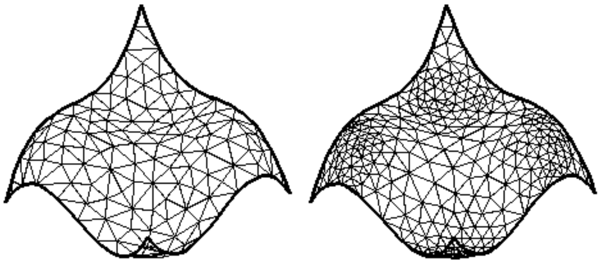
The Triangle **Max Edge** attribute allows you to set largest triangle edge that will be allowed in the grid. If a cell's edge is greater than the specified maximum edge, a point will be inserted into the cell.

This attribute is used to compute triangle areas where each triangle is assumed to an equilateral triangle that has a spacing equal to that of the maximum edge size on each of its three sides.

13.5.3 Maximum Surface Deviation

A point will be inserted into a cell if the estimated maximum distance between the cell's centroid and the underlying database surface exceeds the specified maximum surface deviation. By default, the maximum surface deviation is not applied. When specifying the maximum surface deviation, entering a value of `0` disables this attribute. The maximum surface deviation parameter is not available for blocks and is only applied to domain grids that are constrained to database surfaces.

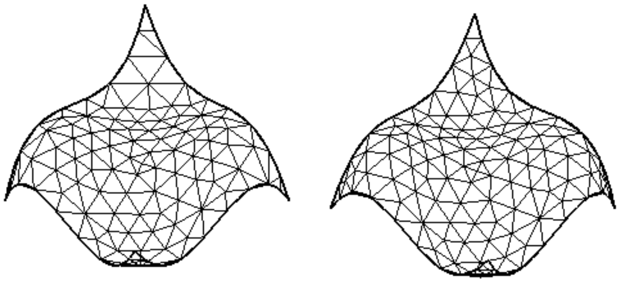
In the figure below, two surface grids have been generated on a database surface with a highly curved shape. In the grid on the left, no surface deviation value has been specified. In the grid on the right, a surface deviation was used to ensure that highly curved regions of the grid are resolved.



13.5.4 Maximum Angular Deviation

A point will be inserted into a cell if the estimated maximum angle between the cell's centroid and the vertex surface normals lying on the database exceeds the specified maximum angular deviation angle. When specifying the maximum angular deviation, entering a value of `0` disables this attribute. The maximum angular deviation parameter is not available for blocks and is only applied to domain grids that are constrained to database surfaces.

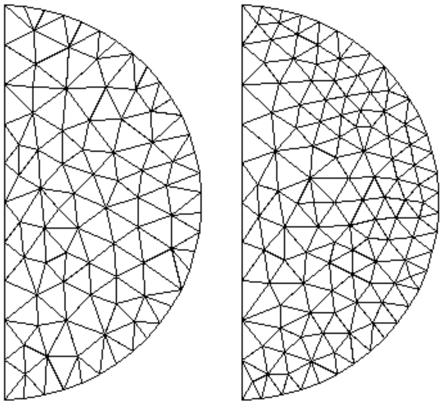
In the figure below, two surface grids have been generated on a database surface with a highly curved shape. In the grid on the left, no angular deviation value has been specified. In the grid on the right, an angular deviation was used to ensure that highly curved regions of the grid are resolved.



13.5.5 Boundary Decay Factor

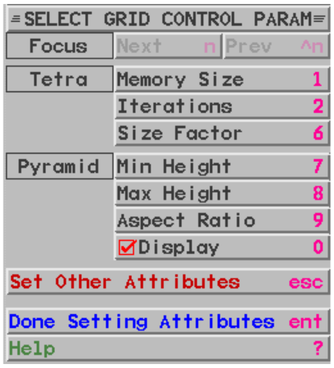
The boundary decay factor qualitatively controls how far into the grid's interior the boundary cell size effects the interior cell size. A decay factor of `0` turns off this attribute; interior cell size is not affected by boundary cell size. A decay factor of `1` applies full boundary influence on the grid's interior. Therefore, valid values for the boundary decay factor fall in the range [0,1].

The figure below illustrates a domain grid with boundary spacings on the circular arc that are smaller than those on the vertical line. The grid on the left has the boundary decay factor turned-off (`0`) while the grid on the right has the boundary decay factor turned-on fully (`1`). Note how the boundary decay factor allows the small cell sizes from the circular arc to influence the size of the interior triangles.



13.6 Tetrahedral Mesh Attributes

The unstructured solver's grid control parameters allow you to establish the criteria by which the solver decides whether a point needs to be inserted into the grid. As long a cell is larger than the minimum cell size, a point will be inserted into it if any of the other grid control parameter values are exceeded.



13.6.1 Tetrahedral Solver Memory Limit

The Tetra **Memory Size** attribute is the maximum amount of memory to be allocated by the tetrahedral solver. The default value is 64 MBytes and may be set to any value greater than or equal to 10 MBytes that the hardware platform can allow. Please note it is possible for a tetrahedral mesh to be returned (although not necessarily a good one) even if the memory limit is exceeded. A rough estimate of memory usage is 1 MByte for every 15,000 tetrahedra.

13.6.2 Tetrahedral Iterations

The Tetra **Iterations** attribute allows you to run the tetrahedral solver when a single run produces a bad mesh and additional iterations can be performed to improve the quality of the grid in inadequate regions. The number of iterations always starts from an initialization, for example iterations = 2 followed by an application of iterations = 4 will result in a total of 6 runs of the tetrahedral solver. Default =1, maximum = 10.

13.6.3 Tetrahedral Size Factor

The Tetra **Size Factor** attribute controls the relative size of the tets on the volume interior. Enter a factor (between 1 and 0); 0 = small, 1 = large. Default = 0.5.

13.7 Pyramid Mesh Attributes

Pyramid cells are created by Gridgen adjacent to any quadrilateral cell on the faces of an unstructured block. Obviously, they result from the use of structured domains during unstructured block face creation (Section 10.4).

The pyramid generation process is as follows:

1. The area of the quadrilateral cell is computed via triangular decomposition.
2. The normal vector to the quadrilateral cell is computed by averaging the cross products at the four corners.
3. The centroid of the quadrilateral cell is computed by averaging the location of the four corner points.
4. The nominal pyramid height is computed as the square root of the quadrilateral area.
5. The true pyramid height is obtained by multiplying the nominal pyramid height by the user specified aspect ratio.
6. The true pyramid height is then limited by the user specified minimum and maximum.

13.7.1 Minimum Pyramid Height

The Pyramid **Min Height** attribute specifies the minimum height allowed for any pyramid cell. By default this limiter is disabled.

13.7.2 Maximum Pyramid Height

The Pyramid **Max Height** attribute specifies the maximum height allowed for any pyramid cell. By default this limiter is disabled.

13.7.3 Pyramid Aspect Ratio

The Pyramid **Aspect Ratio** attribute specifies the aspect ratio that is used to scale the nominal pyramid height to obtain the true pyramid height. By default this value is 1.

13.7.4 Pyramid Display

Whether or not the Pyramid Display toggle button is enabled controls whether or not pyramid cells will be drawn in the Display window when running the unstructured solver. The pyramids are drawn using the same display attributes as the structured domain from which they are extruded.

13.8 Grid Point Insertion on Boundaries

13.8.1 Domains

Grid points are not inserted onto a domain’s boundary by the unstructured solver.

13.8.2 Blocks

Grid points are not inserted onto a block’s boundary by the unstructured solver.

14. Modifying Entities

14.1 Overview

Gridgen provides a variety of tools for modifying database and grid entities. The **Modify** command can be accessed from the Database, Connector, Domain, and Block menus. Using these tools you can change the shape of any entity that you create in Gridgen or import from an external source.

The modification tools work the same way no matter what type of entity you are dealing with; however, some modification commands do not apply to all entity types. The modification commands and the entities they work on are listed in the following table.

Modify Commands And The Entities On Which They Act

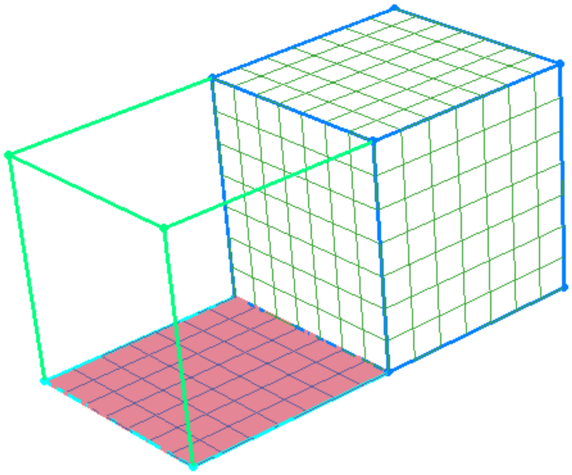
command	database	connector	str domain	unstr domain	str block	unstr block
translate	✓	✓	✓	✓	✓	✓
scale	✓	✓	✓	✓	✓	✓
stretch	✓	✓	✓	✓	✓	✓
rotate	✓	✓	✓	✓	✓	✓
re-extrude			✓		✓	✓
split	✓	✓	✓		✓	
join	✓	✓	✓	✓	✓	
mirror	✓	✓	✓	✓	✓	✓
project		✓	✓	✓		
smooth	✓					
add		✓				
insert		✓				
erase		✓				
edit	✓	✓				
least squares fit		✓				

Before discussing each modification command, the manner in which Gridgen maintains the relationship (called the linkage) between database entities, connectors, domains, and blocks during modification will be described.

14.2 Linkages

Gridgen maintains relationships between database entities and grid entities and between the different grid entities themselves. These relationships, called *linkages*, allow Gridgen to keep connectors and domains attached to the database entities on which they were created or projected and allow Gridgen to keep adjacent grid entities attached while you edit them. It is important that you understand linkages and their implications when you are modifying the database or grid because you will be given the opportunity to unlink an entity from its neighbors.

Linkage effects for each member of Gridgen's data hierarchy are described in the following sections. The grid shown below will serve as the example used throughout the descriptions. There are two square database entities shown in pink (one cannot be seen because it's below the block on the right). The grid contains one structured block (shown on the right with the dark blue connectors), one structured domain (on the database entity in the lower left), and five free connectors (light green).



14.2.1 Database Linkages

Database entities are topologically independent of other database entities. Because of this you do not need to consider linkages to other database entities when modifying the database. One minor exception to this rule is the Group entity (Section 15.2); any modifications to the Group will affect all the members of the group.

The database is topologically linked to connectors and domains. Connectors (their control points and grid points) become linked to a database entity by:

1. creating them using...
   a. a database constrained segment type (Section 7.2.2)
   b. the two point connector (Section 7.3)
   c. **On DB Ents** (Section 7.4) command
2. projection (Section 14.4)

Domains (grid points) become linked to a database entity either by:

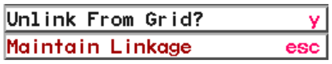
1. assembling them using connectors that are database constrained (Section 9.2 and Section 9.3)

2. creating them using the **On DB Ents** (Section 9.6) command
3. projection, either via the Modify menu (Section 14.4), one of the structured solvers (Section 12.1.5, Section 12.1.6, Section 12.2.6), the unstructured solver (Section 13.3), or one of the extrusion methods (Section 8.2.2).

For purposes of demonstration, assume that in the sample grid shown above we are modifying the database by translating the square database surface (pink, left) to the left. Gridgen detects the following linkages between that database entity and the grid:

- the domain lies entirely on the modified database entity
- the vertical connectors emanating from the domain are attached to the square database entity at their lower nodes
- one edge of the block is shared with the domain and, therefore, is on the database entity

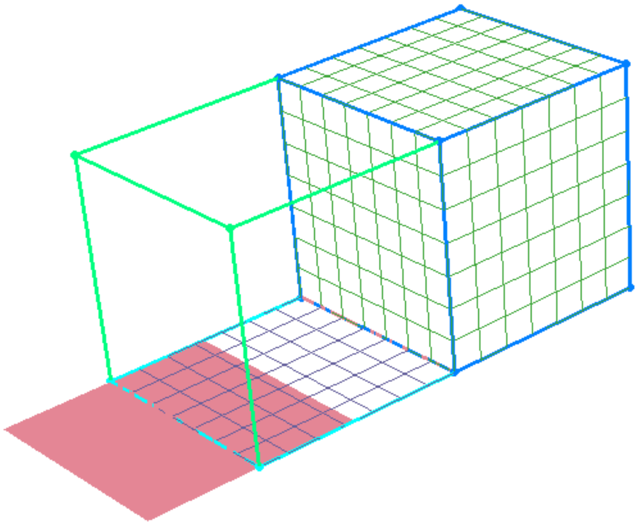
Because of the linkages, Gridgen displays the following menu when the database surface is modified.



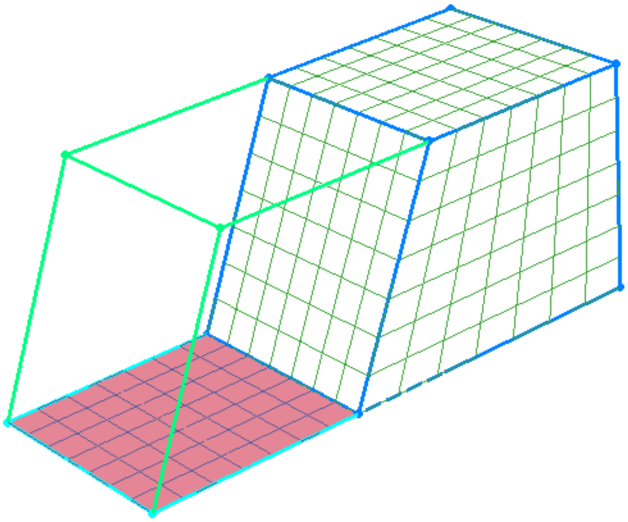
If you choose **Unlink from Grid?** Gridgen does the following.

- Converts each segment in the affected connectors into a general segment (Section 7.2.5).
- Removes the database link from each connector grid point. The point's (`x,y,z`) coordinates do not change but its reference to the database surface's (`u,v`) coordinates is eliminated.
- Removes the database link from each domain grid point. The point's (`x,y,z`) coordinates do not change but its reference to the database surface's (`u,v`) coordinates is eliminated.

With the linkages severed Gridgen modifies the database surface (in this case by translation) without changing the shape of the grid as shown below.

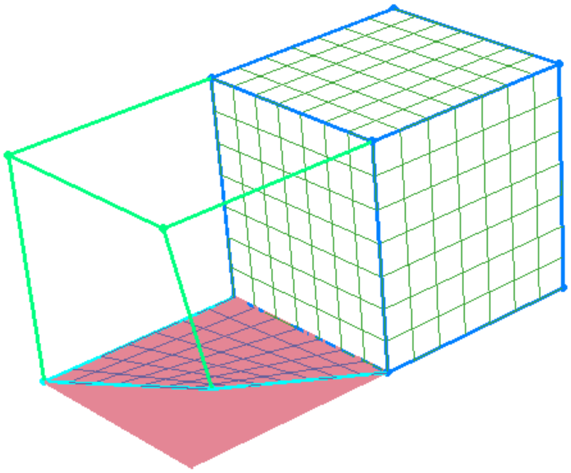


On the other hand, you can press the **Maintain Linkage** button and the grid will stay linked to the database surface and update its shape after the database modification (in this case translation) as shown below.



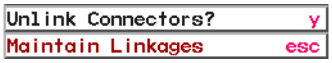
14.2.2 Connector Linkages

You can edit (Section 14.9.4) a database constrained connector and its linkage to the database will be maintained automatically. For example, the figure below illustrates the results of editing one connector belonging to the left most domain: its linkage to the database is maintained automatically. Splitting (Section 14.7) and joining (Section 14.8) also automatically maintain the connector's linkage to the database.

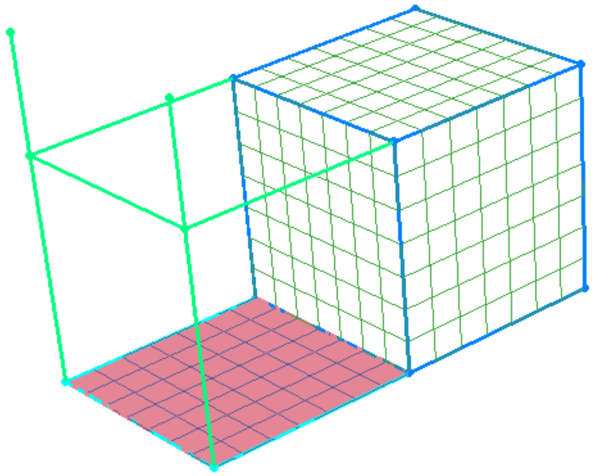


On the other hand, the transformation commands (Section 14.3) will break the linkage between a connector and the database and allow you to move the connector away from the database. After the transformation is complete, the connector is converted to a general type segment (Section 7.2.5.4). A general type segment is a fit of the discretized shape of the connector. Usually it has many more control points than the segment types, which can make it cumbersome to edit. Gridgen will not warn that you are moving a connector off the database, so exercise extra caution when modifying connectors that are on databases.

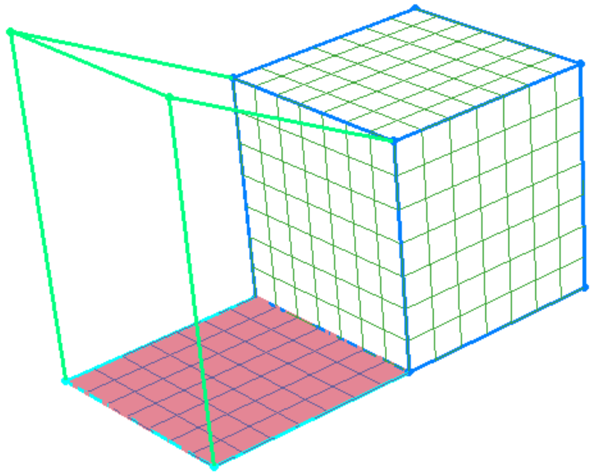
Connectors are linked to other connectors at their nodes. It is possible to break this linkage during modification. Referring to the example grid, suppose that the two vertical connectors on the left were scaled (Section 14.3.2). Since these connectors are linked to other connectors you must choose from the following menu.



If you choose **Unlink Connectors?**, only the selected connectors will be modified (in this case by scaling). As you can see in the figure below, only the two vertical connectors were changed and new nodes were created at their end points. The three horizontal free connectors did not move.

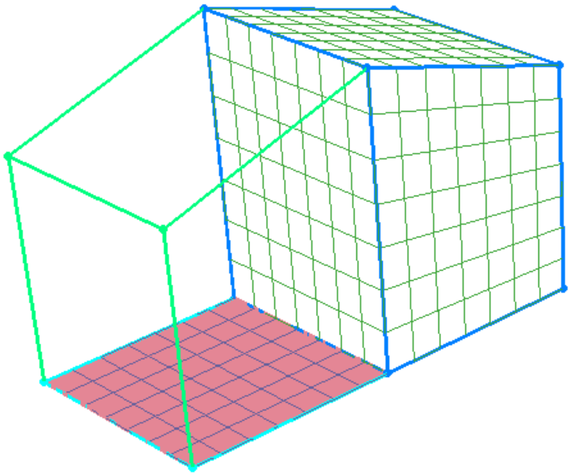


On the other hand, by choosing **Maintain Linkages** you force Gridgen to update all connectors that are linked by their nodes to the modified connectors, resulting in the grid shown below.



A connector is linked to a domain if it is used on the domain's edges. A connector is linked to a block if it is linked to one of the domains that define the block. Fortunately, it is impossible to unlink a connector from a domain or block, even by pressing the **Unlink Connectors?** button. This prevents errors that might potentially delete many higher level elements.

The figure below shows the effect of modifying (by translating upward) a connector that is linked to a domain and a block (the horizontal connector at the upper left of the block). In this case, no matter what button you press the grid entities will remain linked.



Connectors are also linked to domains and blocks by their grid points. Changes in a connector's grid points, whether through a connector shape change or a distribution change (Section 11.3) are propagated automatically to the domains and from the domains to the blocks. The effects on the grids are as follows.

1. For structured domains and blocks...
   a. If the grid has been "locked" (via application of the elliptic PDE solver, the extrusion solver, transformation or copying off a database surface, import from a grid points file) a small perturbation technique is applied to the grid blending the connector's change into the grid's interior. This technique attempts to preserve the current grid's qualities subject to the connector change. However, this technique works best if the changes to the connector are small. Large changes in shape or distribution may require re-application of one of the solvers.
   b. If the grid has only been initialized (via one of the algebraic methods) then that algebraic method will be re-applied.
2. For unstructured domains and blocks...
   a. The grid is re-initialized according to the saved attributes.

14.2.3 Domain Linkages

The same issues that affect connector linkages during modification affect domains. In fact, all of the linkages to other grid components are through the connectors that belong to the domain.

The only difference is the effect on the grid points interior to the domain. Transformations are applied to each individual grid point to obtain its new location. Stretching and scaling transformations will result in grid point distributions that cannot be replicated by any of the grid methods.

In addition, modifying a domain that is on a database can cause the interior grid points to lose their link with the database. If you later apply a grid method (or perform an operation like **ReDimension Grids** that causes them to be applied automatically), the grid could change drastically. This can usually be repaired quickly, but it is something to be aware of when modifying domains.

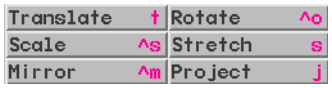
14.2.4 Block Linkages

As is the case for domains, blocks are linked to other blocks by domains that they share (interblock connections). Blocks are linked to the domains that are used on their faces. Blocks are linked to connectors that are used on their edges. The same issues that effect connector and domain linkages, therefore, also effect blocks.

The only difference is the effect on the grid points interior to the block. Transformations are applied to each individual grid point to obtain its new location. Stretching and scaling transformations may result in grid point distributions that cannot be replicated by any of the grid methods.

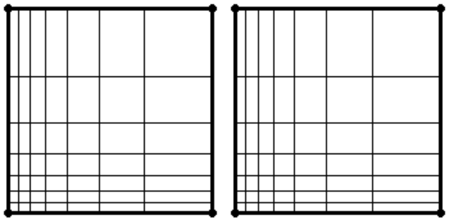
14.3 Transformations

The **Translate**, **Scale**, **Stretch**, **Rotate,** and **Mirror** commands are categorized as *transformation* commands. They may be applied to all entity types and are described in the following subsections.

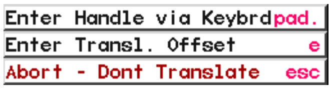


14.3.1 Translate

The **Translate** command lets you alter one or more entities by moving them to a new position in Cartesian space. In the figure below, the domain on the left has been copied and translated to the right.



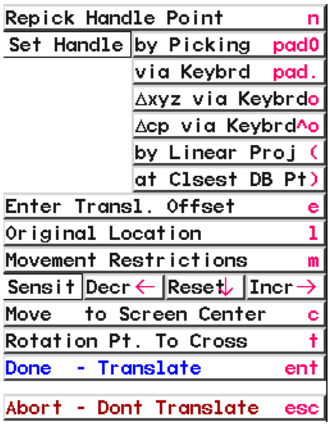
You can control the translation via two mechanisms described below.



The translation can be controlled using the 3D cursor. Immediately upon entering translation mode you may pick a point called the *handle* from the Display window. The handle is a point controlling the translation. The handle can be any pickable point or a point for which you type in the coordinates after pressing the **Enter Handle via Keybrd** button. Translation control is accomplished by moving the 3D cursor on the screen using the mouse with the ***right mouse*** button pressed. During translation the Display window shows the entities' current translated position. The Blackboard indicates the current cross position and translation offset. When the handle is repositioned the entity is translated along the vector defined from the initial to final handle locations.

The **Enter Transl Offset** command allows you to type in the amount of translation in each of the x, y, and z directions.

Other commands used during translation include:



Repick Handle Point - Lets you pick a new handle point if you picked the wrong one the first time.

Set Handle - The buttons following this label are the standard mechanisms for selecting the final location of the handle.

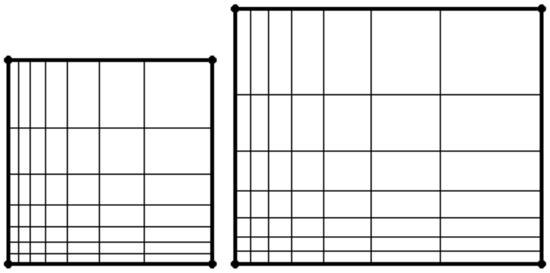
Original Location - Move the handle back to its position prior to movement.

The entity is not copied prior to translating. Use the **Copy** command first if that is what you want. In fact, after copying the entities, the **Copy** command will automatically invoke the Modify menu for the copies.

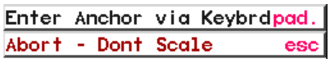
The translation is not saved until the **Done - Replace…** button is pressed.

14.3.2 Scale

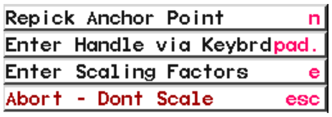
The **Scale** command is used to alter one or more entities by scaling. The scaling factors may be different for each of the three coordinate directions. The example below shows the results of copying and then scaling a domain relative to its lower left corner.



After the entities are selected, you must pick a point called the *anchor* as shown in the menu below. The anchor is a point that will remain fixed during scaling. You can pick the anchor from the Display window using the ***right mouse*** button or you may press the **Enter Anchor via Keybrd** button and type in the 3D coordinates of the anchor point.



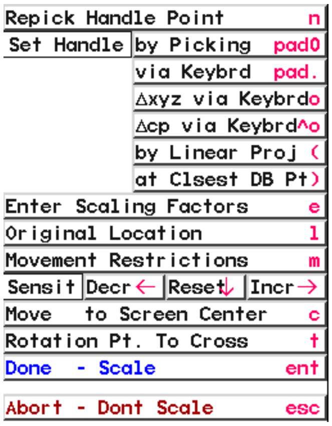
Once the anchor has been selected, two mechanisms may be used to control the scaling as described below.



The scaling can be controlled using the 3D cursor. After selecting the anchor you may also pick a point called the *handle* from the Display window. The handle is a point controlling the scaling. The handle can be any pickable point or a point for which you type in the coordinates after pressing the **Enter Handle via Keybrd** button. Scaling control is accomplished by moving the 3D cursor on the screen using the mouse with the ***right mouse*** button pressed. During scaling the Display window shows the entities' current translated position. The Blackboard indicates the current cross position and scaling factors. The scale factors then multiply the difference between the original point locations and the anchor point to find the new point locations.

The **Enter Scaling Factors** command allows you to type in the coefficients for each of the x, y, and z directions.

Other scaling commands are described below.



Repick Handle Point - Lets you pick a new handle point if you picked the wrong one the first time.

Set Handle - The buttons following this label are the standard mechanisms for selecting the final location of the handle.

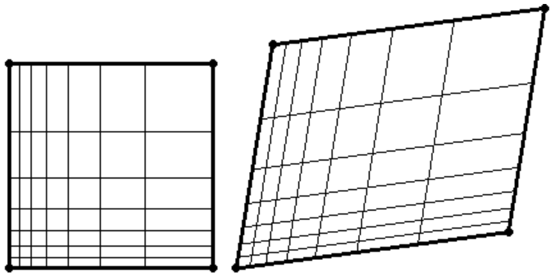
Original Location - Move the handle back to its position prior to movement.

The entity is not copied prior to scaling. Use the **Copy** command first if that is what you want. In fact, after copying the entities, the **Copy** command will invoke the Modify menu for the copies.

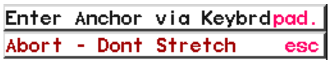
The scaling is not saved until the **Done - Replace…** button is pressed.

14.3.3 Stretch

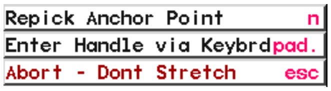
The **Stretch** command is used to alter entities by scaling along a defined vector. Multiple entities can be stretched at one time. The figure below illustrates stretching the domain with the lower left corner fixed and the upper right corner used as the handle.



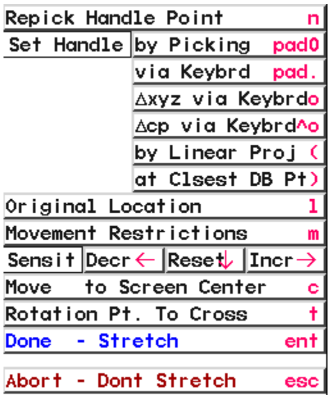
After the entities are selected, you must pick a point called the *anchor*. The anchor is a point that will remain fixed during stretching.



Next, you pick a point called the *handle*. The handle is a point controlling the stretching. Anchors and handles can be any pickable point or points for which you type in coordinates.



Stretching is controlled using the 3D cursor. Stretching control is accomplished by moving the 3D cursor on the screen using the mouse with the ***right mouse*** button pressed. During stretching the Display window shows the entities' current translated position. The Blackboard indicates the current cross position and scaling factors. The scale factors then multiply the difference between the original point locations and the anchor point to find the new point locations.

