1 STRUCTURAL ELEMENTS

1.1 Overview

An important aspect of geomechanical analysis and design is the use of structural support to stabilize a rock or soil mass. Structures of arbitrary geometry and properties, and their interaction with a rock or soil mass, may be modeled with FLAC. This section describes the structural elements available in FLAC. Generic concepts, such as geometry specification, linkage of elements to the grid and to each other, options for specifying end conditions, and specification of properties, are discussed first. Each type of structural element is then described in detail, including a description of the numerical formulation and the properties required for each element type. Example applications are also provided at the end of each section.*

All vector quantities in this section are expressed using indicial notation with respect to a fixed right-handed rectangular Cartesian coordinate system. Thus, the position vector is denoted by $x_i$, where it is understood that the indices range over the set $\{1, 2\}$.

Note that, because FLAC is a two-dimensional code, the three-dimensional effect of regularly spaced elements is accommodated by scaling their material properties in the out-of-plane direction. This procedure is explained in Section 1.9.4.

1.1.1 Types of Structural Elements

Seven forms of structural support may be specified:

1. **Beam Elements** – Beam elements are two-dimensional elements with three degrees of freedom ($x$-translation, $y$-translation and rotation) at each end node. Beam elements can be joined together with one another and/or the grid. Beam elements are used to represent a structural member, including effects of bending resistance and limited bending moments. Tensile and compressive yield strength limits can also be specified. Beams may be used to model a wide variety of supports, such as support struts in an open-cut excavation and yielding arches in a tunnel. Interface elements can be attached on both sides of beam elements in order to simulate the frictional interaction of a foundation wall with a soil or rock. Beam elements attached to sub-grids via interface elements can also simulate the effect of geo-textiles. (See Section 1.2.)

* The data files listed in this volume are created in one of two ways: either by typing in the commands in a text editor; or by generating the model in the GIIC and exporting the file using the File/Export Record menu item. The files are stored in the directory “ITASCA\FLAC600\STRUCTURES” with the extension “.DAT.” A project file is also provided for each example. In order to run an example and compare the results to plots in this volume, open a project file in the GIIC by clicking on the File/Open Project menu item and selecting the project file name (with extension “.PRJ”). Click on the Project Options icon at the top of the Project Tree Record, select Rebuild unsaved states, and the example data file will be run and plots created.
2. **Liner Elements** – Liner elements, like beam elements, are two-dimensional elements with three degrees of freedom (x-translation, y-translation and rotation) at each end node, and these elements can be joined together with one another and/or the grid. Liner elements are also used to represent a structural member in which bending resistance, limited bending moments and yield strengths are important. The primary difference between liner elements and beam elements is that liner elements include bending stresses to check for yielding, whereas beam elements only base the yielding criterion on axial thrust. Liner elements are recommended for modeling tunnel linings, such as concrete or shotcrete liners. (See Section 1.3.)

3. **Cable Elements** – Cable elements are one-dimensional axial elements that may be anchored at a specific point in the grid (point-anchored), or grouted so that the cable element develops forces along its length as the grid deforms. Cable elements can yield in tension or compression, but they cannot sustain a bending moment. If desired, cable elements may be initially pretensioned. Cable elements are used to model a wide variety of supports for which tensile capacity is important, including rockbolts, cable bolts and tiebacks. (See Section 1.4.)

4. **Pile Elements** – Pile elements are two-dimensional elements that can transfer normal and shear forces and bending moments to the grid. Piles offer the combined features of beams and cables. Shear forces act parallel to the element, and normal forces perpendicular to the element. The three-dimensional effect of the pile interaction with the grid can be simulated. A user-defined FISH function describing the load versus deformation at the pile/medium interface normal to the pile can also be specified. The element does not yield axially, but plastic hinges can develop. Pile elements are specifically designed to represent the behavior of foundation piles. (See Section 1.5.)

5. **Rockbolt Elements** – Rockbolt elements, like pile elements, are two-dimensional elements that can transfer normal and shear forces and bending moments to the grid. Rockbolt elements have the same features as pile elements. In addition, rockbolt elements can account for: (1) the effect of changes in confining stress around the reinforcement; (2) the strain-softening behavior of the material between the element and the grid material; and (3) the tensile rupture of the element. Rockbolt elements are well-suited to represent rock reinforcement in which nonlinear effects of confinement, grout or resin bonding, or tensile rupture are important. (See Section 1.6.)

6. **Strip Elements** – Strip elements represent the behavior of thin reinforcing strips placed in layers within a soil embankment to provide structural support. The strip element is similar to the rockbolt element, in that strips can yield in tension or compression, and a tensile failure strain limit can be defined. Strips cannot sustain a bending moment. The shear behavior at the strip/soil interface is defined by a nonlinear shear failure envelope that varies as a function of a user-defined transition confining pressure. Strip elements are designed to be used in the simulation of reinforced earth retaining walls. (See Section 1.7.)

7. **Support Members** – Support members are intended to model hydraulic props, wooden props or wooden packs. In its simplest form, a support member is a spring connected between two boundaries. The spring may be linear, or it may obey an arbitrary relation.
between axial force and axial displacement, as prescribed from a table of values. The support member has no independent degrees of freedom: it simply imposes forces on the boundaries to which it is connected. A support member may also have a width associated with it. In this case, it behaves as if it were composed of several parallel members spread out over the specified width. (See Section 1.8.)

In all cases, the commands necessary to define the structure(s) are quite simple, but they invoke a very powerful and flexible structural logic. This structural logic is developed with the same finite-difference logic as the rest of the code (as opposed to a matrix-solution approach), allowing the structure to accommodate large displacements and to be applied for dynamic as well as static analysis.

### 1.1.2 Geometry

The geometries of all structural elements are defined by their endpoints. The user defines the endpoints for beams, liners, cables, piles, rockbolts and strips, whereas the endpoints for support elements are found automatically by FLAC. Note that cable, pile, rockbolt and strip endpoints have different mechanical behaviors, depending on the form of specification – grid or x,y (see below).

#### 1.1.2.1 Beam, Liner, Cable, Pile, Rockbolt and Strip Elements

The primary format to specify each beam, liner, cable, pile, rockbolt or strip element is of the form

```
STRUCT <type> begin ... end ... <keyword ...>
```

where type is beam, liner, cable, pile, rockbolt or strip. Note that an optional format, using from and to, is also available to facilitate geometry creation for beams and liners along model boundaries. See Section 1.1.2.2.

In general, endpoints may be placed at any location inside or outside the FLAC grid. The beginning and ending locations are identified by the keywords begin and end, respectively. One of three types of linkage may be defined by phrases following begin and end:

- grid = i, j
- x,y
- node = n

grid = i, j denotes that the beginning (or ending) of the element is linked to gridpoint (i, j) of the host medium. For example, in Figure 1.1 two beam elements are connected to the grid at gridpoints (1,2) and (1,3), and at (1,3) and (1,4). The only way that beam and liner elements can interact with the grid is by linking their nodes to gridpoints with the grid keyword, or via a connection through interface elements (see Section 1.1.2.2). If the grid keyword is specified for one end of a cable, then that end of the cable is bonded rigidly to the specified gridpoint; the grout properties are not used at such an attachment point. This also applies for the pile, rockbolt and strip elements and coupling-spring properties.
If \(x\) - and \(y\)-coordinates are specified, the endpoint is located at any selected location within or outside the grid. In Figure 1.1, a cable element (Cable 1) is located with endpoints at \((x = 0, y = 3.5)\) and \((x = 3, y = 3.5)\). If coordinates are used to specify one end of a cable, even if the coordinates coincide exactly with a gridpoint, then grout stiffness and strength will operate at the connection (e.g., the cable may “pull out”). Likewise, if \(x\)- and \(y\)-coordinates are specified for a pile, rockbolt or strip node, the coupling-spring properties will operate.

**Figure 1.1** Placement of element end nodes (FLAC zones are 1 unit square)

\texttt{node} = \(n\) links the beginning (or ending) of the beam, liner, cable, pile, rockbolt or strip to another node of the structure. The node numbers are assigned sequentially, starting with one, when the linkage phrase \texttt{grid} = \(i, j\) or \(x, y\) is used. Alternatively, nodes can be created at any location and assigned node numbers by the user via the command \texttt{STRUCT node n x, y} to position a node at a specific location, or via the command \texttt{STRUCT node n grid i, j} to link the node to a gridpoint. These nodes can then be included in the structural element. Node numbers can be identified by issuing either the \texttt{PRINT struct node position} or \texttt{PLOT struct node} command. For example, cable 2 in Figure 1.1 is defined by endpoints at node numbers 1 and 6. Node 6 is first located at \((x = 3, y = 2)\). Note that all nodes (and all elements) have unique numbers. Example 1.1 shows the commands to produce the geometry in Figure 1.1.

Structural element groups are assigned unique ID numbers automatically. A group is a collection of structural elements of the same type that contains a contiguous set of nodes, and the adjoined element segments have the same property number (see Section 1.1.6). The group ID numbers can be found via the \texttt{PLOT struct number} command. For example, in Example 1.1 there are three groups.
because the two beam elements are connected by a common node (node 2). Note that cable 2 has a separate group ID number even though it is connected to beam 1 at node 2.

### Example 1.1 Specifying structural elements

```plaintext
grid 4 4
model
struct beam begin grid 1,2 end grid 1,3 ; beam 1
struct beam begin grid 1,3 end grid 1,4 ; beam 2
struct cable begin 0,3.5 end 3,3.5 ; cable 1
struct node 6 3,2 ; individual node
struct cable begin node 1 end node 6 ; cable 2
```

A single element may be divided at its creation into a number of smaller elements or segments using the `segment = n` keyword. Each segment represents a structural element. If \( n = 1 \), then only one segment (and, thus, one element) is created. If \( n > 1 \), then FLAC divides the specified beam, liner, cable, pile, rockbolt or strip into \( n \) elements of equal length. The coordinates and node numbers for each element are automatically determined by FLAC. The `PLOT struct element` command plots the individual elements and their identifying numbers.

The most common reason to specify \( n > 1 \) is to improve accuracy, especially with cable, pile, rockbolt and strip elements that are interacting with the host medium. In this case, the distribution of shear forces along the element is, to some extent, a function of the number of nodal points. The following rules-of-thumb can be used to determine the number of element nodal points and, thus, segments for cables, piles, rockbolts and strips.

1. Try to provide approximately one element-nodal point in each `FLAC` zone. The reasoning here is that since the zones are constant-stress elements, it is not necessary to have more than one interaction point within a zone.

2. Try to provide at least two to three structural elements within the development length of the cable or rockbolt. The development length is determined by dividing the specified yield force by the unit bond value. By following this procedure, failure by “pullout” can occur if such conditions arise. For example, if cable elements are too long, then only the yield failure mode of the axial element is possible. (There is no yield in the piles at this time.)

3. If a cable, pile, rockbolt or strip crosses a grid interface, and the calculation is to be performed in large-strain mode, then enough element segments must be provided in the part of the element that is distorted by the interface, so that the proper shear restraint is captured. At least five element segments in this region must be provided.
Structural element segments can be deleted at any time in the calculation process by specifying the keyword phrase `delete n1 n2` following the name of the element type. For example, to delete beam element segments beginning with segment (i.e., element) number 10 and ending with segment number 15, use the command

```
struct beam delete 10 15
```

If only `n1` is given, one element segment is deleted. If neither `n1` nor `n2` is specified, all segments are deleted. If the segments to be deleted are connected by a slave (or master) node, the node must first be "unslaved" (see Section 1.1.5).

All information related to the geometry and properties of structural elements can be printed with the `PRINT struct` command, with appropriate keywords.

### 1.1.2.2 Beam and Liner Elements Created along Boundaries

Beams and liners can only interact with the FLAC grid in one of two ways: either by directly connecting nodes to gridpoints (via the `grid = i, j` linkage), or by using interfaces to connect the nodes to the grid. If a beam or liner is to be attached along a grid boundary, either every node must be individually connected to a corresponding gridpoint, or an interface must be created, with one side of the interface attached to the structural element and the other side attached to the grid. Geometry creation can become quite tedious in either case. See Section 1.1.4 for details on the procedures for linking beams and liners to the grid.

An optional format for geometry creation is available to facilitate generation of beams or liners along grid boundaries. This automation eliminates the requirement for the user to locate and define gridpoints or an interface between the structural nodes and the grid. The format also allows the definition of multiple layers of beams or liners along the boundary.

Specification of beam or liner geometry along a grid boundary is given by a command of the form

```
STRUCT <type> <long> from i1, j1 to i2, j2 <interface ni> <keyword ...>
```

where `type` is `beam` or `liner`. If `interface` is not specified, then structural nodes are created along the boundary, and automatically attached to all the gridpoints along the boundary from gridpoint `(i1, j1)` to gridpoint `(i2, j2)`. If `interface` is specified, then an interface with number `ni` is automatically created between the beam or liner and the grid. Interface properties are specified using the `INTERFACE` command. (See the `INTERFACE` command in Section 1.3 in the Command Reference.) The shortest distance between the specified gridpoints will be taken; the optional keyword `long` forces the longer route to be taken. If `long` is used and `i1 = i2` and `j1 = j2`, then a closed loop of beam or liner elements is created.

Additional layers of beams or liners can be added by specifying the command of the form

```
STRUCT <type> <long> from node n1 <nx> to node n2 interface ni <keyword ...>
```

This command creates another layer adjacent to an existing beam or liner, between the given nodes `n1` and `n2`. The new layer interacts with the existing layer via the interface. The rules for ordering
the nodes are the same as that for the INTERFACE command (i.e., the nodes of the existing beam or liner have to be given so that the new beam or liner approaches from the left).

If the from and to nodes are the same, a closed loop of structural elements is created. In this case, the direction must be specified by using the optional node nx. This node is the next node in the sequence, which conveys the direction required. For example,

```
struct beam from node 4,5 to node 4 interface 12 prop 22
```

The starting and ending node is 4, and the next node to be taken is 5. If there is no neighbor with ID 5, an error is signaled. If the double-node notation is specified for a chain of structural elements that is not closed, then only the direction implied by the pair will be taken; the double-node notation is not recognized in this case.

### 1.1.2.3 Support Elements

A support member is created with a STRUCT command of the following form.

```
STRUCT support x,y <keyword>
```

where x,y is one point on the member. The point must be located in empty space (e.g., occupied by null zones).

Optional keywords may be given as follows:

- **angle a**
  
  The axis of the support member is oriented at a degrees to the x-axis (default = 90°, if omitted).

- **remove**
  
  or

- **delete x,y**
  
  Either keyword causes the existing support member closest to (x,y) to be deleted. The keyword must be the only one given in this case.

- **segment n**
  
  For a member with non-zero width, the segment keyword specifies the number of sub-elements that comprise the member (default = 5).
width \( w \)

The support member spans a width of \( w \) perpendicular to its axis (default = 0).

When the **STRUCT support** command is given, **FLAC** searches for the two nearest boundaries that intersect the axis of the member, and places the member between the two intersection points. An error will be detected if fewer than two boundaries are found. The points of attachment to the **FLAC** grid are preserved regardless of the displacement that occurs subsequently.

### 1.1.3 Connection of Structural Elements to Each Other

Structural connections are provided by specifying the same `grid = i, j, node = n` or `x, y`-coordinates for the elements to be connected. Any beam, liner, cable, pile, rockbolt or strip element can be joined to any other beam, liner, cable, pile, rockbolt or strip element. (In order to create separate, unconnected nodes at the same physical location, use the **STRUCT node** command with a unique node number for each node.) There is also no limit to the number of elements that can be joined at an endpoint. The **PLOT beam**, **PLOT liner**, **PLOT cable**, **PLOT pile**, **PLOT rockbolt** and **PLOT strip** commands can be used with the **PLOT struct node** command to check whether elements are structurally connected to each other. For example, if beams and piles are connected, use the command **PLOT beam red pile white struct node** to identify structural connections. Elements that are connected will have the same structural node numbers. When connecting cables to beams, liners, piles, rockbolts or strips, a pin joint will result if the cable is specified first.