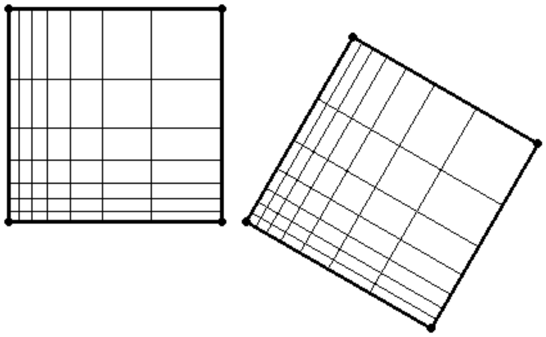


The entity is not copied prior to stretching. Use the **Copy** command first if that is what you want. In fact, after copying the entities, the **Copy** command will invoke the Modify menu for the copies.

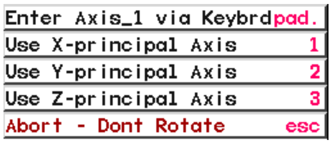
The stretching is not saved until the **Done - Replace…** button is pressed.

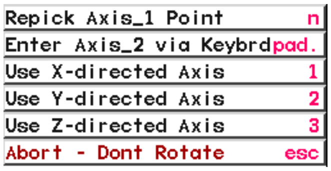
14.3.4 Rotate

The **Rotate** command is used to alter an entity by rotating it about a specified axis. One or more entities may be rotated at a time. In the figure below, the domain is rotated 30 degrees about an axis perpendicular to the page emanating from the lower left corner.

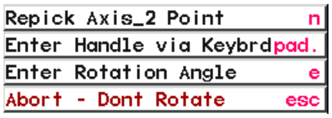


After the entities are selected you define the axis about which to rotate. You can do this by either specifying the two end points of the axis or by choosing to use one of the three principal axes.



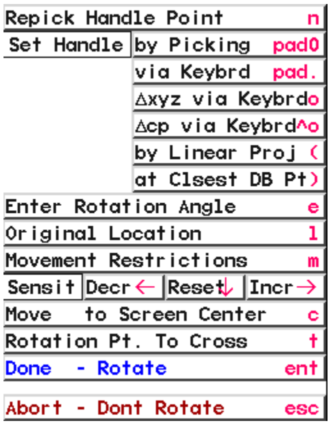


You can control the rotation by pressing the **Enter Rotation Angle** button and then typing in the rotation angle in degrees.



You may also control the rotation by picking a point called the *handle*. Candidate handles are any pickable points on the entity or a point you type in. You then move the cross on the screen using the mouse with the ***right mouse*** button pressed. During rotation the Display window shows the entities' current rotated position. The Blackboard indicates the current cross position and the rotation angle. The entity is rotated by the angle defined by the final and initial handle locations relative to the axis.

Other commands of interest used during rotation include:



Enter Rotation Angle - Enter the numerical value of the rotation in degrees following a right-handed convention around the axis, where the axis is oriented from the first to second axis points. The valid range for rotation angles is -180 < α < 180. The default value for the rotation angle is determined by the current location of the handle. (Once the rotation angle is input, the entities are rotated and the rotation process is complete.)

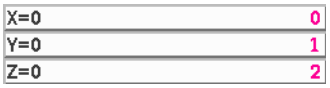
The entity is not copied prior to rotating. Use the **Copy** command first if that is what you want. In fact, after copying the entities, the **Copy** command will invoke the Modify menu for the copies.

The rotation is not saved until the **Done - Replace…** button is pressed.

14.3.5 Mirror

The **Mirror** command is a special version of the scale command to reflect database and grid entities about a plane. For structured grid block entities it also respecifies the mirrored block's computational coordinates to maintain its left or right handedness.

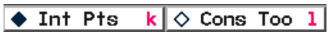
After selecting mirror, you must choose a mirror (symmetry) plane. You can either pick an existing database plane entity from the Display window or use one of the buttons below to select a x, y, or z constant plane through the origin.



14.4 Project

The **Project** command is used to modify a grid's shape by moving its grid points onto the database. Each database entity onto which grid points may be projected is highlighted in salmon when this command is invoked. The candidate database entities for projection can be changed via the **En/Disable** command in the Database menu. Only enabled surface entities can be projected onto. Turning off the entities that the grid points should not be projected onto has two effects. First, it prevents the projection algorithm from placing the grid points on the wrong surface. Second, it reduces the number of surfaces the projection algorithm has to go through and, therefore, the process will run a little faster. It also reduces the visual clutter.

Prior to projecting, you must select whether or not grid points from the entity's boundaries will also be projected. The menu below is an example for projecting domains.

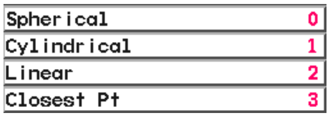


◆ Int Pts k ◇ Cons Too 1

Int Pts - Only the interior points of the domain or connector are projected. If, for example, your domain's connectors are already database constrained (e.g., Line on DB segment types) you should probably enable this radio button. Otherwise, your Line on DB segments will be replaced by General segments (Section 7.2.5.4).

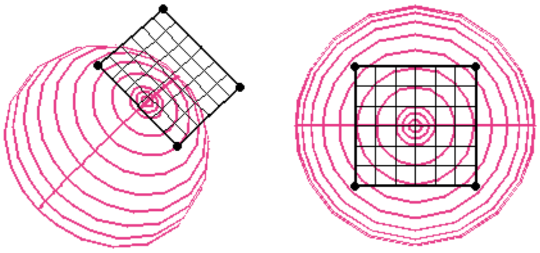
Cons Too, Nods Too - The boundaries of the domain or connector are also projected in addition to the interior points. This would be useful, for example, if you have imported a domain from some external source and want to project the entire entity onto an IGES database.

Four different projection methods are available, each with its own advantages as described in the following sections. The naming convention for the projection methods is based loosely on analogies with coordinate system types. Each method, however, is simply a ray-surface intersection calculation. The methods differ in how the rays are cast.



Spherical 0
Cylindrical 1
Linear 2
Closest Pt 3

In the following subsection, the projection type will be illustrated by showing the effect of projecting the square grid onto the spherical database as shown below.

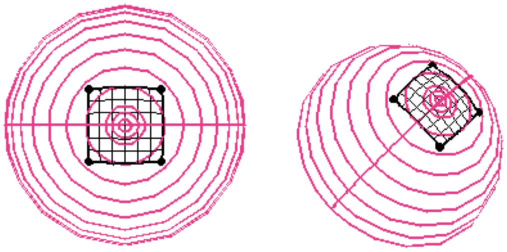


14.4.1 Spherical

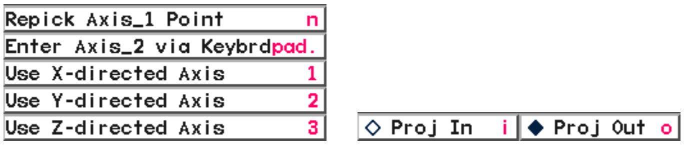
Enter Source via Keybrdpad.

Spherical projection is named for its analogy with a spherical coordinate system. This method projects grid points along rays relative to a user-defined point. The projection rays emanate from the point and pass through each grid point. After pressing the Spherical button, you define the projection point either by picking a point from the Display window or choosing the **Enter Source via Keybrd** button and typing the coordinates of a point. Once the point is defined, you can press the **Repick Source Point** button to change to a different point. Select the **Proj In** or **Proj Out** radio button depending on the relative position of the grid and database and whether you prefer to move the points inward or outward.

Spherical projections are useful for projecting grid points onto blunt geometries such as the nose of a missile. The results of spherically projecting our sample grid onto the database are shown below.



14.4.2 Cylindrical

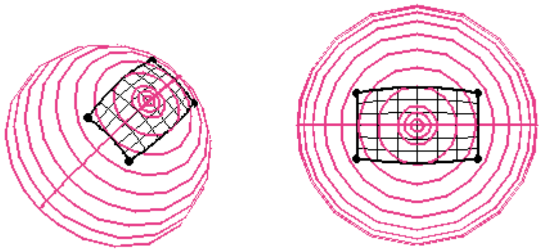


Repick Axis_1 Point n
Enter Axis_2 via Keybrd pad.
Use X-directed Axis 1
Use Y-directed Axis 2
Use Z-directed Axis 3

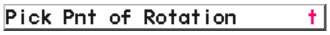
◇ Proj In i | ◆ Proj Out o

Cylindrical projection is named for its analogy with a cylindrical coordinate system. This method projects grid points along rays relative to a user-defined axis of projection. The rays emanate orthogonally from the axis and pass through each grid point. After pressing the **Cylindrical** button, define the projection axis by defining two points on the axis or selecting one of the menu buttons corresponding to the principal axes. Select the **Proj In** and **Proj Out** radio button depending on the relative position of the grid and database and whether you prefer to move the points inward or outward.

Cylindrical projections are useful for projecting grid points onto ducted geometries such as the inside of an aircraft engine intake duct. The results of cylindrically projecting our sample grid onto the database are shown below.



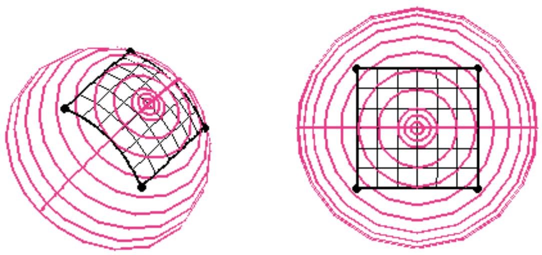
14.4.3 Linear



Pick Pnt of Rotation t

In the Linear projection grid points are projected along parallel rays. The ray cast direction is along the Display window’s line of sight, Gridgen’s screen-z coordinate direction. After pressing the **Linear** button and before pressing the **Done - Project** button, manipulate the image on the screen until you are visually satisfied that rays cast through the grid points and along the line of sight vector will intersect the database in the desired manner. The **Pick Pnt of Rotation** button may be used to change the image’s rotation point in order to more easily obtain the desired projection direction.

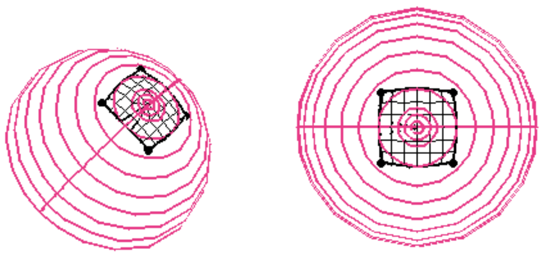
Linear projections are general purpose and can be used in most cases with easily predicted results. It is recommended that you use an orthonormal viewing transformation when orienting the image prior to projection. The results of linearly projecting our sample grid onto the database are shown below.



14.4.4 Closest Point

Closest point projection is orientation independent. Grid points are simply moved to the closest point on the database. This procedure is also known as Normal projection because the rays cast through each grid point will intersect the database orthogonally at the closest point. After choosing the **Closest Pt** button, press the **Done-Project** button.

This projection method is the easiest to apply since it requires the least amount of setup. However, it should be applied with care since it sometimes is not readily apparent beforehand what the closest database location is and bad grids sometimes result. The results of normally projecting our sample grid onto the database are shown below.



14.5 Smooth

The **Smooth** command, found in the Database Modify menu, smooths database curves through fitting. A smoothing tolerance is available so that the maximum deviation of the smoothed curve from the original can be prescribed by the user.

14.6 Re-extrude

The **Re-Extrude** command, available in the Domain and Block Modify menus, provides a way to re-extrude previously extruded prism and hexahedral domains and blocks. This command may be applied to single or multiple domains or blocks. Note that re-extrude is only available when all domains or blocks selected for modification have been previously extruded.

Pressing the **Re-Extrude** button places domains or blocks eligible for re-extrusion in the state that they existed in before being saved as domains or blocks. The full extrusion menu is available to extrude the domains or blocks forwards, backwards, or completely over (Section 8.1). The beginning extrusion attributes are inherited from the original domain or block.

If a domain selected for re-extrusion belongs to a block, the re-extruded domain has to contain the same number of extrusion steps as the original domain. The resulting re-extruded domain will not be saved if the number of extrusion steps differ.

Changes to the total number of extrusion steps for domains or blocks will cause those domains or blocks to be automatically unlinked from the grid when they are saved(see Section 14.2). No change to the total number of extrusion steps will cause a prompt to appear for the user to decide whether to maintain linkage or not.

Certain conditions will make selected domains or blocks ineligible for re-extrusion:

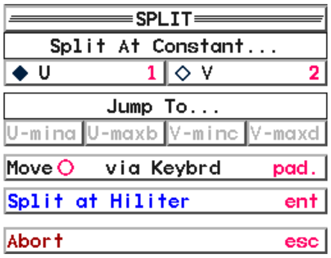
- A domain or block split since its original extrusion.
- A domain or block extruded along a connector path
- Multiple domains or blocks whose extrusion types are not identical.
- Multiple domains or blocks of different grid type (i.e. structured, unstructured).
- Multiple domains or blocks having an unmatched number of extrusion steps.

14.7 Split

The **Split** command is used to divide a database or grid entity into two separate entities.

14.7.1 Database Entities

Database curve and surface entities can be split at any parametric u or v location. Trimmed surfaces may not be split. The hiliter may be moved along the entity by moving the mouse with the ***right mouse*** button pressed. When the desired split location has been hilited, the entity is split by pressing the **Split at Hiliter** button.



Other commands of interest during database entity splitting include the following.

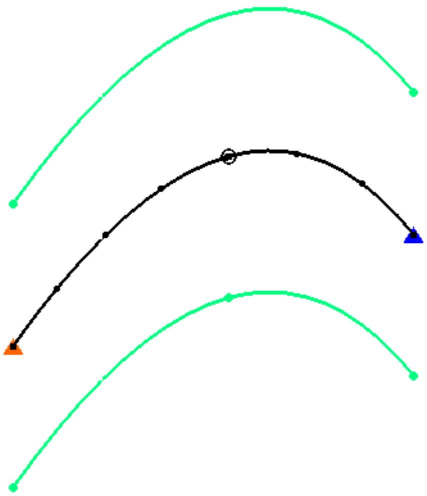
Split At Constant **U** - highlight u constant lines on a database surface for splitting.

Split At Constant **V** - highlight v constant lines on a database surface for splitting.

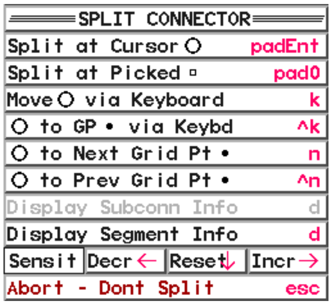
Move ❍ via Keybrd - Type-in the (u,v) coordinates to which the highlighter should be moved.

14.7.2 Connectors

A connector is split by moving the highlighter along its length by moving the mouse with the ***right mouse*** button pressed. When the desired split location has been highlighted, the connector is split by pressing the **Split at Cursor ❍** button. A new node will appear at the split location. The splitting process is illustrated in the following figure. From top to bottom the figure shows: the original connector, the connector during splitting showing the cursor over the split location, the final two connectors.



After the connector has been split once, the cursor is moved to the connector that is the “ending” half of the original connector, allowing you to continue splitting. The “ending” half is graphically denoted by the filled blue triangle. Terminate the splitting process by pressing the **Abort - Dont Split** button.



Other commands of interest during connector splitting allow you greater control over the split location.

Split at Picked □ - Pressing this button will split the connector at the control point beneath the cursor’s current location.

Move ❍ via Keyboard - You may type-in the x, y, or z, (or u and v for database constrained connectors) coordinate to which the cursor should be moved. In the case of a multi-valued connector, the cursor is moved to the typed-in location closest to the cursor’s original location.

❍ to GP • via Keybd - You may type-in the index of the connector grid point to which the cursor should be moved.

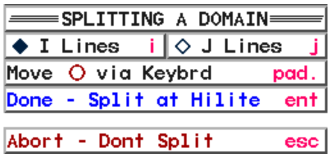
❍ to Next Grid Pt • - Move the cursor along the connector to the next grid point, in the direction from the connector’s beginning to ending.

❍ to Prev Grid Pt • - Move the cursor along the connector to the previous grid point, in the direction from the connector’s ending to beginning.

14.7.3 Domains

When the **Split** command is invoked for a structured domain the highlighter appears on the domain. Unstructured domains may not be split. The highlighter location is the split location. The highlighter can be moved along the range of possible split locations by moving the mouse with the ***right mouse*** button pressed.

The highlighter can be changed to other coordinate families by pressing the **I Lines** or **J Lines** buttons. When the proper split line is identified by the highlighter, press the **Split At Hiliter** button to perform the actual splitting. A necessary side effect of splitting a domain is that the two connectors on the domain's edges that are opposite the split direction will also be split.



SPLITTING A DOMAIN

◆ I Lines i | ◇ J Lines j

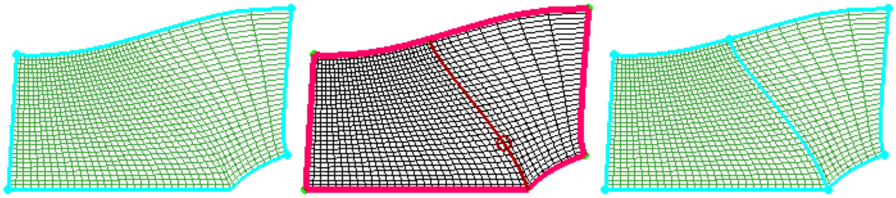
Move ○ via Keybrd pad.

Done - Split at Hilite ent

Abort - Dont Split esc

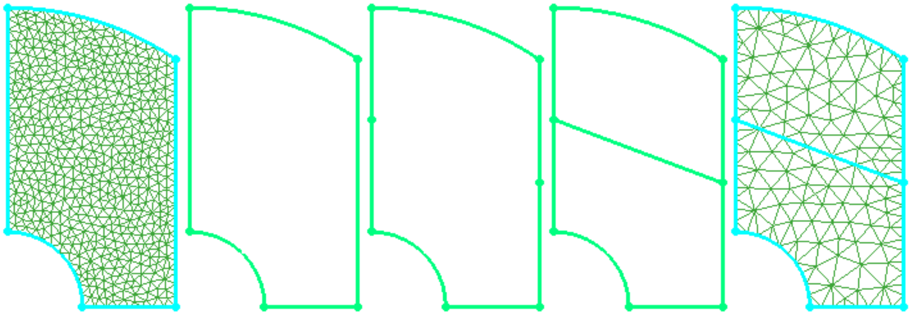
The **Move ❍ via Keybrd** button may be used to type-in the index of the grid line to which the hiliter should be moved.

The figures below illustrate the before, during, and after stages of a domain being split (from left to right, respectively).



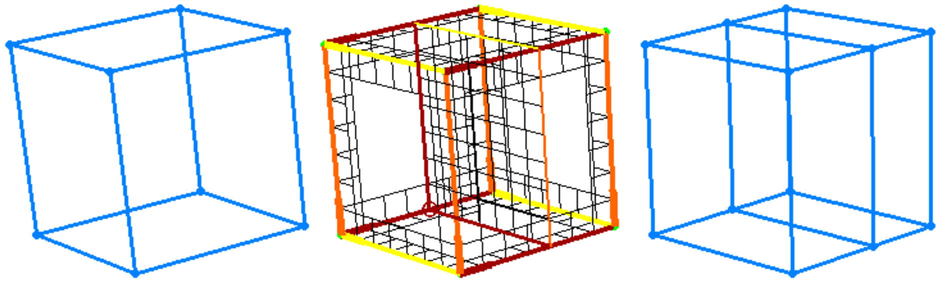
Only structured domains can be split. The equivalent functionality for an unstructured domain is illustrated in the figure below. From left to right:

1. Start with the original unstructured domain.
2. Delete the unstructured domain but not its connectors.
3. Split the connectors and create new nodes at either end of the split line (the two vertical connectors in the example).
4. Create a new connector for the split line (in this case the line slanting downward and to the right).
5. Create two new unstructured domains by assembling the edges.

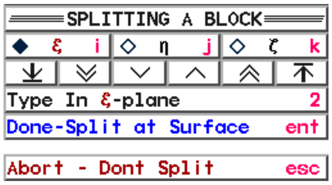


14.7.4 Blocks

When the **Split** command is invoked for a structured block the highlighter appears in the Display window. The hgihlighter location is the split location. The highlighter can be moved along the range of possible split locations by pressing the ***left arrow*** and ***right arrow*** keys to decrease and increase, respectively, the split location. The grid index to be split can be changed to other coordinate families by pressing the **I, J,** or **K** buttons. When the proper split line is identified by the highlighter, press the **Done - Split At Surface** button to perform the actual splitting. A necessary side effect of splitting a block is that the four domains on the block's boundaries that are opposite the split direction will also be split. The figures below show a simple structured block before, during, and after splitting (from left to right, respectively).



Other commands of interest during block splitting are shown below.



The VCR menu buttons may be used to move the highlighter and the **Type In I-plane** button may be used to type-in the index of the grid location to which the highlighter should be moved.

Only structured blocks may be split.

14.8 Join

The **Join** command is used to assemble individual entities into a larger entity. If the **Join** menu button is not available, exit to the Main menu for the entity type, select **Modify**, pick one of the entities to be joined, and select the **Join** menu button.

14.8.1 Database Entities

When database entities are joined you simply select the two entities to be joined. They must both either be curves or surfaces. In order for surfaces to be joined, they must share an entire edge, including knots and control points. In order for curves to be joined, they must share an end point. The polynomial degree, parametrization, and rationality of the entities must also be identical. If the entities cannot be joined you are warned. If the joining is successful, a new entity consisting of the two joined entities is created. The original entities remain intact.

14.8.2 Connectors

When joining connectors, Gridgen highlights in the Display window those connectors that are candidates for joining to the current connector. You simply pick the connector to join using the mouse. After each join operation Gridgen stays in joining mode so you may join several connectors at one time. Gridgen remains in joining mode until you press the **Abort** button or until there are no more candidate connectors left for joining.

It is not possible to join connectors that span multiple domains. It is also not possible to join two connectors if their common node is also shared by one or more other connectors. This is due to the fact that joining implicitly deletes nodes.

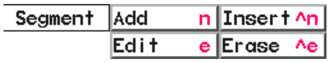
14.8.3 Domains

Domain joining proceeds in the same manner as block face creation (Section 10.5) in that many domains can be joined to create a new domain. Structured domains must be joined in a manner that results in a computationally rectangular region, whereas no such restrictions are place on joining unstructured domains. It is not possible to join domains that span multiple blocks, regardless of how they fit together. This is because the collection of joined domains is replaced by a single domain that would not fit into either block. If you attempt to join domains in this manner, a warning message is displayed. All connectors on the interior of the joined domain (previously on the perimeters of the constituent domains) are deleted. For structured domains, each constituent domain becomes a subdomain in the joined domain.

14.8.4 Blocks

Only blocks that share an entire common face may be joined. Unstructured blocks may not be joined. Once a block has been picked for joining, you simply pick the block to which it will be joined.

14.9 Connector Segment Modification



The commands Add, Insert, Erase and Edit allow you to modify the component segments of a connector and are described in the following subsections.

14.9.1 Add

The **Add** command is used to add a segment to the end of an existing connector. See Section 7.2 for information on adding segments to connectors.

14.9.2 Insert

Inserting a segment into a connector is very similar to adding a segment, except that the inserted segment is positioned in front of any existing segment instead of at the connector's end. When the **Insert** command is selected, you will be prompted to select the existing segment in front of which the new segment will be located. Operation then continues as if adding the segment. When the segment is completed, the start of the following segment is automatically moved to the end of the new segment to maintain continuity.

If the segment following the inserted segment is a database segment and hence is constrained to lie on a database entity, the end point of the inserted segment is moved automatically to the beginning of the database segment.

When inserting between two segments, the first control point of the inserted segment is automatically set to the end of the preceding segment and does not need to be redefined. The exceptions are the inserted database segments, which initialize the new segment's beginning point to the closest point on the entity to the previous segment and modify the last point of the previous segment so that continuity is maintained. Also, when inserting in front of the first segment, you must set the first control point.

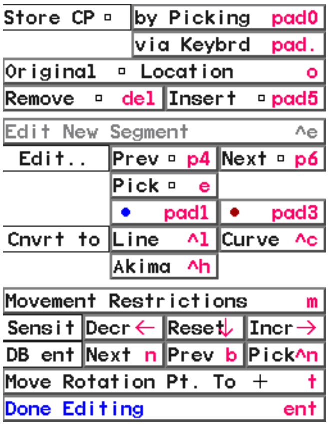
14.9.3 Erase

The **Erase** command is used to remove segments from a connector. Position the cursor (displayed as a pair of scissors) over the segment to erase and hit the ***right mouse*** button. Once a segment is erased, it cannot be restored, so caution is advised. If the segment after the erased one is not a database segment, its first control point will be moved to the end of the previous segment to maintain continuity. If the following segment is a database segment, the last point on the segment in front of the erased segment will be moved to the first point on the segment beyond the erased segment. If the segments on both sides of the erased segment are differing surface segments, they will be connected by a branch segment.

14.9.4 Edit

The **Edit** command is used when individual control points of a segment need to be re-positioned. If there is more than one segment in the definition, first pick the segment to edit. Then pick the control point to edit (via the mouse). Once a point is picked it may be moved via mouse control in the same manner as during the segment's creation. However, the edited point need not be saved explicitly, since the edit point is constantly replaced in permanent memory with the cursor location. Thus, when editing is exited, the edit point will be stored in the existing cursor location. If the point being edited is a segment end point adjacent to another segment, the adjacent segment will be modified when edit mode is exited or when another point is edited, whichever comes first.

Other commands of interest during segment editing include:



Store CP □ **by Picking**, Store CP □ **via Keybrd** - These commands allow you to save a control point in a new location by picking an existing point or by typing-in a value via the keyboard. A button for storing the control point at the current cursor location is not present because the control point is implicitly being stored at the cursor location as you edit the segment and move the cursor.

Original □ Location - The edit point will be returned to the location it occupied before it was selected for editing.

Remove □ - This command will remove the edit point from the segment definition. Naturally, this command will become available only when the resulting segment definition will remain viable without the deleted point. For example, it is not possible to remove any of the three non-sliding control points on a Circular Arc segment (since exactly three are needed to define the segment). Also, the removal of a control point on a surface segment could force an additional point to be removed if only one point remains on the current surface after the edit point is removed.

Insert □ - This command will add a control point equidistant from the current edit point and the control point preceding that point in the segment definition, and will make the new point the edit point. It will only be available when the current control point is not the first one in the segment. On Conic Section and 3 Pt Circle segments, this command creates a "slide" control point positioned initially between the two control points and is displayed by a small solid circle. Slide control points are moved along the segment via mouse control to any position between the two surrounding control points within 1% of the points themselves, but will not modify the Cartesian shape of the segment. The shapes of the conic sections are controlled only by the true control points, not by "slide" points. Slide points are primarily used to define points along the segment that will be used as break points in the distribution of grid points on the connector.

Edit New Segment - Pressing this button while editing one segment of a multi-segmented connector will allow you to pick another segment to edit.

Edit **Prev** □ - This menu button will move the edit point to the control point immediately preceding the current edit point in the current segment definition.

Edit **Next** □ - This menu button will move the edit point to the control point immediately following the current edit point in the current segment definition.

Edit **Pick** □ - This command will allow you to pick a control point on the current segment to be the new edit point.

Edit • - (Bezier segments only) Control is passed from the current control point to the left or right side slope point of the control point, depending on the color selected. Mouse movement will then control the selected slope point in the usual manner.

DB Ent **Next** - (database segments only) This command will move the current edit point to the next entity in the list of toggled entities. Under some circumstances this may cause the addition and/or deletion of control points surrounding the edit point, but only when such a modification is necessary to maintain a viable definition on a surface.

DB Ent **Prev** - (database segments only) This command will move the current edit point to the previous entity in the list of toggled entities. Under some circumstances this may cause the addition and/or deletion of control points surrounding the edit point, but only when such a modification is necessary to maintain a viable definition on a surface.

DB Ent **Pick** - (database segments only) This command will move the current edit point to the entity picked graphically by you. Under some instances the picked entity will cause control points surrounding the edit point to be added and/or deleted in order to maintain a viable definition on a surface.

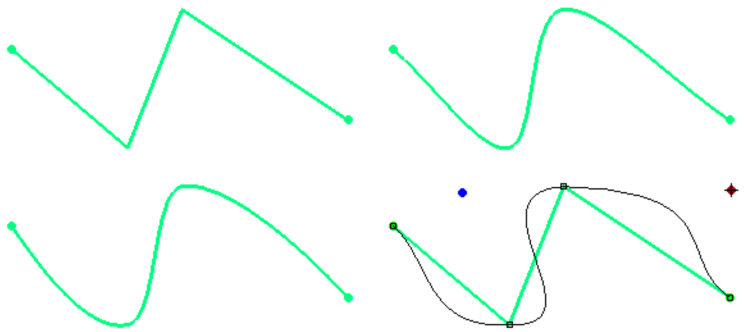
Return to Control Point - (Bezier segments only) Control is passed from the control point's slope point back to the control point itself.

Cnvrt to **Line** - (not for Circles or Conics) The current segment will be converted to a Line segment. If the segment is a Bezier type, you will be asked to confirm the request, since all previously set slope conditions on the segment will be overwritten. In the figure below, the upper left connector uses linear slopes.

Cnvrt to **Curve** - (not for Circles or Conics) The current segment will be converted to a Curve segment. If the segment is a Bezier type, you will be asked to confirm the request, since all previously set slope conditions on the segment will be overwritten. In the figure below, the upper right connector uses curve slopes.

Cnvrt to **Akima** - (not for Circles or Conics) The current segment will be converted to an Akima Curve segment. If the segment is a Bezier type, you will be asked to confirm the request, since all previously set slope conditions on the segment will be overwritten. In the figure below, the lower left connector uses Akima slopes.

Cnvrt to **Bezier** - (not for Circles or Conics) The current segment will be converted to a general Bezier segment. In so doing, you will be asked to decide whether or not to remove all slope constraints on the segment. A "yes" response will set the slope condition on all slope points on the segment to "free", so that further modification of the control point will maintain the same relative tangent vector. A negative response will maintain either all "linear" or "cubic" slope conditions on all slope points, depending on the segment type before conversion. This will allow editing of control points to proceed as in the strict Line or Curve case. In the figure below, the lower right connector has been converted to Bezier form and the slope points (blue and red filled circles) are being manually edited.

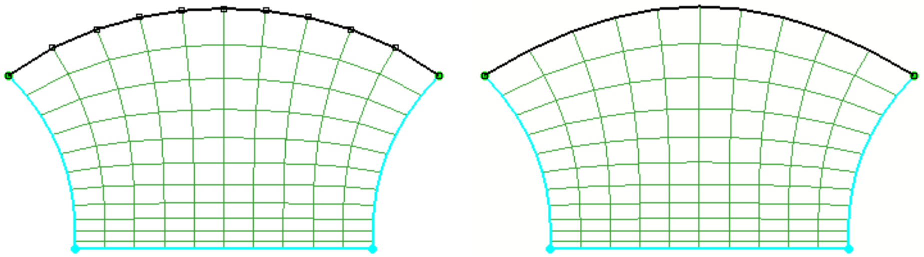


Done Editing - The segment editor will be exited.

14.10 Connector Least Squares Fit

A dimensioned connector can be modified by replacing its shape with a least squares fit of its grid points. This command is most useful for converting the connectors that are created by the extrusion methods (Section 8) into forms that are more easily editable. After pressing the **Lst Sq Fit** button you are prompted for the tolerance to use when fitting the grid points.

For example, consider the case shown below. The grid on the left was extruded hyperbolically from the horizontal straight line on the bottom. The resulting arc connector on the top is consists of a line segment with 11 control points, each of which corresponds to a grid point. Suppose you wanted to edit the shape of this connector. Each control point would have to be moved individually and the resulting curve would lack smoothness since the segment is linear. The same grid is shown on the right after the arc connector on the top has been run through the **Lst Sq Fit** command. Note that there are only two control points: the beginning and the ending. Furthermore, the segment is a Bezier curve (Section 7.2.5) that retains its smoothness during editing.



15. Miscellaneous

15.1 Name

Names can be assigned to database entities and blocks. Gridgen uses database entity names to associate grid with the database. When you create a connector on a database entity or project grids onto the database, each grid point is tagged with the name of the database entity it is on and its parametric coordinates. Block names are used to identify particular blocks and for sorting. Block sort order is used to determine the order in which blocks are exported. The blocks at the top of the sort list are exported first.

The naming processes for database entities and blocks are slightly different. They are each described in the following sections.

15.1.1 Database Entities

Each entity in the database will be assigned a unique name by Gridgen at the time of import or creation. You can also assign a name to an entity via the **Name** command. Having unique entity names is important because Gridgen associates grid elements (connectors, domains, and blocks) with the database entities via the entity name.

It is always necessary for the database file to be imported before the corresponding Gridgen file. Gridgen will look for the database entity name that matches the entity name stored in the Gridgen file.

The grid - database associativity via name also allows portions of the database to be varied easily. For example, a new body shape could be substituted for the original by simply importing a new database file with the same entity names but different shapes, and importing the old Gridgen file. Since Gridgen thinks the database and grid match (since all of the entity names are the same) all components are regenerated in terms of the new database shape.

Here is the strategy Gridgen uses to set the default entity names.

For IGES files:

1. Use the name entity if specified.
2. Use the subfigure name for subfigure instance entities.
3. Use the entity label and subscript directory entry fields. If the label field is blank then use a string of the form “entity type”; for example, “conic” for a conic section. If the subscript field is blank then use a string of the form “sequence number”; for example, “2” for the second entity in the file.

For all other database file types:

1. Use a string that is a concatenation of base name of file, entity type, and sequence number in file. For example, the second entity in a network file called `wing.net` would be “wing-psurface-2”.
2. If the default name is already taken, use a string of the form “Entity-#” where # is the entity number.

If the **Name** command is used to assign names to database entities, then one of two approaches must be taken when re-running Gridgen:

1. Import the entities as before and repeat the naming process prior to importing the Gridgen file.
2. Save the database in a Gridgen composite file. The composite file includes entity names.

15.1.2 Blocks

Each block must have a unique name of no more than 15 characters. Gridgen automatically assigns a name to each block when it is created. The default block name consists of combinations of the uppercase alphabetic characters. For example, the first 26 blocks created will be named A, B... Z. You can assign a more descriptive name to a block using the **ReName** command in the Block Modify menu. Gridgen checks for duplicate block names and, if necessary, you will be prompted to enter another name.

It is strongly recommended you not assign block names that contain the block number, for example `block19`. Because Gridgen's blocks change numerical order quite frequently due to grid editing, the name will very quickly become different from the block number. In other words, the block with name `block19` may actually be block number 2.

15.2 Group

The Group command allows you to select a set of a particular entity type and apply a name to that set to use as a handle for other regular operations.

15.2.1 Database Entities

To group a set of database entities follow this procedure starting from the DATABASE COMMANDS menu:

1. Press the **Group** button.
2. Select the entities to be included in the group.
3. Press the **Done** button.
4. Enter the group name via the keyboard.

The Group database command allows you to assemble one or more entities into a logical collection. Group entities are useful for enabling and disabling a large number of entities for grid point projection, for modifying many entities at once, or for changing the display of many entities.

Two points should be made regarding database groups in conjunction with the layer manager. First, database groups are a picking mechanism only, and as such, do not belong to a layer. Second, if any child of a group is disabled or is in a layer that is off, the group itself will be unavailable for picking or selection. For more information about the layer manager, see Section 5 of the User Manual.

15.2.2 Connectors, Domains, and Blocks

To group a set of connectors, domains or blocks entities follow the procedure below:

1. Press the **Define** button.
2. Type in the group name via keyboard.
3. Select the entities to be included in the group.
4. Press the **Done** button.

The grid entity Group command allows you to assemble one or more entities into a logical collection. Grid entity groups are useful for modifying many entities at once, for changing the display of many entities, and for many other regular operations. A group, always available for picking, may contain entities existing in database layers (Section 5) which are not currently visible. A group of a particular entity type is available for all operations that can be used on that entity type and groups are homogeneous.

The Group **Undef** command allows you to remove entities from the Group. This command does not delete the entities from themselves.

15.3 Copy

To copy an entity follow this procedure.

1. Press the **Copy** button.
2. Select the entity or entities to copy.
3. Press the **Done** button.
4. Perform any entity transformations (e.g., translation, etc.).
5. Press the **Done** button.

The **Copy** command allows you to duplicate entities. Note that after selecting and copying the entities Gridgen immediately puts you into the Modify menu (Section 14) so that you can translate, scale, or otherwise change the copies.

15.3.1 Database Entities

When copying database entities it is not necessary to modify the copy since two or more identical entities can occupy the same physical space.

The string “-copy” is appended to the original entity name. If several copies are made of the same entity the string “-copyN” is appended to the name where “N” indicates the nth copy.

15.3.2 Connectors, Domains, and Blocks

Copies of connectors, domains, and blocks must be modified so they are not coincident with their originals. This is because Gridgen culls duplicate grid entities after the copy operation is complete.

Block copies are re-named according to the default block naming scheme (Section 10).

15.4 Delete

To delete an entity or entities follow this procedure.

1. Press the **Delete** button.
2. Select the entity or entities to delete.
3. Press the **Done** button.

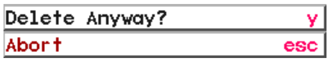
The **Delete** command allows you to remove entities from Gridgen. Note that there is no undo command; once an entity is deleted, it is gone.

15.4.1 Database Entities

If any of the database entities picked for deletion have been used by connectors or domains the referenced connectors will be graphically highlighted in the Display window and the following text will appear in the Message window.

```
WARNING:
The entities picked for deletion
include some that are referenced by the
hilited connectors.  Deleting the db
entities will sever the linkage between
the grid and db.
```

You must then choose from a button menu whether to proceed with the deletion and un-link the grid from the database (**Delete Anyway?**) or **Abort** and keep things the way they are.



15.4.2 Connectors

If any of the connectors picked for deletion are used in domains or blocks then those domains and blocks will be graphically highlighted in the Display window, the number of domains and blocks to be deleted by deleting this connector will be tallied in the Blackboard, and the following text will appear in the Message window.

```
DELETE CONNECTORS:
Deleting these connectors
will cause all domains and
blocks using them to be
destroyed.  Select an
option below.
```

You must then choose from a button menu whether to proceed with the connector deletion and the implicit domain and block deletion (**Delete Connectors**) or **Abort** and keep things the way they are.

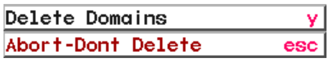
15.4.3 Domains

If any of the domains picked for deletion are used in blocks then those blocks will be graphically highlighted in the Display window, the number of blocks to be deleted by deleting this domain will be tallied in the Blackboard, and the following text will appear in the Message window.

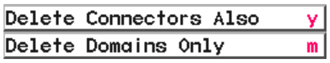
```
DELETE DOMAINS:
Deleting these domains
```

```
will cause all blocks
using them to be
destroyed.  Select an
option below.
```

You must then choose from a button menu whether to proceed with the domain deletion and the implicit deletion (**Delete Domains**) or **Abort** and keep things the way they are.

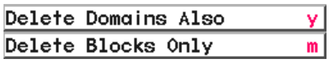


You are also given the option to delete either just the domain representation (**Delete Domains Only**) or the domain representations and the domains' connectors **(Delete Connectors Also**). Should you choose to delete connectors then only the connectors unique to the deleted domains will be deleted.

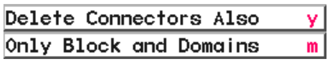


15.4.4 Blocks

When deleting blocks you are given the option to delete either just the block representation (**Delete Blocks Only**) or the block representations and the blocks' domains **(Delete Domains Also**). Should you choose to delete domains then only the domains unique to the deleted blocks will be deleted.



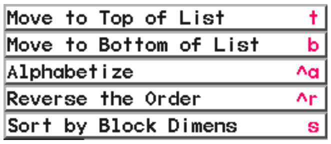
You are also given the option to delete the blocks' domains **(Delete Domains Also**). Should you choose to delete domains then only the domains unique to the deleted blocks will be deleted.



15.5 Sort

The **Sort** command is used to arrange the numerical sequence of the blocks in the system. This command is useful for arranging the grid in a manner that will be better suited for the analysis software. Sorting is also useful for examining the block sizes in a large grid.

The first step after pressing the **Sort** button is to select from the Display window or Browser the block(s) to be moved in the numerical sequence. Once these blocks are selected, they can then be arranged using one of the following sorting criteria.



Move to Top of List -This will move all selected blocks to the top of the overall block list. If 6 of 10 blocks were selected, for example, they will occupy the first six out of 10 positions in the block list.

Move to Bottom of List - This will move all selected blocks to the bottom of the overall block list. If 6 of 10 blocks were select, for example, they will occupy the last six out of 10 positions in the block list.

Alphabetize - This will alphabetize the selected blocks and rearrange them relative to one another, keeping all non-queued blocks in their original position. For example, if blocks 1, 2 and 4 were selected in the list below (shown in bold), their alphabetized sort is as follows.

Example of alphabetic block sorting.

block number	before	after
1	wing_a (18x29x11)	axle (10x10x10)
2	fuselage (21x18x45)	fuselage (21x18x45)
3	aileron (21x31x19)	aileron (21x31x19)
4	axle (10x10x10)	wing_a (18x29x11)

Reverse the Order - This will reverse the order of the picked blocks relative to one another, keeping all non-queued blocks in their original position. For example, if blocks 1, 2 and 4 were queued in the list below, their new arrangement is as follows.

Example of reversing the sort order.

block number	before	after
1	wing_a (18x29x11)	axle (10x10x10)
2	fuselage (21x18x45)	fuselage (21x18x45)
3	aileron (21x31x19)	aileron (21x31x19)
4	axle (10x10x10)	wing_a (18x29x11)

Sort By Block Dimens - This will rearrange the order of the selected blocks relative to one another from the largest block size to the smallest size, leaving all non-selected blocks in their original position.

Once a sorting criterion is picked, the blocks will be rearranged in the Blackboard display as dictated. You may then select a new set of blocks to sort and repeat this process, accept the new arrangement with the **Done Selecting** menu button, or revert back to the original order with the **Abort** menu button.

15.6 Enable/Disable

The **En/Disable** command is used to temporarily “turn off” entities. There are two reasons for disabling entities:

1. disabled entities are rendered in a minimal sense, reducing visual clutter in the Display window
2. disabled entities are not loaded into the pick list, which can substantially improve the graphical response during picking, especially for large grids

En/Disable †

Upon invoking the **En/Disable** command, the Browser lists each entity according to its enable/disable status. The browser text will be white if the entity is enabled. The browser text will be salmon if the entity is disabled. Picking an entity will immediately toggle its enable/disable status and the Display window image will change accordingly.

The **En/Disable** command is also available via the Display menu.

15.6.1 Database Entities

Disabled database entities are not rendered by default. When a disabled database entity is enabled, its previous display attributes are recalled, and the entity is rendered as before.

15.6.2 Connectors

Connectors cannot be disabled.

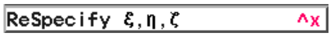
15.6.3 Domains

Only domains that are not used in blocks may be disabled. To disable a domain used in a block, you must disable the entire block. The display of a disabled domain’s grid points is turned off and the domain’s connectors are rendered in a half-tone color.

15.6.4 Blocks

A disabled block’s connectors are rendered in a half-tone color and the grid point display of each of its constituent domains is turned off.

15.7 ReSpecify ξ, η and ζ

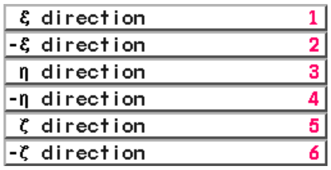


ReSpecify ξ,η,ζ ^x

The **ReSpecify** ξ, η, ζ command is used to specify the orientation of a structured block’s computational coordinate axes. Recall that connector, domain, and block creation proceed without regard for the block’s eventual computational directions. This is the only command in Gridgen in which the computational axis system is important. This command is useful for ensuring that the block’s axis orientation is consistent with the requirements of the your analysis software package. Specifying the computational axis system in no way affects the grid, analysis software boundary conditions, or block construction details and may be applied as many times as necessary.

ReSpecify ξ, η, ζ is located in Modify Blocks menu. When it is invoked all connectors on the twelve edges of the block are highlighted in pink and are grouped according to the three computational directions in the block. Position the cursor over an edge and all edges in the same group of like-indexed block edges (in other words, the four block edges with the same varying index) will be highlighted. After you press the ***right***

mouse button to select an edge, you will select from the button menu whether the connector group is positive or negative ξ, η, or ζ directed as shown below.



The arrows on the connectors in the Display window indicate the direction of increasing computational index. After the direction is set, the selected connectors are redrawn using the following color code.

Computational coordinate color scheme.

computational coordinate	**color**
ξ	red
η	orange
ζ	yellow

Next, select one of the two remaining edge groups with the ***right mouse*** button and select its computational coordinate from the menu. The connectors are again redrawn to reflect their new orientation. The last step is to select from the menu the computational coordinate of the third and final edge group.

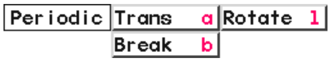
15.8 *Align* ξ, η, ζ

The **Align** ξ, η, ζ command in the Modify Blocks menu allows you to specify the computational coordinate axis systems of one or more structured blocks so that, where possible, they are aligned with the ξ, η, ζ axes of a block of your choosing. You might use this command if your chosen analysis software package restricts the type of inter-block connections.

For this command to function properly, all of the selected blocks must be connected to one another so that the coordinate system from the chosen block can be propagated throughout the blocks. Any physically isolated blocks will not be affected by this command. Also, the master block must be selected in the group for modification.

15.9 *Periodicity*

The Periodic commands may be used to slave two connectors or two domains together to ensure periodicity in your grid.



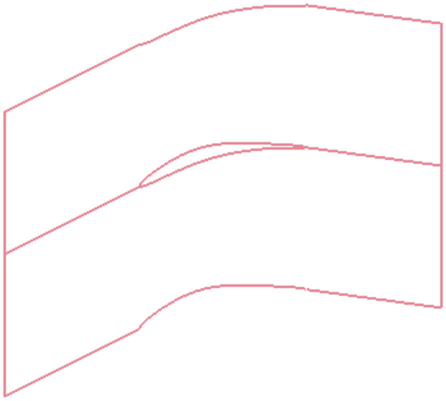
The Periodic **Trans** and **Rotate** commands allow you to copy an existing connector or domain, and transform it via translation or rotation (respectively) so that future changes (transformations, editing, dimension, distribution) to the original grid are automatically made to the periodic copy and vice versa. Please note that periodicity is only enforced in pairs - no more than two grids or two domains may be periodic with each other.

Periodic **Trans** - Pressing this button lets you pick an existing connector or domain that is then copied and transformed using Gridgen's translate (Section 14.3.1) command. Any subsequent changes to either grid will be reflected in the other.

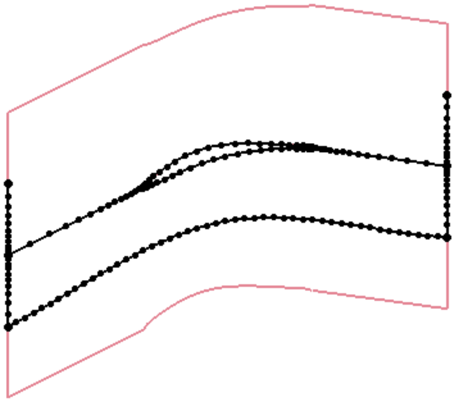
Periodic **Rotate** - Pressing this button lets you pick an existing connector or domain that is then copied and transformed using Gridgen's rotate (Section 14.3.4) command. Any subsequent changes to either grid will be reflected in the other.

Periodic **Break** - This command invokes the Browser with a list of every periodic connector or domain pair. You then select the periodic pair that you wish to break (delete). Pressing the **Done** button then separates the pair and each grid may then be modified independently.

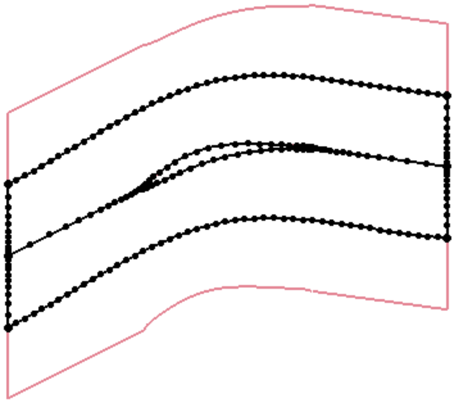
A sample implementation of periodicity is described below. Consider the two dimension blade-centered passage shown below on which we want to generate a simple blade-centered H mesh.



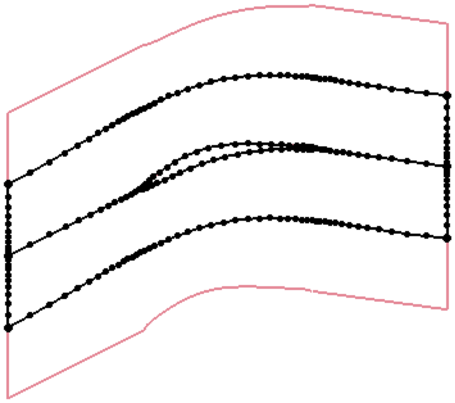
First, using Gridgen's normal tools, we can create the connectors shown below.



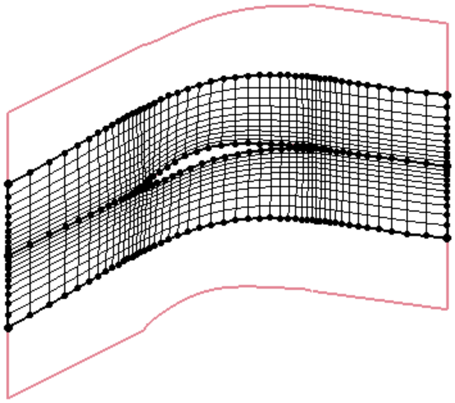
The goal is to ensure that the lower connector is periodic with the upper boundary of the grid. We can accomplish this by entering the Connector menu, pressing Periodic **Trans**, selecting the lower connector and translating it into the upper position resulting in the image below.



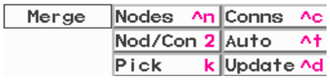
We have now created the boundaries of the surface grids and established that the upper and lower boundaries are to be periodic. With periodicity ensured, we can change the distribution of points along the lower connector and the distribution will automatically change on the upper connector as well (see below).



The surface grids for the passages are shown below.



15.10 Merge

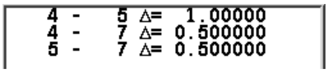


These commands for removing coincident and overlapping connectors are available on the Connector menu and also from within domain and block face assembly.

15.10.1 Nodes

The Merge **Nodes** command allows you to select node pairs to be merged into a single (`x,y,z`) point. This merging is not a topological merging as happens when using the **Pick an Unlinked Con** command during domain assembly (Section 9.4). Nor does this command alter the node tolerance in the Defaults menu (Section 3.7.1). Rather, this command alters the physical location of one of the nodes of the selected pair to match the coordinates of the other node. If one or both of the nodes are database constrained a branch segment is added.

After pressing the Merge **Nodes** button, you must enter the distance within which two existing nodes must lie in order to be candidates for merging. After you enter this value, Gridgen lists in the Browser all node pairs that lie within this tolerance.



```
4 -    5 Δ=  1.00000
4 -    7 Δ= 0.500000
5 -    7 Δ= 0.500000
```

Each Browser item consists of three bits of information. The two integer numbers separated by dashes are the two node numbers. The symbol between the two node numbers indicates whether the nodes are database constrained as follows:

`-` - Neither node is database constrained.

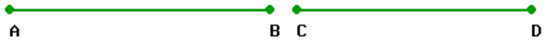
= - Both nodes are database constrained.

> - Only the first node is database constrained.

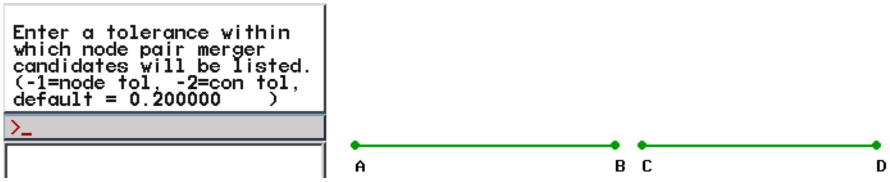
< - Only the second node is database constrained.

The real number on the right is the actual distance between the two nodes. For example, the second line shown above indicates that nodes 4 and 7 are separated by a distance of 0.5. Pointing at the Browser entry using the mouse highlights the two nodes in the Display window. To merge nodes, select the pairs to be merged from the Browser and press **Done**.

Consider the example shown below. There are two connectors, each consisting of a single straight line segment. The labeled nodes are: A (x=0), B (x=1), C (x=1.1), and D (x=2). Therefore the gap between B and C is 0.1. Perhaps this gap exists because connectors AB and CD were created on sloppy database entities with a gap. We wish to merge nodes B and C using the Merge **Nodes** command.

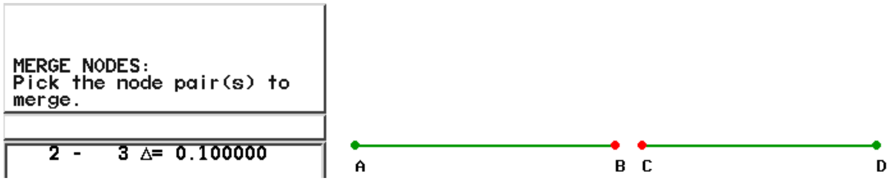


Enter the node merging tolerance. Every node pair closer to each other than the tolerance is a candidate merger pair. In the example shown below, we know that the distance between B and C is 0.1 so we enter a tolerance of 0.2 to be sure the test identifies node pair B-C as a candidate.



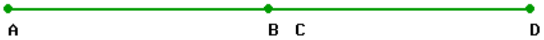
```
Enter a tolerance within
which node pair merger
candidates will be listed.
(-1=node tol, -2=con tol,
default = 0.200000    )
>_
```

In the figure below, Gridgen shows in the Browser the node pair within the tolerance and the actual distance between the nodes. You can then select from the Browser or Display window the pair to merge.



```
MERGE NODES:
Pick the node pair(s) to
merge.

  2 -    3 Δ= 0.100000
```

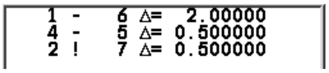
After picking node pair B-C and pressing **Done**, Gridgen merges the two nodes together. In the example shown below node C has been relocated to node B (x=1). Now the two connectors match end to end and can be used to create a domain.



15.10.2 Connectors

The Merge **Conns** command allows you to select pairs of dimensioned connectors to be merged together into a single connector. This merging is not topological; one of the connectors is replaced with the other. Merging connectors is good for removing coincident connectors from your grid. This command does not alter the connector tolerance in the Defaults menu (Section 3.7.2).

After pressing the Merge **Conns** button you must enter the distance within which two existing connectors must lie in order to be candidates for merging. If the two connectors have the same dimension, Gridgen compares corresponding grid points on the two connectors. If the maximum distance between grid points is less than the tolerance you entered, the connector pair becomes a candidate for merging. If the two connectors have different dimensions Gridgen compares a small sampling of the grid points on one connector to the shape of the other connector and vice versa. If the maximum distance computed between the two connectors is less than the tolerance you entered the connector pair becomes a candidate for merging. After all the comparisons are complete Gridgen lists in the Browser all candidate connector pairs.

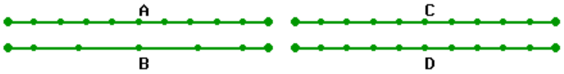


```
  1 -    6 Δ=  2.00000
  4 -    5 Δ= 0.500000
  2 !    7 Δ= 0.500000
```

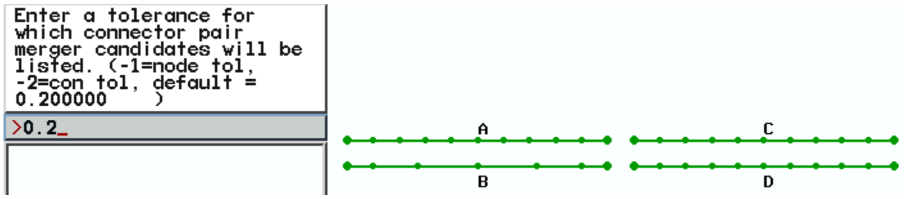
Each Browser item consists of three bits of information. The two integer numbers separated by dashes are the two connector numbers. The real number on the right is the actual maximum distance between the two connectors. For example, the second line shown above indicates that connectors 4 and 5 are separated at most by a distance of 0.5. Pointing at the Browser entry using the mouse highlights the two connectors in the Display window. To merge connectors, select the pairs to be merged from Browser and press **Done**.

In the Browser illustration above, line 3 contains an exclamation point (!). This means that the two connectors do not have the same dimension. However, they will still be merged if you select them. Connectors used in structured domains are not candidates for merging if they have different dimensions.

Consider the example shown below with four connectors A, B, C, and D. Each connector is 1 unit long, connectors A and B are vertically separated by 0.1 units as are connectors C and D. Note that all connectors have 11 points except for B which has 7. Imagine, for example, connectors A and B are supposed to be coincident as are C and D, but they were created on a sloppy database resulting in the vertical gap we want to eliminate.