### 1.1.4 Linkage of Structural Elements to the Grid

In order for beam or liner elements to interact with the model grid, they must be explicitly linked to the grid. A beam or liner element can be linked to the grid with the `grid = i, j` option in the **STRUCT** command. It should be noted that, even though a beam or liner element node has the same coordinates as a grid coordinate, the beam or liner will *not* interact with the grid unless the `grid = i, j` option is specified. Further, only the endpoints of the series of elements will be linked to the grid if more than one segment is requested in a single **STRUCT beam** or **STRUCT liner** command; separate **STRUCT beam** and **STRUCT liner** commands *must* be given if all nodes are to be connected to gridpoints.

The beam or liner can also be linked to the medium by connecting the beam or liner to an interface. This is done with the **from node n1 to node n2** keyword phrase in the **INTERFACE** command. Beams and liners may interact with the grid, and beams and liners may interact with each other via interfaces. (See the **INTERFACE** command in Section 1.3 in the **Command Reference** for rules to include beam elements at interfaces.)

Connection of beam and liner elements to a grid can be a tedious operation. The process to connect beams or liners to a grid boundary is automated, as explained in Section 1.1.2.2.

Cable, pile, rockbolt and strip elements can interact with the grid via the shear coupling springs (and normal coupling springs in the case of a pile or rockbolt). Elastic stiffness properties and
cohesive and stress-dependent frictional properties describe the interaction between the elements and the grid. If all the parameters are zero, these elements will not interact with the grid. If a cable, pile, rockbolt or strip node is placed with the grid keyword, then it will be rigidly connected to that gridpoint, and the springs will have no effect at that point.

**1.1.5 End Conditions**

The supplemental command `STRUCT node = n` keyword provides options for describing the end conditions of beam, liner, cable, pile, rockbolt and strip elements. The options include:

1. free or fixed \( x \)- and \( y \)-displacements or rotations;
2. pin joints;
3. applied velocities;
4. applied loads or moments; and
5. slaved nodes.

These options are given by the following qualifying keywords following the node number \( n \).

- **fix** \(<x> <y> <r>\)
  
  This option allows beam, liner, cable, pile, rockbolt or strip node \( n \) to have fixed \( x \)- and/or \( y \)-velocities or (for beam, liner, pile and rockbolt nodes) fixed angular velocities (e.g., a beam, liner, pile or rockbolt end may be locked in place or allowed to rotate).

- **free** \(<x> <y> <r>\)
  
  This removes the constraint set by the fix keyword. (The default condition is free.)

- **initial** keyword

  Certain node variables can be assigned initial values. The following keywords apply:

  - **rvel** \( value \)
    
    rotational velocity for beam, liner, pile or rockbolt nodes

  - **xdis** \( value \)
    
    \( x \)-displacement for beam, liner, cable, pile, rockbolt or strip nodes
\[ x_{vel} \] \text{ value}

\[ x \]-velocity for beam, liner, cable, pile, rockbolt or strip nodes

\[ y_{dis} \] \text{ value}

\[ y \]-displacement for beam, liner, cable, pile, rockbolt or strip nodes

\[ y_{vel} \] \text{ value}

\[ y \]-velocity for beam, liner, cable, pile, rockbolt or strip nodes

\textbf{load} \quad \text{ } fx, fy, \text{ } mom

This allows the user to apply \( x \)- and/or \( y \)-direction forces or moments to node \( n \) for beams, liners, piles and rockbolts, and \( x \)- and/or \( y \)-direction forces to node \( n \) for cables or strips. (\( \text{mom} \) may be omitted for cables and strips.)

\textbf{pin}

This establishes a pin connection at node \( n \) (i.e., frees moments for beams, liners and piles).

\textbf{slave} \quad \text{<x> <y> <m>}

This option sets the slave condition of node \( n \) to node \( m \) in the \( x \)- and/or \( y \)-direction for beams, liners, cables, piles, rockbolts and strips. If neither \( x \) nor \( y \) is specified, both directions are “slaved”; the rotational degree-of-freedom cannot be “slaved.” Note that the stiffness assigned to slaved nodes will still influence the critical timestep. Thus, the modulus may be reduced to increase the timestep for slaved nodes.

\textbf{unslave} \quad \text{<x> <y>}

This option removes the slave condition of node \( n \) in the \( x \)- and/or \( y \)-direction.

For the keywords \textbf{fix}, \textbf{free}, \textbf{initial}, \textbf{load} and \textbf{pin}, a range of nodes can be selected instead of a single node. Replace \text{node} \( n \) with \text{node range} \( n1 \ \text{n2} \) to specify a range of nodes, where \( n1 \) is the first node and \( n2 \) is the last node in the range.

When using the \textbf{initial} keyword for cable nodes, the velocity can only be assigned in the axial direction of the element; the cable \textit{must} be aligned in either the \( x \)- or \( y \)-direction to specify an axial velocity.

It should be noted that element nodes that are linked to a \textit{FLAC} gridpoint will have the same kinematic restraints as the gridpoint. A consequence of this is that appropriate symmetry conditions used for
the FLAC grid will also work for the elements. However, beam, liner, pile and rockbolt elements that terminate at a symmetry line should also have their rotations fixed at the symmetry line.

### 1.1.6 Material Properties

Property numbers are assigned to elements with the **property np** keyword. If this keyword is not specified, the default is \( np = 1 \). Each different type of element is then assigned geometric and material properties by using the **STRUCT property = np** command. For example, if a beam is to be used (i.e., one structural cross-section), the commands may be:

```plaintext
struc beam ... prop = 3
struc prop=3  e=200e9  i=2.3e-5  a=4.8e-3
```

for a \( W_6 \times 25 \) beam (in SI units).*

Property numbers are assigned to a structural-element group. As mentioned in Section 1.1.2.1, a group is a contiguous set of structural-element segments of the same type that have the same property number. If two separate groups of the same type of element, and with the same property number, are connected, they will become one group with the same group number. If two separate groups with different property numbers are connected, each group will still retain its own group number. Group numbers can be determined by giving the **PLOT struct number** command.

When structural elements are created in the **GIIC**, property numbers are assigned using the following property number ranges:

- **beam**: 1001 – 1999
- **cable**: 2001 – 2999
- **pile**: 3001 – 3999
- **rockbolt**: 4001 – 4999
- **liner**: 5001 – 5999
- **support**: 6001 – 6999
- **strip**: 7001 – 7999

If a FLAC data file is imported into the **GIIC**, the property numbers should correspond to these ranges in order for the structural element properties dialogs to be active in the **GIIC**. The property numbers specified in the data files listed in this chapter correspond to these ranges.

Structural nodes and coupling springs are assigned a property number that corresponds to the element segment that is connected to the structural nodes. Any properties attributed to the connecting node (e.g., a plastic hinge node) will be taken from the last element segment created. This is also an important consideration when a property number is changed for a portion of the element group (see the **STRUCT chprop** command, below).

---

It is possible to specify different property numbers within a structural element group. This is done with the `STRUCT chprop np range nel1 nel2` command, where `np` is the new property number. Element segments with ID numbers between and including the element numbers `nel1` and `nel2` have their property numbers changed to the new property number. Separate group ID numbers will be created automatically within the original group, corresponding to the different property numbers.

Property numbers for structural nodes are stored separately and are determined based upon adjacent element segment property numbers when cycling begins. When `STRUCT chprop` is given, all nodes and coupling springs associated with the element segments that have property numbers changed are also affected. The command `PRINT struct node info` can be given after performing `CYCLE 1` in order to check property numbers of nodes. Note that property numbers of nodes can only be changed with the `STRUCT chprop` command before any cycling is done. A `FISH` function can be used to change individual property numbers of nodes at any time in the calculation. For an example, see Example 1.21.

The mass density (density keyword) must also be given if the weight of the structure is to be taken into account under gravity or dynamic loading. Gravity forces, $F_i$, are included in structural elements based on the formula

$$ F_i = \rho A L g_i $$

(1.1)

where:  
$\rho$ = mass density of element;  
$A$ = cross-sectional area of element;  
$L$ = element length; and  
$g_i$ = gravitational acceleration vector.

The required properties for each structural element type are described in the following sections. It should be noted that all quantities must be given in a consistent set of units (see Table 1.1).
<table>
<thead>
<tr>
<th>Property</th>
<th>Unit</th>
<th>SI</th>
<th>Imperial</th>
</tr>
</thead>
<tbody>
<tr>
<td>area</td>
<td>length^2</td>
<td>m^2 m^2 m^2 cm^2</td>
<td>ft^2 in^2</td>
</tr>
<tr>
<td>axial or shear stiffness</td>
<td>force/ disp</td>
<td>N/m kN/m MN/m Mdynes/cm</td>
<td>lbf/ ft lbf/ in</td>
</tr>
<tr>
<td>bond stiffness</td>
<td>force/length/ disp</td>
<td>N/m/m kN/m/m MN/m/m Mdynes/cm/cm</td>
<td>lbf/ ft/ ft lbf/ in/ in</td>
</tr>
<tr>
<td>bond strength</td>
<td>force/length</td>
<td>N/m kN/m MN/m Mdynes/cm</td>
<td>lbf/ ft lbf/ in</td>
</tr>
<tr>
<td>exposed perimeter</td>
<td>length</td>
<td>m m m cm</td>
<td>ft in</td>
</tr>
<tr>
<td>moment of inertia</td>
<td>length^4</td>
<td>m^4 m^4 m^4 cm^4</td>
<td>ft^4 in^4</td>
</tr>
<tr>
<td>plastic moment</td>
<td>force-length</td>
<td>N-m kN-m MN-m Mdynes.cm</td>
<td>ft-lbf in-lbf</td>
</tr>
<tr>
<td>yield strength</td>
<td>force</td>
<td>N kN MN Mdynes</td>
<td>lbf lbf</td>
</tr>
<tr>
<td>Young’s modulus</td>
<td>stress</td>
<td>Pa kPa MPa bar</td>
<td>lbf/ ft^2 psi</td>
</tr>
</tbody>
</table>

where: 
1 bar = 10^6 dynes/cm^2 = 10^5 N/m^2 = 10^5 Pa,
1 atm = 1.013 bars = 14.7 psi = 2116 lbf/ft^2 = 1.01325 x 10^5 Pa,
1 slug = 1 lbf s^2/ft = 14.59 kg,
1 snail = 1 lbf s^2/in, and
1 gravity = 9.81 m/s^2 = 981 cm/s^2 = 32.17 ft/s^2.
1.1.7 Plastic Moments and Hinges

Inelastic bending is simulated in beams, piles and rockbolts by specifying a limiting plastic moment.* If a plastic moment is specified, the value may be calculated as follows. Consider a flexural member of width $b$ and height $h$. If the member is composed of a material that behaves in an elastic-perfectly plastic manner, the elastic and plastic resisting moments can be computed. The moment necessary to produce yield stress, $\sigma_y$, in the outer fibers is defined as the elastic moment, $M^E$, and is calculated as

$$M^E = \sigma_y \frac{bh^2}{6}$$  \hspace{1cm} (1.2)

For yielding to occur throughout the section, the yield stress must act on the entire section, and the location of the resultant force on one-half of the section must be $h/4$ from the neutral surface. The resisting moment, defined as the plastic moment, $M^P$, is

$$M^P = \sigma_y \frac{bh^2}{4}$$  \hspace{1cm} (1.3)

The preceding discussion assumes a section that is symmetric about the neutral axis. However, if the section is not symmetric (for example, a T-section), or if the stress-strain relations for tension and compression differ appreciably (for example, reinforced concrete), the neutral axis shifts away from the fibers which first yield, and it is necessary to relocate the neutral axis before the resisting moment can be evaluated. The neutral axis may be found by integrating the stress profile over the section and solving for the location of the axis at which stress is zero. In some cases, the integral can be expressed in terms of one unknown, in which case, the solution may not be difficult. However, if the stress-strain relation for the material does not resemble an ideal elasto-plastic diagram, the solution may involve a number of trials. Nearly all texts on reinforced concrete or steel design provide procedures and examples for calculating plastic moments.

The present formulation in FLAC assumes that beam, pile and rockbolt elements behave elastically until they reach the plastic moment. This assumption is reasonably valid for symmetric rolled-steel sections, because the difference between $M^P$ and $M^E$ is not large. However, for reinforced concrete, the plastic moment may be as much as an order of magnitude greater than the elastic moment.

The section at which the plastic moment occurs can continue to deform without inducing additional resistance after it reaches $M^P$. The plastic-moment capacity sets the limit for the internal moments of structural-element segments for beams, piles and rockbolts. In order to limit the moment that is

---

* Alternatively, inelastic behavior can be specified for liner elements by employing an elastic-plastic material model that incorporates bending resistance, limiting bending moments and yield strengths of the liner material. See Section 1.3.1.
transmitted between element segments, the moment capacity at the nodes must also be restricted. The condition of increasing deformation with a limiting resisting moment that results in a discontinuity in the rotational motion is called a plastic hinge. Potential plastic hinges are prescribed for beam, pile or rockbolt nodes by the user. (See the command `STRUCT hinge nel1 nel2`.) If the limiting moment is reached at elements connected by a plastic hinge node, then a discontinuity in the rotational motion will develop. See Section 1.2.4.4 for an example application of plastic hinges for beams, and Section 1.6.4.2 for an example application of plastic hinges for rockbolts.

Once plastic rotation occurs at a particular location in a structure, this spot is weakened, and future deformation will tend to occur at the same location. In order for plastic rotation to localize at a single node, it may be necessary to specify a softening hinge. If the hinge is non-softening, reversal in the loading can cause a plastic rotation to occur at a different node. This can result in the buildup of equal and opposite rotations that produce kinks in the structure.

A softening plastic hinge can be simulated by specifying a plastic moment-angular displacement relation using `FISH`. `FISH` access to the plastic moment is provided via the `SET pmom_func` command. This function provides arguments (using the special function `fc_arg` – see Section 2.5.5 in the `FISH volume`) to access the average axial force in elements adjacent to the hinge node, the hinge rotation and the plastic moment. An example application of softening plastic hinges for piles is given in Section 1.5.4.3.
1.2 Beam Elements

1.2.1 Formulation

The beam elements in FLAC are standard two-dimensional beam elements with 3 degrees of freedom (two displacements and one rotation) at each end node (Figure 1.2). A typical beam element is defined by its material and geometric properties, which are assumed to be constant for each element. In general, the beam is assumed to behave as a linearly elastic material with both an axial tensile and compressive failure limit. If desired, a maximum moment (plastic moment) may also be specified. The beam is considered to have a symmetric cross-section (Figure 1.3) with area, $A$, length, $L$, second moment of area, $I$, and is defined by its endpoints, “a” and “b.”

Figure 1.2 Nomenclature for beam elements
Figure 1.3  Rectangular beam cross-section with second moment of area, \( I \), and cross-sectional area, \( A \)

The orientation of the beam in two-dimensional space is defined by its unit vectors, \( n_i, t_i \), where (Figure 1.4):

\[
\begin{align*}
  t_1 &= \frac{\Delta x}{z} = \cos \theta, \quad z = (\Delta x^2 + \Delta y^2)^{1/2} \\
  t_2 &= \frac{\Delta y}{z} = \sin \theta \\
  n_1 &= -t_2 = -\sin \theta \\
  n_2 &= t_1 = \cos \theta
\end{align*}
\]
The force vector, $F_i$, at each node can be resolved into tangential (axial) and normal (shear) component vectors:

\[
F_i = F_i^t + F_i^n = (F_{ij} t_j) t_i + (F_{ij} n_j) n_i = F_i^t t_i + F_i^n n_i
\]

where:
- $F_{ij} t_j = |F_i^t| = F^t$; and
- $F_{ij} n_j = |F_i^n| = F^n$.