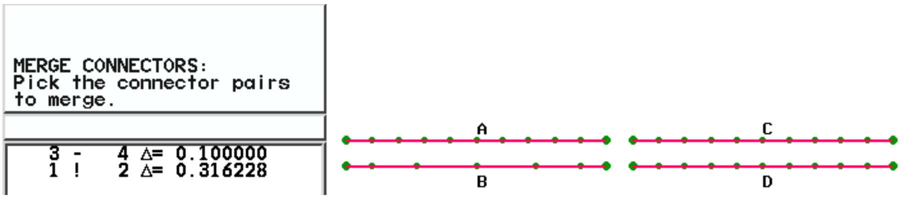


After pressing the Merge **Conns** button, you enter the connector merge tolerance. In this example we use a value of 0.2.



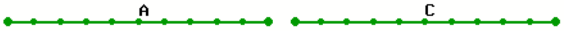
```
Enter a tolerance for
which connector pair
merger candidates will be
listed. (-1=node tol,
-2=con tol, default =
0.200000    )
>0.2_
```

The Browser then lists all connector pairs that match within the tolerance of 0.2. Pairs A-B and C-D are candidates, and Gridgen recognizes A and B do not have the same dimension.



```
MERGE CONNECTORS:
Pick the connector pairs
to merge.

   3 -   4 Δ= 0.100000
   1 !   2 Δ= 0.316228
```

After you select both pairs for merging Gridgen eliminates connector B in favor of connector A and eliminates connector D in favor of connector C as shown below.



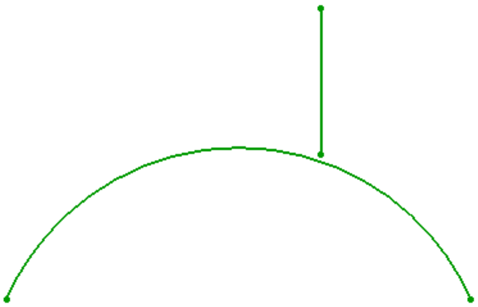
15.10.3 Nodes and Connectors

The Merge **Nod/Con** command combines the Merge **Nodes** command with a pre-processing steps that splits overlapping connectors.

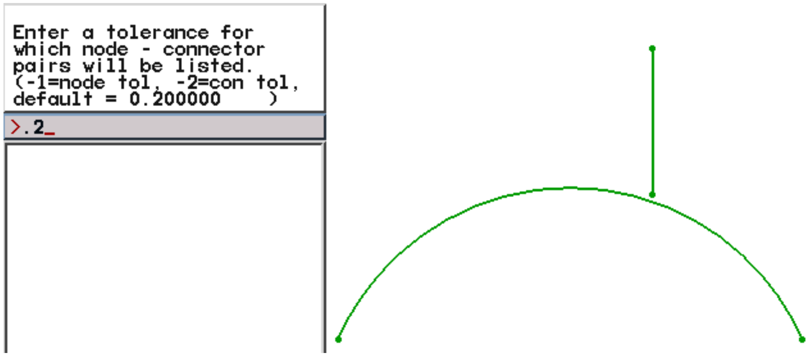
After pressing the Merge **Nod/Con,** button you enter the distance within which a node and a connector must lie in order to be candidates for merging. After you enter this value, Gridgen lists in the Browser all node-connector pairs that are within this tolerance.

Each Browser item consists of three bits of information. The two integer numbers separated by dashes are the node and connector numbers, respectively. The real number on the right is the actual maximum distance between the node and connector. To merge the node and connector, select the pairs to be merged from the Browser and press **Done**. The connector is split at the location closest to the node, and the new node on the connector is merged with the node.

For example, consider the grid below with two connectors: a circular arc and a vertical line. The lower end point of the vertical line is very close to the circle and we wish to make the two connectors intersect. Perhaps the gap resulted from a sloppy database model from which the connectors were originally created.

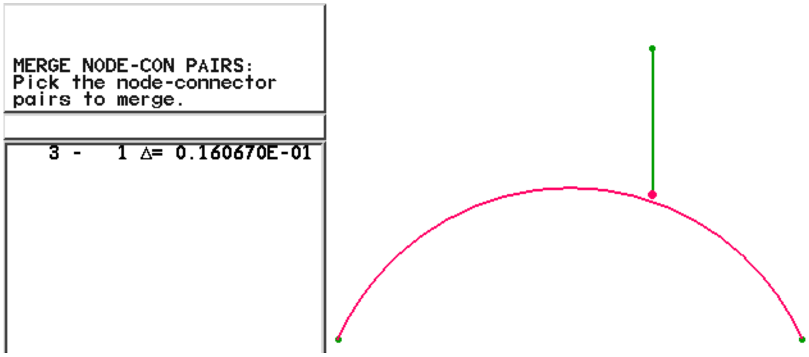


After pressing the Merge **Nodes and Conns** button we enter a tolerance of 0.2, which is large enough to locate the connector and node we wish to merge in this example (the horizontal distance between the circle's end points is 1.0).



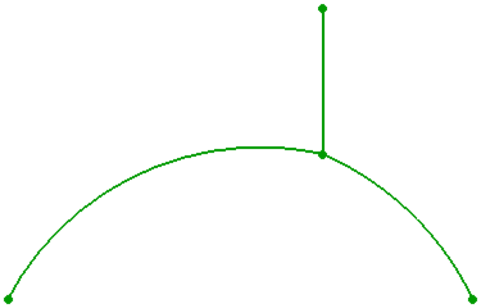
```
Enter a tolerance for
which node - connector
pairs will be listed.
(-1=node tol, -2=con tol,
default = 0.200000     )
>.2_
```

The Browser then lists all node - connector pairs that fall within the specified tolerance. In this case, it finds the node at the lower end of the vertical connector and the circular arc connector.



```
MERGE NODE-CON PAIRS:
Pick the node-connector
pairs to merge.

   3 -    1 Δ= 0.160670E-01
```

After selecting the node - connector pair from the Browser or Display window and pressing Done Gridgen splits the arc at the location closest to the node and merges the node with the new node resulting in the three connectors shown below.



15.10.4 Automatic

If you select **Merge Auto**, you can specify a merge tolerance. Gridgen will then apply the merge connectors and merge connectors and nodes commands with that tolerance to try to create a watertight geometry.

15.10.5 Pick

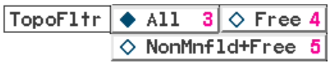
If you select **Merge Pick**, you can pick two connectors from the Display Window, and Gridgen will merge them together. After you pick the second connector, Gridgen will highlight both connectors, one in pink and one in orange. You will be given the option to merge the connectors or abort the merge operation.

15.10.6 Update Doms

This command updates the triangulation of domains after merging connector. Only used on unstructured domains.

15.10.7 Topological Filter Commands

You may filter the list of “mergeable” connectors by enabling the menu buttons on the **TopoFltr** menu prior to pressing one of the Merge command buttons. By default **TopoFltr** will consider all connectors as candidates for merging.



All - When this button is enabled (default) all nodes and connectors are pickable.

Free - When this button is enabled only connectors used by a single domain are pickable.

NonMnfld+Free - When this button is enabled all connectors that meet the above **Free** criteria as well as connectors used by one or more domains are considered.

15.11 Intersect

The **Intersect** command allows you to create new point and curve database entities by intersecting existing entities. The following table lists the entity combinations that can be intersected along with the results of the intersection operation.

Intersection Results

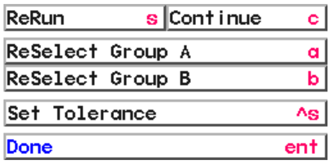
entity A	entity B	results
curve	curve	points and new knots added to each curve
curve	plane	points and new knots added to the curve
curve	surface	points and new knots added to the curve
surface	plane	curves
surface	surface	curves

The only manner in which curve entities are changed during intersection is that a new knot (control point) is added to the curve's definition. The new knot does not alter the curve's shape, but it does provide a pickable location for subsequent operations. Curves are not split by intersecting.

Curve - curve intersections are computed relative to a plane. Gridgen uses the plane perpendicular to the line of sight (parallel to the plane of the screen). The implication is that Gridgen computes both true and apparent curve - curve intersections. A true intersection is one in which the curves truly intersect in 3D space. In this instance, Gridgen creates one new point entity and adds a knot to each curve. An apparent intersection is one in which the curves intersect only in the plane onto which they are projected for intersection. For example, consider intersecting the line `(0,0,0)` - `(1,0,0)` with the line `(0.5,-0.5,1)` - `(0.5,0.5,1)`. These lines do not truly intersect - the first is defined in the `z=0` plane and the second in the `z=1` plane. However, they do apparently intersect when projected into the `z=0` plane. In this instance, Gridgen creates two new points (one at the apparent intersection on each curve) and adds a knot to each curve.

Database point entities are not candidates for intersection. Nor are plane-plane intersections allowed. Connectors, domains, and blocks cannot be intersected.

The intersection process is controlled by selecting two groups of entities called Group A and Group B. (You do not have to create group entities and use them for computing intersections; the term "group" in this discussion refers to the two collections of entities that you pick for intersecting.) Group A may consist of a single curve or surface, as can Group B. Every entity in Group A will be intersected with every entity in Group B. Each new entity resulting from the intersections will be drawn in the Display window in yellow as it is computed. The Blackboard window will display a horizontal bar that indicates the percent of the entity pairings for which intersections have been attempted.



Commands of interest during intersection include:

Run, **ReRun** - The **Run** command starts the intersection process and is available only after both Groups A and B have been selected. The **ReRun** command allows you to restart the intersection process by deleting any current intersections and re-computing intersections starting with the first entity of group A and the first entity of group B. You would use the **ReRun** command if you started the intersection process but then paused it after realizing you forgot to set the appropriate tolerance, for example.

Continue - After the intersection process has been paused, pressing the **Continue** button will re-start the process from where it was stopped. If you change the intersection tolerance and then press continue, the new tolerance will only be applied to intersections computed after the **Continue** button was pressed.

Pause - Pressing this button will stop the intersection process.

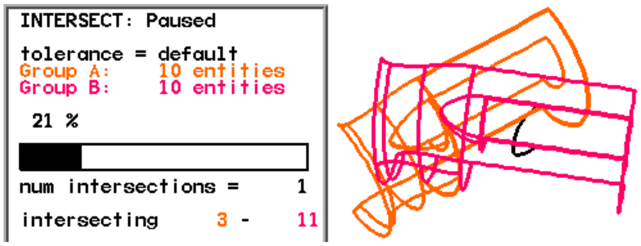
Select Group A - Pressing this button allows you to select or re-select the first group of entities to be intersected. All entities of Group A are intersected with all entities of Group B. Group A entities are colored orange.

Select Group B - Pressing this button allows you to select or re-select the second group of entities to be intersected. All entities of Group A are intersected with all entities of Group B. Group B entities are colored pink.

Set Tolerance - Intersections are computed within a tolerance; the computed intersections will vary from the theoretical intersections by no more than the tolerance. By default, intersections that result in a curve use the surface-fit tolerance (Section 3.7.1). Intersections that result in a point use the same-point tolerance (Section 3.7.1). Pressing the **Set Tolerance** button allows you to enter your own tolerance value for the intersection calculations.

Abort - When the intersection process is paused, pressing the **Abort** button will exit the intersection command and delete any intersection curves it had computed.

Done - When the intersection process is paused, pressing the **Done** button will exit the intersection command and save any intersection curves it had computed.



```
INTERSECT: Paused

tolerance = default
Group A:     10 entities
Group B:     10 entities

 21 %

num intersections =      1

intersecting      3 -    11
```

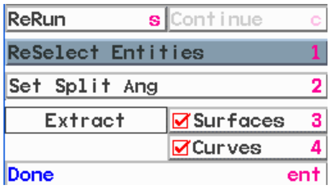
The above illustration shows many of the graphical features of the intersection procedure. The Group A and Group B entity numbers currently being intersected are listed in the Blackboard and their images in the Display window are highlighted with a thicker line width relative to other members of the group. New intersections immediately appear in the Display window, yellow in color, as does a numerical tally of the number of intersections in the Blackboard. The percent complete bar, the horizontal bar yellow in color, indicates the progress of the intersection procedure based on the total number of combinations of Group A and B entities.

Grid entities (connectors, domains, and blocks) may not be intersected.

15.12 Feature Extraction

The **Feature Ext** command is used to subdivide and extract curves from database shell entities to make them easier to use. Many shell entities only contain shape information. They do not have any internal topology to attach grid to specific locations. Using feature extraction, you can create curves at natural breaks in the geometry and subdivide shells into smaller pieces along these feature lines.

The feature extraction process is controlled by entering a split angle. Database shell entities are composed of triangular facets. If the angle between two adjacent facets if greater than the entered split angle, the edge they share is identified as a feature, and Gridgen will consider it a candidate for curve extraction and surface splitting. You can control whether curves are extracted, surfaces are split, or both. Controls available for feature extraction are:



Run, **ReRun** - The **Run** command starts the feature extraction process and is available only after a shell has been selected. The **ReRun** command allows you to restart the feature extraction process by deleting any current extracted curves and split surfaces and re-feature extracting the original shells. This would be used if you wanted to feature extract using a different split angle.

Continue - After the feature extraction process has been paused, pressing the **Continue** button will re-start the process from where it was stopped. If you change the split angle and then press continue, the new split angle will only be applied to shells remaining to be extracted after the **Continue** button was pressed.

Pause - Pressing this button will stop the feature extraction process.

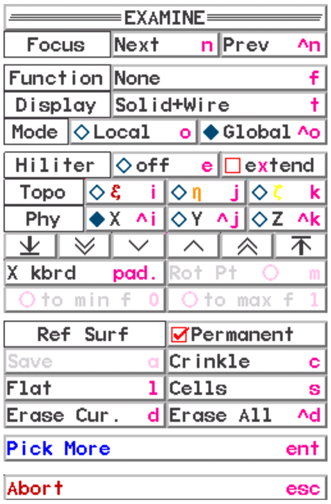
Select Entities, ReSelect Entities - Invoking this allows you to select a set of database shell entities for feature extraction.

Set Split Ang - After pressing this button, you will be prompted to enter an angle in degrees. If the angle between two adjacent triangle facets in a shell is greater than the angle you type here, the edge between them will be a candidate feature. Whether it is actually extracted as a curve or surface edge depends on the setting of the **Extract Surfaces** and **Extract Curves** toggles.

Extract Surfaces - If this button is selected, Gridgen will look for closed boundaries of features that can be used to subdivide the input database shell entity. The shell will subdivided if the features form a closed loop or a loop that closes with the boundary of the shell.

Extract Curves - If this button is selected, Gridgen assemble features into database curves (polylines). Contiguous features are joined into a single curve for easier reference during grid construction and other database operations.

16. Diagnostic Functions



The **Examine** command provides you with three sets of information regarding the selected entities and function:

1. a colored display in the Display window,
2. a tabular listing in the Blackboard, and
3. the hiliter (if enabled).

When the Examine command is invoked it automatically displays the diagnostic measure you last examined or a simple listing of entity attributes (if this is the first time the command has been invoked).

16.1 Focus Entity

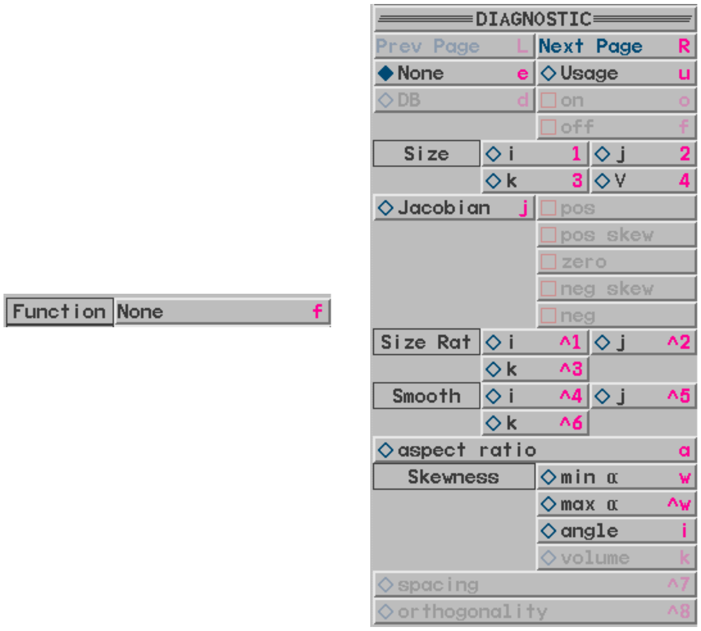
The focus entity is the entity that is emphasized and hilited (if the hiliter is enabled) in the Display window. If either Local mode (Section 16.4) or the hiliter (Section 16.5) are enabled, the focus entity is also the entity to which the Blackboard text corresponds.

Focus **Next -** Moves the hiliter (if enabled) and the Blackboard text (if in Local mode) to the next selected entity in numerical order. If focusing on the last selected entity, focus will be moved to the first selected entity.

Focus **Prev -** Moves the hiliter (if enabled) and the Blackboard text (if in Local mode) to the previous selected entity in numerical order. If focusing on the first selected entity, focus will be moved to the last selected entity.

16.2 Diagnostic Function

The term diagnostic function refers to the particular type of information to be calculated and displayed for the selected entities. The current diagnostic function is printed on the button next to the label Function.



The default diagnostic function is **None** which simply provides entity dimension information in the Blackboard. Pressing the button next to the Function label allows you to select the diagnostic function to visualize in the Display window for the selected entities. Each Gridgen entity type (connector, domain, block) saves the last diagnostic function examined and subsequent invocations of the **Examine** command will begin with that diagnostic function. Note that all diagnostic functions are not available for every Gridgen entity or mesh type.

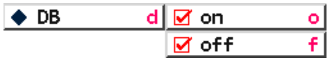
16.2.1 None

The focus entity (Local mode) or all selected entities (Global mode) will be hilited in the Display window and size information will be listed in the Blackboard. Function **None** in Global mode is similar to the **Summarize** command found in previous Gridgen versions, which has been replaced by this feature.

16.2.2 Usage

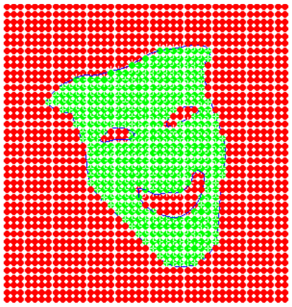
Selecting Function **Usage** will highlight in the Display window the focus entity (Local mode) or all selected entities (Global mode) along with entities of other types that use or are used by the examined entities. Tabular information about how the entity is used is listed in the Blackboard.

16.2.3 Database

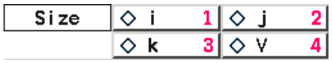


For connectors and domains only, selecting **Function DB** causes grid points that are on and/or off the database to be rendered in the Display window and listed in the Blackboard. The toggle buttons **on** and **off** indicate whether or not to render grids points that are on or off the database, respectively. Keep in mind that Gridgen considers a point to be off the database if the point does not have an explicit reference to a database entity and parametric coordinates. A point flagged as off may actually be within the same point tolerance of the database.

The example below shows the database function displayed on a single quadrilateral domain that has been projected onto a trimmed surface in the database that is shaped like a mask. It is very easy to distinguish the points on the database (green) from those off the database (red).



16.2.4 Size



For grid entities only, the cell size is rendered in the Display window and a tabular summary is listed in the Blackboard.

For structured domains, the radio buttons **i**, **j**, and **A** indicate which cell size you would like to examine: the i, j direction lengths or the cell area, respectively. Only the cell area measure is available for unstructured domains. Quadrilateral cell area is computed by taking the minimum of two areas computed by triangulating the cell along its two diagonals.

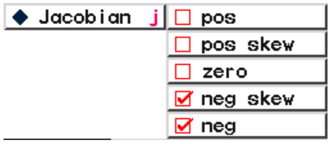
For structured blocks, the radio buttons **i**, **j**, **k**, and **V** indicate which cell size you would like to examine: the i, j, k direction lengths or the cell volume, respectively. Only the cell volume measure is available for unstructured blocks. Hexahedral cell volume is computed using the method of Davies and Salmond (Reference 6) which uses pyramid decomposition.

The **Size** function is a point based measure. Gridgen uses the current i-j-k coordinate differenced with the index+1 point in the computational direction chosen. Furthermore, this function will be calculated across entity boundaries as long as the adjacent entities are loaded into **Examine** as well. Otherwise, the stencil direction will be reversed as necessary at the entity boundary. If a topological hiliter (Section 16.5) is active, the second index+1 point used in the calculation will be rendered with an open diamond adjacent to the hiliter intersection.

Since this function is directional along computational coordinates, adjacent entities must be computationally aligned (Section 15.8) in order for reference surfaces to exhibit a continuous variation in the function across entity boundaries. Non-aligned entities will display a discontinuity across their common boundary. Block

(2D or 3D) adjacency is determined by block connections using a common boundary. If a connection between two blocks is overridden with a baffle type boundary condition, no adjacent entity information will be available at that boundary.

16.2.5 Jacobian



◆ Jacobian j

□ pos

□ pos skew

□ zero

☑ neg skew

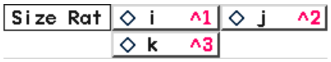
☑ neg

For structured blocks and prism blocks the Jacobian is computed for each cell. The Jacobian computation for hexahedra consists of computing the triple product at each of the cell's corners using only other grid points from that cell and then computing the average of the corner Jacobians. For prisms the triple product is pre-multiplied by a coefficient of 0.5. The toggle buttons for **pos**, **pos skew**, **zero**, **neg skew**, **neg** indicate whether or not to render positive, positive skew, zero, negative skew, and negative Jacobian cells in the Display window, respectively. The cell Jacobians are classified according to the table below.

Cell Jacobian Classifications

Jacobian classification	**menu button**	**average value of corner Jacobians**	**signs of corner Jacobians**
positive	**pos**	positive	all positive
positive skew	**pos sku**	positive	mixed signs
zero	**zero**	zero	n/a
negative skew	**neg sku**	negative	mixed signs
negative	**neg**	negative	all negative

16.2.6 Size Ratio



Size Rat

◇ i ^1

◇ j ^2

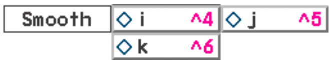
◇ k ^3

For structured grid entities, the ratio of adjacent cell sizes is rendered in the Display window and a tabular summary is listed in the Blackboard. The radio buttons **i**, **j**, and **k** indicate the direction in which the cell size ratios are to be computed: the `i`, `j`, or `k` computational direction, respectively.

The **Size Ratio** function is a point based measure. Gridgen computes the ratio between the distance from the current i-j-k coordinate to the index+1 point and the distance from the current i-j-k to the index-1 point in the computational direction chosen. The larger of the two distances is always divided into the smaller, so all values range between 0 and 1. Furthermore, this function will be calculated across entity boundaries as long as the adjacent entities are loaded into **Examine** as well. Otherwise, the value at boundaries will be set to 1. If a topological hiliter (Section 16.5) is active, the index+1 and index-1 points used in the calculation will be rendered with open diamonds on either side of the hiliter intersection.

Since this function is directional along computational coordinates, adjacent entities must be computationally aligned (Section 15.8) in order for reference surfaces to exhibit a continuous variation in the function across entity boundaries. Non-aligned entities will display a discontinuity across their common boundary. Block (2D or 3D) adjacency is determined by block connections using a common boundary. If a connection between two blocks is overridden with a baffle type boundary condition, no adjacent entity information will be available at that boundary.

16.2.7 Smoothness



For structured grid entities, the normalized turning angle from one side of a gridpoint to the opposite is rendered in the Display window and a tabular summary is listed in the Blackboard. The radio buttons **i**, **j**, and **k** indicate the direction in which the turning angle measure is to be computed: the `i`, `j`, or `k` computational direction, respectively.

The Smooth function is a point based measure. Gridgen computes the angle turned while traversing three adjacent coordinates: index-1, the current i-j-k, and index+1 in the computational direction chosen. A value of 1 indicates there is no turn, or the three points are colinear. A value of 0.5 indicates a 90 degree turn. And a value of 0 indicates a 180 degree turn, or the grid is folded onto itself. Furthermore, this function will be calculated across entity boundaries as long as the adjacent entities are loaded into **Examine** as well. Otherwise, the value at boundaries will be set to 1. If a topological hiliter (Section 16.5) is active, the index+1 and index-1 points used in the calculation will be rendered with open diamonds on either side of the hiliter intersection.

Since this function is directional along computational coordinates, adjacent entities must be computationally aligned (Section 15.8) in order for reference surfaces to exhibit a continuous variation in the function across entity boundaries. Non-aligned entities will display a discontinuity across their common boundary. Block (2D or 3D) adjacency is determined by block connections using a common boundary. If a connection between two blocks is overridden with a baffle type boundary condition, no adjacent entity information will be available at that boundary.

16.2.8 Aspect Ratio

For grid entities, the cell aspect ratio is rendered in the Display window and a tabular summary is listed in the Blackboard.

Quadrilateral cell aspect ratio is computed from the ratio of the average length and average width. The aspect ratio is always greater than or equal to 1 with a value of 1 representing a square.

Hexahedral cell aspect ratio is computed from the ratio of the maximum of the length, width, and height and the minimum of the length, width, and height. The aspect ratio is always greater than or equal to 1 with a value of 1 representing a cube.

Triangular cell aspect ratio is computed as the ratio of the radius of the cell’s circumscribing circle to 2 times the radius of the inscribed circle.

Tetrahedral cell aspect ratio is computed as the ratio of the radius of the cell’s circumscribing sphere to 3 times the radius of the inscribed sphere.

Prism aspect ratio is the ratio of the average height of the prism and the average length of the base’s (triangle) edges. The aspect ratio of a prism can be less than 1.

Pyramid aspect ratio is the ratio of the height of the pyramid and the average length of the base's (quadrilateral) edges. The aspect ratio of a pyramid can be less than1.

16.2.9 Skewness

For grid entities, the skewness is rendered in the Display window and a tabular summary is listed in the Blackboard.

The **min** α skewness function is represented by the cell's included minimum angle in degrees for all types of grid.

The **max** α skewness function is represented by the cell's included maximum angle in degrees for all types of grid.

The **angle** skewness function is represented as the maximum ratio of the cell's included angle to the angle of an equilateral element. The **angle** skewness applies to all cell types and is available for domains and blocks. The **angle** skewness varies between 0 (good) and 1 (bad). It is recommended this skewness measure be kept below 0.8 for a good grid, values below 0.9 are acceptable, depending on the solver.

The **angle** skewness is computed as follows:

max [(Qmax - Qe)/(180 - Qe), (Qe - Qmin)/Qe]

where:

Qmax = largest angle in cell, in degrees

Qmin = smallest angle in cell, in degrees

Qe = angle for equilateral element (i.e. 60 for triangle, 90 for quad, etc.)

The **volume** skewness function is represented as a ratio of the cell's volume to the optimum cell volume and only applies to triangles and tetrahedral. The **volume** skewness varies between 0 (good) and 1 (bad). It is recommended this skewness measure be kept below 0.8 for a good grid, values below 0.9 are acceptable, depending on the solver

The **volume** skewness is computed as follows:

(Optimal Cell Size - Actual Cell Size) / (Optimal Cell Size)

where the Optimal Cell Size is the **area** (for 2D) or **volume** (for 3D) of an equilateral cell with the same circumradius as the actual cell.

The **angle** and **volume** functions are based on Fluent®'s equiangle and equivolume grid quality criteria.

16.2.10 Spacing

For structured boundary condition regions (domains) the spacing of gridpoints adjacent to the region is rendered in the Display window and a tabular summary is listed in the Blackboard. This function is intended to assist those building viscous structured grids since it in essence provides "near wall spacing" in a very accessible manner. Since this function is considered a boundary diagnostic, it is only available from the ANALY-

SIS S/W COMMANDS menu.

The **spacing** function is a measure of the distance from each point on the structured boundary to the adjacent point on the interior of the associated block. Normals to the boundary are rendered for further directional clarification. Spacing values at individual points on the boundary can be obtained by enabling the hiliter.

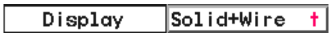
16.2.11 Orthogonality

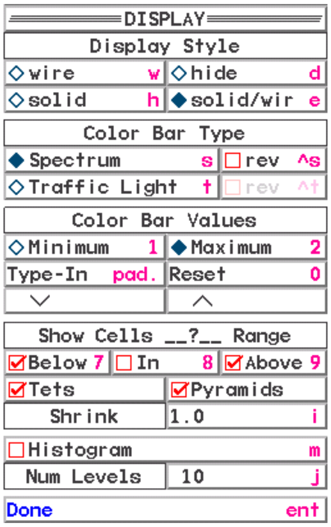
For structured boundary condition regions (domains) the orthogonality of gridpoints adjacent to the region is rendered in the Display window and a tabular summary is listed in the Blackboard. This function is intended to assist those building viscous structured grids since it in essence provides “near wall orthogonality” in a very accessible manner. Since this function is considered a boundary diagnostic, it is only available from the ANALYSIS S/W COMMANDS menu.