Thus:

\[
F_i^{[a]} = F_i^{[a]} t_i + F_i^{[a]} n_i
\]
\[
F_i^{[b]} = F_i^{[b]} t_i + F_i^{[b]} n_i
\]

where the superscripts $[a]$ and $[b]$ identify the ends of the beam.
The component axial and shear forces and moments at each node are given by the stiffness matrix for a flexural element:

\[
\begin{bmatrix}
F^t[a] \\
F^n[a] \\
M^a \\
F^t[b] \\
F^n[b] \\
M^b
\end{bmatrix}
= K
\begin{bmatrix}
u^t[a] \\
u^n[a] \\
\theta[a] \\
u^t[b] \\
u^n[b] \\
\theta[b]
\end{bmatrix}
\] (1.7)

where: 
- \(u^t[a]\) = tangential (axial) displacement at a;
- \(u^n[a]\) = normal (shear) displacement at a;
- \(u^t[b]\) = tangential (axial) displacement at b;
- \(u^n[b]\) = normal (shear) displacement at b;
- \(\theta[a]\) = rotation at a;
- \(\theta[b]\) = rotation at b; and

\[
K = \frac{E}{L}
\begin{bmatrix}
A & & & & & & & & & \text{SYM.} \\
0 & \frac{12I}{L^2} & & & & & & & & \\
0 & \frac{6I}{L} & 4I & & & & & & & \\
-A & 0 & 0 & A & & & & & & \\
0 & -\frac{12I}{L^2} & -\frac{6I}{L} & 0 & \frac{12I}{L^2} & & & & & \\
0 & \frac{6I}{L} & 2I & 0 & -\frac{6I}{L} & 4I
\end{bmatrix}
\]
Various moment-release conditions (i.e., pinned joints) may be applied at each end node. These conditions are:

\[ M^{[a]} = 0; \]
\[ M^{[b]} = 0; \text{ and} \]
\[ M^{[a]} = M^{[b]} = 0. \]

If \( M^{[a]} = 0 \),

\[ \theta^{[a]} = \frac{1}{2} \left[ \frac{3}{L} \left( u^{n[b]} - u^{n[a]} \right) - \theta^{[b]} \right] \]

If \( M^{[b]} = 0 \),

\[ \theta^{[b]} = \frac{1}{2} \left[ \frac{3}{L} \left( u^{n[b]} - u^{n[a]} \right) - \theta^{[a]} \right] \]

If \( M^{[a]} = M^{[b]} = 0 \),

\[ \theta^{[a]} = \theta^{[b]} = 0 \]

The motion of beam-element nodes uses logic similar to that used for gridpoint translation, described in Section 1.3.3.5 in Theory and Background. Beam-element nodes that are attached to gridpoints contribute their shear and axial forces to the gridpoints to which they are attached, and translate with those gridpoints. Moments are transmitted within the beam at the points of attachment (provided the nodes are not pinned at these points), but there is no moment transmission between the grid and beam at the gridpoint (i.e., a beam attached to the grid is pin-jointed at the beam-grid connection).

### 1.2.2 Beam-Element Properties

The beam elements used in FLAC require the following input parameters (beam property keywords are shown in parentheses):

1. elastic modulus (\( e \)) [stress];
2. cross-sectional area (area or height and width or radius) [length^2];
3. second moment of area (\( i \)) [length^4] (commonly referred to as the moment of inertia);
4. spacing (spacing) [length] (optional – if not specified, beams are considered to be continuous in the out-of-plane direction);
(5) plastic moment \( (\text{pmom}) \) [force-length] (optional – if not specified, the moment capacity is assumed to be infinite);

(6) axial peak tensile yield strength \( (\text{syield}) \) [stress] (optional – if not specified, the tensile yield strength is assumed to be infinite);

(7) axial residual tensile yield strength \( (\text{syresid}) \) [stress] (optional – if not specified, the residual tensile yield strength is zero);

(8) axial compressive yield strength \( (\text{sycomp}) \) [stress] (optional – if not specified, the compressive yield strength is assumed to be infinite);

(9) density \( (\text{density}) \) [mass/volume] (optional – used for dynamic analysis and gravity loading); and

(10) thermal expansion coefficient \( (\text{thexp}) \) (optional – used for thermal analysis).

For beam elements, the height and width of the element cross-section (or the radius for a circular cross-section) can also be prescribed instead of the area and moment of inertia. The area and moment of inertia will then be calculated automatically.

Beam-element properties are easily calculated or obtained from handbooks. For example, typical values for structural steel are 200 GPa for Young’s modulus, and 0.3 for Poisson’s ratio. For concrete, typical values are 25 to 35 GPa for Young’s modulus, 0.15 to 0.2 for Poisson’s ratio, and 2100 to 2400 kg/m\(^3\) for mass density. Composite systems, such as reinforced concrete, should be based on the transformed section. Note that the beam element formulation is a plane-stress formulation. If the beam is representing a structure that is continuous in the direction perpendicular to the analysis plane (e.g., a concrete tunnel lining), the value specified for \( E \) should be divided by \( (1 - \nu^2) \) to account for plane-strain conditions.

If spaced reinforcement is to be simulated (e.g., spaced struts along a retaining wall), the spacing in the out-of-plane direction can be prescribed. The spacing parameter is used to automatically scale properties and parameters to account for the effect of the distribution of the beams over a regularly spaced pattern. See Section 1.9.4 for more information on the simulation of spaced reinforcement. Note that the actual beam properties, not scaled properties, are entered in FLAC when spacing is given.

Axial tensile and compressive yield strength limits can be specified for beams. The yield criterion is only based on axial thrust. (For yielding behavior that includes bending stresses, the liner structural element should be used. See Section 1.3.1.) A residual tensile strength limit can also be specified for tensile failure. Note that this formulation does not consider shear failure. Failure by shear can be checked by printing or plotting the shear force, dividing by the cross-sectional area and comparing the resultant shear stress with the maximum shear strength available.

A limiting plastic moment and plastic hinge condition can be prescribed for beam nodes. See Section 1.1.7 for details. Softening relations for plastic hinges can also be defined by the user.
The effect of linear thermal expansion is implemented in the beam formulation. The temperature change occurs as a result of either heat conduction or temperature re-initialization in the FLAC grid (for **CONFIG thermal**). It is assumed that the grid temperature is communicated instantaneously to the structural elements. The temperature change generates thermal expansion/contraction in the structural element axial direction; the effect of beam lateral expansion is neglected, and no other coupling takes place. The effect of heat conduction in the structural element is not considered.

The incremental axial force generated by thermal expansion in a beam element is calculated using the formula (note that compression is positive for axial forces)

\[ \Delta F = E A \alpha \Delta T \]  

where \( E \) is the Young’s modulus of the element, \( A \) is the cross-sectional area, \( \alpha \) is the linear thermal expansion coefficient, and \( \Delta T \) is the temperature increment for the element.

The structural element nodal temperature increment is determined by interpolation of nodal temperature increments in the host zone and stored in a structural node offset. The temperature change in a structural element is calculated as the average of values at the two nodes. The thermal expansion of a beam element is computed incrementally as the product of the thermal linear expansion coefficient, temperature change for the step, and element length. Thermal strains, thermal strain increments and temperatures at structural nodes are not stored.

### 1.2.3 Commands Associated with Beam Elements

All the commands associated with beam elements are listed in Table 1.2, below. This includes the commands associated with the generation of beams and those required to monitor histories, plot and print beam-element variables. See Section 1.3 in the **Command Reference** for a detailed explanation of these commands.
### Table 1.2 Commands associated with beam elements

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>beam</td>
<td>keyword</td>
</tr>
<tr>
<td>begin</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>from</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n1 &lt;nx&gt;</td>
</tr>
<tr>
<td>to</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n2</td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>segment</td>
<td>ns</td>
</tr>
<tr>
<td>interface</td>
<td>ni</td>
</tr>
<tr>
<td>delete</td>
<td>&lt;n1 n2&gt;</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>node</td>
<td>n*</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>fix</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>free</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>initial</td>
<td>keyword</td>
</tr>
<tr>
<td>xdis</td>
<td>value</td>
</tr>
<tr>
<td>xvel</td>
<td>value</td>
</tr>
<tr>
<td>ydis</td>
<td>value</td>
</tr>
<tr>
<td>yvel</td>
<td>value</td>
</tr>
<tr>
<td>rvel</td>
<td>value</td>
</tr>
<tr>
<td>load</td>
<td>fx fy m</td>
</tr>
<tr>
<td>pin</td>
<td></td>
</tr>
<tr>
<td>slave</td>
<td>&lt;x&gt; &lt;y&gt; m</td>
</tr>
<tr>
<td>unslave</td>
<td>&lt;x&gt; &lt;y&gt;</td>
</tr>
</tbody>
</table>

* For the keywords fix, free, initial, load and pin, a range of nodes can be specified with the phrase range n1 n2.
<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>prop</th>
<th>np</th>
<th>keyword</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>area</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>density</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>e</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>height</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>pmom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>radius</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>spacing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sycomp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>syield</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>syresid</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>thexp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>width</td>
<td></td>
</tr>
<tr>
<td>chprop</td>
<td>np</td>
<td>range</td>
<td>nel1 nel2</td>
<td></td>
</tr>
<tr>
<td>hinge</td>
<td>nel1 nel2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HISTORY</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td>aaccel</td>
</tr>
<tr>
<td></td>
<td>adisp</td>
</tr>
<tr>
<td></td>
<td>avel</td>
</tr>
<tr>
<td></td>
<td>xaccel</td>
</tr>
<tr>
<td></td>
<td>xdisp</td>
</tr>
<tr>
<td></td>
<td>xvel</td>
</tr>
<tr>
<td></td>
<td>yaccel</td>
</tr>
<tr>
<td></td>
<td>ydisp</td>
</tr>
<tr>
<td></td>
<td>yvel</td>
</tr>
<tr>
<td>element</td>
<td>nel</td>
</tr>
<tr>
<td></td>
<td>keyword</td>
</tr>
<tr>
<td></td>
<td>axial</td>
</tr>
<tr>
<td></td>
<td>moment1</td>
</tr>
<tr>
<td></td>
<td>moment2</td>
</tr>
<tr>
<td></td>
<td>shear</td>
</tr>
</tbody>
</table>
Table 1.2  Commands associated with beam elements (continued)

<table>
<thead>
<tr>
<th>PLOT</th>
<th>beam</th>
<th>structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;beam&gt; keyword</td>
<td></td>
</tr>
<tr>
<td></td>
<td>adisp &lt;ng &lt;ng2*&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>avel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>axial &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>element</td>
<td></td>
</tr>
<tr>
<td></td>
<td>load</td>
<td></td>
</tr>
<tr>
<td></td>
<td>location &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>material</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mome &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>node</td>
<td></td>
</tr>
<tr>
<td></td>
<td>number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sdisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shear &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>strain &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>svel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xdisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xvel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ydisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yvel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRINT</th>
<th>structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>keyword</td>
</tr>
<tr>
<td></td>
<td>beam</td>
</tr>
<tr>
<td></td>
<td>hinge</td>
</tr>
<tr>
<td></td>
<td>node</td>
</tr>
<tr>
<td></td>
<td>property</td>
</tr>
<tr>
<td></td>
<td>beam</td>
</tr>
</tbody>
</table>

* A range of group ID numbers can be specified for plotting by giving a beginning number ng and an ending number ng2. All groups within this range will be plotted.
1.2.4 Example Applications

A few simple examples are given to illustrate the implementation of the structural element commands for beam elements.

1.2.4.1 Simple Beam – Two Equal Concentrated Loads

A simply supported beam is loaded by two equal concentrated loads symmetrically placed, as shown in Figure 1.5. The shear and moment diagrams for this configuration are also shown in the figure. The shear force magnitude, \( V \), is equal to the applied concentrated load \( P \). The maximum moment, \( M_{\text{max}} \), occurs between the two loads and is equal to \( Pa \). The maximum deflection of the beam, \( \Delta_{\text{max}} \), occurs at the center and is given by AISC (1980) pp. 2-116 as

\[
\Delta_{\text{max}} = \frac{Pa}{24EI} (3L^2 - 4a^2)
\]  

(1.12)

where:  
\( E \) = Young’s modulus; and  
\( I \) = second moment of inertia.

![Figure 1.5 Simply supported beam with two equal concentrated loads (distance in units of meters)](image)
The following properties are used in this example:

- cross-sectional area \((A)\) 0.006 m\(^3\)
- Young’s modulus \((E)\) 200 GPa
- Poisson’s ratio \((\nu)\) 0.30
- second moment of inertia \((I)\) 200 \(\times\) 10\(^{-6}\) m\(^4\)

Point loads of \(P = 10,000\) N are applied at the two locations shown in Figure 1.5.

The FLAC model consists of 10 beam segments and 11 nodes, as shown in Figure 1.6. The beam is created with three separate **STRUCT beam** commands, to ensure that nodes will lie exactly at the beam third points. Also, four segments are created in the middle third to ensure that a node will lie at the exact beam center so that the displacement of this node can be compared with \(\Delta_{\text{max}}\). Simple supports are specified at the beam end nodes by restricting translation in the \(y\)-direction. Two point loads acting in the negative \(y\)-direction are applied at the beam third points. The data file for this model is listed in Example 1.2.

![Figure 1.6 FLAC model for simple beam problem showing segment ID numbers](FLAC Version 6.0)
**Example 1.2  Simple beam – two equal concentrated loads**

```plaintext
struct node 1 0.0,0.0
struct node 2 3.0,0.0
struct node 3 6.0,0.0
struct node 4 9.0,0.0
struct beam begin node 1 end node 2 seg 3 prop 1001
struct beam begin node 2 end node 3 seg 4 prop 1001
struct beam begin node 3 end node 4 seg 3 prop 1001
struct prop 1001 e 2.0E11 area 0.0060 I 2.0E-4
struct node 1 fix y
struct node 4 fix y
struct node 2 load 0.0,-10000.0 0.0
struct node 3 load 0.0,-10000.0 0.0
history 1 node 8 ydisplace
history 4 element 1 moment2
history 5 element 2 moment1
history 999 unbalanced
solve
save se_01_02.sav
```

The displacement field is shown in Figure 1.7. The maximum displacement occurs at the beam center and equals $6.468 \times 10^{-3}$ m, which is within 0.1% of the theoretical value of Eq. (1.12).

Figures 1.8 and 1.9 show the shear force and moment distributions, which also correspond with the theoretical solutions.

The evolution of the moment at $x = 1$ is shown in Figure 1.10 to reach a steady-state value of 10,000 N-m. In this plot, we overlay two histories, one of which has sampled the moment acting at the right end of beam segment 1, and the other has sampled the moment acting at the left end of segment 2. If expressed in a consistent system, these two values should be identical, and the plot demonstrates that they are. Note that the moment acting on the right end of segment 1 has a positive sign, while the moment acting on the left end of segment 2 has a negative sign. This is the correct behavior that satisfies equilibrium.
**Figure 1.7** Displacement field of simple beam

**Figure 1.8** Shear force distribution in simple beam
**Figure 1.9**  Moment distribution in simple beam

**Figure 1.10**  Evolution of moment at $x = 1$ in simple beam
A cantilever beam is subjected to an applied moment at its tip, as shown in Figure 1.11. This problem is an example of geometric nonlinearity, whereby deformations significantly alter the location of loads, so that equilibrium equations must be written with respect to the deformed geometry. Such problems can be solved by running FLAC in large-strain mode. The large-strain \( y \)-direction deflection at the beam tip (assuming that the material remains linearly elastic) is given by Cook et al. (1989), pp. 529-531 as

\[
v_{\text{tip}} = \frac{EI}{M} \left( 1 - \cos \left( \frac{ML}{EI} \right) \right)
\]  

(1.13)

where: \( E \) = Young’s modulus; and \( I \) = second moment of inertia.

The following properties and loading conditions are used in this example:

- Cross-sectional area \( A \) = 0.006 m\(^3\)
- Young’s modulus \( E \) = 200 GPa
- Poisson’s ratio \( \nu \) = 0.30
- Second moment of inertia \( I \) = 200 \( \times 10^{-6} \) m\(^4\)
- Beam length \( L \) = 10 m
- Applied moment at tip \( M \) = 5 \( \times 10^6 \) N-m

For these conditions, the theoretical tip deflection, \( v_{\text{tip}} \), is given by Eq. (1.13) to be 5.477 m.

The FLAC model consists of 10 segments and 11 nodes. The left end is fully fixed, and a moment vector aligned with the \( z \)-direction is applied to the node at the beam tip. The data file for this example is listed in Example 1.3.
Example 1.3  Cantilever beam with applied moment at tip

```
struct node 1 0.0,0.0
struct node 2 10.0,0.0
struct beam begin node 1 end node 2 seg 10 prop 1001
struct node 1 fix x y r
struct node 2 load 0.0,0.0 5000000.0
history 1 node 2 ydisplace
struct prop 1001 e 2.0E11 area 0.0060 I 2.0E-4
set large
set force 100
history 999 unbalanced
solve
save se_01_03.sav
```

The final structural configuration is shown in Figure 1.12. The y-direction deflection at the beam tip equals 5.505 m, which is within 0.5% of the analytical solution.
1.2.4.3  Buckling of an Axially Loaded Beam

This example demonstrates the buckling behavior of an axially loaded beam with small initial deflection. The beam rests on a base and is fixed in lateral translation at its ends. A global system of coordinates is defined with the $y$-axis pointing upwards, oriented along the axis of the beam, and with origin at the base of the beam. This solution is taken from Massonnet (1960). The beam initial shape is defined by the equation

$$x_0 = f_0 \sin \left( \frac{\pi y}{l} \right)$$

(1.14)

where $l$ is beam length, and $f_0$ is maximum initial deflection.