The **orthogonality** function is a measure of the minimum angle between the line formed by a point on the structured boundary and the first point on the interior of the associated block and the surface of the boundary. The maximum value or perfect orthogonality is 90 degrees. The minimum value is 0 degrees. Normals to the boundary are rendered for further directional clarification. Orthogonality values at individual points on the boundary can be obtained by enabling the hiliter.

16.3 Display Attributes

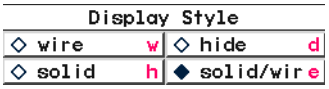
The current Display Style for the diagnostic function is shown on the menu button next to the Display label.





The manner in which the diagnostic function is rendered (for example, colored grid lines versus solid fill) and the values of the colors used for the rendering are controlled by the DISPLAY menu. The default display style is **Solid+Wire** which draws the diagnostic measure by filling in each cell with a color and outlining each cell in the background color. Each diagnostic measure saves its last display attributes and subsequent rendering of that diagnostic function will be initially rendered using the saved settings.

16.3.1 Display Style

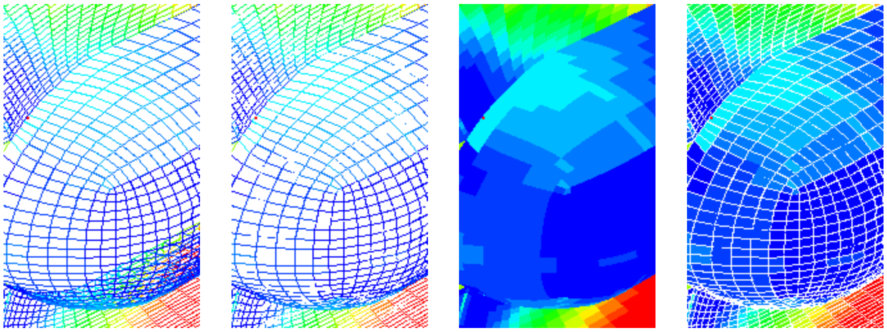


Display Style

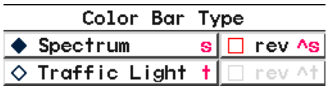
◇ wire w | ◇ hide d

◇ solid h | ◆ solid/wir e

The radio buttons **wire**, **hide**, **solid**, and **solid/wir** under the label Display Style indicate whether the diagnostic function is to be rendered in the Display window by colored grid lines (left), colored grid lines with hidden lines removed (2nd from left), solid shading of filled grid cells (2nd from right), or combined solid shading with wireframe (right).



16.3.2 Color Bar Type

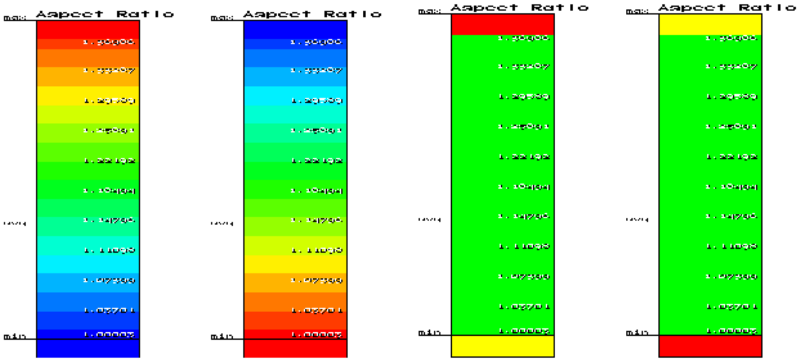


Color Bar Type

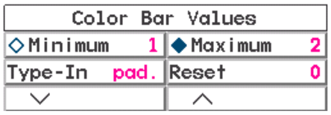
◆ Spectrum s | □ rev ^s

◇ Traffic Light t | □ rev ^t

The color bar type indicates the palette from which the colors used to render the diagnostic function are chosen. The two palette types are indicated by the radio buttons **Spectrum** and **Traffic Light** labeled Color Bar Type. The **Spectrum** color bar (below left) gives a “rainbow” effect, with colors varying from blue to red. The **Traffic Light** color bar (below 2nd from right) gives a “good-bad” effect, with in-range, above-range, and below-range cells rendered in green, red, and yellow, respectively. The radio buttons **rev** for each color bar type allow you to reverse the order of colors in the color bar. For example, the **Spectrum** color bar uses by default blue for the low values and red for the high values. If the **rev** button is toggled on, red will be used for the low values and blue for the high values (below 2nd from left). The reversed version of the traffic light color bar is shown on the right below.

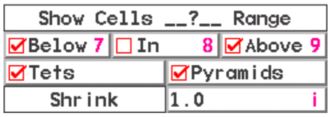


16.3.3 Color Bar Values



The buttons following this label allow you to change the range of diagnostic function values represented by the color bar. By default, when a diagnostic function is first rendered, the color bar values are computed to cover the entire range of the current diagnostic function's values so that all cells are in range. Those minimum and maximum color bar values are then saved and re-used each time that function is examined. You may change either the minimum or maximum color bar value (depending on which of the **Minimum** or **Maximum** radio buttons is selected) by entering the value via the keyboard (**Type-In**), by re-computing it according to the current function value range for the currently examined entities (**Reset**), or by decreasing or increasing by 10 percent of the current color bar range using the arrow buttons.

16.3.4 Show Cells

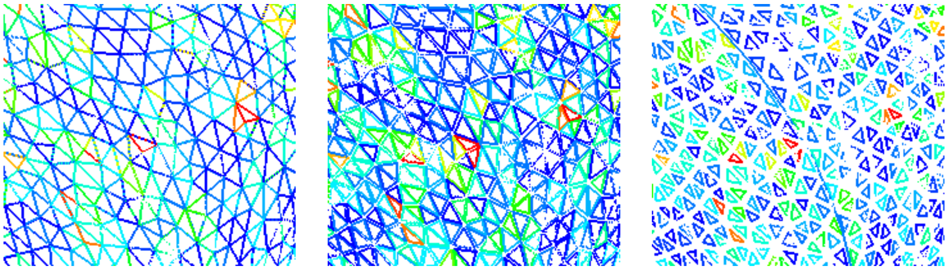


The **Below**, **In**, and **Above** buttons labeled Show Cells __?__ Range allow you to specify which cells are to be rendered in the Display window according to the cell's diagnostic function value: below range, in range, or above range with respect to the color bar values, respectively. For connectors and domains the default condition is to show all cells. For blocks the default condition is to show everything except for in range cells (otherwise every cell in the volume grid would be displayed).

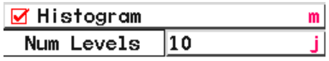
The toggle buttons **Tets** and **Pyramids** let you set whether or not tetrahedral and pyramid cells, respectively, will be shown in the Display window when examining hybrid unstructured blocks.

Pressing the button labeled Shrink lets you enter a real number between 0 and 1 (inclusive) that determines how much the grid cells being rendered in the Display window are shrunk about their centroids. The shrink factor has no effect whatsoever on the grid or the values of the diagnostic functions. It simply provides an interesting visual effect that helps you distinguish a cell from its neighbors.

Triangular grid cells colored by aspect ratio and rendered with wireframe style are shown below with shrink factors of 1.0, 0.8, and 0.5, respectively. Entering a value of zero (0.0) causes the cells to rendered by only a dot at their centroids. These dots have a fixed size regardless of model zoom level, and therefore this shrink factor value may aid in locating extremely small cells in the Display window.



16.3.5 Histogram

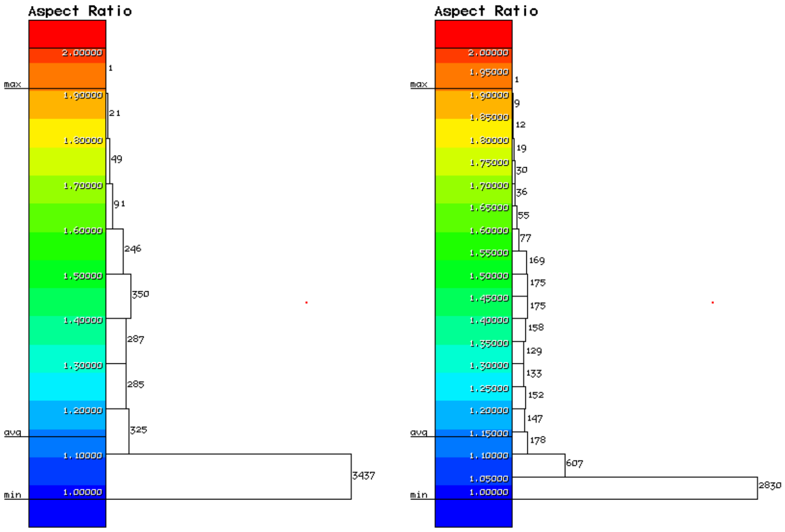


☑ Histogram m

Num Levels 10 j

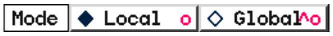
Enabling this toggle button adds a histogram to the color bar in the Display window, with horizontal bars representing the number of cells with diagnostic function values in the color bar range that is graphically cut by the bar.

Num Levels - By default histograms are drawn with 10 horizontal bars (levels). You may enter a new number of levels (bars) with which the histogram is to be drawn. Histograms with 10 (left) and 20 (right) levels are shown below.



16.4 Mode

The examine mode refers to whether the Examine command is being used in a very low-level mode (**Local**) to examine the variation in a grid's diagnostic functions or whether it is being used in a very high level mode (**Global**) to assess the quality of the entire grid (or large portions of it).



Mode ◆ Local o ◇ Global^o

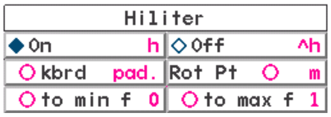
Mode **Local** - Blackboard text corresponds to focus entity only.

Mode **Global** - Blackboard text corresponds to all selected entities.

16.5 Hiliter

The hiliter consists of a small pink glyph in the Display window corresponding to a specific location on an entity. You can move the hiliter by using the mouse with the ***right mouse*** button pressed. Volume grid hiliters may also be moved using the VCR buttons and arrow hot keys. The hiliter is not available for unstructured surface grids. Information about the hilited point is listed in the Blackboard.

16.5.1 Connectors



For connectors the hiliter is only the open circle glyph.

On - Enables the hiliter for curve and surface entities.

Off - Disables the hiliter for curve and surface entities.

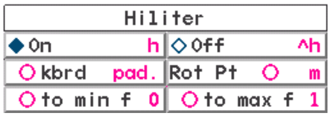
❍ **kbrd** - Move the hiliter to the typed-in point.

Rot Pt ❍ - Move the rotation point to the current hilited point.

❍ **to min f** - Move the hiliter to the location of the minimum value of the current diagnostic function (if appropriate).

❍ **to max f** - Move the hiliter to the location of the maximum value of the current diagnostic function (if appropriate).

16.5.2 Domains



For structured surface grids and database surfaces two isolines, lines of constant computational (grid) or parametric (database) coordinate, are drawn through the circle hiliter glyph. The glyph used for the hiliter depends on whether the current diagnostic function is point or cell based. Point based diagnostic functions utilize an open circle (❍). Cell based diagnostic functions use a glyph that outlines the cell.

The hiliter controls for domains are similar to those for connectors (Reference 16.5.1).

16.5.3 Blocks

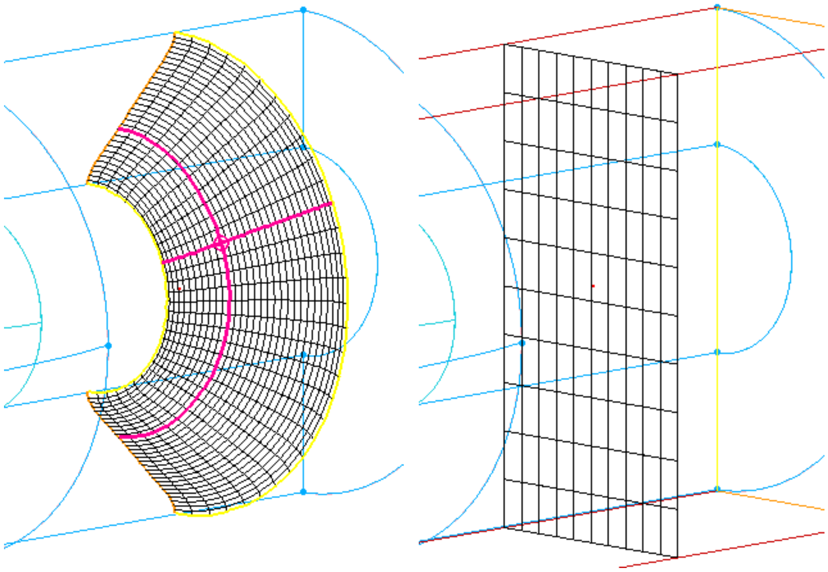
Two types of hiliter are available for volume grids: topological (**Topo**) and physical (**Phy**).

For structured volume grids, the topological hiliter displays a constant computational coordinate plane (ξ, η, or ζ) on which the hiliter isolines and circle glyph are drawn. The glyph used for the topological hiliter depends on whether the current diagnostic function is point or cell based. Point based diagnostic functions utilize an open circle (❍). Cell based diagnostic functions use a glyph that outlines the cell. If a diagnostic function has been selected, the topological hiliter will be directly rendered by color variation for that function.

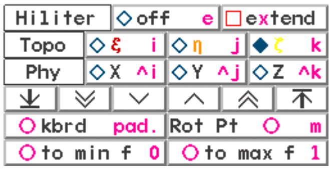
For a prism block, the topological hiliter displays planes of triangular surface grids in the extrusion direction.

For both structured, unstructured, and hybrid volume grids, the physical hiliter allows you to position a constant cartesian coordinate plane (x, y, or z) at which a cut will be made through the volume grid. There are no hiliter isolines or circular point or cell glyphs rendered on a physical hiliter. Also, no diagnostic function data is rendered directly on a physical hiliter, but instead is rendered on the various cuts that can be made at the hiliter location.

Topological (left) and physical (right) hiliters are shown below for a structured grid. The topological hiliter cuts through the grid at ξ=54 while the physical hiliter cuts the grid at x=45.0.



Commands that allow you to manipulate the hiliter for blocks are described below.



Off - Pressing this radio button turns off the hiliter.

extend - Toggling on this button causes a topological hiliter to be extended beyond the **Local** block into all appropriate blocks currently picked for **Examine.**

ξ, η, ζ - Pressing these buttons enables the hiliter on ξ, η, and ζ constant planes of a structured block, respectively.

X, Y, Z - Pressing these buttons enables the hiliter on x,y, and z constant planes of block, respectively.

❍ **kbrd** - Move the hiliter to the typed-in point.

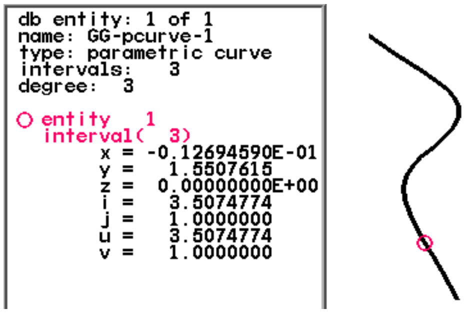
X kbrd - Move the unstructured block hiliter by entering a the x, y, or z constant plane value.

Rot Pt ❍ - Move the rotation point to the current hilited point.

❍ **to min** - Move the hiliter to the location of the minimum value of the current diagnostic function (if appropriate).

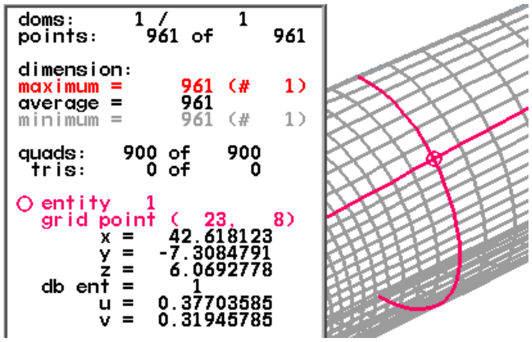
❍ **to max** - Move the hiliter to the location of the maximum value of the current diagnostic function (if appropriate).

An example of the hiliter for a curve entity with Blackboard information is shown below.



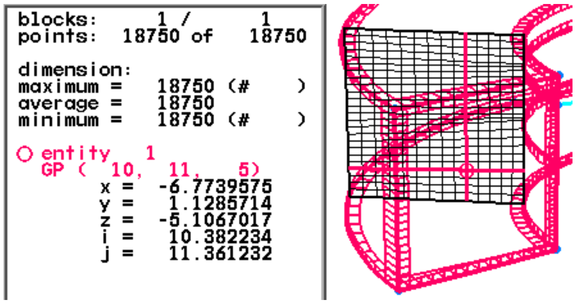
```
db entity: 1 of 1
name: GG-pcurve-1
type: parametric curve
intervals:   3
degree:  3

○ entity   1
  interval(  3)
        x = -0.12694590E-01
        y =   1.5507615
        z =  0.00000000E+00
        i =   3.5074774
        j =   1.0000000
        u =   3.5074774
        v =   1.0000000
```

An example of the hiliter for a surface entity with Blackboard information is shown below.



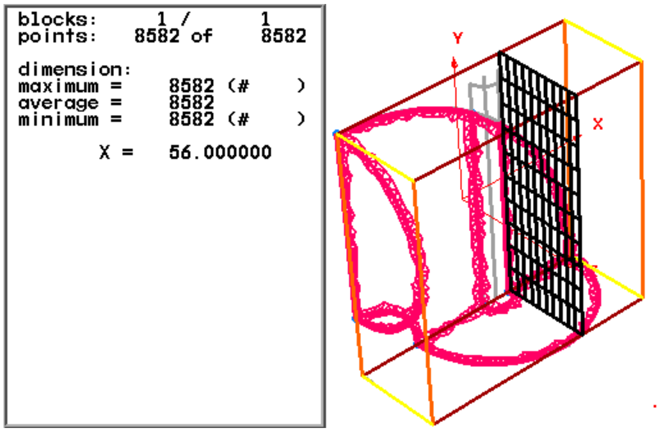
```
doms:       1 /        1
points:     961 of       961

dimension:
maximum =       961 (#    1)
average =       961
minimum =       961 (#    1)

quads:    900 of    900
 tris:      0 of      0

○ entity   1
  grid point (  23,    8)
        x =   42.618123
        y =  -7.3084791
        z =   6.0692778
  db ent =      1
        u =  0.37703585
        v =  0.31945785
```

An example of the hiliter for a structured block with Blackboard information is shown below.



```
blocks:      1 /        1
points:  18750 of   18750

dimension:
maximum =   18750 (#     )
average =   18750
minimum =   18750 (#     )

○ entity   1
  GP (  10,  11,   5)
        x =  -6.7739575
        y =   1.1285714
        z =  -5.1067017
        i =   10.382234
        j =   11.361232
```

An example of the hiliter for an unstructured block with Blackboard information is shown below.

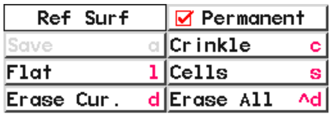


```
blocks:      1 /       1
points:   8582 of    8582

dimension:
maximum =    8582 (#    )
average =    8582
minimum =    8582 (#    )

       X =   56.000000
```

16.6 Reference Surfaces

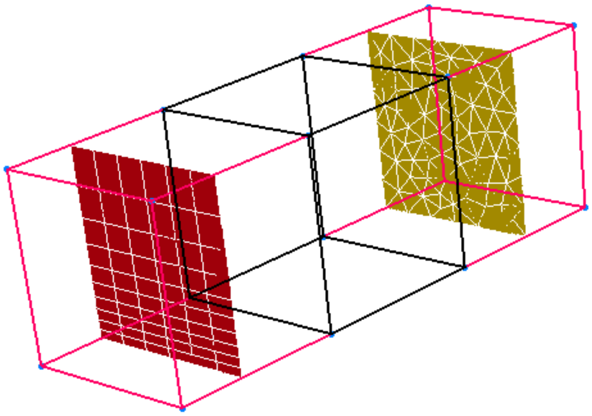
A reference surface is the result of cutting a volume grid with the hiliter. Reference surfaces are always rendered to reflect the current diagnostic function drawn with the current display attributes. You may create multiple reference surfaces. In local mode, the physical hiliter only cuts the focus block when creating a reference surface. In global mode, the physical hiliter cuts all blocks when creating a reference surface.

Commands for controlling reference surfaces are described below and illustrated for a 3 block grid consisting of a structured block, a hybrid unstructured block (contains both tetrahedra and pyramids), and a prism block.

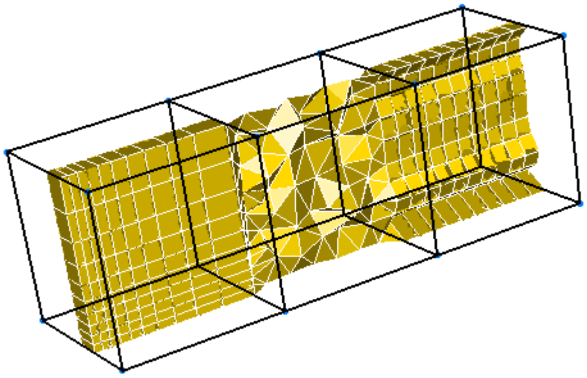


Permanent - This toggle button sets the state of the reference surfaces that you create. When the permanent state is toggled-on, each new reference surface you create persists until you exit the **Examine** command. Otherwise, if the permanent state is toggled-off, each new reference surface replaces the previous one.

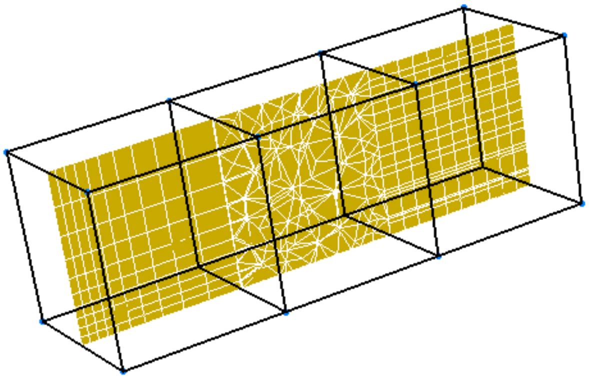
Save - This command saves the current structured block topological hiliter location as a reference surface. The illustration below shows **Save**d reference surfaces through a structured block (left, red) and a prism block (yellow, right). The center block is unstructured; therefore, no computational surface is available.



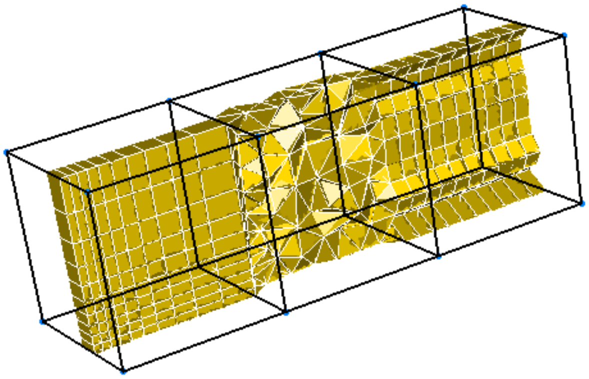
Crinkle - Gridgen finds the cells in the volume mesh that are intersected by a x, y, or z physical hiliter. For every touched tetrahedral cell that has only three of its four faces cut by the plane, the fourth un-cut face becomes part of the crinkle surface. The selection of the uncut face is also biased based on which half of the volume grids extent box the hiliter is in. The resulting reference surface is not flat. Rather, it has the appearance of a crinkled sheet of aluminum foil, hence the name. An example of a crinkle surface is shown below for a structured, unstructured, and prism block (left to right).



Flat - Gridgen calculates a flat surface through a volume grid at the location of a x, y, or z physical hiliter. Gridgen creates the reference surface from the actual intersection of the hiliter with every cell in the volume mesh. The resulting surface is planar as shown in the example below.



Cells - Gridgen calculates a complete cell surface through a volume grid at the location of a x, y, or z physical hiliter. The reference surface is composed of every cell touched by the hiliter. Not only is the reference surface not planar, it has thickness as shown below.



Erase Cur. - To erase only one of several reference surfaces currently displayed, you would position the hiliter over that surface and press this button.

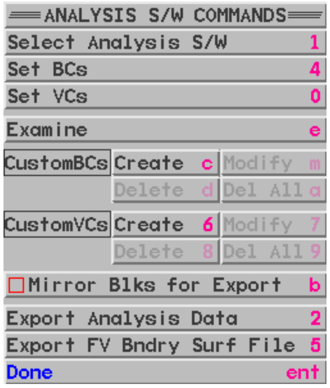
Erase All - Pressing this button erases all of the reference surfaces.

17. Analysis Software Interfaces

A powerful feature of Gridgen is its ability to customize the grid for a particular analysis software package. When you choose one of the supported analysis codes Gridgen will:

- use the analysis software's nomenclature
- enforce the analysis software's grid topology requirements
- allow you to apply the analysis software's boundary and volume conditions
- export boundary condition, volume condition, and grid data in files that may be used directly in the analysis software

The following sections discuss each of these capabilities in more detail following the order of the ANALYSIS S/W COMMANDS menu as shown below.



17.1 Select Analysis Software

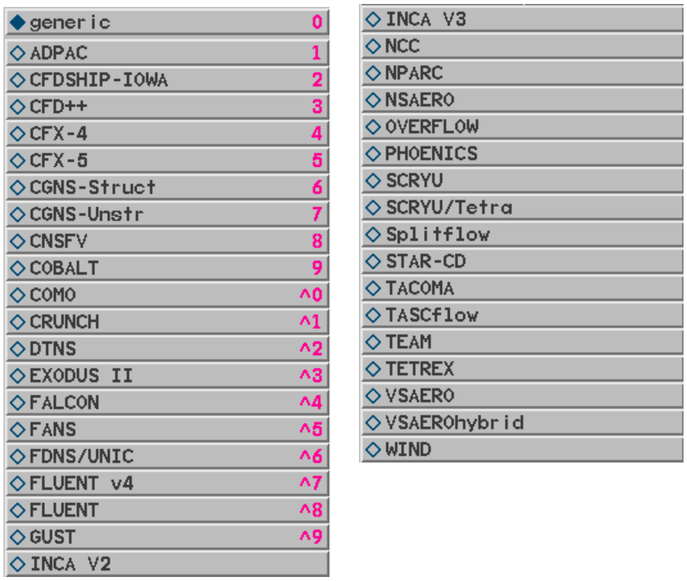
Selecting your analysis software (ASW) will usually be your first step in creating a grid. By choosing your ASW first, you ensure that Gridgen will create a grid that meets your ASW's requirements. You can change analysis software (including 2D and 3D) at any time, but all boundary condition information you have set will be lost. Depending on what types of cells the new analysis software selection supports, some domain and/or block definitions may also be lost.

You first specify whether your grid is 2D or 3D by using the first row of radio buttons.

2D - The grid will consist of computationally two-dimensional blocks comprised of surface cells (quadrilaterals and triangles) only. Structured 2D blocks will have dimensions I x J x 1. Note that to set BCs for a 2D grid, or to specify the index orientations for a 2D grid, you need to create a 2D block.

3D - The grid will consist of computationally three-dimensional blocks comprised of volume cells (hexahedra, pyramids, prisms and tetrahedra). Structured 3D blocks will have dimensions I x J x K.

Gridgen provides support for an extensive list of analysis software packages as shown in the following figure and list.



generic is Gridgen's general purpose CFD-style solver facsimile. You may use it to export grid and boundary condition data and then translate to your own format using the descriptions in Section 18.10. This is the default ASW.

ADPAC is an end-user proprietary CFD solver for structured, multi-block grids.

CFDSHIP-IOWA is a structured grid CFD code from the Iowa Institute of Hydraulic Research at the University of Iowa.

CFD++ (from Metacomp Technologies) uses a unified-grid treatment for all grid and cell topologies including patched-nonaligned and overset grids.

CFX-4 (from AEA Technology) supports structured, multi-block grids only.

CFX-5 (from AEA Technology) supports structured, unstructured and hybrid grids.

CGNS-Struct is the CFD General Notation System (CGNS) standard for transferring structured grid information.

CGNS-Unstr is the CFD General Notation System (CGNS) standard for transferring unstructured grid information.

CNSFV (from NASA Ames Research Center (Reference 11)) supports structured, multi-block grids only.

Cobalt (from the United States Air Force Research Laboratory) supports unstructured grids only.

COMO is an end-user proprietary CFD solver that supports hybrids grids.

CRUNCH (from CRAFT Tech) supports unstructured hybrid grids only.

DTNS (from NSWC Carderock Division) supports structured, multi-block grids only.

EXODUS II is a finite element data model from Sandia National Laboratories (Reference 18).

FALCON is an end-user proprietary CFD solver for structured, multi-block grids.

FANS (from NSWC Carderock Division) supports structured, multi-block grids only.

FDNS/UNIC (from Engineering Sciences, Inc.) supports structured, multi-block grids only.

FLUENT v4 (from Fluent, Inc.) supports structured, multi-block grids only.