The additional deflection taken by the beam under an axial load $P$ is predicted from linear stability analysis by the equation

$$x - x_0 = f_0 \sin \left( \frac{\pi y}{l} \right) \frac{P_{cr}}{P} - 1$$

(1.15)

where $P_{cr}$ is the minimum critical load for buckling.

The minimum critical load is defined as

$$P_{cr} = \frac{\pi^2 EI}{l^2}$$

(1.16)

where $E$ is Young’s modulus and $I$ is moment of inertia for lateral flexion.

The additional deflection at the center of the beam $f$ is thus

$$f = \frac{f_0}{P_{cr}/P - 1}$$

(1.17)

For this example, the beam is 200 m long, and the maximum amplitude of initial deflection is 1 cm (or 0.005% of the beam length). Young’s modulus is 257 MPa, and the moment of inertia for lateral flexion is 5.333 m$^4$. The beam is modeled using 100 elements, translation is fixed in both directions at the base, and in the $x$-direction at the top. An axial load is applied in increments at the beam top until the critical load is reached. After each increment, the model is cycled to mechanical equilibrium and the load deflection is recorded in a table (see Example 1.4).

The beam deflection is seen to increase beyond measure as the load converges to the minimum critical value, as expected. A comparison between analytical solution and numerical prediction for
additional deflection at the center of the beam is presented in Figure 1.13. As may be seen, the match between the solutions is very good.

Figure 1.13  Load deflection curve

Example 1.4  Axial loading of a beam with small initial deflection

; Axial loading of a beam with small initial deflection.
; Load versus additional deflection at beam half length:
; comparison between
; - numerical prediction  (table 1)
; - minimum critical load  (table 3)
; - analytical solution  (table 4)
; from Linear Stability Analysis in [Massonnet,
;  Resistance des Materiaux, 1960]
;
def setup
  _h   = 200.
  _E   = 2.57e8
  _I   = 5.333
  _A   = 4.
  Pcr  = (pi/_H)^2 * _E * _I
  _f0  = 0.01
  _steps = 10
\begin{verbatim}
ii = 0
deltaP = Pcr / float(_steps + 1)
_nid2 = 1
x2val = 0.0
y2val = 0.0
_nid1 = _nid2
x1val = x2val
y1val = y2val
command
  struct node _nid1 x1val y1val
endcommand
loop jj (1,100)
  y2val = float(jj)*h/100.
  x2val = _f0 * sin(pi*y2val/_h)
  _nid2 = _nid1 + 1
  command
    struct node _nid2 x2val y2val
    struct beam begin node _nid1 end node _nid2 seg 1 prop 1001
  end_command
  _nid1 = _nid2
  x1val = x2val
  y1val = y2val
end_loop
command
  set = large
  struct prop 1001 e _E area _A I _I
  struct node 1 fix x y
  struct node _nid2 fix x
  his 1 node _nid2 xdisp
  his 2 node _nid2 ydisp
end_command
end
setup
call str.fin
def store_it
  np = imem(str_pnt+$ksnode)
loop while np # 0
  _id = imem(np+$kndid)
  if _id = 51 then
    xval = fmem(np+$kndx)
    deflec = abs(xval) - _f0
    table(1,deflec) = thisload ; numerical sol.
    xval = _f0/(Pcr/thisload - 1.)
    table(4,xval) = thisload ; analytic sol.
    table(3,xval) = Pcr
  endif
end
\end{verbatim}
np = imem(np)
end_loop
end

def load_deflection
    thisload = deltaP * float(ii)
    mthisload = -thisload
    command
        struct node _nid2  load 0. mthisload 0.
        solve sratio 1e-7 force 0.0 step 50000000
    end_command
    store_it
end
set ii = 1
load_deflection
save f1.sav
set ii=2
load_deflection
save f2.sav
set ii=3
load_deflection
save f3.sav
set ii=4
load_deflection
save f4.sav
set ii=5
load_deflection
save f5.sav
set ii=6
load_deflection
save f6.sav
set ii=7
load_deflection
save f7.sav
set ii=8
load_deflection
save f8.sav
set ii=9
load_deflection
save f9.sav
set ii=10
load_deflection
save f10.sav
set ii=10.2
load_deflection
save f10p2.sav
set ii=10.4

FLAC Version 6.0
1.2.4.4 Plastic Hinge Formation in a Beam Structure

Simple examples are presented to illustrate the development of a plastic hinge in beam elements. The first example is a concentrated load, $P$, applied at the center of a 10 m long, simply supported beam with a plastic-moment capacity, $M_P$, of 25 kN-m. The system, along with the shear and moment diagrams, is shown in Figure 1.14. From these shear and moment diagrams, we find that the specified plastic-moment capacity corresponds to a maximum vertical load of 10 kN and a maximum shear force of 5 kN. If we apply a constant vertical velocity to the beam center, we expect that the limiting values of moment and shear force will be 25 kN-m and 5 kN, respectively.

![Figure 1.14 Simple beam with single concentrated load](image)
The *FLAC* model is created by issuing a single `STRUCT beam` command and specifying two segments. The beam is assigned the same properties as in Section 1.2.4.1 but, in addition, the plastic-moment capacity is set to 25 kN-m with the `STRUCT prop pmom` command. Simple supports are specified at the beam ends by restricting translation in the y-direction. A constant vertical velocity is applied at the center node, and the moment and shear force acting at the right end of beam segment 1 and left end of segment 2 are monitored during the calculation to determine when the limiting value is reached. Note that we specify combined local damping for this problem (SET `st damping struct combined`) in order to eliminate the ringing that can occur with the default damping scheme when the system is driven by a constant motion.

**Example 1.5 Plastic hinge formation**

```
struct node 1 0.0,0.0
struct node 2 10.0,0.0
struct beam begin node 1 end node 2 seg 2 prop 1001
struct prop 1001 e 2.0E11 area 0.0060 I 2.0E-4 pmom 25000.0
struct node 1 fix y
struct node 2 fix y
struct node 3 fix y initial yvel=-5.0E-6
struct hinge 1 2
history 1 node 3 ydisplace
history 2 element 1 moment1
history 3 element 1 moment2
history 4 element 2 moment1
history 5 element 2 moment2
history 6 element 1 shear
history 7 element 2 shear
history 8 node 3 adisplacement
set st damping struct=combined 0.8
set large
history 999 unbalanced
cycle 3000
save se_01_05.sav
```

We find that the limiting values of moment and shear force are equal to the analytical values of 25 kN-m and 5 kN, respectively (see Figures 1.15 and 1.16). Also, the moment and shear force distributions correspond with the analytical solution (see Figures 1.17 and 1.18). We also see that a discontinuity in the rotational motion develops at the center location; the rotation at the center is nonzero (PRINT `struct node hinge`).

Note that if the command `STRUCT hinge 1 2` is removed from Example 1.5, the limiting values of moment and shear force also match the analytical values. However, the rotation at the center load is now zero. If it is only necessary to determine the solution of limiting plastic moment, then it is not necessary to define locations of plastic hinges. However, plastic hinges should be specified when running in large-strain mode in order to calculate the post-failure behavior of the structure.
Figure 1.15  Moment at right end of segment 1 and left end of segment 2 versus applied center displacement

Figure 1.16  Shear force at right end of segment 1 and left end of segment 2 versus applied center displacement
Figure 1.17  Moment distribution at limit condition

Figure 1.18  Shear force distribution at limit condition
If we continue loading a structure in a large-strain fashion, then the addition of the plastic hinge will allow a discontinuity to develop in the rotation at the center after the plastic moment has been reached. We illustrate this behavior by modifying the previous example to represent a cantilever beam (fixed at the left end) with a vertical load applied at the free end (see Example 1.6). The problem is run in large-strain mode.

The final structural configuration and moment distribution are shown in Figure 1.19. We see that a discontinuity develops in the rotation at the beam center.

**Example 1.6  Cantilever beam with a plastic hinge**

```
struct node 1 0.0,0.0
struct node 2 10.0,0.0
struct beam begin node 1 end node 2 seg 2 prop 1001
struct prop 1001 e 2.06E11 area 0.0060 I 2.0E-4 pmom 25000.0
struct node 1 fix x y r
struct node 2 load 0 -5.5e3 0
struct hinge 1 2
history 1 node 3 ydisplace
history 2 element 1 moment1
history 3 element 1 moment2
history 4 element 2 moment1
history 5 element 2 moment2
history 6 element 1 shear
history 7 element 2 shear
history 8 node 3 adisplacement
set st_damping struct=combined 0.8
set large
history 999 unbalanced
cycle 3000
save se_01_06.sav
```
Figure 1.19  Final structural configuration and moment distribution in beam cantilever with plastic hinge
1.2.4.5  **Braced Excavation**

The tutorial example in Section 2.2.4 in the User’s Guide illustrates a collapsing trench. Here, we support this trench with two struts that brace the excavation walls. The following command sequence is for the simple case in which the braces are placed immediately upon excavation.

Note that the supports are assumed to be installed at 1 meter spacing. The spacing of supports at a different interval can be modeled by scaling the material properties of the structural elements. (See Section 1.9.4.)

**Example 1.7  Braced support of a vertical excavation**

```plaintext
grid 5,5
m mohr
prop b=1e8  s=.3e8  d=1000  fric=35  coh=0.0  ten=0.0
fix y  j=1
fix x  i=1
fix x  i=6
set large
hist nstep=1
hist ydis  i=3  j=6
set grav=9.81
solve elastic
; excavate trench and install braces
model null  i=3  j=3,5
init xdis=0  ydis=0
; properties for W6x25 beam in SI units
struc prop=1001  E=200e9  I=2.3e-5  area=4.8e-3
struc beam beg gr=3,6  end gr=4,6  seg=3  pr=1001
struc beam beg gr=3,4  end gr=4,4  seg=3  pr=1001
step 100
plot hold xv fill z  int=5e-6  dis max=1e-2  bou beam lmag
```
Figure 1.20 illustrates the effect of the two braces (compare to Figure 2.31 in the User’s Guide). Trench collapse will still occur for this model, but the failure region is reduced. Additional bracing and/or sheet piling (represented by vertical beams, liners or piles) may be used to stabilize the trench.

1.2.4.6 Cross-Braced Structure on Soil Foundation

This example illustrates the loading of a foundation by a surface structure. Here, a simple cross-braced platform is constructed on a concrete slab which rests on a soil mass. The structure is loaded with vertical point loads on the supporting columns. The object is to examine the loads and moments in the structure, as well as the stresses and displacements induced in the soil mass.

The command structure in Example 1.8 is used to set up and run this problem.

Example 1.8 Cross-braced structure on soil foundation

```
; a simple cross-braced structure on a soil
grid 10,10
; the soil mass
me
prop s=.3e8 b=1e8 d=1600
fix x i=1
fix x i=11
```
fix y  j=1
set grav=9.81
solve
save se_01_07a.sav
; let soil equilibrate under gravity
; build structure
; concrete slab
struc prop=1001 E=17.58e9  I=0.0104  a=.5
struc prop=1002 E=200e9  I=2.3e-5  a=4.8e-3
struc beam beg gr 5,11 end gr 7,11 seg=1 pr=1001
struc beam beg node 1 end 4,13 seg=2 pr=1002
struc beam beg 4,13 end 6,13 seg=2 pr=1002
struc beam beg 6,13 end 6,10 seg=2 pr=1002
struc node=8 5.0,11.5
struc beam beg node=8 end node=1 seg=1 pr=1002
struc beam beg node=8 end node=4 seg=1 pr=1002
struc beam beg node=8 end node=6 seg=1 pr=1002
struc beam beg node=8 end node=2 seg=1 pr=1002
struc node=1  fix r
struc node=2  fix r
struc node=4  Load 0 -1e6 0
struc node=6  Load 0 -1e6 0
; check structure
plot hold beam grid
pause
; check linkage
pr struc beam
hist node 1 adisp avel xdisp xvel ydisp yvel
hist node 1 ndisp nforce sdisp yforce
hist ele 2 axial moment 1 moment 2
solve
plot hold syy fill beam bou
print struc beam
save se_01_07b.sav

The problem configuration is shown in Figure 1.21. The effect of loading the soil with the structure is illustrated in Figure 1.22. The forces and moments developed in the structure are available via the command PRINT struct beam.
Figure 1.21  FLAC grid and cross-braced structure

Figure 1.22  Vertical stresses after structural loading
1.2.4.7  *Shaft Excavation with Structural Lining*

This example problem examines the loads that develop in shotcrete and concrete liners for a circular shaft in a biaxial stress field. Here, beam elements are used to represent the lining in direct contact with the rock mass. Interface elements could also be used to simulate the effect of slip between the lining and the rock by adding the optional `interface` keyword to the `STRUCT beam` command.

For this case, the development of the grid and liner is accomplished by the commands given in Example 1.9.

### Example 1.9  Shaft excavation with a structural lining

```plaintext
config extra 1
grid 15,15
m mohr
  gen 14,14 14,60 60,60 60,14 rat 1.2 1.2 i=8,16 j=8,16
  gen 14,-30, 14,14 60,14 60,-30 rat 1.2 .833 i=8,16 j=1,8
  gen -30,14 -30,60 60,60 14,14 rat .833 1.2 i=1,8 j=8,16
  gen circle 14,14 4
  gen adjust
prop s=5.75e9 b=6.6e9 d=2000 coh=1e7 fric=35
  fix x y i=1
  fix x y i=16
  fix x y j=1
  fix x y j=16
ini sxx=-60e6 syy=-30e6 szz=-60e6
solve elastic
save se_01_09a.sav
mod null region=8,8
struc prop=1001 E=13.8e9 I=2.8e-4 a=.15
  struct beam long from 9,9 to 9,9 prop 1001
step 500
wind 0 28 0 28
sclin 1 0,0 28,28
  plot hold xdisp int 2e-3 struct axial fill max -3e7 beam white
  save se_01_09b.sav
```

The x-displacement contours around the lined shaft and axial forces in the liner are illustrated in Figure 1.23. Note that the negative maximum value for the axial force given in the `PLOT struct` command changes the sense of the axial force plot so that the plot is displayed outside the shaft boundary.
Figure 1.23  x-displacement contours around lined shaft and axial forces in liner

The loads and moments in the liner elements are listed (from PRINT struct beam) in Example 1.10. Note that the moments are equal and opposite in sign at connecting nodes between element segments, which indicates that the model is at an equilibrium state.