FLUENT (from Fluent, Inc.) supports hybrid grids.

GASP (from AeroSoft, Inc.) supports structured, multi-blocks grids only.

GUST (from AeroSoft, Inc.) supports hybrid grids.

INCA v2 (from Amtec Engineering) supports structured multi-block grids only.

INCA v3 (from Amtec Engineering) supports hybrid grids.

NCC (from NASA Glen Research Center) supports hybrid grids.

NPARC versions 2 and 3 (from the NPARC Alliance (Reference 5)) supports structured, multi-block grids only.

OVERFLOW (from NASA Ames Research Center (Reference 3)) supports structured, overlapping, multi-block grids only.

PHOENICS from Concentration, Heat & Momentum Limited supports structured, multiblock grids only.

SCRYU (from Software Cradle, Ltd.) supports structured, multi-block grids only.

SCRYU/Tetra (from Software Cradle, Ltd.) supports unstructured tetrahedral grids only.

Splitflow (from Lockheed Martin, Corp.) supports 3D unstructured Cartesian Hex grids.

STAR-CD (from Computational Dynamics) supports hybrid grids.

TACOMA is an end-user proprietary CFD solver for structured, multi-block grids.

TASCflow (from AEA Technology Engineering Software) supports structured, multi-block grids only.

TEAM (from USAF Wright Laboratories (Reference 8, Reference 14)) supports structured, multi-block grids only.

TETREX (from Tetra Research Corp.) supports hybrid grids.

VSAERO (from Analytical Methods, Inc.) supports surface grids only.

VSAEROhybrid (from Analytical Methods, Inc.) supports unstructured and structured surface grids only.

WIND (from the NPARC Alliance) supports structured, multi-block grids only.

After choosing the ASW, its dimension, and pressing the **Done** button, you must confirm your selections since any boundary conditions that have already been set will be cleared when the analysis software is changed. Furthermore, all 3D blocks will be deleted when a 2D analysis software package is selected. Likewise, if structured blocks exist in the model and an ASW that supports only unstructured elements is chosen, the structured topology will be deleted as necessary.

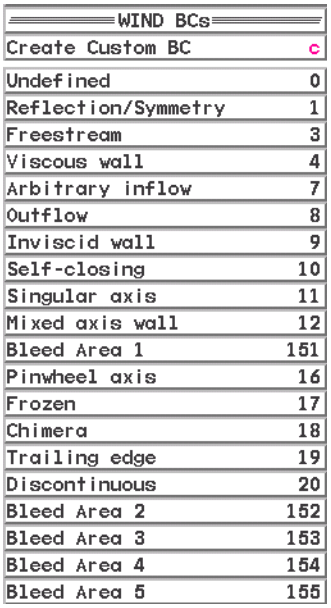
Please note: Gridgen's support for any particular analysis software package is not an endorsement of that package. Furthermore, selection of a suitable analysis package is the sole responsibility of the user. You are strongly advised to consult the documentation for each analysis software package prior to using it with Gridgen.

17.2 Set BCs

The **Set BCs** command allows you to specify that a particular boundary condition (BC) is to be applied to one or more block boundaries. It is important to note that BCs can only be applied to blocks. In 2D, the BC regions (the block boundaries) are connectors. In 3D, the BC regions (the block boundaries) are domains. Note that only one boundary condition may be applied to a particular region. For regions containing domains, block deletion will cause each domain within the block to be tagged with that region's BC. When a new block is created, the new region containing that domain, or domains, will be assigned the tagged BC.

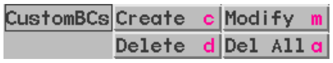
There is no method for manually setting an inter-block connection; inter-block connections are set automatically by Gridgen where adjacent blocks share a common domain (3D) or connector (2D). Boundary conditions, including inter-block connections, may be overwritten by simply applying another BC. This allows for the specification of infinitely thin walls, for example. If the Browser text is blank where an inter-block connection is expected, check the type of connection (relative orientation of block indices) with what is supported by the analysis software. Gridgen is programmed to eliminate connection types not supported by the current analysis software.

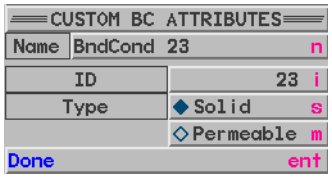
The list of available boundary conditions is dependent on the chosen solver (Section 17.1). For example, boundary conditions for the WIND solver are shown below.



WIND BCs	
Create Custom BC	c
Undefined	0
Reflection/Symmetry	1
Freestream	3
Viscous wall	4
Arbitrary inflow	7
Outflow	8
Inviscid wall	9
Self-closing	10
Singular axis	11
Mixed axis wall	12
Bleed Area 1	151
Pinwheel axis	16
Frozen	17
Chimera	18
Trailing edge	19
Discontinuous	20
Bleed Area 2	152
Bleed Area 3	153
Bleed Area 4	154
Bleed Area 5	155

However, if the boundary condition you wish to apply is not available, you may press the **Create** button and define your own boundary condition as shown below (see Section 17.5).





CUSTOM BC ATTRIBUTES

Name	BndCond 23	n
ID	23	i
Type	◆ Solid	s
	◇ Permeable	m
Done		ent

Name **Text** - The button labeled Name shows the default name for the new boundary condition. If you press the button, you may type in your own name which will then appear on the button.

ID **Number** - Pressing the button labeled ID lets you type in the numeric code to be used for the new boundary condition. This numeric value is shown on the button.

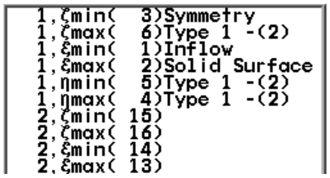
Type - This setting is used primarily for FIELDVIEW export (FV-UNS). The two options available are described below.

Type **Solid** - Enabling this toggle button indicates the boundary condition you are creating is a solid boundary. The boundary condition type is not used by all analysis software packages.

Type **Permeable** - Enabling this toggle button indicates the boundary condition you are creating is a permeable boundary. The boundary condition type is not used by all analysis software packages.

17.2.1 Structured Grids

Upon invoking the **Set BCs** command, you select BC regions from the Browser window. For example, the Browser text may look like:

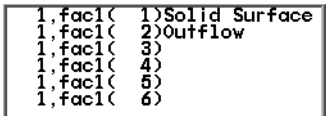


```
1,ζmin(  3)Symmetry
1,ζmax(  6)Type 1 -(2)
1,ξmin(  1)Inflow
1,ξmax(  2)Solid Surface
1,ηmin(  5)Type 1 -(2)
1,ηmax(  4)Type 1 -(2)
2,ζmin( 15)
2,ζmax( 16)
2,ξmin( 14)
2,ξmax( 13)
```

The numeral in the first column (1 and 2 in this example) represents the BC region's block number. Each region in the block is then listed according to the type of computational index that is constant on it. The number in the parentheses next to the computational index is the entity number of the boundary condition region: in 2D it is the connector number, in 3D it is the domain number. Finally, the text to the right of the entity number is the boundary condition type applied to that region. Gridgen automatically determines the inter-block connections and they are denoted by the text `Type 1` in the Browser. The data that follows the inter-block connection text indicates the block number to which the region connects. After selecting the regions BCs will be applied to, you then choose a specific BC from a menu that is customized for the current ASW.

17.2.2 Unstructured Grids

Upon invoking the **Set BCs** command, you select BC regions from the Browser Window. For example, the Browser text may look like:



```
1,fac1(  1)Solid Surface
1,fac1(  2)Outflow
1,fac1(  3)
1,fac1(  4)
1,fac1(  5)
1,fac1(  6)
```

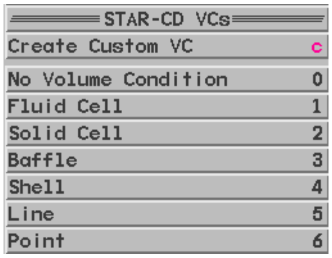
The numeral in the first column is the block number which is followed by the string "`facN`" where `N` is the face number in the block. The number in the parentheses next to the face number is the entity number of the boundary condition region: in 2D it is the connector number, in 3D it is the domain number. Finally, the text to the right of the entity number is the boundary condition type applied to that region. Gridgen automatically determines the inter-block connections and they are denoted by the text `Type 1` in the Browser. The data that follows the inter-block connection text indicates the block number to which the region connects. After selecting the regions BCs will be applied to, you then choose a specific BC from a menu that is customized for the current ASW.

17.3 Set VCs

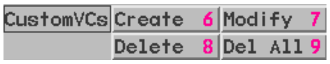
The **Set VCs** command allows you to apply a volume condition (VC) to a block. A VC represents a flow solver condition assigned only to the interior cells of a block. Each block may only have one applied VC.

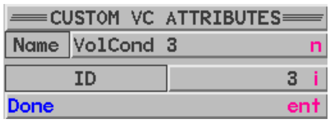
For export, VC's can be used to export blocks singly by setting each block's VC to **No Volume Condition** or they can be used to export merged blocks if all blocks are of the same VC (except **No Volume Condition**).

The list of available volume conditions is dependent on the chosen solver (Section 17.1). For example, volume conditions for the **Star-CD** solver are shown below.



However, if the volume condition you wish to apply is not available, you may press the **Create** button and define your own volume condition as shown below (see Section 17.6).



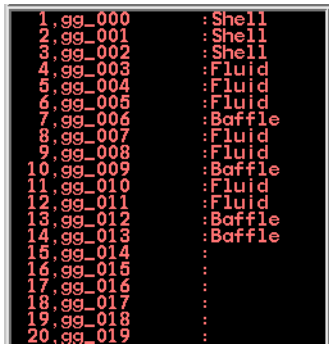


Name **Text** - The button labeled Name shows the default name for the new volume condition. If you press the button, you may type in your own name which will then appear on the button.

ID **Number** - Pressing the button labeled ID lets you type in the numeric code to be used for the new volume condition. This numeric value is shown on the button.

17.3.1 Applying VCs

Upon invoking the **Set VCs** command, you select blocks from the Browser window. For example, the Browser text may look like:



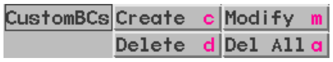
```
 1,gg_000        :Shell
 2,gg_001        :Shell
 3,gg_002        :Shell
 4,gg_003        :Fluid
 5,gg_004        :Fluid
 6,gg_005        :Fluid
 7,gg_006        :Baffle
 8,gg_007        :Fluid
 9,gg_008        :Fluid
10,gg_009        :Baffle
11,gg_010        :Fluid
12,gg_011        :Fluid
13,gg_012        :Baffle
14,gg_013        :Baffle
15,gg_014        :
16,gg_015        :
17,gg_016        :
18,gg_017        :
19,gg_018        :
20,gg_019        :
```

The numeral in the first column (1 through 20 in this example) represents the block number. Each region in the block is then listed according to the name assigned to the block either by Gridgen or the user. The text to the right of the colon is the volume condition type applied to that region. After selecting the regions, VC's will be applied to, you then choose a specific VC from a menu that is customized for the current ASW. For any newly created or imported block, Gridgen will automatically assign the default VC of **FLUID**.

17.4 Examine

It is possible to obtain diagnostic information for block boundaries regarding wall-adjacent cell spacing and orthogonality. A detailed description of the **Examine** command and the functions specific to boundaries can be found in Section 16.

17.5 Custom BCs, Creating, Editing & Deletion



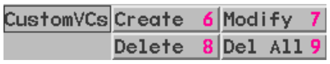
Create - Pressing this button lets you create a new custom boundary condition as described in Section 17.2.

Modify - Allows a selected custom boundary condition to be changed to a new Name, ID, or Type.

Delete - Removes selected custom boundary condition from the list of available boundary conditions for the current analysis software. Any BC regions already specified with a deleted custom boundary condition return to the unspecified state.

Del All - Removes all the custom boundary conditions from the list of available boundary conditions for the current analysis software. Any BC regions already specified with a deleted custom boundary condition return to the unspecified state.

17.6 Custom VCs, Creating, Editing & Deletion



Create - Pressing this button lets you create a new custom volume condition as described in Section 17.3.

Modify - Allows a selected custom volume condition to be changed to a new Name or ID.

Delete - Removes selected custom volume condition from the list of available volume conditions for the current analysis software. Any VC regions already specified with a deleted custom volume condition return to the unspecified state.

Del All - Removes all the custom volume conditions from the list of available volume conditions for the current analysis software. Any VC regions already specified with a deleted custom volume condition return to the unspecified state.

17.7 Mirror Blocks for Export

The **Mirror Blks for Export** toggle, when enabled, instructs Gridgen to mirror selected blocks only on export for your analysis software. This setting, the selected blocks for mirroring, and the chosen mirror plane only persist while the user is in the ANALYSIS S/W COMMANDS menu. Once the menu is exited, or the **Mirror Blks for Export** toggle is turned off, all settings related to this command are cleared from memory.

Turning on this toggle immediately places you in the block Browser to select the blocks you wish to mirror on export. Once the blocks are selected, you are placed in a final menu used to determine the mirror plane. See Section 14.3.5 for more details on the mirror plane selection.

17.8 Export Analysis Data, Grid, and BCs

These commands allow you to export analysis software data to a file or files including boundary condition, volume condition, and grid data. Analysis software requiring a boundary condition file and a grid file will be supported by the command pair: **Export Analysis BCs** and **Export Analysis Grid**. Analysis software requiring exactly one comprehensive data file or more than the customary two grid and BC files will be supported by a single command: **Export Analysis Data**.

See Section 2.5 for additional information regarding use of the file Browser. These commands may also be invoked via a button or buttons in the INPUT/OUTPUT COMMANDS menu.

The following sections describe the analysis data export in the context of specific analysis software packages.

17.8.1 generic

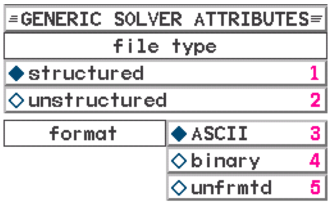
Gridgen's generic ASW is not really a software package at all, but a facsimile of one that is included in Gridgen to give you the ability to create and export grids and boundary conditions in a generic format. Using the file descriptions in Section 18, you may then write your own translator to convert from the generic format to your own format. Gridgen's generic solver has been patterned after a computational fluid dynamics (CFD) code and supports hybrid grids.

The generic ASW supports both structured and unstructured grid output formats (with the exception of 2D unstructured). The structured grid output maintains the computational coordinates of the grid, whereas, the unstructured grid output format breaks the entire grid into a list of points and element connectivity.

The generic structured format allows you to export separate grid and boundary condition files. The format of the generic structured BC file may be found in Section 18.8. The generic structured grid file only supports quadrilateral and hexahedral grid cells and is a structured, multi-block grid with Gridgen style, ASCII, binary or unformatted format, and double precision as described in Section 18.10.

The generic unstructured format includes the grid and the boundary conditions as named regions exported in FIELDVIEW Unstructured format. The unstructured generic file supports hexahedral and tetrahedral (volumetric) elements only. See Section 18.7 for more information.

If your grid contains any unstructured blocks, the unstructured format is automatically chosen as the generic format. If all blocks contain structured grids, you are given the option to export the grid as either structured or unstructured and choose the format, as shown below.



17.8.2 ADPAC

ADPAC supports structured, multi-block grids only. Gridgen exports two files for ADPAC; the boundary data (.boundata) file, and the grid (.mesh) file. Any physical values, such as pressure and temperature, required by the boundary data file are set to zero. It is up to you to edit this file and supply the correct values prior to running ADPAC. ADPAC's grid file is a Fortran unformatted, double precision, PLOT3D, multigrid file. Only 3D grids supported.

17.8.3 CFDSHIP-Iowa

CFDSHIP-Iowa supports 2D and 3D multi-block structured grids.

17.8.4 CFD++

CFD++ supports all cell types that can be created in Gridgen. Both 2D and 3D grids are supported. Multiple data files are written in big-endian binary regardless of platform.

17.8.5 CFX-4

CFX-4 supports structured 2D and 3D grids. Gridgen creates an input file for CFX-4 containing the global data, patch data, glue data, and grid points. Gridgen does not support CFX-4's volumetric conditions.

The following CFX-4 physical types may be used in the creation of custom bc's: SYMMET, INLET, OUTLET, PRESS, and WALL. If a custom bc is created that begins with a physical type keyword, then the physical type for that bc will be set to the keyword type. If no keyword is found at the beginning of a custom bc name, the default physical type USER2D will be assigned. For example, if a customer creates a bc called OUTLET_CHANNEL, the bc patch will written out as OUTLET_CHANNEL of physical type OUTLET.

Two-dimensional grids are assumed to lie in the x-y plane. The grid is extruded by 1.0 units in the z direction to create a 3D grid of one cell thickness.

17.8.6 CFX-5

Gridgen exports fully hybrid grids containing hexahedra, tetrahedra, prisms, and pyramids.

User-defined VC's are allowed. Gridgen includes VC names in CFX-5 export.

17.8.7 CGNS-Struct

This ASW selection supports 2D and 3D structured only.

User-defined VC's are allowed. Gridgen includes VC names in CGNS-Struct export.

17.8.8 CGNS-Unstr

All cell types are supported for 2D and 3D grids.

User-defined VC's are allowed. Gridgen includes VC names in CGNS-Unstr export.

17.8.9 CNSFV

Although CNSFV permits only a single boundary condition per block face, Gridgen does not restrict the grid construction in this manner. Only the `ZONAL` and `BOUNDCND` namelists are written by Gridgen.

CNSFV's grid file is a PLOT3D, multi-grid, Fortran unformatted dataset. Only 3D grids supported.

17.8.10 Cobalt

Gridgen exports structured and unstructured volume grids to Cobalt's grid file. The volume grid from all of the blocks are agglomerated into one zone for output to the grid file. Cobalt's boundary condition file contains place holders for all of the assigned boundary condition patches. However, you must modify the boundary condition file prior to use in Cobalt. Note that while you may set periodic BCs on certain grid regions (patches), Gridgen makes no effort to ensure the grid faces are indeed physically periodic. Both 2D and 3D grids supported.

17.8.11 COMO

COMO is a proprietary file format based on the PATRAN Neutral File. Gridgen exports all cell types for both 2D and 3D grids.

17.8.12 CRUNCH

Crunch export supports only 3D grids of all cell types. That means you may create a grid containing both structured and unstructured blocks. Gridgen exports a generic BC and a double precision FV-UNS file.

17.8.13 DTNS

Only the formatted boundary condition and inter-block connection data is written by Gridgen.

A Fortran unformatted DTNS-specific grid file is created for each block. Only 3D structured grids supported.

17.8.14 Exodus II

All cell types are supported for 2D and 3D grids.

17.8.15 FALCON

FALCON is a proprietary format.

17.8.16 FANS

Only the formatted boundary condition and inter-block connection data is written by Gridgen.

FANS' volume grid file is multi-block, Gridgen-style, Fortran unformatted, and double precision. Only 3D structured grids supported.

17.8.17 FDNS/UNIC

FDNS/UNIC supports 2D and 3D structured, multi-block grids only. Two-dimensional grids must be created in the `x-y` plane. Place holders for each applied boundary condition are written to the boundary condition file. However, you must modify the input variables before running FDNS.

17.8.18 FLUENT v4

FLUENT v4 requires that your multiple block grid fit together into one large block. If Gridgen cannot assemble your multiple block grid into a single block grid when exporting data for FLUENT v4, you will receive an error message. Use the **Set BCs** command to check for inter-block connections that are not set properly.

FLUENT v4 does not allow poles (line singularities). Instead, use a small cylinder of finite radius. Gridgen does not detect and warn of the existence of poles during export. Both 2D and 3D structured grids are supported.

17.8.19 FLUENT

FLUENT export supports 2D and 3D grids of all cell types. Export format is for the most current version of FLUENT. Gridgen does not write all of the data necessary for Periodic and Periodic Shadow boundary conditions to work properly. You must make modifications to these BCs if you use them.

Two dimensional grids are assumed to lie in the `x-y` plane.

Volume condition export is supported for FLUENT. Blocks with common VC's are automatically merged in the AS/W file with the VC used as the zone name. To have separate blocks written out with assigned VC, set a different VC to each block to avoid automatic merge. Alternatively, setting block VC's to **No Volume Condition** will cause the blocks to be exported separately under their block names.

Wherever Gridgen detects a pole (line singularity) in a structured grid, it will convert the cell from a hex (quad) to a wedge (triangle) when exporting it for FLUENT. Whenever Gridgen detects a completely singular face (point singularity) in a structured grid, it will convert hexes to pyramids.

17.8.20 GASP

GASP export supports 2D and 3D structured grids only. Three files are written by Gridgen: GASP's .grd,

.inp, and .sln files.

Two-dimensional grids are assumed to lie in the `x-y` plane. Since GASP expects 2D grids to be one cell thick Gridgen extrudes the grid by 1.0 units in the `z` direction.

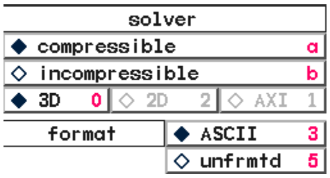
GASP export is not supported under MS Windows or Linux.

17.8.21 GUST

GUST supports 3D unstructured grids comprised of both hexahedral and tetrahedral cells. That means you may use a grid comprised of both structured and unstructured blocks in GUST. The volume grids from each of the blocks are agglomerated into a single block for output. Gridgen's **Set BCs** command is used to assign GUST groups for later application of boundary conditions in GUST.

17.8.22 INCA v2

INCA v2 export supports 2D and 3D structured grids only. When you export data for INCA Version 2, you must first set several parameters specifying exactly how you would prefer the file to be formatted. This menu is shown and described below.



solver
◆ compressible a
◇ incompressible b
◆ 3D 0 ◇ 2D 2 ◇ AXI 1
format ◆ ASCII 3
◇ unfrmtd 5

in/compressible - selects the compressibility of the flow for use with the SOLVERS.ENABLED input in the INCA.SETUP namelist and is used by some of the boundary conditions.

3D/2D/AXI - the flow dimensionality is set by Gridgen's current setting from when the ASW was specified. However, if the blocks are 2D then you're given the option to choose between 2D and axi-symmetric.

format - how the grid file will be formatted, either ASCII or Fortran unformatted.

Then the following INCA namelists are written:

```
RUN.CONTROL
INCA.SETUP
then for each block
      ZONE.SETUP
      BC.PATCH.SETUP
      INTRZN.PATCH.SETUP (if necessary)
UNIFORM.FLOW (initial conditions)
boundary condition specific namelists
NS.GENERAL
CHEM.SETUP
NS.ZONE
```

The flowfield data and other parameters are initialized to zero. Gridgen computes the index ranges for the patches by adding two boundary cells in each computational direction. INCA's grid file is PLOT3D style. Note that even though the boundary conditions are written assuming a two-cell block overlap, only the true grid without overlap is exported by this command.

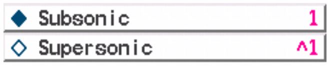
17.8.23 INCA v3

INCA v3 export supports 3D grids of both hexahedral and tetrahedral cells. That means you may create a grid containing both structured and unstructured blocks for use in INCA v3. The volume grid from each of the blocks are agglomerated into one block for output.

INCA v3 export is supported on SGI platforms only under IRIX 6.5.

17.8.24 NCC

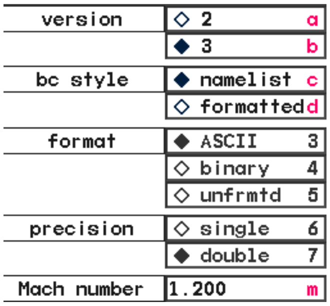
NCC export supports 2D and 3D grids of all cell types with the exception of pyramids. When exporting BC data for NCC you must specify how you would prefer the BC file to be formatted. This menu is shown below, choose either the supersonic or subsonic attribute.



◆ Subsonic 1
◇ Supersonic ^1

17.8.25 NPARC

NPARC export supports 2D and 3D structured grids only. When you export BC data for NPARC you must first set several parameters specifying exactly how you would prefer the BC file to be formatted. This menu is shown and described below.



version	◇ 2	a
	◆ 3	b
bc style	◆ namelist	c
	◇ formatted	d
format	◆ ASCII	3
	◇ binary	4
	◇ unfrmtd	5
precision	◇ single	6
	◆ double	7
Mach number	1.200	m

version 2 - The BC data will be written for NPARC Version 2 which uses type 70 inter-block connection and requires connecting blocks to overlap by one cell. To meet this requirement Gridgen extrapolates each block by one cell in the -J, +J, -K, +K, -L, and +L directions. The indices in the BC file will reflect this extrapolation.

version 3 - The BC data will be written for NPARC Version 3 which uses the Type 77 inter-block connection and does *not* require connecting blocks to overlap.

bc style namelist - Input `nbcseg` is set to 0 so that NPARC's boundary conditions will be written using the `BOUNDS` namelist.

bc style formatted - Input `nbcseg` is set so that NPARC's boundary conditions are written in a formatted tabular list.

format ASCII - NPARC's restart file will have ASCII format.

format binary - NPARC's restart file will have binary format.

format unformatted - NPARC's restart file will have Fortran unformatted format.

precision single - NPARC's restart file will have single precision (4 byte) real words.

precision double - NPARC's restart file will have double precision (8 byte) real words.

Mach number - This command allows you to enter the Mach number that will be used to compute the initial flowfield in the restart file. The flowfield is computed using the technique described in the NPARC

User’s Guide (Reference 5).

Gridgen’s formatting of the NPARC BC file includes:

```
A 2 line comment string.
An INPUTS namelist containing NBLOCKS.
An empty TURBIN namelist.
For each block…
      A one-line comment containing the block name.
      A BLOCK namelist containing JMAX, KMAX, and LMAX
        (if 3D blocks), NBCSEG, and NPSEG.
      Either the BOUNDS namelists or formatted BC data.
      INVISC and LAMIN namelist inputs.
```

Note that the `PRESS` and `TEMP` inputs required by some BCs are not set by Gridgen. If namelist style is used for the BCs, then Gridgen sets `PRESS` and `TEMP` equal to `?P` and `?T`, respectively, when required by the BC. If formatted style is used for the BCs, then Gridgen includes the strings `( PRESS )` and `( TEMP )` in the appropriate fields. It is up to you to supply the correct values for `PRESS` and `TEMP`.

Also note that for NPARC a block interface or connection must be a block max face to a block min face. Block connections not of this standard will be flagged as Type 3 (non-conformal) BC’s.

NPARC’s grid file actually contains both grid and flowfield data. Its format is described in Reference 5.

17.8.26 OVERFLOW

OVERFLOW export supports 3D structured grids only. Gridgen writes only OVERFLOW’s `BCINP` namelist.

Please note that while Gridgen can create overlapping grids as required for OVERFLOW, the hole cutting and other overlap calculations must be computed using some other software such as PEGSUS, OVERFLOW’s pre-processor. In fact, the grid exported via OVERFLOW’s **Export Analysis Grid** command is actually formatted for PEGSUS. The PEGSUS grid file is Fortran unformatted and you are given a choice of single or double precision. This grid file also contains the block names.

17.8.27 PHOENICS

PHOENICS grid format is ASCII, PLOT3D, but with each grid block written out to a separate file. A boundary condition file is exported by Gridgen in the PHOENICS command language.

17.8.28 SCRYU

SCRYU export supports 2D and 3D grids containing hexahedral, quadrilateral, prism, and triangle elements. Boundary condition regions are written to the “S” file in “REGN” format.

17.8.29 SCRYU/Tetra

Tetrahedral elements only are supported and written to a NASTRAN formatted file. No boundary condition information is exported.

17.8.30 Splitflow

Splitflow export supports only 2D grids with unstructured, triangulated domains. 2D blocks must be created out of all unstructured domains (sub-parts) for Splitflow export. Within the Splitflow export file, sub-part names and material IDs are listed as the block names and numbers respectively.

Volume condition export, using both the VC name and number, is supported for Splitflow. The volume condition **SOLID** is used as the default.

17.8.31 STAR-CD

STAR-CD export supports 2D and 3D grids of all cell types. Two dimensional grids are assumed to lie in the x-y plane.

Gridgen writes three files for Star-CD's units 14, 15, 23, and 24. The name of the files that Gridgen exports is obtained from the base name of the file that you select (via the Browser or keyboard) with the numerals "14", "15", "23", and "24" appended, respectively. The files have no extension.

Volume condition export is supported for STAR-CD. Blocks with common VC's are automatically merged in the AS/W file with the VC used as the zone name. To have separate blocks written out with assigned VC, set a different VC to each block to avoid automatic merge. Alternatively, setting block VC's to **No Volume Condition** will cause the blocks to be exported separately under their block names.

Wherever Gridgen detects a singularity in a structured grid it will convert the cell from a hex (quad) to a wedge (triangle) when exporting it for use with STAR-CD.

17.8.32 TACOMA

TACOMA export supports 2D and 3D structured grids only.

17.8.33 TASCflow

TASCflow export supports 3D structured grids only. Gridgen writes two files: TASCflow's .grd and .bcf files. You are given a choice as to whether these files will be ASCII or unformatted.