Example 1.10 Results of PRINT struct beam for tunnel liner example

<p>| Structural element data ... |
|-----------------------------|------------------------|-----------------|-----------------|-----------------|-----------------|-------------|</p>
<table>
<thead>
<tr>
<th>Elem ID</th>
<th>Nod1</th>
<th>Nod2</th>
<th>Prop</th>
<th>F-shear</th>
<th>F-axial</th>
<th>Mom-1</th>
<th>Mom-2</th>
<th>strain</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>1001</td>
<td>4.886E+03</td>
<td>4.827E+06</td>
<td>1.039E+04</td>
<td>4.572E+03</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>1001</td>
<td>-4.523E+03</td>
<td>4.419E+06</td>
<td>-5.094E+03</td>
<td>-1.039E+04</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>1001</td>
<td>1.218E+03</td>
<td>9.035E+06</td>
<td>5.904E+03</td>
<td>1.815E+03</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>1001</td>
<td>-2.065E+02</td>
<td>8.112E+06</td>
<td>-2.447E+03</td>
<td>1.815E+03</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1001</td>
<td>3.124E+03</td>
<td>4.093E+06</td>
<td>7.116E+03</td>
<td>2.447E+03</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1001</td>
<td>-3.287E+03</td>
<td>4.122E+06</td>
<td>-1.734E+03</td>
<td>-7.116E+03</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1001</td>
<td>-8.458E+01</td>
<td>7.684E+06</td>
<td>-2.024E+03</td>
<td>1.734E+03</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1001</td>
<td>-8.322E+02</td>
<td>8.839E+06</td>
<td>-4.572E+03</td>
<td>2.024E+03</td>
</tr>
</tbody>
</table>
1.2.4.8 Modeling Geo-textiles

Geo-textiles are used for a variety of purposes in geotechnical engineering, including the construction of reinforced embankments, retaining soil structures, and impermeable barriers at the base of leaching piles at mines. In order to effectively model geo-textiles, it is necessary to account for both the behavior of the flexible fabric and its interaction with the soil above and below. One way to perform such modeling is to use beam elements attached to sub-grids on both sides with interfaces. By assigning the beam a zero moment of inertia, it will act like a flexible member that takes no moments. Sliding is possible on both sides of the “fabric,” with friction angles determined by the two interface properties. Pullout and large-strain effects are also possible. Example 1.11 illustrates the approach:

**Example 1.11 Geo-textile model**

```plaintext
grid 5 7
model elas
prop dens 1000 bulk 2e8 shear 1e8
model null j 4
ini y add -1 j=5,8
struct beam beg 0,3 end 5,3 seg 8 prop 1001
struct prop 1001 a=0.01 i=0 e=1e10
int 1 as from node 1 to node 9 bs from 6,5 to 1,5
int 1 kn 1e8 ks 1e8 fric 0.5
int 2 as from node 9 to node 1 bs from 1,4 to 6,4
int 2 kn 1e8 ks 1e8 fric 0.5
fix x y j=1
fix x y j=8
ini yvel -0.5e-4 j=8
ini yvel 0.5e-4 j=1
set large
cyc 200
ini xvel 0 yvel 0
cyc 800
save se_01_11a.sav
ini xvel 1e-4 j=8
cyc 2000
plot hol bou stres struc axial fil whit max -450 iface 1 sstress fill red
save se_01_11b.sav
```

In this simple example, loading is first applied normal to the beam-element geo-textile. Then the upper sub-grid is displaced relative to the lower sub-grid so that sliding occurs along the geo-textile fabric. The axial forces in the beam elements and the shear stress along the lower interface are plotted in Figure 1.24.
If this model is used for a groundwater flow analysis, the interface/fabric/interface layer will act as an impermeable barrier to fluid flow. However, with a FISH function that controls APPLY discharge sources on opposing grid segments, the effect of a leaky layer may be modeled.

![Diagram of geo-textile beam and shear stresses](image)

**Figure 1.24** Axial forces in the geo-textile beam and shear stresses along the lower interface

This simulation of geo-textiles assumes that the fabric separates the soil above from the soil below the fabric. Alternatively, the reinforcement may be embedded within the soil, such as soil nails or a geo-grid. In this case, the reinforcement acts to improve the shear resistance of the soil, and cable elements may provide a more reasonable representation. See Section 1.4.6.2 for an example application.
1.2.4.9  Tunnel with Yielding Steel Arch and Interface

A 4 m wide by 4 m high tunnel is located at a shallow depth. The tunnel is supported by a yielding steel arch which has a axial capacity that is limited by the friction joint in the top section steel set of the arch. The maximum yield stress is 4 MPa, which results in a maximum axial force of 7.2 kN for the given arch dimensions. The properties for the arch and parameters for the problem setting are shown in the data file listed in Example 1.12.

The friction joints along the arch are specified by using the `STRUCT chprop` command to assign lower compressive-yield properties at quarter points along the arch. The results shown in Figure 1.25 demonstrate that the axial force in the arch segments representing friction joints (group IDs 2 and 4 corresponding to segments 5 and 8) is 7.2 kN.

**Example 1.12 Tunnel with yielding arch and interface**

```plaintext
title Horseshoe excavation with steel arch and interface
grid 10 10
m m
gen arc 5 5 7 5 180
prop dens 2000 bulk 33.333e6 shear 20.0e6 fric 35 coh=50e3
fix y j 1
fix x i 1
fix x i 11
set grav 10
solve
ini xdisp 0 ydisp 0
m null i 4 7 j 4 6
m null i 5 6 j 7
; steel arch lining TH-13
struct prop 1001 dens 8000 e 2e1 i=1.37e-6 area=0.0018 sycomp=400e6
struct beam long from 4 4 to 8 4 int 1 prop 1001
interf 1 kn 2e8 ks 2e8 fric 35 coh 50e3
; yielding segments in crown
struct prop 1002 dens 8000 e 2e11 i=1.37e-6 area=0.0018 sycomp=4e6
struct chprop 1002 range 5 5
struct chprop 1002 range 8 8
; histories
hist unbal
def closure
  closure=(ydisp(6,4)-ydisp(6,8))/(y(6,8)-y(6,4))
end
hist closure
solve
```

Figure 1.25  Distribution of axial forces along yielding arch
1.3 Liner Elements

1.3.1 Formulation

The liner-element formulation is similar to the beam-element formulation as described in Section 1.2.1. Liners are two-dimensional elements with 3 degrees of freedom (two displacement and one rotation) at each end node.

In contrast to the beam elements, liner elements include an elastic-plastic material model that incorporates bending resistance, limiting bending moments and yield strengths of the liner material.* This model can simulate inelastic behavior representative of common surface-lining materials. This includes materials that behave in a ductile manner, such as steel, as well as un-reinforced and reinforced cementitious materials, such as concrete and shotcrete, that can exhibit either brittle or ductile behavior. Note that shear failure is not included in either the beam or liner material model.

The behavior of the elastic-plastic material model can be shown on a moment-thrust interaction diagram, such as that given in Figure 1.26. Moment-thrust diagrams are commonly used in the design of concrete columns. These diagrams illustrate the maximum force that can be applied to a typical section for various eccentricities. The ultimate failure envelopes for un-reinforced and reinforced cementitious materials are similar. However, reinforced materials have a residual capacity that remains after failure at the ultimate load. Un-reinforced cementitious materials typically have no residual capacity.

* The liner material model was developed in collaboration with Geocontrol S.A., Madrid, Spain, and the Norwegian Geotechnical Institute, Oslo, Norway, for application to the analysis of yielding arch supports in tunnels. For further information on the model, see Chryssanthakis et al. (1997).
Interaction diagrams can be constructed by knowing or specifying the section geometry and compressive and tensile strengths (in terms of stress) for the material. The thickness and compressive and tensile strengths are input, and the model uses this information to determine the ultimate capacity for various eccentricities ($e$ in Figure 1.26). As the calculation progresses, the axial forces and moments in the structural elements are compared to the ultimate capacity. When a node reaches the ultimate capacity, a “fracture” flag is set, indicating that all future evaluations for that node will use the “cracked” failure envelope and the residual strength capacity.

The following procedure is used to calculate liner forces and moments. First, incremental forces and moments are calculated from incremental displacements during a timestep, assuming a linearly elastic stress-strain relation. These incremental forces and moments are added to the total values for axial force and moment in the liner element at that step. If these values plot outside the ultimate failure envelope, such as that shown in Figure 1.26, then the total force and moment are adjusted to return the values to the failure surface. The axial force and moment adjustment is an interpolation based upon the location of the unadjusted force-moment point relative to the balanced point and the slope of the failure envelope. Note that this is not related to a plasticity flow rule.

If a residual strength is specified for the liner, then future adjustments for the force and moment will return values to the “cracked” failure envelope.

### 1.3.2 Liner-Element Properties

The liner elements used in FLAC require the following input parameters (liner property keywords are shown in parentheses):

1. elastic modulus ($E$) [stress];
2. Poisson’s ratio ($\nu$);
3. cross-sectional area ($A$ or $h$ and $w$) [length$^2$];
4. thickness ($t$) [length];
5. second moment of area ($I$) [length$^4$] (commonly referred to as the moment of inertia);
6. cross-sectional shape factor ($S$) – see Figure 1.27;
7. spacing ($s$) [length] (optional – if not specified, liners are considered to be continuous in the out-of-plane direction);
8. axial peak tensile yield strength ($f_y$) [stress] (optional – if not specified, the tensile yield strength is assumed to be infinite);
9. axial residual tensile yield strength ($f_{yres}$) [stress] (optional – if not specified, the residual tensile yield strength is zero);
(10) axial compressive yield strength (sycomp) [stress] (optional – if not specified, the compressive yield strength is assumed to be infinite);

(11) density (density) [mass/volume] (optional – used for dynamic analysis and gravity loading); and

(12) thermal expansion coefficient (thexp) (optional – used for thermal analysis).

For liner elements, the height and width of the element cross-section can be given instead of the area, thickness and moment of inertia. The area and moment of inertia will then be calculated automatically. Note that either the thickness or height parameter must be given to calculate the bending stresses for the liner yielding criterion.

The shape factor adjusts the cross-sectional area to account for shear deformations induced by different liner shapes. For rectangular shapes, the shape factor is $5/6$. This is the default value if the shape factor is not specified. Figure 1.27 lists shape factors and inertial moments for various shapes. See Boresi et al. (1993) for additional information on shape factors.

![Figure 1.27 Shape factors and inertial moments for different shapes](image)

Liner-element properties are calculated or obtained from handbooks in the same manner as beam-element properties. See Section 1.2.2 for recommendations on determining liner properties.

Axial tensile and compressive yield strength limits can be specified for liners. The yield criterion is based on axial thrust and bending stresses. A residual tensile strength limit can also be specified for tensile failure. The criterion is illustrated by the moment-thrust diagram in Figure 1.26. Note
that this formulation does not consider shear failure. Failure by shear can be checked by printing or plotting the shear force, dividing by the cross-sectional area and comparing the resultant shear stress with the maximum shear strength available.

The effect of linear thermal expansion is also implemented in the liner formulation. This formulation is identical to that for beam elements. See Section 1.2.2 for details.

1.3.3 Commands Associated with Liner Elements

All the commands associated with liner elements are listed in Table 1.3, below. This includes the commands associated with the generation of liners and those required to monitor histories, plot and print rockbolt-element variables. See Section 1.3 in the Command Reference for a detailed explanation of these commands.
### Table 1.3  Commands associated with liner elements

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>liner</td>
<td>keyword</td>
</tr>
<tr>
<td>begin</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>xy</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>xy</td>
<td></td>
</tr>
<tr>
<td>from</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n1 &lt;nx&gt;</td>
</tr>
<tr>
<td>to</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n2</td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>segment</td>
<td>ns</td>
</tr>
<tr>
<td>interface</td>
<td>ni</td>
</tr>
<tr>
<td>delete</td>
<td>&lt;n1 n2&gt;</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>xy</td>
<td></td>
</tr>
<tr>
<td>node</td>
<td>n*</td>
</tr>
<tr>
<td>keyword</td>
<td></td>
</tr>
<tr>
<td>fix</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>free</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>initial</td>
<td>keyword</td>
</tr>
<tr>
<td>xdis</td>
<td>value</td>
</tr>
<tr>
<td>xvel</td>
<td>value</td>
</tr>
<tr>
<td>ydis</td>
<td>value</td>
</tr>
<tr>
<td>yvel</td>
<td>value</td>
</tr>
<tr>
<td>rvel</td>
<td>value</td>
</tr>
<tr>
<td>load</td>
<td>fx fy m</td>
</tr>
<tr>
<td>pin</td>
<td></td>
</tr>
<tr>
<td>slave</td>
<td>&lt;x&gt; &lt;y&gt; m</td>
</tr>
<tr>
<td>unslave</td>
<td>&lt;x&gt; &lt;y&gt;</td>
</tr>
</tbody>
</table>

* For the keywords **fix**, **free**, **initial**, **load** and **pin**, a range of nodes can be specified with the phrase **range n1 n2**.
**Table 1.3  Commands associated with liner elements (continued)**

<table>
<thead>
<tr>
<th>STRUCTURE prop np</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>area</td>
<td>value</td>
</tr>
<tr>
<td>density e</td>
<td>value</td>
</tr>
<tr>
<td>height i</td>
<td>value</td>
</tr>
<tr>
<td>pratio</td>
<td>value</td>
</tr>
<tr>
<td>shape</td>
<td>value</td>
</tr>
<tr>
<td>spacing</td>
<td>value</td>
</tr>
<tr>
<td>sycomp</td>
<td>value</td>
</tr>
<tr>
<td>syield</td>
<td>value</td>
</tr>
<tr>
<td>syresid</td>
<td>value</td>
</tr>
<tr>
<td>thexp</td>
<td>value</td>
</tr>
<tr>
<td>thickness</td>
<td>value</td>
</tr>
<tr>
<td>width</td>
<td>value</td>
</tr>
<tr>
<td>chprop np range</td>
<td>nel1 nel2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HISTORY node n</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaccel</td>
<td></td>
</tr>
<tr>
<td>adisp</td>
<td></td>
</tr>
<tr>
<td>avel</td>
<td></td>
</tr>
<tr>
<td>xaccel</td>
<td></td>
</tr>
<tr>
<td>xdisp</td>
<td></td>
</tr>
<tr>
<td>xvel</td>
<td></td>
</tr>
<tr>
<td>yaccel</td>
<td></td>
</tr>
<tr>
<td>ydisp</td>
<td></td>
</tr>
<tr>
<td>yvel</td>
<td></td>
</tr>
<tr>
<td>element nel</td>
<td>keyword</td>
</tr>
<tr>
<td>axial</td>
<td></td>
</tr>
<tr>
<td>moment1</td>
<td></td>
</tr>
<tr>
<td>moment2</td>
<td></td>
</tr>
<tr>
<td>shear</td>
<td></td>
</tr>
</tbody>
</table>
### Table 1.3  Commands associated with liner elements (continued)

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT</td>
<td>liner</td>
</tr>
<tr>
<td>structure</td>
<td>&lt;liner&gt;</td>
</tr>
<tr>
<td></td>
<td>keyword</td>
</tr>
<tr>
<td></td>
<td>adisp &lt;ng &lt;ng2*&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>avel &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>axial &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>element</td>
</tr>
<tr>
<td></td>
<td>load</td>
</tr>
<tr>
<td></td>
<td>location &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>material</td>
</tr>
<tr>
<td></td>
<td>mome &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>node</td>
</tr>
<tr>
<td></td>
<td>number</td>
</tr>
<tr>
<td></td>
<td>sdisp</td>
</tr>
<tr>
<td></td>
<td>shear &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>strain &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>svel</td>
</tr>
<tr>
<td></td>
<td>xdisp &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>xvel &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>ydisp &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td></td>
<td>yvel &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT</td>
<td>structure</td>
</tr>
<tr>
<td></td>
<td>keyword</td>
</tr>
<tr>
<td></td>
<td>liner</td>
</tr>
<tr>
<td></td>
<td>node</td>
</tr>
<tr>
<td></td>
<td>property</td>
</tr>
<tr>
<td></td>
<td>liner</td>
</tr>
</tbody>
</table>

* A range of group ID numbers can be specified for plotting by giving a beginning number `ng` and an ending number `ng2`. All groups within this range will be plotted.
1.3.4 Example Applications

Simple examples are given to illustrate the implementation of the structural element commands for liners.

1.3.4.1 Reinforced Beam Test

A common method for testing shotcrete in the laboratory is to saw beams from shotcrete panels and test them in third-point loading. The advantage of third-point loading is that the test specimen experiences a constant moment over the middle third of the beam. The tensile stress at failure is given by

\[
\sigma_t = \frac{Pl}{bh^2}
\]  

(1.18)

where \( P \) is the applied load, \( l \) is the length, \( b \) is the width, and \( h \) is the height (or thickness) of the test beam. The force applied to the beam and the deflection (displacement) of the beam are monitored to produce a force-displacement curve. The maximum applied load, and hence the maximum tensile strength of the shotcrete, is determined from this curve.