A TASCflow `name.lun` file must exist in order for Gridgen to export files for TASCflow. Gridgen first checks for a `name.lun` file in the current directory and then in the directory pointed to by the `TASC_UTILITY` environment variable. If Gridgen does not find a `name.lun` file in either of these locations, it will create one in the current directory.

TASCflow unformatted export is not supported under Linux. Glyph requests for unformatted export under Linux will automatically be converted to ASCII.

17.8.34 TEAM

TEAM export supports 2D and 3D structured grids only. Please note the following:

- The first line of boundary condition data for each block always has the strings `'INITIAL FLOW'` set to `'FREE'` and `'EQUATION_SET'` set to `'EULER'`.
- The additional line of data for the `SOLID_V` boundary condition is always output as `'TGRAD n' 'FIXED f' 'I node'`.
- The additional line of input for the MFR, Mach Number, Pressure Ratio, and Speed Ratio Inlet boundary conditions are always output as `'MFR n'`, `'MACH n'`, `'PRATIO n'`, and `'QRATIO n'`, respectively.
- The additional line of input for the `EXHAUST` boundary condition is always output as '`NPR n' 'TRATIO n' 'NX n' 'NY n' 'NZ n' 'EXITMACH n'`.
- If your grid is 2D, two `SYMMETRY_XZ` boundary conditions will be added on the Kmin and Kmax block faces.

TEAM's grid file is ASCII, double precision and its format is described in Reference 8.

You must create 2D grids for TEAM in the x-z plane. When the grid is exported it is automatically extruded in the y-direction by 1.0 unit to create a 1-cell thick block.

17.8.35 TETREX

TETREX export supports both 2D and 3D grids of all cell types.

17.8.36 VSAERO

Since VSAERO is a panel method, only 2D structured blocks are supported and boundary conditions need not be applied. A complete VSAERO file is written but all non-grid data are set to zero.

17.8.37 VSAEROhybrid

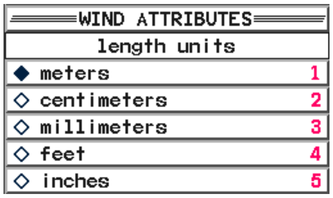
VSAEROhybrid export supports 2D structured and unstructured blocks. Blocks must be either structured or unstructured. Hybrid blocks are not supported.

17.8.38 WIND

WIND export supports 2D and 3D structured grids. Gridgen exports the .cgd "Common Grid Data" file used by WIND which contains the grid and boundary conditions. When exporting the grid for WIND, you must select the grid units from a menu for inclusion into the `.cgd` file. Please note that selecting units for WIND does *not* scale your grid. It simply indicates to WIND what the units are of the grid that you have already made.

WIND export in Gridgen now uses ADF Library Version A02011. As a result, executables based on older ADF libraries are no longer supported.

WIND export is not supported under HP-UX.



WIND ATTRIBUTES
length units
◆ meters 1
◇ centimeters 2
◇ millimeters 3
◇ feet 4
◇ inches 5

18. File Formats

18.1 Introduction

Several of the files that may be imported and exported by Gridgen are described in this section. Please note the distinction between binary and Fortran unformatted files. Fortran unformatted files contain extra data at the beginning and end of each record whereas the binary file contains only the data. Furthermore, please note that binary and Fortran unformatted files are not portable among all of Gridgen's supported hardware platforms due to an issue called *endian-ness* (whether the bits in word are stored from highest to lowest or lowest to highest). The Gridgen file, however, is portable among all supported hardware platforms because it contains the endian-ness setting with which it was written so that it can be reversed (if necessary) by the receiving system.

Native byte formats for Gridgen's supported hardware platforms are as follows:

1. SGI, HP, SUN and IBM have the Big Endian byte format.
2. PC (Windows and LINUX) and DEC have the Little Endian byte format.

18.2 Gridgen File

The Gridgen file (default extension .gg) contains all of the data used by Gridgen to create and maintain the grid including connectors, domains, blocks, analysis software boundary conditions, display options, grid attributes, and database references (entity names and parametric coordinates). The Gridgen file does *not* contain the databases, even those created in Gridgen. If you create database entities in Gridgen (for example, using the **Copy** command in the DATABASE COMMANDS menu), you must save them in a database file.

A detailed description of the Gridgen file is omitted due to its complexity. The Gridgen file is written in binary format.

If the grid has been generated using a database, then the database file must be imported before the Gridgen file.

18.3 GRIDBLOCK Files

Gridgen can only import GRIDBLOCK files from GRIDGEN Versions 8 and 6 using the Gridgen **Import** command in the INPUT/OUTPUT COMMANDS menu (Section 4.1.3). However, these file types are subject to the following restrictions.

GRIDBLOCK Version 8 files (default extension `.blk`) are fully compatible. However, due to the manner in which older versions of GRIDGEN maintained adherence of the grid to the database, the database entity groups that are automatically created by Gridgen upon network or PATRAN Neutral file import must be deleted (using the **UnGroup** command in the DATABASE COMMANDS menu) prior to GRIDBLOCK Version 8 file import.

Only the connector shapes from GRIDBLOCK Version 6 (default extension `.blk`) files will be imported. Furthermore, there is no adherence of connectors to the database and all of Version 6's ellipse segments are converted to circular arc segments (the shape may change).

If the grid has been generated using a database, then the database file must be imported before the Gridgen file.

18.4 Database Network File

A database network file (default extension `.net`) consists of one or more bilinear surfaces (wireframes) to be used as part of the geometry model (database). Gridgen can import (Section 4.4) and export (Section 4.5) database network files in binary, Fortran unformatted, and ASCII formats and in Gridgen or PLOT3D style. Gridgen reads from a Gridgen style network file until the end-of-file is encountered. Each network in the file must consist of at least 2x2 points (a single quadrilateral).

The formats described below are also those used for importing (Section 4.6) and exporting (Section 4.7) structured surface grids.

A code fragment for exporting a network file in Gridgen style is shown below.

```
c.....nmax is number of networks
c.....ni(n) number of points in the i direction for the nth network
c.....nj(n) number of points in the j direction for the nth network

      integer ni(nmax), nj(nmax)
      real x(imax,jmax,nmax),
     &     y(imax,jmax,nmax),
     &     z(imax,jmax,nmax)

      do 10 n = 1, nmax
          write(1) ni(n), nj(n)
          write(1) ( ( x(i,j,n), i=1,ni(n) ), j=1,nj(n) ),
     &             ( ( y(i,j,n), i=1,ni(n) ), j=1,nj(n) ),
     &             ( ( z(i,j,n), i=1,ni(n) ), j=1,nj(n) )
   10 continue
```

A code fragment for exporting a network file in PLOT3D style is shown below. Note this is PLOT3D's 3D, multi-grid format with `kmax=1` and no i-blanking. This is the only PLOT3D format used by Gridgen.

```
c.....nmax is number of networks
c.....ni(n) number of points in the i direction for the nth network
c.....nj(n) number of points in the j direction for the nth network

      integer ni(nmax), nj(nmax)
      real x(imax,jmax,nmax),
     &     y(imax,jmax,nmax),
     &     z(imax,jmax,nmax)

      write(1) nmax
      write(1) ( ( ni(n), nj(n), 1 ), n=1,nmax )
      do 10 n = 1, nmax
          write(1) ( ( x(i,j,n), i=1,ni(n) ), j=1,nj(n) ),
     &             ( ( y(i,j,n), i=1,ni(n) ), j=1,nj(n) ),
     &             ( ( z(i,j,n), i=1,ni(n) ), j=1,nj(n) )
   10 continue
```

18.5 Segment File

The segment file (default extension `.dat`) consists of one or more discrete, linear curves (segments). Segments can be imported (Section 4.4) and exported (Section 4.5) as database curves, imported (Section 4.6) and exported (Section 4.7) as connector grid points, and imported as a connector segment type (Section

7.2.4). When importing a segment file Gridgen reads until the end of file is reached. The segment file must be ASCII.

A code fragment for writing a segment file is shown below.

```
c.....nmax is the number of segments
c.....ni(n) is the number of points on segment n
c.....imax is the number of points

      integer nmax
      integer ni(nmax)
      real x(imax), y(imax), z(imax)

      do n = 1, nmax
         write(1,*) ni(n)
      do i = 1, ni(n)
            write(1,*) x(i,n), y(i,n), z(i,n)
      end do
       end do
```

18.6 Composite Database File

Gridgen's composite database file is provided as a means of consolidating all database entities and database display information in a single file. Since not all entities are supported by each of the file types (for example, conic arcs may only be imported via an IGES file) the composite file provides a means of exporting (Section 4.5) an entire geometry model that may have originated in several file formats in a single, convenient file. The composite file is also a convenient mechanism for storing database entities that are created or modified by Gridgen (for example, use of the **Copy** and **Translate** commands in the DATABASE COMMANDS menu). Finally, the composite file also stores the database display data (which entities are enabled, how they are displayed and in what color) and entity names and groups created in Gridgen. The bottom line is the composite database file should be used whenever any changes have been made to the database after import.

A detailed description of the composite file is omitted due to its complexity. The composite file is written only in ASCII format. Binary, and either single or double precision composite database files written by older Gridgen versions may be imported.

18.7 FIELDVIEW Unstructured File

Tetrahedral volume grids may be imported (Section 4.6) and structured, unstructured, hybrid grids may be exported (Section 4.7) in the FIELDVIEW Unstructured File (FV-UNS) format. This neutral file is used by Intelligent Light's (http://www.ilight.com) CFD post-processing software of the same name. Please contact Intelligent Light for further information on the format of this file.

18.8 Generic Flow Solver BC File

The generic flow solver's (Section 17.8.1) boundary condition file is an ASCII file used to export boundary condition data for un-supported analysis software. The code fragment below describes this file's contents in detail to assist you in preparing a translator if necessary. Note, the generic flow solver boundary condition file is for structured grids only. For unstructured grids, the generic format (grid and boundary condition) is in FV-UNS file format.

```
write flow_solver_id

write number_of_blocks

do number_of_blocks
```

```
               write block_imax, block_jmax,
                     (and block_kmax if 3D), block_name

      c........a region is a boundary condition or interblock connection
               write number_of_regions_on_block

               do number_of_regions_on_block

                  write source_region_imin, source_region_imax,
                        source_region_jmin, source_region_jmax,
                        (and source_region_kmin, source_region_kmax if
3D),
                        bc_number

                  if region is a connection (bc_number<0) then

      c..............For an interblock connection, there are three
      c..............pairs of indices that must be matched (three on
      c..............the source face and three on the target face).

      c..............The index that's constant on both the source and
      c..............target faces is easy to determine simply by
      c..............comparing the min and max indices and finding the
      c..............pair in which min and max are the same.

      c..............On each of the source and target faces, Gridgen
      c..............writes one of the two remaining min-max index
      c..............combinations as negative numbers.  The negative
      c..............min-max pair on each face correspond.  Once you
      c..............determine the corresponding negative indices
      c..............just convert them to positive numbers.

      c..............That leaves one pair of indices that align.

      c..............Keep in mind that the max index may actually
      c..............be less than the min index.  You can either use
      c..............a negative increment to step through the index
      c..............or you can swap the min and max indices on BOTH
      c..............source and the matching target indices.

                     write target_region_imin, target_region_imax,
                           target_region_jmin, target_region_jmax,
                          (and target_region_kmin, target_region_kmax if
3D),
                           target_block_number

                  endif

               end do number_of_regions_on_block

            end do number_of_blocks
```

The boundary condition codes used by the generic flow solver are described in the table below.

Generic ASW Boundary Conditions

value	description
0	none
1	interblock connection
2	solid surface
3	symmetry
4	farfield
5	inflow
6	outflow
71	i pole
72	j pole
73	k pole
8	generic #1
9	generic #2
10	generic #3

18.9 Surface Grid File

Surface grid files are used to import and export grid data. Surface grid files are identical in content to database network files (Section 18.4).

18.10 Volume Grid File

The volume grid file is used to import (Section 4.6) and export (Section 4.7) volume grid data. The volume grid file may be written in Gridgen or PLOT3D style, binary, Fortran unformatted, or ASCII format, single or double precision.

A Fortran code fragment for a Gridgen style volume grid file follows:

```
c.....number of grid points in block
      integer ni(nmax), nj(nmax), nk(nmax)
c.....grid point coordinates
      real x(imax,jmax,kmax), y(imax,jmax,kmax), z(imax,jmax,kmax)

      write(1) nblocks
      do mb = 1, nblocks
         write(1) ni(mb), nj(mb), nk(mb)
         write(1) (((x(i,j,k),i=1,ni(mb)),j=1,nj(mb)),k=1,nk(mb)),
     &            (((y(i,j,k),i=1,ni(mb)),j=1,nj(mb)),k=1,nk(mb)),
     &            (((z(i,j,k),i=1,ni(mb)),j=1,nj(mb)),k=1,nk(mb))
      end do
```

19. References

1. Abolhassani, J., Sadrehaghighi, I., Smith, R.E., and Tiwari, S.N., “Application of Lagrangian Blending Functions for Grid Generation Around Airplane Geometries”, *Journal of Aircraft*, Vol. 27, No. 10, October 1990, pp. 873-877.

2. Advanced Visual Systems, Inc., “Unstructured Cell Data”, *AVS User’s Guide*, pp. 2-39, Release 4, May 1992, Part Number 320-0011-02, Rev B.

3. Buning, Pieter G., et al, *OVERFLOW User's Manual, Version 1.6ak*, NASA Ames Research Center, Sep. 1992.

4. Chan, W.M., and Steger, J.L., “A Generalized Scheme for Three-Dimensional Hyperbolic Grid Generation”, AIAA paper no. 91-1588-CP, 1991.

5. Cooper, G.K., and Sirbaugh, J.R., "PARC Code: Theory and Usage", AEDC TR-89-15, United States Air Force Arnold Engineering and Development Center, 1989.

6. Davies, D.E., Salmond, Deborah J., “Calculation of the Volume of a General Hexahedron for Flow Predictions”, *AIAA Journal*, Vol. 23, No. 6, June 1985, pp 954-956.

7. Ehrlich, Louis W., “An Ad Hoc SOR Method”, *Journal of Computational Physics*, Vol. 44, pp. 31-45, 1981.

8. Goble, B.D., Raj, P., Kinnard, T.A., *Three-dimensional Euler/Navier-Stokes Aerodynamic Method (TEAM) Upgrade: Version 713 User's Manual*, United States Air Force Wright Laboratory, November 1993.

9. Gordon, W.J., "Blending Function Methods of Bivariate and Multivariate Interpolation", *SIAM Journal of Numerical Analysis*, Vol. 8, No. 158, 1971.

10. Hilgenstock, A., "A Fast Method For The Elliptic Generation of Three-Dimensional Grids With Full Boundary Control", *Numerical Grid Generation in Computational Fluid Mechanics `88*, ed. by Sengupta, S., et al, Pineridge Press Ltd., Swansea, UK, 1988, pp. 137-146.

11. Klopfer, G.H. and Molvik, G.A., "Conservative Multizonal Interface Algorithm for the 3D Navier-Stokes Equations", AIAA Paper 91-1601-CP, June, 1991.

12. Matus, R.J., Steinbrenner, J.P., and Chawner, J.R., “Preprocessing Complex Geometries for Computational Analysis”, presented at the ASME Energy Sources and Technology Conference, 01-03 Feb. 1999.

13. PDA Engineering, Inc., *PATRAN Plus User Manual*, Vols. I and II, Release 2.5, Publication Number 2191025, 1990.

14. Raj, P., Sikora, J.S., and Olling, C.R., *Three-dimensional Euler/Navier-Stokes Aerodynamic Method (TEAM)*, Vols. I-III, AFWAL TR-87-3074, United States Air Force Wright Research and Development Center, June 1989.

15. Reed, K., "The Initial Graphics Exchange Specification (IGES)", NISTIR 4412, National Institute of Standards and Technology, September 1991.

16. Reymond, Michael, and Miller, Mark, *MSC/NASTRAN Quick Reference Guide Version 69*, The MacNeal-Schwendler Corp., 1996.

17. Rizzi, A., and Eriksson, L.E., "Transfinite Mesh Generation and Damped Euler Equation Algorithm for Transonic Flow Around Wing-Body Configurations", AIAA paper no. 81-0999, 1981.

18. Shoof, Larry A., and Yarberry, Victor R., “EXODUS II: A Finite Element Data Model”, Sandia Report SAND92-2137, September 1994.

19. Smith, R.E., "Algebraic Grid Generation", *Numerical Grid Generation*, ed. by Thompson, J., North-Holland, NY, 1982, pp. 137-170.

20. Soni, B.K., "Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics", AIAA paper no. 85-1526.

21. Sorenson, R.L., *The 3DGRAPE Book: Theory, Users' Manual, Examples*, NASA TM-102224, July 1989.

22. Steinbrenner, J.P., Chawner, J.R., and Matus, R.J., “Evolution of Gridgen’s Data Hierarchy to Unstructured Grids”, from 6th International Conference on Numerical Grid Generation in Computational Field Simulations, ed. by Cross, M., et al, publ. May 1998 by NSF Research Center for CFS, Mississippi, pp. 833-842.

23. Steinbrenner, J.P., and Chawner, J.R., “Gridgen’s Synergistic Implementation of CAD and Grid Geometry Modeling”, from *Numerical Grid Generation in Computational Field Simulations*, ed. by Soni, B.K., et al, publ. by Engineering Research Center for CFS, Mississippi, 1996, pp. 363-372.

24. Steinbrenner, J.P., and Chawner, J.R., “Automatic Structured Grid Generation Using Gridgen (Some Restrictions Apply)”, NASA CP-3291, *Surface Modeling, Grid Generation, and Related Issues in CFD Solutions*, NASA Lewis Research Center, Cleveland, OH May 1995.

25. Steinbrenner, J.P., and Chawner, J.R., “Incorporation of a Hierarchical Grid Component Structure into GRIDGEN”, AIAA paper no. 93-0429, AIAA 31st Aerospace Sciences Mtg., Reno, NV, Jan. 1993.

26. Steinbrenner, J.P. and Chawner, J.R., “Recent Enhancements to the GRIDGEN Structured Grid Generation System”, NASA CP-3142, *Software Systems for Surface Modeling and Grid Generation*, ed. by Smith, R.E., NASA Langley Research Center, Apr. 1992, pp. 253-271.

27. Steinbrenner, J.P., Chawner, J.R., and Anderson, D.A., “Enhancements to the GRIDGEN System for Increased User Efficiency and Grid Quality”, AIAA paper no. 92-0662, AIAA 30th Aerospace Sciences Mtg, Reno, NV, Jan. 1992.

28. Steinbrenner, J.P., and Anderson, D.A., “Grid-Generation Methodology in Applied Aerodynamics”, from *Applied Computational Aerodynamics*, ed. by Henne, P.A., AIAA, 1990, pp. 91-130.

29. Steinbrenner, J.P., Chawner, J.R., and Fouts, C.L., “The GRIDGEN 3D Multiple Block Grid Generation System”, WRDC-TR-90-3022, Vols. I and II, Wright Research and Development Center, Wright-Patterson AFB, OH, Jul. 1990 and Oct. 1990.

30. Steinbrenner, J.P. Chawner, J.R., and Fouts, C.L., “Multiple Block Grid Generation in the Interactive Environment”, AIAA paper no. 90-1602, AIAA 21st Fluid Dynamics Conference, Seattle, WA Jun. 1990.

31. Steinbrenner, J.P., Chawner, J.R., and Fouts, C.L., “A Structured Approach to Interactive Multiple Block Grid Generation”, from AGARD CP-464, *Applications of Mesh Generation to Complex 3-D Configurations*, Mar. 1990, pp. 8-1 - 8-12.

32. Steinbrenner, J.P. and Anderson, D.A., “Three-Dimensional Parametric Block Grid Regeneration With Localized Solution Adaption”, from Numerical Grid Generation in Computational Fluid Dynamics ‘88, ed. by Sengupta, S., et al, Pineridge Press Ltd., Swansea, UK, 1988, pp. 539-548.

33. Steinbrenner, J.P., Karman, S.L. Jr., and Chawner, J.R., “Generation of Multiple Block Grids for Arbitrary 3D Geometries”, from AGARDograph No. 309, *Three Dimension Grid Generaion for Complex Configurations - Recent Progress*, ed. by Yoshihara, H., Mar. 1988, pp. 40-55.

34. Thomas, P.D., and Middlecoff, J.F., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", *AIAA Journal*, Vol. 18, pp. 652-656, 1979.

35. Thompson, J.F., Thames, F.C., and Mastin, C.W., "Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Fields Containing Any Number of Arbitrary Two Dimensional Bodies", NASA CR-2729, July 1977.

36. Vinokur, M., "On One-Dimensional Stretching Functions for Finite-Difference Calculations", NASA CR 3313, 1980.

37. von Lavante, E., “Elliptic Generation of Two-Dimensional Grids For Internal Flow Calculations”, *TEES Technical Report* No. 12415-82-01, March 1982

38. Walatka, P.P., and Buning, P.G., "PLOT3D User's Manual", NASA TM-101067, 1989.

39. White, J.A., "Elliptic Grid Generation With Orthogonality and Spacing Control On An Arbitrary Number of Boundaries", AIAA paper no. 90-1568, June 1990.

40. Winslow, A., "Equipotential Zoning of Two-Dimensional Meshes", *Journal of Computational Physics*, Vol. 149, pp. 153-172, 1966.

41. http://www.vrml.org/, “Home of the VRML Consortium.”

42. 3D Systems, Inc., “StereoLithography Interface Specification”, October 1989.

43. Intelligent Light, “*FIELDVIEW User’s Manual, Software Release Version 5.5*”, November 1996.

44. E. von Lavante, “Elliptic Generation of Two-Dimensional Grids for Internal Flow Calculations”, Texas Engine;ering Experiment Station, *TEES Technical Report* No. 12415-82-01, 25 March 1982.

20. Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Numerics

2 Point Connectors *7-10*

A

ACIS SAT
- import as database *4-7*

Add Segment *7-2*
ADPAC asw *17-8*
algebraic solver *12-1*
- automatic initialization
  - blocks *10-8*
  - domains *9-3*

Align I,J,K *15-8*
Analysis Software
- ADPAC *17-8*
- CFD++ *17-8*
- CFDSHIP-Iowa *17-8*
- CFX-4 *17-9*
- CFX-5 *17-9*
- CGNS-structured *17-9*
- CGNS-unstructured *17-9*
- CNSFV *17-9*
- Cobalt *17-9*
- COMO *17-9*
- CRUNCH *17-9*
- current *2-1*
- data file export *17-7*
- DTNS *17-10*
- Exodus II *17-10*
- export
  - boundary conditions *4-19*
  - data *4-19*
  - grid *4-19*
- Falcon *17-10*
- FANS *17-10*
- FDNS/UNIC *17-10*
- FLUENT v4 *17-10*
- FLUENT v5 *17-10*
- GASP *17-10*
- generic *17-8*
- GUST *17-11*
- INCA v2 *17-11*
- INCA v3 *17-11*
- mirror for export *17-7*
- NCC *17-12*
- NPARC *17-12*
- OVERFLOW *17-13*
- PHOENICS *17-13*
- requirements of *1-4*
- SCRYU *17-13*
- SCRYU/Tetra *17-13*
- selection *17-1*
- setting boundary conditions *17-3*, *17-5*
- SPLITFLOW *17-13*
- STAR-CD *17-14*
- TACOMA *17-14*
- TASCflow *17-14*
- TEAM *17-14*
- TETREX *17-14*
- VSAERO *17-15*
- WIND *17-15*

Analysis Software Commands *17-1*
anchor
- scale *14-8*
- stretch *14-10*

angle constraint
- elliptic PDE solver *12-11*

angular deviation
- maximum *3-4*
  - default *3-2*

append grids *4-3*
ASCII mesh import *4-15*
Aspect Ratio diagnostic function *16-5*
associativity of grid and database *4-2*
Auto Complete *9-7*

Auto Next Edge *9-7*
Auto Save *9-8*
axis
- 3D cursor movement *2-16*
- Align I,J,K command *15-8*
- body axes *2-14*
- computational *15-8*
- cylindrical projection *14-14*
- for Rotate command *14-11*
- image manipulation *2-14*
- plane *6-6*
- ReSpecify I,J,K command *15-7*
- Rot Axes command *2-20*
- rotation point *2-19*
- screen axes *2-14*
- surface of revolution *6-10*
- XYZ Axes command *2-20*

B

background control function
- default *3-3*
- setting *12-8*
- type
  - fixed grid *12-8*
  - LaPlace *12-8*
  - Thomas-Middlecoff *12-8*

backward compatibility of restart files *4-3*
binary mesh import *4-15*
Blackboard *2-1*
- 3D cursor coordinates *2-16*
- block face creation *10-1*
- circular arc creation *7-6*
- copying connector dimension *11-2*
- database point creation *6-2*
- delete *15-4*
- domain creation *9-2*, *9-4*
- during database curve extraction *6-4*
- during database intersection *15-17*
- during elliptic PDE solver *12-6*
- during subdomain creation *12-14*, *12-15*
- during unstructured solver *13-2*
- face linkages *10-7*
- intersection *15-16*
- location of *2-21*
- rotate *14-11*
- setting elliptic PDE attributes *12-6*
- translation *14-7*, *14-9*, *14-10*

block
- algebraic solver *12-1*
- changing grid density *11-3*
- copy *15-3*
- create *10-1*
  - structured *10-1*
  - unstructured *10-3*
    - adding faces *10-3*
    - linkage *10-6*
    - reorient faces *10-6*
- data hierarchy *14-6*
- default settings *3-4*
  - maximum pyramid height *3-5*
  - minimum pyramid height *3-5*
  - pyramid aspect ratio *3-5*
  - tetrahedral mesh size factor *3-5*
  - tetrahedral mesher memory limit *3-5*
- delete *15-5*
- disable *15-7*
- display *2-20*
- elliptic PDE solver
  - background control function
    - default *3-3*
  - foreground control function
    - default *3-3*
- enable *15-7*
- hierarchy *1-1*, *1-2*
- linkage *14-6*
- maintaining linkages *14-6*
- modify
  - join *14-20*
  - mirror *14-12*
  - rotate *14-10*
  - scale *14-8*
  - split *14-18*
  - stretch *14-9*
  - translate *14-7*
- name *15-2*
- number of *2-1*
- preparing to run batch elliptic smoothing *4-20*
- Quick Menu *2-7*
- sort *15-5*
- structured
  - 2D *10-1*

computational coordinates *10-7*
restrictions *10-1*
BMP file *2-27*
boundary
connectors on database boundaries via on DB Entities command *7-2*
rendering domain boundary *2-26*
boundary conditions
custom
Create *17-7*
Delete *17-7*
Delete All *17-7*
Modify *17-7*
Examine *17-6*
export *4-19*
for elliptic PDE solver *12-13*
setting *17-3*, *17-5*
structured grids *17-5*, *17-6*
unstructured grids *17-5*
branch segment *7-3*
break point
create *11-11*
definition *11-6*
delete *11-11*
setting spacing constraint *11-7*
setting spacing constraint at *11-7*
uses for *11-10*
Browser window *2-1*
Button Menu *2-1*

C

capturing screen images *2-27*
CATIA v4
import as database *4-7*
Cells reference surface *16-16*
CFD++ asw *17-8*
CFDSHIP-Iowa asw *17-8*
CFX-4 asw *17-9*
CFX-5 asw *17-9*
CGNS-structured asw *17-9*
CGNS-unstructured asw *17-9*
chord height deviation
maximum *3-4*
default *3-3*
CNSFV asw *17-9*
Cobalt asw *17-9*
Color Bar Type for Examine *16-8*
Color Bar Values for Examine *16-9*
combine separate restart files *4-3*
COMO asw *17-9*
composite database file import *4-7*
computational coordinates *10-7*
align with chosen block *15-8*
respecifying *15-7*
conic rho value *6-9*
connector
adding break points *11-10*
adding subconnectors *11-10*
approximating shape *14-23*
beginning spacing *11-7*
break point *11-6*
changing spacing and distribution *11-13*
color coding *7-1*
control point
edit *14-21*
copy *15-3*
create
2 point connectors *7-10*
add segment *7-2*
miscellaneous commands *7-9*
on database entities *7-11*
periodic *15-8*
segment
3D Space Akima *7-3*
3D Space Curve *7-3*
3D Space Line *7-2*
copy *7-7*
DB cnstr Akima *7-3*
DB cnstr Curve *7-3*
DB cnstr Line *7-3*
import *7-7*
pole *7-8*
creating *7-1*
data hierarchy *14-4*
definition *1-2*, *7-1*
delete *15-4*
dimension
chordal deviation *11-3*
default *3-1*
maximum angular deviation *3-2*
maximum chord height deviation *3-3*
maximum turning angle *11-3*

disable *15-7*
display grid points *2-20*
display status *2-20*
distribution
    break point *11-10*
    Set Δs Values command *11-13*
    spacing constraint *11-6*
    spacing continuity *11-7*
    spline type *11-9*
    unconstraining *11-7*
distribution function *11-7*
    copy *11-8*
    copy from network *11-8*
    equal spacing *11-8*
    general *11-9*
    geometric progression *11-8*
    hyperbolic tangent *11-8*
    maximum deviation *11-9*
    maximum turning angle *11-9*
    monotonic rational quadratic spline *11-8*
    MRQS *11-8*
edit segment *14-21*
ending spacing *11-7*
erase segment *14-20*
hierarchy *1-1*
insert segment *14-20*
least squares fit *14-23*
line width *2-21*
linkage *14-4*
merging duplicates *15-12*
modify
    join *14-19*
    mirror *14-12*
    rotate *14-10*
    scale *14-8*
    split *14-16*
    stretch *14-9*
    translate *14-7*
number of *2-1*
periodic *15-8*
project *14-13*
Quick Menu *2-7*
redimensioning *11-1*
reducing number of control points *14-23*
segment
    3D Space
        Akima *7-3*
        curve *7-3*
        line *7-2*
    branch *7-3*
    database constrained
        Akima *7-3*
        curve *7-3*
        line *7-3*
    insert *14-20*
    rational *7-4*
        circular arc *7-6*
            slide points *7-6*
        conic *7-4*
segment file *4-7*
setting average spacing *11-1*
setting number of grid points *11-1*
shape *7-1*
smoothing shape *14-23*
spacing constraint
    default
        beginning *3-3*
        ending *3-3*
split
    using Merge Nod/Con *15-13*
subconnector *11-6*
constraint
    slope
        Bezier curve *14-22*
control point
    Add At Cross *2-18*
    Add By Closest Point Projection *2-18*
    Add By Linear Projection *2-18*
    Add by Picking *2-18*
    Add via Keyboard *2-18*
    Add via Offset *2-18*
    adding during distribution *11-11*
    convert to break point *11-10*
    database curve fitting *6-5*
    database surface fitting *6-12*
    definition *7-1*
    edit *14-21*
    Import segment type *7-7*
    pole segment *7-8*
    via 3D cursor *2-17*
converting structured grids to unstructured *9-8*
coordinates
    determining for a pickable point *2-28*
copy

  - block *15-3*
  - break point spacing constraint *11-7*
  - connector *15-3*
  - connector dimension *11-2*
  - database entity *15-3*
  - domain *15-3*
- **create**
  - block *10-1*
    - import *4-13*
    - structured *10-1*
    - unstructured *10-3*
      - linkage *10-6*
      - reorient faces *10-6*
  - break point *11-11*
  - connector
    - 2 point *7-10*
    - add segment *7-2*
    - import *4-13*
    - on database entities
      - database
        - placing connectors on *7-11*
    - segment
      - 3D Space Akima *7-3*
      - 3D Space Curve *7-3*
      - 3D Space Line *7-2*
      - copy *7-7*
      - DB cnstr Akima *7-3*
      - DB cnstr Curve *7-3*
      - DB cnstr Line *7-3*
      - import *7-7*
      - pole *7-8*
  - control points *2-17*
  - database
    - fit curve *6-5*
    - fit surface *6-12*
    - linear sweep surface *6-11*
    - note *6-14*
    - plane *6-5*
    - polyconic surface *6-9*
    - ruled surface *6-7*
    - surface *6-7*
    - surface of revolution *6-10*
  - database entity
    - curve *6-2*
      - akima *6-3*
      - circle *6-3*
      - conic *6-3*
      - curve *6-3*
      - intersect *15-15*
      - line *6-3*
    - note *6-14*
    - plane
      - 3-point *6-6*
      - coefficient *6-7*
      - constant *6-5*
      - point-normal *6-6*
    - point
      - average *6-2*
      - intersect *15-15*
    - surface *6-7*
  - domain *9-1*
    - import *4-13*
    - structured *9-1*
    - unstructured
      - parametric *9-7*
      - parametric fit *9-7*
      - standard *9-7*
- **Crinkle reference surface** *16-15*
- **CRUNCH asw** *17-9*
- **cursor** *2-15*
  - 3D *2-16*
    - centering *2-19*
    - movement *2-16*
  - during picking *2-8*
  - panning *2-15*
  - rotation *2-15*
  - zooming *2-15*

D

- **data hierarchy**
  - block *14-6*
  - connector *14-4*
  - database *14-2*
  - domain *14-6*
- **database** *6-1*
  - associativity to grid *4-2*
  - constraining domains to *12-11*
  - create
    - Akima curve *6-3*
    - conic section *6-3*
    - curve *6-2*
    - extract curve from surface *6-4*
    - fit curve *6-5*

        - fit surface *6-12*
        - line *6-3*
        - linear sweep surface *6-11*
        - note *6-14*
        - offset curve *6-3*
        - plane *6-5*
        - point *6-1*
        - polyconic surface *6-9*
        - ruled surface *6-7*
        - surface *6-7*
        - surface of revolution *6-10*
    - creating domains on *9-9*
    - creating entities *6-1*
    - data hierarchy *14-2*
    - definition *1-1*
    - display *2-20*
    - gaps
        - closing automatically *15-11*
    - group *15-2*
    - hierarchy *1-1*
    - intersection *15-15*
    - layer manager *5-1*
    - linkage *14-2*
    - modify
        - join *14-19*
        - mirror *14-12*
        - rotate *14-10*
        - scale *14-8*
        - split *14-16*
        - stretch *14-9*
        - translate *14-7*
    - show enabled entities *2-20*
    - STL
        - feature extraction *15-17*

Database diagnostic function *16-3*

database entity
- copy *15-3*
- default name *15-1*
- delete *15-4*
- disable *15-7*
- enable *15-7*
- import
    - composite *4-7*
    - IGES *4-9*
    - PATRAN *4-10*
    - segment *4-7*
    - wireframe *4-6*
- intersecting *15-15*
- name *15-1*
- name assigned on import *15-1*
- shell
    - extracting curves *15-18*
    - extracting surfaces *15-18*
    - feature extraction *15-17*

database hierarchy
- during modification *14-2*

database parametrization
- default display *3-6*

decay factor
- unstructured solver *13-7*

default
- angular deviation
    - maximum *3-4*
- average connector spacing *3-1*
- background control function *3-3*
- connector dimension *3-1*
- connector spacing
    - beginning *3-3*
    - ending *3-3*
- database parameter display *3-6*
- database parametrization display *3-6*
- decay factor *3-4*
- distribution function *3-3*
- domain initialization *3-4*
- foreground control function *3-3*
- join angle *3-5*
- maximum angular deviation *3-2*
- maximum chord height deviation *3-3*, *3-4*
- pyramid aspect ratio *3-5*
- pyramid height
    - maximum *3-5*
    - minimum *3-5*
- QuickSave file name *3-6*
- split angle *3-5*
- tetrahedral mesh size factor *3-5*
- tetrahedral mesher memory limit *3-5*
- tolerances *3-6*
- triangle area
    - maximum *3-4*
    - minimum *3-4*

Defaults menu *3-1*

delete
- automatically created group entities for Version 8 compatibility *4-11*
- block *15-5*

    - when changing analysis software *17-3*
  - break point *11-11*
  - connector *15-4*
  - control point during segment creation *7-10*
  - control point during segment editing *14-21*
  - database entity *15-4*
  - domain *15-4*
  - face
    - during block creation *10-6*
  - last 2 Point Connector *7-11*
  - overwrite during Gridgen file import *4-3*
- **dimension**
  - definition *7-1*
- **disable**
  - block *15-7*
  - connector *15-7*
  - database entities *15-7*
  - domain *15-7*
- **display attributes for Examine** *16-7*
- **display editing** *2-22*
- **Display menu** *2-19*
- **display style** *2-23*
  - hidden line removal *2-24*
  - off *2-23*
  - shaded *2-24*
  - shaded/wireframe *2-24*
  - solid *2-24*
  - solid/wireframe *2-24*
  - wireframe *2-23*
- **Display Style for Examine** *16-8*
- **Display window** *2-1*
  - background color *2-21*
  - linewidth *2-21*
- **distance between two pickable points** *2-28*
- **distribution**
  - definition *7-1*
  - function
    - default *3-3*
  - Set Δs Values command *11-13*
- **distribution function** *11-7*
  - copy *11-8*
  - copy from network *11-8*
  - equal spacing *11-8*
  - general *11-9*
  - geometric progression *11-8*
  - hyperbolic tangent *11-8*
  - maximum deviation *11-9*
  - maximum turning angle *11-9*
  - monotonic rational quadratic spline *11-8*
  - MRQS *11-8*
- **domain**
  - adding to block face *10-1*, *10-3*
  - automatic grid initialization *9-3*, *9-7*
  - background control function
    - default *3-3*
  - bad types *9-10*
  - changing grid density *11-3*
  - color codes *9-4*
  - constraining to a database *9-9*
  - copy *15-3*
  - create *9-1*
    - Auto Complete *9-7*
    - Auto Next Edge *9-7*
    - Auto Save *9-8*
    - Next Edge *9-8*
    - On Database Entities *9-9*
    - periodic *15-8*
    - Previous Edge *9-8*
    - Reorient Edge *9-8*
    - structured *9-1*
    - unstructured *9-4*
      - example *9-4*
      - from structured *9-8*
      - multiple loop *9-4*
      - parametric *9-7*
      - parametric fit *9-7*
      - standard *9-7*
      - with holes *9-4*
  - create during block creation *10-6*
  - data hierarchy *14-6*
  - default settings *3-4*
    - decay factor *3-4*
    - initialization *3-4*
    - maximum angular deviation *3-4*
    - maximum chord height deviation *3-4*
    - maximum triangle area *3-4*
    - minimum triangle area *3-4*
  - delete *15-4*
  - disable *15-7*
  - display *2-20*
  - duplicate *9-11*
  - enable *15-7*

- enable, disable *2-20*
- foreground control function
  - default *3-3*
- hierarchy *1-1*, *1-2*
- linkage *14-6*
- linkages during modification *14-6*
- modify
  - join *14-20*
  - mirror *14-12*
  - rotate *14-10*
  - scale *14-8*
  - split *14-17*
  - stretch *14-9*
  - translate *14-7*
- number of *2-1*
- overlapping *9-12*
- overlapping and adjacent *9-11*
- periodic *15-8*
- prohibited *9-12*
- project *14-13*
- Quick Menu *2-7*
- should be avoided *9-10*
- structured
  - algebraic *9-3*
  - algebraic grid method *12-1*
  - database constrained *9-3*
  - pole *9-10*
  - transfinite interpolation *9-3*
    - parametric *9-3*
    - parametric fit *9-3*
    - standard *9-4*
- topology
  - default settings *3-5*
    - join angle *3-5*
    - split angle *3-5*

DTNS asw *17-10*

duplicate grid entities

- removing *15-11*

E

elliptic partial differential equations *12-4*

elliptic PDE solver

- attributes *12-6*
  - background control function *12-8*
    - type
      - fixed grid *12-8*
      - LaPlace *12-8*
      - Thomas-Middlecoff *12-8*
  - boundary condition *12-13*
    - fixed *12-13*
    - floating *12-13*
    - orthogonal *12-14*
  - foreground control function *12-9*
    - angle constraint *12-11*
    - form *12-9*
      - Hilgenstock-White *12-10*
      - Steger-Sorenson *12-10*
    - selecting boundaries for *12-9*
    - setting *12-9*
    - spacing constraint *12-10*
  - relaxation factor *12-7*
  - shape *12-11*
    - database *12-12*
    - fixed *12-12*
    - free *12-12*
- background control function
  - default *3-3*
- creating subgrids for *12-14*
- foreground control function
  - default *3-3*
- preparing Glyph files for batch run *4-20*
- running *12-4*

enable

- block *15-7*
- database entities *15-7*
- domain *15-7*

enable, disable

- in Display menu *2-20*

entity

- display
  - display style *2-23*
  - render density *2-26*
  - render mode *2-25*
- display of *2-22*
- selecting *2-8*
- selecting multiple *2-8*
- selection
  - By Text *2-11*
  - from Browser *2-9*
  - from Display window *2-8*
  - get 1st *2-11*
  - get last *2-11*
  - in Box *2-11*

on Box *2-11*
Pick by Expression *2-11*
Set to off *2-11*
Set to on *2-11*
Set to tog *2-11*
Sort By *2-11*
special controls *2-10*
entity selection
by page *2-11*
Erase reference surfaces *16-16*
Examine *16-1*
block hiliter *16-11*
Cells *16-16*
connector hiliter *16-11*
Crinkle *16-15*
Diagnostic Function *16-2*
Aspect Ratio *16-5*
Database *16-3*
Jacobian *16-4*
None *16-2*
orthogonality *16-7*
Size *16-3*
Size Ratio *16-4*
skewness *16-6*
Smooth *16-5*
Spacing *16-6*
Usage *16-2*
Display
Color Bar Type *16-8*
Color Bar Values *16-9*
Display Style *16-8*
Histogram *16-10*
Show Cells *16-9*
Display Attributes *16-7*
domain hiliter *16-11*
Erase *16-16*
extend *16-12*
Flat *16-15*
Focus *16-1*
Hiliter *16-10*
Mode *16-10*
Permanent *16-14*
Save *16-14*
examine
block face *10-6*
Exodus II asw *17-10*
export
analysis software data *4-19*, *17-7*
analysis software grid *4-19*
boundary condition file *4-19*
FIELDVIEW boundary surface file *4-19*
FVUNS grid *4-17*
Glyph block smoothing batch files *4-20*
Glyph rc file *4-19*
NASTRAN grid *4-17*
PATRAN grid *4-17*
PLOT3D grid *4-17*
QuickSave file *3-6*
Quick Menu *2-7*
selecting file name for *2-12*
STL grid *4-17*
VRML grid *4-17*
extend Examine toggle *16-12*
extracting curve from surface *6-4*
extracting curves from shell database entities *15-18*
extracting surfaces from shell database entities *15-18*
extrusion
see grid extrusion

F

Falcon asw *17-10*
FANS asw *17-10*
FDNS/UNIC asw *17-10*
feature extraction *15-17*
FIELDVIEW
boundary surface file export *4-19*
file style attribute *4-15*, *4-17*
file
analysis bcs
export *4-19*
analysis data
export *4-19*
analysis grid
export *4-19*
attributes
format *4-6*, *4-15*, *4-17*
ASCII *4-6*, *4-15*, *4-18*
binary *4-6*, *4-15*, *4-18*
unformatted *4-6*, *4-15*, *4-18*
precision *4-6*, *4-15*, *4-18*
style *4-14*

FVUNS 4-15, 4-17
Gridgen 4-14, 4-17
NASTRAN 4-14, 4-17
PATRAN 4-17
PLOT3D 4-14, 4-17
STL 4-8, 4-15, 4-17
VRML 4-8, 4-15, 4-17
type
composite 4-7
IGES 4-9
network 4-6
PATRAN 4-10
segment 4-7
database
import 4-4
export image file
display command for 2-22
Glyph
export rc file 4-19
Glyph batch block smoothing
export 4-20
Gridgen
quick import 4-3
Input/Output menu
Quick Key for 2-5
QuickSave
Quick Key for 2-6
setting name 3-6
QuickSave file
Quick Menu 2-7
selecting name for 2-12
Flat reference surface 16-15
FLUENT v4 asw 17-10
FLUENT v5 asw 17-10
foreground control function
angle constraint 12-11
default 3-3
form 12-9
Hilgenstock-White 12-10
Steger-Sorenson 12-10
selecting boundaries for 12-9
setting 12-9
spacing constraint 12-10
Fortran unformatted mesh import 4-15
frame buffer 2-21
FVUNS
grid export to 4-17
import as grid 4-15

G

gaps
closing automatically 15-11
GASP asw 17-10
generatrix
linear sweep surface 6-11
offset curve 6-3
surface of revolution 6-10
generic asw 17-8
Global Mode for Examine 16-10
Glyph
export
files for batch block smoothing 4-20
rc file 4-19
grid
associativity to database 4-2
types 1-1
grid extrusion 8-1
attributes 8-4
aspect ratio 8-5
boundary conditions 8-11
marching direction 8-7
smoothing parameters 8-9
step size 8-5
surface shape 8-4
automatic cell quality checking 8-12
hyperbolic (Hyp) 8-2
normal (Nrml) 8-2
path based (Path) 8-3
rotational (Rot) 8-2
stop criteria 8-12
translational (Tran) 8-3
grid point
changing number of 11-1
grid spacing
global change 11-13
Gridgen
history 1-5
Version 6 1-5
Version 7 1-5
Version 8 1-5
Version 9 1-5
group
automatically created during database file import 4-11
connector groups in ReDimension Grids command 11-3

database entity groups during intersection *15-16*
IGES composite curves *4-10*
in browser via sorting *2-11*
of commands via labels *2-8*
of radio buttons *2-4*
Un/Group database command *15-2*
GUST asw *17-11*

H

handle
rotate *14-11*
stretch *14-10*
help
menu button for *2-3*
Quick Menu *2-7*
using on-line help *2-27*
hierarchy *1-1*
Hiliter
block Examine *16-11*
connector Examine *16-11*
domain Examine *16-11*
Examine *16-10*
Histogram for Examine *16-10*
Histogram Levels *16-10*
hot keys *2-3*
hyperbolic extrusion *8-2*
attributes *8-4*

I

IGES
database import *4-9*
entities supported by Gridgen *4-9*
image capture *2-22*
image manipulation
axis systems *2-14*
demote *2-22*
devices *2-14*
glide toggle *2-14*
hot keys for *2-15*
light source *2-22*
panning *2-14*
Quick Menu keys for *2-6*
recomputing view limits *2-21*
restoring views *2-22*
rotating *2-14*
saving views *2-22*
screen capture *2-22*
sparse mode
image manipulation
of large models *2-22*
view transformation *2-21*
zoom box *2-15*
zooming *2-14*
import
ACIS SAT file as database *4-7*
ASCII *4-15*
binary *4-15*
CATIA v4 as database *4-7*
composite database file *4-7*
curves *4-13*
Fortran unformatted *4-15*
FVUNS as grid *4-15*
Gridgen restart file *4-1*
grids from other sources *4-13*
IGES database *4-9*
NASTRAN as grid *4-14*
PATRAN as database *4-10*
PLOT3D as grid *4-14*
Pro/E prt/asm file as database *4-7*
Quick Import Gridgen file *4-3*
Quick Key for Input/Output menu *2-5*
STEP as database *4-7*
STL as database *4-8*
STL as grid *4-15*
surface grids from other sources *4-13*
VRML as database *4-8*
VRML as grid *4-15*
INCA v2 asw *17-11*
INCA v3 asw *17-11*
Incorporate Volume Grid command *4-20*
intersect *15-15*

J

Jacobian diagnostic function *16-4*
join
block *14-20*

connector *14-19*
database *14-19*
domain *14-20*
join angle *3-5*

L

layer manager *5-1*
layers *5-1*
Linear TFI *12-2*
linkage *14-2*
block *14-6*
connector *14-4*
database *14-2*
domain *14-6*
Local Mode for Examine *16-10*

M

menu
split *2-8*
menu button
grayed-out *2-3*
Next Page *2-8*
Prev Page *2-8*
Quick Key *2-5*
Quick Menu *2-6*
radio *2-4*
selecting *2-3*
small text *2-20*
toggle *2-4*
VCR *2-4*
merge
connectors *15-11*, *15-12*
nodes *15-11*
nodes and connectors *15-13*
topological filter *15-15*
Message window *2-1*
mirror *14-12*
for export only *17-7*
model size *3-7*
modify *14-1*
database hierarchy *14-2*
linkage *14-2*
mirror *14-12*
rotate *14-10*
scale *14-8*
stretch *14-9*
transformations *14-7*
translate *14-7*
mouse *2-2*
bell *2-22*
glide *2-14*
left button picking *2-22*
sensitivity
decrease *2-19*
increase *2-19*
reset *2-19*
warp
disabling *2-21*
warping *2-8*

N

NASTRAN
grid export to *4-17*
import as grid *4-14*
NCC asw *17-12*
Next Edge *9-8*
node
definition *7-1*
display status *2-20*
merging duplicates *15-11*
tolerance *3-6*
None diagnostic function *16-2*
normal (algebraic) extrusion *8-2*
attributes *8-4*
NPARC asw *17-12*

O

On Database Entities *9-9*
Ortho TFI *12-3*
orthogonal
closest point projection *14-15*
elliptic PDE boundary condition *12-14*
foreground control functions *12-9*
Ortho TFI *12-3*

- **Orthogonality diagnostic function** *16-7*
- **OVERFLOW asw** *17-13*

P

- **paper and report references** *19-1*
- **Parametric Fit TFI** *12-4*
- **Parametric TFI** *12-3*
- **path based extrusion** *8-3*
  - attributes *8-4*
- **PATRAN**
  - grid export to *4-17*
  - import as database *4-10*
- **periodic** *15-8*
- **Permanent reference surface** *16-14*
- **PHOENICS** *17-13*
- **picking** *2-8*
  - by page *2-11*
  - By Text *2-11*
  - from Browser *2-9*
  - from Display window *2-8*
  - get 1st *2-11*
  - get last *2-11*
  - in Box *2-11*
  - on Box *2-11*
  - Pick by Expression *2-11*
  - Set to off *2-11*
  - Set to on *2-11*
  - Set to tog *2-11*
  - Sort By *2-11*
  - special controls *2-10*
- **PLOT3D**
  - grid export to *4-17*
  - import as grid *4-14*
- **PNG file, PostScript file** *2-27*
- **point**
  - anchor *14-8*, *14-10*
  - break point *11-10*
  - closest point projection *14-15*
  - control point
    - create *2-17*
    - definition *7-1*
    - edit *14-21*
  - create offset curve through *6-3*
  - created during intersection *15-15*
  - database
    - 3-point plane *6-6*
    - create
      - average *6-2*
    - fit curve through *6-5*
    - fit surface through *6-12*
    - point-normal plane *6-6*
  - grid point
    - algebraic methods *12-1*
    - display of *2-23*
    - effect of modifying an entity on *14-6*, *14-7*
  - handle *14-10*, *14-11*
  - node
    - definition *7-1*
  - number of database render points *2-27*
  - origin for spherical projection *14-13*
  - rotation point
    - move to 3D cursor *2-19*
    - pick *2-21*
  - same grid point tolerance *3-7*
  - same point (node) tolerance *3-7*
  - slide point *7-6*
  - slope point *14-22*
- **Polar TFI** *12-3*
- **pole domains** *9-10*
- **Pole TFI** *12-4*
- **Previous Edge** *9-8*
- **Pro/E prt/asm**
  - import as database *4-7*
- **project**
  - command
    - closest point *14-15*
    - cylindrical *14-14*
    - linear *14-14*
    - spherical *14-13*
  - control point during Add by Picking *2-18*
  - during elliptic PDE solver iteration *12-12*
  - elliptic PDE solver attribute
    - type *12-13*
  - extrusion attribute *8-4*
  - in elliptic PDE solver *12-5*
  - unstructured solver attribute
    - type *13-4*
- **pyramid**
  - attributes *13-9*
  - cell creation *13-8*

pyramid aspect ratio *3-5*
pyramid height
  maximum *3-5*
  minimum *3-5*

Q

Quick Keys *2-5*
Quick Menu *2-6*
  enabling *2-21*

R

radio button *2-4*
ReDimension *11-1*
ReDimension Grids *11-3*
reference surfaces for Examine *16-14*
references *19-1*
render density *2-26*
  curve *2-27*
  grid *2-26*
render mode *2-25*
  intervals *2-25*
  isolines *2-25*
  outline *2-25*
  triangles *2-26*
Reorient Edge *9-8*
report and paper references *19-1*
ReSpecify I,J,K *15-7*
restart file
  backward compatibility *4-3*
  combining multiple restarts *4-3*
restart file import *4-1*
RGB file *2-27*
rotate
  move rotation point to 3D cursor *2-19*
  pick rotation point *2-21*
rotation point
  changing *2-28*
rotational extrusion *8-2*
  attributes *8-4*

S

Save
  reference surface *16-14*
scale
  scaled distribution function *11-3*
screen capture *2-22*, *2-27*
SCRYU asw *17-13*
SCRYU/Tetra asw *17-13*
segment
  3D Space
    Akima *7-3*
    Bezier *7-8*
    curve *7-3*
    line *7-2*
  add *7-2*
  branch *7-3*
  database constrained
    Akima *7-3*
    Bezier *7-8*
    curve *7-3*
    line *7-3*
  definition *1-2*, *7-1*
  edit *14-21*
  format of segment file *18-2*
  general *7-8*
  import database from segment file *4-7*
  pole *7-8*
  rational *7-4*
    circular arc *7-6*
      slide points *7-6*
    conic *7-4*
  segment file *4-7*
  using 3D cursor with *2-17*
Set BCs *17-3*, *17-5*
Set Δs Values *11-13*
Show Cells for Examine *16-9*
Show EnbldDB *2-20*
Size diagnostic function *16-3*
Size Ratio diagnostic function *16-4*
Skewness diagnostic function *16-6*
Smooth diagnostic function *16-5*
spacing
  average
    default *3-1*
  beginning
    default *3-3*
  ending

default *3-3*

spacing constraint *11-6*

beginning spacing *11-7*

elliptic PDE solver *12-9*, *12-10*

ending spacing *11-7*

unconstrained *11-7*

Spacing diagnostic function *16-6*

spine

polyconic surface *6-9*

split

block *14-18*

connector *14-16*

connector into subconnectors *11-10*

creates General segment *7-8*

database *14-16*

database entity *14-16*

domain *14-17*

split angle *3-5*

Standard TFI *12-2*

STAR-CD asw *17-14*

Status window *2-1*

Steger-Sorenson *12-10*

STEP

import as database *4-7*

Stereo Lithography (STL)

grid export to *4-17*

import as database *4-8*

import as grid *4-15*

structured grid *12-1*

algebraic methods *12-1*

creating subgrids for *12-14*

Linear TFI *12-2*

Ortho TFI *12-3*

Parametric Fit TFI *12-4*

Parametric TFI *12-3*

Polar TFI *12-3*

standard TFI *12-2*

elliptic methods *12-4*

attributes *12-6*

background control function *12-8*

type

fixed grid *12-8*

LaPlace *12-8*

Thomas-Middlecoff *12-8*

boundary condition *12-13*

fixed *12-13*

floating *12-13*

orthogonal *12-14*

foreground control function *12-9*

angle constraint *12-11*

form *12-9*

Hilgenstock-White *12-10*

Steger-Sorenson *12-10*

selecting boundaries for *12-9*

setting *12-9*

spacing constraint *12-10*

relaxation factor *12-7*

shape *12-11*

database *12-12*

fixed *12-12*

free *12-12*

creating subgrids for *12-14*

running *12-4*

transfinite interpolation *12-2*

structured grids

domains

create *9-1*

subconnector

copy dimension from *11-2*

copying distribution from *11-8*

definition *11-6*

dimensioning *11-11*

summarize

Status window *2-1*

T

TACOMA asw *17-14*

TASCflow asw *17-14*

TEAM asw *17-14*

technical references *19-1*

tetrahedra

attributes *13-7*

tetrahedral mesh size factor *3-5*

tetrahedral mesher memory limit *3-5*

TETREX asw *17-14*

Text Input window *2-1*

three-dimensional (3D) grids *17-1*

- **toggle button** *2-4*
- **tolerance** *3-6*
  - connector *3-6*
  - grid point *3-6*
  - model size *3-7*
  - node *3-6*
- **TopoFltr** *15-15*
- **transfinite interpolation** *12-1*
- **transformations** *14-7*
- **translational extrusion** *8-3*
  - attributes *8-4*
- **triangle area**
  - maximum *3-4*
  - minimum *3-4*
- **two-dimensional (2D) grids** *17-1*

U

- **unstructured grids**
  - attributes
    - shape *13-4*
  - block
    - create *10-3*
  - boundary point insertion *13-9*
  - convert from structured *9-8*
  - domains
    - create *9-4*
  - solver *13-1*
- **unstructured solver** *13-1*
  - attributes *13-2*
    - boundary decay factor *13-7*
    - maximum angular deviation *13-6*
    - maximum surface deviation *13-6*
    - maximum triangle size *13-6*
    - minimum triangle size *13-6*
    - shape *13-4*
    - smoothing *13-5*
  - projection *13-4*
- **Usage diagnostic function** *16-2*

V

- **VCR button** *2-4*
- **Version window** *2-1*
- **VRML**
  - grid export to *4-17*
  - import as database *4-8*
  - import as grid *4-15*
- **VSAERO asw** *17-15*

W

- **WIND asw** *17-15*
- **window**
  - Blackboard *2-1*
  - Browser *2-1*
    - picking items from *2-9*
  - Button Menu *2-1*
  - Display *2-1*
  - Message *2-1*
  - Status *2-1*
  - Text Input *2-1*
  - Version *2-1*

Z

- **zoom box** *2-15*
  - undo *2-16*
  - using orthonormal view *2-21*