Two beam bending tests are shown below: one with the residual tensile strength equal to the peak tensile strength; and one with the residual tensile strength reduced. A FISH function, Pforce, is used to monitor the force, \( P \), that develops during the tests. The data file, showing the properties and dimensions used in the tests, is listed in Example 1.13:

**Example 1.13 Reinforced beam test**

```plaintext
grid 20 1
mo el
  gen (0.0,0.0) (0.0,0.0225) (0.45,0.0225) (0.45,0.0)
  pro bul 1e9 she 1e9 den 1000
  fix y i 2 j 2
  fix y i 20 j 2
  def Pforce
    Pforce=yforce(2,2)+yforce(20,2)
  end
save bend0.sav
; syresid = syield = 3.2 MPa
stru liner from 2 2 to 20 2 prop 5001
struct prop 5001 dens 1000 e .204e9 area 0.0225 i .042e-3 thick 0.15
struct prop 5001 pratio 0.2 syield 3.2e6 syresid 3.2e6 sycomp 25e6
stru node 7 ini yvel -1e-8
stru node 7 fix y
stru node 13 ini yvel -1e-8
stru node 13 fix y
```

FLAC Version 6.0
In the first test, the residual tensile strength is set equal to the peak tensile strength (3.2 MPa). Using Eq. (1.18), the peak applied load is calculated to be 26,667 N for the given model dimensions. A constant moment develops over the middle third of the model, as shown in Figure 1.28. The load-deflection plot shown in Figure 1.29 shows that the peak load corresponds to the value calculated by Eq. (1.18).
**Figure 1.28**  Moment distribution in third-point loading test

**Figure 1.29**  Load-deflection plot for residual tensile strength equal to peak tensile strength
In the second test, the residual tensile strength is reduced to 1 MPa. The resultant load-deflection curve is shown in Figure 1.30. Substituting 1 MPa into Eq. (1.18), the applied load $P$ is calculated to be 8333 N, which agrees with the residual value shown in the figure.

**Figure 1.30** Load-deflection plot for residual tensile strength reduced
1.3.4.2 Plastic Hinge Formation in a Liner Structure

The liner element material model simulates the development of a limiting plastic moment and plastic hinge when the failure limit is reached. The example presented in Example 1.5 is repeated, using liner elements to illustrate the ability of the liner model to produce the same limiting values of moment and shear force. In order to produce a plastic-moment capacity of 25 kN-m, the axial tensile yield strength for the liner element is specified as 62.5 MPa, based upon Eq. (1.2). Otherwise, the data file, shown in Example 1.14, is similar to that for the beam-element model.

**Example 1.14 Plastic hinge formation in a liner structure**

```plaintext
struct node 1 0.0,0.0
struct node 2 10.0,0.0
struct liner begin node 1 end node 2 seg 2 prop 5001
struct prop 5001 e 2.0E11 pratio 0.3 area 0.0060 I 2.0E-4 thick 1.0 &
    syield 6.25e7 syresid 6.25e7 sycomp 1e10
struct node 1 fix y
struct node 2 fix y
struct node 3 fix y initial yvel -5.0e-6
history 1 node 3 ydisplace
history 2 element 1 moment1
history 3 element 1 moment2
history 4 element 2 moment1
history 5 element 2 moment2
history 6 element 1 shear
history 7 element 2 shear
history 8 node 3 adisplacement
set st damping struct=combined 0.8
set large
history 999 unbalanced
cycle 3000
save se_01_13.sav
```

We find that the results for the liner-element model are the same as those for the beam-element model. The limiting values of moment and shear force equal the analytical values of 25 kN-m and 5 kN, respectively, as shown in Figures 1.31 and 1.32. The moment and shear force distributions correspond with the analytical solution (compare Figures 1.33 and 1.34 to Figure 1.14).
Figure 1.31  Moment at right end of segment 1 and left end of segment 2 versus applied center displacement

Figure 1.32  Shear force at right end of segment 1 and left end of segment 2 versus applied center displacement
Figure 1.33  Moment distribution at limit condition

Figure 1.34  Shear force distribution at limit condition
The development of a discontinuity in the rotational motion for liner elements is illustrated for the loading of a cantilever beam. This example is similar to the beam-element test described in Example 1.6. The example is a cantilever beam (fixed at the left end) with a vertical load applied at the free end (see Example 1.15). The problem is run in large-strain mode. The cantilever is composed of two liner element segments. The segment with the fixed end is prescribed high-strength properties. The second segment is prescribed lower values for the axial and residual tensile yield properties, in order to allow failure to develop in this segment.

The final structural configuration and moment distribution are shown in Figure 1.35. We see that a discontinuity develops in the rotation at the beam center.

### Example 1.15 Cantilever beam with a plastic hinge, using a liner element

```
struct node 1 0.0,0.0
struct node 2 10.0,0.0
struct liner begin node 1 end node 2 seg 2 prop 5001
struct prop 5001 e 2.06E11 pratio 0.30 area 0.0060 I 2.0E-4 thick 1.0 &
        syield 6.25e7 syresid 6.25e7 sycomp 1e10
struct prop 5002 e 2.06E11 pratio 0.30 area 0.0060 I 2.0E-4 thick 1.0 &
        syield 1e10 syresid 1e10 sycomp 1e10
struct chprop 5002 range 1 1
struct node 1 fix x y r
struct node 2 load 0 -5.5e3 0
history 1 node 3 ydisplace
history 2 element 1 moment1
history 3 element 1 moment2
history 4 element 2 moment1
history 5 element 2 moment2
history 6 element 1 shear
history 7 element 2 shear
history 8 node 3 adisplacement
set st_damping struct=combined 0.8
set large
history 999 unbalanced
cycle 3000
save se_01_14.sav
```
Figure 1.35 Final structural configuration and moment distribution in liner cantilever with plastic hinge
1.3.4.3 Tunnel with Un-reinforced Shotcrete Lining

A horseshoe-shaped tunnel, 4 m wide by 4 m high, is located at a shallow depth. Three types of lining support are illustrated in this example: (1) shotcrete lining bonded to the rock; (2) shotcrete lining with sliding interface with the rock; and (3) two layers of shotcrete lining with a sliding interface with the rock. The data file for this example is listed in Example 1.16. A FISH function, closure, to calculate the tunnel closure for each case is included in this file.

Example 1.16 Tunnel with un-reinforced shotcrete lining

```plaintext
grid 10 10
m m
gen arc 5 5 7 5 180
prop dens 2000 bulk 33.333e6 shear 20.0e6 fric 35 coh=50e3
fix y j 1
fix x i 1
fix x i 11
set grav 10
solve
save lining0.sav
;
ini xdisp 0 ydisp 0
m null i 4 7 j 4 6
m null i 5 6 j 7
; un-reinforced shotcrete lining
struct prop 5001 dens 2100 e 20e9 thick=0.15 area=0.15 pr=0.2
struct prop 5001 syield=4e6 sycomp=40e6
struct liner long from44 to84 prop 5001
; histories
hist unbal
def closure
    closure=100.0*(ydisp(6,4)-ydisp(6,8))/(y(6,8)-y(6,4))
end
hist closure
solve
save lining1.sav
;
rest lining0.sav
ini xdisp 0 ydisp 0
m null i 4 7 j 4 6
m null i 5 6 j 7
; un-reinforced shotcrete lining
struct prop 5001 dens 2100 e 20e9 thick=0.15 area=0.15 pr=0.2
struct prop 5001 syield=4e6 sycomp=40e6
struct liner long from44 to84 int1 prop 5001
interf 1 kn 1e8 ks 1e8 fric 35 coh 50e3
```
In the first case, the tunnel is lined on the sidewalls and arch with 150 mm of shotcrete. The shotcrete is assumed to be fully bonded to the surrounding rock. The compressive strength is assumed to be 40 MPa, and the tensile strength is assumed to be 4 MPa. No residual tensile strength is specified (i.e., the shotcrete is un-reinforced). The calculated vertical tunnel closure in this case is approximately 0.23%. The distribution of axial forces in the shotcrete is shown in Figure 1.36. The maximum axial force is approximately 137 kN.
In the second case, the shotcrete is identical to the previous case, except that a slipping interface is connected between the lining and the surrounding rock. The interface is assumed to have the same shear strength properties as the rock. In this case, the vertical tunnel closure is calculated as approximately 0.37%. The maximum axial force is approximately 64 kN, as shown in Figure 1.37.

In the third case, the shotcrete is installed in two layers. The first layer is 50 mm thick, and the second layer is 100 mm thick. The second layer covers only the crown of the tunnel. The first layer is assumed to be “glued” to the second, and connected to the rock with an interface. The interface has the same properties as in the second case. The vertical tunnel closure in the third case is approximately 0.36%. The distribution of axial forces in both linings is shown in Figure 1.38. The maximum axial force is 59 kN in layer 1, and 11 kN in layer 2.
Figure 1.37  Distribution of axial forces in un-reinforced shotcrete lining with interface

Figure 1.38  Distribution of axial forces in un-reinforced shotcrete lining with two layers
1.4 Cable Elements

1.4.1 Formulation

Cable and bolt reinforcements in rock and soil have two somewhat different functions:

In hard rock subjected to low magnitude in-situ stress fields, failure is often localized and limited to wedges of rock directly adjacent to openings. The effect of the rockbolt reinforcement here is to provide a local resistance at the joint surfaces to resist wedge displacement. The bending, as well as the axial stiffness of the reinforcement, may be important in resisting shear deformations. In FLAC, this type of bolt action may be modeled using rockbolt elements which have a flexural rigidity. (See Section 1.6.)

If bending effects are not important, cable elements are sufficient because they allow the modeling of a shearing resistance along their length, as provided by the shear resistance (bond) between the grout and the cable, or the grout and the host medium. The cable element formulation in FLAC considers more than just the local effect of the reinforcement – its effect in resisting deformation is accounted for along its entire length. The cable element formulation is useful in modeling reinforcement systems (e.g., cable bolts) in which the bonding agent (grout) may fail in shear over some length of the reinforcement. The numerical formulation for reinforcement which accounts for this shear behavior of the grout annulus is described here.

In the discussion of the formulation that follows, the host material is assumed to be rock. However, the formulation applies equally to soil as the host material.

The cable is assumed to be divided into a number of segments of length, L, with nodal points located at each segment end. The mass of each segment is lumped at the nodal points, as in the continuum formulation of FLAC.

1.4.1.1 Axial Behavior

The axial behavior of conventional reinforcement systems may be assumed to be governed entirely by the reinforcing element itself. The reinforcing element is usually steel, and may be either a bar or cable. Because the reinforcing element is slender, it offers little bending resistance (particularly in the case of cable), and is treated as a one-dimensional member with capacity to sustain uniaxial tension. (Compression is also allowed. However, when modeling support that is primarily loaded in compression, pile elements are recommended.) A one-dimensional constitutive model is adequate for describing the axial behavior of the reinforcing element. In the present formulation, the axial stiffness is described in terms of the reinforcement cross-sectional area, A (area), and Young’s modulus, E (E).
The incremental axial force, \( \Delta F' \), is calculated from the incremental axial displacement by

\[
\Delta F' = -\frac{E A}{L} \Delta u'
\]

where:

\[
\Delta u' = \Delta u_1 t_i = \Delta u_1 t_1 + \Delta u_2 t_2 = (u_1^{[b]} - u_1^{[a]}) t_1 + (u_2^{[b]} - u_2^{[a]}) t_2.
\]

\( u_1^{[a]}, u_1^{[b]}, \) etc. are as shown in Figure 1.2. The superscripts \([a]\), \([b]\) refer to the nodes. The direction cosines \( t_1, t_2 \) refer to the tangential (axial) direction of the cable.

A tensile yield-force limit (yield) and a compressive yield-force limit (ycomp) can be assigned to the cable. Accordingly, cable forces that are greater than the tensile or compressive limits (Figure 1.39) cannot develop. If either yield or ycomp is not specified, the cable will have zero strength for loading in that direction.

![Figure 1.39 Cable material behavior for cable elements](image)

In evaluating the axial forces that develop in the reinforcement, displacements are computed at nodal points along the axis of the reinforcement, as shown in Figure 1.40. Out-of-balance forces at each nodal point, as well as shear forces contributed through shear interaction along the grout annulus, are computed from axial forces in the reinforcement. Axial displacements are computed based on accelerations from integration of the laws of motion using the computed out-of-balance axial force and a mass lumped at each nodal point.
1.4.1.2 Shear Behavior of Grout Annulus

The shear behavior of the grout annulus is represented as a spring-slider system located at the nodal points shown in Figure 1.40. The shear behavior of the grout annulus, during relative displacement between the reinforcing/grout interface and the grout/medium interface, is described numerically by the grout shear stiffness ($K_{\text{bond}}$ in Figure 1.41(b)) – i.e.,

\[
\frac{F_s}{L} = K_{\text{bond}} \left( u_c - u_m \right)
\]  

(1.20)

where:  
\begin{align*}
F_s & = \text{shear force that develops in the grout} \\
& \quad \text{(i.e., along the interface between the cable element and the grid);} \\
K_{\text{bond}} & = \text{grout shear stiffness (}K_{\text{bond}}\text{);} \\
u_c & = \text{axial displacement of the cable;} \\
u_m & = \text{axial displacement of the medium (soil or rock); and} \\
L & = \text{contributing element length.}
\end{align*}
The maximum shear force that can be developed in the grout, per length of element, is a function of the cohesive strength of the grout and the stress-dependent frictional resistance of the grout. The following relation is used to determine the maximum shear force:

\[
\frac{F_s^{\text{max}}}{L} = S_{\text{bond}} + \sigma'_c \times \tan(S_{\text{friction}}) \times \text{perimeter}
\]  

(1.21)

where:
- \( S_{\text{bond}} \) = intrinsic shear strength or cohesion (\( s_{\text{bond}} \));
- \( \sigma'_c \) = mean effective confining stress normal to the element;
- \( S_{\text{friction}} \) = friction angle (\( s_{\text{friction}} \)); and
- \( \text{perimeter} \) = exposed angle of the element (\( \text{perimeter} \)).

The mean effective confining stress normal to the element is defined by the equation

\[
\sigma'_c = -\left(\frac{\sigma_{nn} + \sigma_{zz}}{2} + p\right)
\]

(1.22)

where:
- \( p \) = pore pressure;
- \( \sigma_{zz} \) = out-of-plane stress; and
- \( \sigma_{nn} = \sigma_{xx} n_1^2 + \sigma_{yy} n_2^2 + 2 \sigma_{xy} n_1 n_2 \),
- \( n_i \) = unit vectors as defined in Eq. (1.4).

The limiting shear-force relation is depicted by the diagram in Figure 1.41(a). The input properties are shown in bold type on this figure.
In computing the relative displacement at the grout/medium interface, an interpolation scheme is used to calculate the displacement of the medium in the cable axial direction at the cable node. Each cable node is assumed to exist within an individual FLAC zone (hereafter referred to as the host zone). The interpolation scheme uses weighting factors that are determined by the distance to each of the gridpoints of the host zone. The calculation of the weighting factors is based on satisfying moment equilibrium. The same interpolation scheme is used to apply forces developed at the grout/medium interface back to the host zone gridpoints.

For example, in computing the axial displacement of the grout/medium interface, the following interpolation scheme is used. Consider reinforcement passing through a constant-strain finite difference triangle (subzone) making up part of the intact medium, as shown in Figure 1.42(a). The incremental $x$-component of displacement ($\Delta u_{xp}$) at the nodal point is given by

$$\Delta u_{xp} = W_1 \Delta u_{x1} + W_2 \Delta u_{x2} + W_3 \Delta u_{x3}$$

(1.23)

where $\Delta u_{x1}$, $\Delta u_{x2}$, $\Delta u_{x3}$ are the incremental gridpoint displacements; and $W_1$, $W_2$, $W_3$ are weighting factors.

A similar expression is used for $y$-component displacements. The weighting factors $W_1$, $W_2$, $W_3$ are computed from the position of the nodal point within the triangle, as follows:

$$W_1 = A_1 / A_T$$

(1.24)

where $A_T$ is the total area of the finite-difference triangle; and $A_1$ is the area of the triangle in Figure 1.42(b).

Incremental $x$- and $y$-displacements (Eq. (1.23)) are used at each calculation step to determine the new local reinforcing orientation. The axial component of displacement of the grout/medium interface is computed from the current orientation of the reinforcing segment.

Forces generated at the grout/medium interface ($F_{xp}$, $F_{yp}$) are distributed back to gridpoints according to the same weighting factors used previously – i.e.:

$$F_{x1} = W_1 \cdot F_{xp}$$

$$F_{x2} = W_2 \cdot F_{xp}$$

$$F_{x3} = W_3 \cdot F_{xp}$$

(1.25)

where $F_{x1}$, $F_{x2}$ and $F_{x3}$ are forces applied to the gridpoints.
1.4.1.3 Normal Behavior at Grout Interface

As explained above, an interpolated estimate of grid velocity is made at each cable node. The velocity component normal to the average axial cable direction is transferred directly to the node (i.e., the cable node is “slaved” to the grid motion in the normal direction). The node exerts no normal force on the grid if the cable segments on either side of the node are colinear. However, if the segments make an angle with each other, then a proportion of their axial forces will act in the mean normal direction. This net force acts on both the grid and the cable node (in opposite directions). Thus, an initially straight cable can sustain normal loading if it is allowed finite deflection, using FLAC’s large-strain mode.
1.4.2 Cable-Element Properties

The cable elements used in FLAC require the following input parameters (cable property keywords are shown in parentheses):

1. cross-sectional area (area or radius) [length²] of the cable;
2. density (density) [mass/volume] of the cable (optional – used for dynamic analysis and gravity loading);
3. elastic modulus (e) [stress] of the cable;
4. spacing (spacing) [length] (optional – if not specified, cables are considered to be continuous in the out-of-plane direction);
5. tensile yield strength (yield) [force] of the cable (if not specified, the tensile yield strength is zero);
6. compressive yield strength (ycomp) [force] of the cable (if not specified, the compressive yield strength is zero);
7. exposed perimeter (perimeter) [length] of the cable;
8. stiffness of the grout (kbond) [force/cable length/displacement];
9. cohesive strength of the grout (sbond) [force/cable length];
10. frictional resistance of the grout (sfriction) [degrees]; and
11. thermal expansion coefficient (thexp) (optional – used for thermal analysis).

The cable radius, rather than the area, can also be specified; the cross-sectional area will then be calculated automatically. The cable perimeter must be specified separately if the frictional resistance of the grout is to be considered.

If spaced reinforcement is to be simulated (e.g., soil nails installed on a regular spacing), the spacing in the out-of-plane direction can be prescribed. The spacing parameter is used to automatically scale properties and parameters to account for the effect of the distribution of the cables over a regularly spaced pattern. See Section 1.9.4 for more information on the simulation of spaced reinforcement. Note that the actual cable properties, not scaled properties, are entered in FLAC when spacing is given.

The area, modulus and yield strength of the cable are usually readily available from handbooks, manufacturer’s specifications, etc. The properties related to the grout are more difficult to estimate. The grout annulus is assumed to behave as an elastic-perfectly plastic solid. As a result of relative shear displacement, \( u^t \), between the tendon surface and the borehole surface, the shear force, \( F^t \), mobilized per length of cable is related to the grout stiffness, \( K_{bond} \) – i.e.,

\[
F^t = K_{bond} u^t
\]  

(1.26)
Usually, $K_{\text{bond}}$ can be measured directly in laboratory pullout tests. Alternatively, the stiffness can be calculated from a numerical estimate for the elastic shear stress, $\tau_G$, obtained from an equation describing the shear stress at the grout/rock interface (St. John and Van Dillen 1983):

$$\tau_G = \frac{G}{D/2 + t} \frac{\Delta u}{\ln(1 + 2t/D)}$$  \hspace{1cm} (1.27)

where: $\Delta u =$ relative displacement between the element and the surrounding material; $G =$ grout shear modulus; $D =$ reinforcing diameter; and $t =$ annulus thickness.

Consequently, the grout shear stiffness, $K_{\text{bond}}$, is simply given by

$$K_{\text{bond}} = \frac{2\pi G}{\ln (1 + 2t/D)}$$  \hspace{1cm} (1.28)

In many cases, the following expression has been found to provide a reasonable estimate of $K_{\text{bond}}$ for use in FLAC:

$$K_{\text{bond}} \simeq \frac{2\pi G}{10 \ln (1 + 2t/D)}$$  \hspace{1cm} (1.29)

The one-tenth factor helps to account for the relative shear displacement that occurs between the host-zone gridpoints and the borehole surface. This relative shear displacement is not accounted for in the present formulation.

The maximum shear force per cable length in the grout is determined by Eq. (1.21). The values for bond cohesive strength and friction angle can be estimated from the results of pullout tests conducted at different confining pressures; or, should such results be unavailable, the maximum force per length may be approximated from the peak shear strength (St. John and Van Dillen 1983):

$$\tau_{\text{peak}} = \tau_I Q_B$$  \hspace{1cm} (1.30)

where $\tau_I$ is approximately one-half of the uniaxial compressive strength of the weaker of the rock and grout, and $Q_B$ is the quality of the bond between the grout and rock ($Q_B = 1$ for perfect bonding).

Neglecting frictional confinement effects, $S_{\text{bond}}$, may then be obtained from

$$S_{\text{bond}} = \pi (D + 2t) \tau_{\text{peak}}$$  \hspace{1cm} (1.31)
Failure of reinforcing systems does not always occur at the grout/rock interface. Failure may occur at the reinforcing/grout interface, as is often true for cable reinforcing. In such cases, the shear stress should be evaluated at this interface. This means that the expression $(D + 2t)$ is replaced by $(D)$ in Eq. (1.31).

The calculation of cable-element properties is demonstrated by the following example. A 25.4 mm (1 inch)-diameter locked-coil cable was installed at 2.5 m spacing, perpendicular to the plane of analysis. The reinforcing system is characterized by the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cable diameter $(D)$</td>
<td>25.4 mm</td>
</tr>
<tr>
<td>hole diameter $(D + 2t)$</td>
<td>38 mm</td>
</tr>
<tr>
<td>cable modulus $(E)$</td>
<td>98.6 GPa</td>
</tr>
<tr>
<td>cable ultimate tensile capacity</td>
<td>0.548 MN</td>
</tr>
<tr>
<td>grout compressive strength</td>
<td>20 MPa</td>
</tr>
<tr>
<td>grout shear modulus $(G_g)$</td>
<td>9 GPa</td>
</tr>
<tr>
<td>friction (ignored)</td>
<td>0</td>
</tr>
</tbody>
</table>

Two independent methods are used in evaluating the maximum shear force in the grout. In the first method, the bond shear strength is assumed to be one-half the uniaxial compressive strength of the grout. If the grout-material compressive strength is 20 MPa, and the grout is weaker than the surrounding rock, the grout shear strength is then 10 MPa.

In the second method, reported pullout data is used to estimate the grout shear strength. The report presents results for 15.9 mm (5/8 inch)-diameter steel cables grouted with a 0.15 m (5.9 inch) bond length in holes of varying depths. The testing indicated capacities of roughly 70 kN. If a surface area of 0.0075 m$^2$ (0.15 m $\times$ 0.05 m) is assumed for the cables, then the calculated maximum shear strength of the grout is

$$
\frac{70 \times 10^3 \text{ N}}{0.0075 \text{ m}^2} = 9.33 \times 10^6 \text{ N/m}^2 = 9.33 \text{ MPa}
$$
This value agrees closely with the 10 MPa estimated above, and either value could be used. Assuming failure occurs at the cable/grout interface, the maximum bond force per length is (using Eq. (1.31) with \(D + 2t\) replaced by \(D\))

\[
S_{\text{bond}} = \pi (0.0254 \text{ m}) (10 \text{ MPa}) = 800 \text{ kN/m}
\]

The bond stiffness, \(K_{\text{bond}}\), is estimated from Eq. (1.29). For the assumed values shown above, a bond stiffness of \(1.5 \times 10^{10} \text{ N/m/m}\) is calculated.

Values for \(K_{\text{bond}}, S_{\text{bond}}, E\) and tensile yield force are divided by 2.5 to account for the 2.5 m spacing of cables perpendicular to the modeled cross-section (see Section 1.9.4). This is performed automatically when the spacing parameter is specified.

The final input properties for FLAC are:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_{\text{bond}})</td>
<td>(1.5 \times 10^{10} \text{ N/m/m})</td>
</tr>
<tr>
<td>(S_{\text{bond}})</td>
<td>(8.0 \times 10^{5} \text{ N/m})</td>
</tr>
<tr>
<td>(E)</td>
<td>98.6 GPa</td>
</tr>
<tr>
<td>(\text{yield})</td>
<td>(5.48 \times 10^{5} \text{ N})</td>
</tr>
<tr>
<td>(\text{area})</td>
<td>(5 \times 10^{-4} \text{ m}^2)</td>
</tr>
<tr>
<td>(s_{\text{friction}})</td>
<td>0</td>
</tr>
<tr>
<td>(\text{spacing})</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Note that other researchers have reported values for \(k_{\text{bond}}\), based upon the results of pull tests, that are approximately one order of magnitude lower than those calculated using Eq. (1.29) (e.g., Ruest and Martin 2002).

Another example estimation of grout properties from pullout tests is presented in Section 9 in the Examples volume.

The effect of linear thermal expansion is implemented in the cable formulation. The temperature change occurs as a result of either heat conduction or temperature re-initialization in the FLAC grid (for CONFIG thermal).

It is assumed that the grid temperature is communicated instantaneously to the structural elements. The temperature change generates thermal expansion/contraction in the structural element axial direction; the effect of the cable lateral expansion is neglected, and no other coupling takes place. The effect of heat conduction in the structural element is not considered.

The incremental axial force generated by thermal expansion in a cable element is calculated using the formula (note that compression is positive for axial forces)

\[
\Delta F = E \ A \ \alpha \ \Delta T \tag{1.32}
\]
where \( E \) is the Young’s modulus of the element, \( A \) is the cross-sectional area, \( \alpha \) is the linear thermal expansion coefficient, and \( \Delta T \) is the temperature increment for the element.

The structural element nodal temperature increment is determined by interpolation of nodal temperature increments in the host zone and stored in a structural node offset. The temperature change in a structural element is calculated as the average of values at the two nodes. The thermal expansion of a cable element is computed incrementally as the product of the thermal linear expansion coefficient, temperature change for the step, and element length. Thermal strains, thermal strain increments and temperatures at structural nodes are not stored.

### 1.4.3 Mean Effective Confining Stress

The out-of-plane stress component, \( \sigma_{zz} \), can be included or excluded from the calculation of mean effective confining stress for cable elements, \( \sigma'_c \) (see Eq. (1.21)).

Use the keyword \texttt{szz on} or \texttt{off} with the \texttt{STRUCT prop} command to turn the \( \sigma_{zz} \) component on or off. By default, \( \sigma_{zz} \) is included.

### 1.4.4 Pretensioning Cable Elements

Cable elements may be pretensioned in \textit{FLAC} by using the optional keyword \texttt{tens = t} with the \texttt{STRUCT cable} command. A positive value for \( t \) assigns an axial force into the cable element(s) described by that \texttt{STRUCT} command. It is important to note that the cable with specified pretension is unlikely to be initially in equilibrium with other elements or the \textit{FLAC} grid to which it is linked. In other words, some displacement of the cable nodes and linked elements or nodes or gridpoints is probably required to achieve equilibrium. These displacements will likely result in some loss of the initial pretension.

In practice, pretensioned elements may be fully grouted, or they may be left un-grouted over part of their length. In either case, some anchorage length is provided (usually at the far end) to support the element during pretensioning. To simulate this pretensioning in \textit{FLAC}, several separate \texttt{STRUCT} commands are required. One \texttt{STRUCT} command is used to define the geometry for the anchorage. A \texttt{STRUCT prop} command is then used to define the anchorage properties. A third \texttt{STRUCT} command is used to define the free (i.e., un-bonded) section and the pretension force. Note that the anchorage section and free section would be linked by a common cable node. Another \texttt{STRUCT prop} command (or a \texttt{STRUCT chprop} command) is used to specify the properties for the free length. In most cases, the free length would have \( S_{bond} = 0 \) (i.e., un-bonded). An example of this procedure is shown in Section 6 in the \textit{Examples volume}.

As an alternative to specifying a pretension to the free length, a load can be applied to the free end using the \texttt{STRUCT node n load fx fy} command. After equilibrating forces have developed in the anchorage, the loaded node can be connected to the \textit{FLAC} grid or another element.

The procedure for subsequent “grouting” of the free length is to simply change the \( K_{bond} \) and \( S_{bond} \) values for the free section to appropriate values for a grouted section.
1.4.5 Commands Associated with Cable Elements

All the commands associated with cable elements are listed in Table 1.4. This includes the commands associated with the generation of cables, and those required to monitor histories, plot and print cable-element variables. See Section 1.3 in the Command Reference for a detailed explanation of these commands.
Table 1.4 Commands associated with cable elements

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>cable</td>
<td>keyword</td>
</tr>
<tr>
<td>begin</td>
<td></td>
</tr>
<tr>
<td>grid</td>
<td>ij</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>xy</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>grid</td>
<td>ij</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>xy</td>
<td></td>
</tr>
<tr>
<td>tension</td>
<td>value</td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>segment</td>
<td>ns</td>
</tr>
<tr>
<td>delete</td>
<td>&lt;n1 n2&gt;</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>xy</td>
<td></td>
</tr>
<tr>
<td>node</td>
<td>n*</td>
</tr>
<tr>
<td>keyword</td>
<td></td>
</tr>
<tr>
<td>fix</td>
<td>&lt;x&gt; &lt;y&gt;</td>
</tr>
<tr>
<td>free</td>
<td>&lt;x&gt; &lt;y&gt;</td>
</tr>
<tr>
<td>initial</td>
<td>keyword</td>
</tr>
<tr>
<td>xdis</td>
<td>value</td>
</tr>
<tr>
<td>xvel</td>
<td>value</td>
</tr>
<tr>
<td>ydis</td>
<td>value</td>
</tr>
<tr>
<td>yvel</td>
<td>value</td>
</tr>
<tr>
<td>load</td>
<td>fx fy</td>
</tr>
<tr>
<td>slave</td>
<td>&lt;x&gt; &lt;y&gt; m</td>
</tr>
<tr>
<td>unslave</td>
<td>&lt;x&gt; &lt;y&gt;</td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>keyword</td>
<td></td>
</tr>
<tr>
<td>area</td>
<td>value</td>
</tr>
<tr>
<td>density</td>
<td>value</td>
</tr>
<tr>
<td>e</td>
<td>value</td>
</tr>
<tr>
<td>kbond</td>
<td>value</td>
</tr>
<tr>
<td>perimeter</td>
<td>value</td>
</tr>
<tr>
<td>radius</td>
<td>value</td>
</tr>
<tr>
<td>sbond</td>
<td>value</td>
</tr>
<tr>
<td>sfriction</td>
<td>value</td>
</tr>
<tr>
<td>spacing</td>
<td>value</td>
</tr>
<tr>
<td>szz</td>
<td>on/off</td>
</tr>
<tr>
<td>thexp</td>
<td>value</td>
</tr>
<tr>
<td>ycomp</td>
<td>value</td>
</tr>
<tr>
<td>yield</td>
<td>value</td>
</tr>
<tr>
<td>chprop</td>
<td>np</td>
</tr>
<tr>
<td>range</td>
<td>nel1 nel2</td>
</tr>
</tbody>
</table>

* For the keywords fix, free, initial and load, a range of nodes can be specified with the phrase range n1 n2.
### Table 1.4 Commands associated with cable elements (continued)

<table>
<thead>
<tr>
<th>HISTORY</th>
<th>keyword</th>
<th>node</th>
<th>n</th>
<th>element</th>
<th>nel</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>node</td>
<td></td>
<td></td>
<td>node</td>
<td></td>
<td>node</td>
</tr>
<tr>
<td>sbond</td>
<td></td>
<td></td>
<td></td>
<td>sbond</td>
<td></td>
<td>sbond</td>
</tr>
<tr>
<td>sdisp</td>
<td></td>
<td></td>
<td></td>
<td>sdisp</td>
<td></td>
<td>sdisp</td>
</tr>
<tr>
<td>sforce</td>
<td></td>
<td></td>
<td></td>
<td>sforce</td>
<td></td>
<td>sforce</td>
</tr>
<tr>
<td>xaccel</td>
<td></td>
<td></td>
<td></td>
<td>xaccel</td>
<td></td>
<td>xaccel</td>
</tr>
<tr>
<td>xdisp</td>
<td></td>
<td></td>
<td></td>
<td>xdisp</td>
<td></td>
<td>xdisp</td>
</tr>
<tr>
<td>xvel</td>
<td></td>
<td></td>
<td></td>
<td>xvel</td>
<td></td>
<td>xvel</td>
</tr>
<tr>
<td>yaccel</td>
<td></td>
<td></td>
<td></td>
<td>yaccel</td>
<td></td>
<td>yaccel</td>
</tr>
<tr>
<td>ydisp</td>
<td></td>
<td></td>
<td></td>
<td>ydisp</td>
<td></td>
<td>ydisp</td>
</tr>
<tr>
<td>yvel</td>
<td></td>
<td></td>
<td></td>
<td>yvel</td>
<td></td>
<td>yvel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLOT</th>
<th>cable</th>
<th>structure</th>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td></td>
<td>&lt;cable&gt; keyword</td>
<td></td>
</tr>
<tr>
<td>sbond</td>
<td></td>
<td>axial $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td>sdisp</td>
<td></td>
<td>bond $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td>sforce</td>
<td></td>
<td>cs_sdisp $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td>xaccel</td>
<td></td>
<td>cs_sforce $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td>xdisp</td>
<td></td>
<td>element</td>
<td></td>
</tr>
<tr>
<td>xvel</td>
<td></td>
<td>fyaxial $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td>yaccel</td>
<td></td>
<td>load</td>
<td></td>
</tr>
<tr>
<td>ydisp</td>
<td></td>
<td>location $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td>yvel</td>
<td></td>
<td>material</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>node</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>strain $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>svel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>xdisp $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>xvel $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ydisp $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>yvel $&lt;ng &lt;ng2*&gt;&gt;$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRINT</th>
<th>structure</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>cable</td>
<td>property cable</td>
</tr>
</tbody>
</table>

* A range of group ID numbers can be specified for plotting by giving a beginning number $ng$ and an ending number $ng2$. All groups within this range will be plotted.
1.4.6 Example Applications

Simple examples are given below to demonstrate the implementation of cable elements in FLAC. Additional examples can be found in the Examples volume.

1.4.6.1 Reinforced Beam with Vertical Crack

This problem examines the behavior of a lightly reinforced beam subject to gravity loading. A vertical crack is created through the midpoint of the beam. Cable elements are used to represent the reinforcement. The input commands for this problem are given in Example 1.17:

```
Example 1.17 Reinforced beam with vertical crack
conf p_str
grid 13 3
me
model null i 7
gen 0 0 0 2 5.5 2 5.5 0 i 1 7
gen 5.5 0 5.5 2 11 2 11 0 i 8 14
int 1 aside from 7 1 to 7 4 bside from 8 1 to 8 4
int 1 kn 1e10 ks 1e10 fric 0.0
prop s .3e9 b 1e9 d 2400
set large
fix y j 1 i 1
fix y j 1 i 14
set grav 10.0
his yd i 7 j 4
struct cable beg .1 .1 end 10.9 .1 seg 13 pro 2001
stru pro 2001 yi 1e6 kb 1e10 sb 1e7 e 200e9 a 2e-3
hist node 7 xvel yvel xdisp ydisp
hist node 7 sbond nbond sforce nforce sdisp ndisp
hist elem 7 axial shear
step 5000
plot grid struc axial fill disp fix
ret
```

Figure 1.43 shows the axial force distribution that develops along the cable reinforcement. The vertical centerline displacement is 1.2 cm.
Figure 1.43  Axial force in cable reinforcement and displacement of beam
1.4.6.2 Soil Nailing

This example demonstrates the ability of cable elements to simulate support provided by materials such as geo-grids or soil nails in the construction of reinforced embankments. In this example, three layers of soil nails are installed on a 0.5 m spacing within a vertical embankment. Two cases are examined: (1) only cohesive resistance is assumed between the nails and the soil; and (2) both cohesive and frictional resistance are included. The data file is listed in Example 1.18. The command

```plaintext
struct prop 2001 sfric 20 peri 0.314
```

is added in the second case to include the effect of the frictional resistance.

**Example 1.18 Soil nailing support**

```plaintext
grid 11 11
m m
prop dens 2000 bulk 5e9 shear 1e9
prop coh 4e4 fr 30
;
fix x y j=1
fix x i=1
fix x i=12
;
ini syy -2.2e5 var 0 2.2e5
ini sxx -1.32e5 var 0 1.32e5
ini szz -0.88e5 var 0 0.88e5
set grav 10
;
hist xdisp i=1 j=12
;
hist 999 unbal
solve elastic
save se_01_17a.sav
; exclude cable friction
free x i=1 j 2 12
struct cable begin 0,3.5 end 8,3.5 seg 8 prop 2001
struct cable begin 0,6.5 end 8,6.5 seg 8 prop 2001
struct cable begin 0,9.5 end 8,9.5 seg 8 prop 2001
struct prop 2001 e=200e9 a=8.5e-3 yield=1e10 kbond=7e6 sbond=1e2 spac 0.5
;
step 2000
save se_01_17b.sav
; include cable friction
restore se_01_17a.sav
free x i=1 j 2 12
struct cable begin 0,3.5 end 8,3.5 seg 8 prop 2001
```
In the first case, the soil nails are not sufficient to support the embankment. Figure 1.44 shows the axial forces in the cables, and indicates that bond yield has been reached at all cable nodes. By typing the command `PRINT struct node spring`, it can also be seen that the maximum shear force per length of cable element has been reached at all nodes.

In the second case, by including a frictional resistance at the nail/soil interface of 20°, the nails are now sufficient to stabilize the embankment. Figure 1.45 plots the axial forces in the cables for this case. Note that significantly higher axial forces can now develop as a result of the frictional resistance.
Figure 1.44  Axial forces in nails with only cohesive strength at soil/nail interface

Figure 1.45  Axial forces in nails with both frictional and cohesive strength at soil/nail interface
1.5 Pile Elements

1.5.1 Formulation

The pile elements in FLAC combine the behaviors of beam elements and cable elements. Piles are two-dimensional elements with 3 degrees of freedom (two displacements and one rotation) at each end node. The formulation for the pile element is identical to that for beams, as described in Section 1.2.1. A pile element segment is treated as a linearly elastic material with no axial yield. However, plastic moments and hinges can be specified in the same way as beams.

Piles interact with the FLAC grid via shear and normal coupling springs. The coupling springs are nonlinear connectors that transfer forces and motion between the pile elements and the grid at the pile element nodes. The formulation is similar to that for cable elements. The behavior of the shear coupling springs is identical to the representation for the shear behavior of grout, as described for cable elements in Section 1.4.1.2. The behavior of the normal coupling springs includes the capability to model load reversal and the formation of a gap between the pile and the grid. The normal coupling springs are primarily intended to simulate the effect of the medium squeezing around the pile. A force-displacement law for the normal springs can also be defined externally by a FISH function. The formulations for the shear and normal coupling springs are described below.

The coupling springs associated with FLAC’s pile elements are similar to the load/displacement relations provided by “p-y curves” (e.g., see Coduto 1994). However, p-y curves are intended to capture (in a crude way) the interaction of the pile with the whole soil mass, while FLAC’s coupling springs represent the local interaction of the soil and pile elements. See Section 1.5.4.4 for further discussion on this topic.

The pile formulation simulates a row of equally spaced piles in plane-strain symmetry. See Section 1.9.4 for property-scaling rules to simulate the effect of spacing.

* The coupling springs can also simulate the effect of a continuous wall/medium contact. However, it is recommended that, for this case, beam elements with interface elements attached on both sides of the beams be used, because interfaces provide a better representation of the effect of wall/soil separation. For example, see the diaphragm wall example (Section 11 in the Examples volume).

† Note that pile elements cannot be used to simulate a single vertical pile because the structural element formulation does not apply to axisymmetric geometry (see Section 1.9.1). For such a case, the three-dimensional program FLAC3D is recommended.
1.5.1.1 Behavior of Shear Coupling Springs

The shear behavior of the pile/grid interface is represented as a spring-slider system at the pile nodal points. The system is similar to that illustrated for the cable/grid interface in Figure 1.40. The shear behavior of the interface during relative displacement between the pile nodes and the grid is described numerically by the coupling spring shear stiffness (\(c_{sstiff}\) in Figure 1.46(b)) – i.e.,

\[
\frac{F_s}{L} = c_{sstiff} (u_p - u_m)
\]  

(1.33)

where:

- \(F_s\) = shear force that develops in the shear coupling spring (i.e., along the interface between the pile element and the grid);
- \(c_{sstiff}\) = coupling spring shear stiffness (\(c_{sstiff}\));
- \(u_p\) = axial displacement of the pile;
- \(u_m\) = axial displacement of the medium (soil or rock); and
- \(L\) = contributing element length.

![Shear strength criterion and shear force versus displacement](image)

**Figure 1.46** Material behavior of shear coupling spring for pile elements

The maximum shear force that can be developed along the pile/grid interface is a function of the cohesive strength of the interface and the stress-dependent frictional resistance along the interface. The following relation is used to determine the maximum shear force per length of the pile:

\[
\frac{F_{s,max}}{L} = c_{scoh} + \sigma'_c \times \tan(c_{sflic}) \times \text{perimeter}
\]  

(1.34)
where: 

- \( c_{s\text{coh}} \) = cohesive strength of the shear coupling spring (\( c_{s\text{coh}} \));
- \( \sigma'_c \) = mean effective confining stress normal to the pile element;
- \( c_{s\text{fric}} \) = friction angle of the shear coupling spring (\( c_{s\text{fric}} \)); and
- \( \text{perimeter} \) = exposed perimeter of the element (\( \text{perimeter} \)).

The mean effective confining stress normal to the element is defined by the equation

\[
\sigma'_c = -\left( \frac{\sigma_{nn} + \sigma_{zz}}{2} + p \right)
\]

where:
- \( p \) = pore pressure;
- \( \sigma_{zz} \) = out-of-plane stress; and
- \( \sigma_{nn} = \sigma_{xx} n_1^2 + \sigma_{yy} n_2^2 + 2 \sigma_{xy} n_1 n_2 \),
- \( n_i \) = unit vectors as defined in Eq. (1.4).

The limiting shear-force relation is depicted by the diagram in Figure 1.46(a). The input properties are shown in bold type on this figure.

The same interpolation scheme as that employed for the cable elements is used to calculate the displacement of the grid in the pile axial direction at the pile node.

### 1.5.1.2 Behavior of Normal Coupling Springs

The normal behavior of the pile/grid interface is represented by a linear spring with a limiting normal force that is dependent on the direction of movement of the pile node. The normal behavior during the relative normal displacement between the pile nodes and the grid is described numerically by the coupling spring normal stiffness (\( c_{s\text{nstiff}} \) in Figure 1.47(b)) – i.e.,

\[
\frac{F_n}{L} = c_{s\text{nstiff}} (u_p^n - u_m^n)
\]

where: 
- \( F_n \) = normal force that develops in the normal coupling spring (i.e., along the interface between the pile element and the grid);
- \( c_{s\text{nstiff}} \) = coupling spring normal stiffness (\( c_{s\text{nstiff}} \));
- \( u_p^n \) = displacement of the pile normal to the axial direction of the pile;
- \( u_m^n \) = displacement of the medium (soil or rock) normal to the axial direction of the pile; and
- \( L \) = contributing element length.
A limiting normal force can be prescribed to simulate the localized three-dimensional effect of the pile pushing through the grid (e.g., a soil being squeezed around a single pile). The limiting force is a function of a normal cohesive strength and a stress-dependent frictional resistance between the pile and the grid. The following relation is used to determine the maximum normal force per length of the pile:

$$\frac{F_{n}^{\text{max}}}{L} = cs_{\text{ncoh}} + \sigma_c' \times \tan(cs_{\text{nfric}}) \times \text{perimeter}$$  \hspace{1cm} (1.37)

where:
- $cs_{\text{ncoh}}$ = cohesive strength of the normal coupling spring (cs.ncoh), which is dependent on the direction of loading;
- $\sigma_c'$ = mean effective confining stress normal to the pile element;
- $cs_{\text{nfric}}$ = friction angle of the normal coupling spring (cs.nfric); and
- perimeter = exposed perimeter of the element (perimeter).

The mean effective confining stress normal to the element is defined by Eq. (1.35).

The limiting normal-force relation is shown in the diagram in Figure 1.47(a). The cohesive strength is defined by two property keywords (cs.ncoh and cs.nten). The value that will be used in Eq. (1.37) depends on the direction of motion of the pile node. Conceptually, a single normal spring is considered to be located at each pile node. Positive normal motion is defined to be to the left when facing along the pile element in the direction of node $n$ to node $n+1$. The sign convention is shown in Figure 1.48. Displacement of the node in the positive normal direction is considered as a positive...
displacement with the spring in compression, and `cs_ncoh` is used; displacement in the negative normal direction is considered as a negative displacement with the spring in tension, and `cs_nten` is used. The dual cohesion parameters are useful for cases in which different conditions exist on both sides of the pile (e.g., the pile acts as a retaining wall). If `cs_nten` is not specified, then its value defaults to that for `cs_ncoh`, and the response is the same for normal movement in either direction.

![Figure 1.48 Sign convention for compressive strength of normal coupling springs](image)

1.5.2 Pile-Element Properties

The pile elements in FLAC require the following input parameters (pile property keywords are shown in parentheses):

1. cross-sectional area (`area` or `height` and `width` or `radius`) [length$^2$] of the pile;

2. second moment of area (I) [length$^4$] (commonly referred to as the moment of inertia) of the pile;

3. density (`density`) [mass/volume] of the pile (optional – used for dynamic analysis and gravity loading);

4. elastic modulus (`e`) [stress] of the pile;

5. spacing (`spacing`) [length] (optional – if not specified, piles are considered to be continuous in the out-of-plane direction);

6. plastic moment (`pmom`) [force-length] (optional – if not specified, the moment capacity is assumed to be infinite);
STRUCTURAL ELEMENTS

(7) exposed perimeter (perimeter) [length] of the pile (i.e., the length of the pile surface that is in contact with the medium);

(8) stiffness of shear coupling spring (cs_stiff) [force/pile length/displacement];

(9) cohesive strength of the shear coupling spring (cs_coh) [force/pile length];

(10) frictional resistance of the shear coupling spring (cs_fric) [degrees];

(11) stiffness of normal coupling spring (cs_nstiff) [force/pile length/displacement];

(12) cohesive (and tensile) strength [force/pile length] of the normal coupling spring (cs_ncoh and cs_nten);

(13) frictional resistance of the normal coupling spring (cs_nfric) [degrees]; and

(14) normal gap (cs_ngap) formation between the pile and the medium.

The height and width of the pile element cross-section (or the radius for a circular cross-section) can also be prescribed instead of the area and moment of inertia. The area and moment of inertia will then be calculated automatically.

Pile element properties are determined in a fashion similar to that used for beam elements. (See Section 1.2.2.)

A limiting plastic moment and plastic hinge condition can be prescribed for pile nodes. See Section 1.1.7 for details. Softening relations for plastic hinges can also be defined by the user. An example is given in Section 1.5.4.3.

The exposed perimeter of a pile element and the properties of the coupling springs should be chosen to represent the behavior of the pile/medium interface commensurate with the problem being analyzed. For piles in soil, the pile/soil interaction can be expressed in terms of a shear response along the length of the pile shaft as a result of axial loading (e.g., a friction pile) or in terms of a normal response when the direction of loading is perpendicular to the pile axis (e.g., piles used to stabilize a slope).

Pile/soil interaction will depend on whether the pile was driven or cast-in-place. The interaction is expressed in terms of the shear resistance that can develop along the length of the pile. For example, driven friction piles receive most of their support by friction or adhesion from the soil along the pile shaft. A cast-in-place point-bearing pile, on the other hand, receives the majority of its support from soil near the tip of the pile.

In many cases, properties needed to describe the site-specific response of the pile/soil interaction will not be available. However, a reasonable understanding of the soil properties at the site is usually provided from standard in-situ and laboratory tests. In such cases, the pile/soil shear response can be estimated from the soil properties. If the failure associated with the pile/soil response is assumed to occur in the soil, then the lower limits for cs_fric and cs_coh can be related to the angle of internal friction of the soil (for cs_fric) and the soil cohesion times the perimeter of the pile (for cs_coh). If failure is assumed to occur at the pile/soil interface, the values for cs_fric and cs_coh may be reduced to reflect the smoothness of the pile surface.
When a pile is loaded laterally, a gap may open on one side between the pile and the medium. If the load is reversed, the pile first has to close this gap before it can load the medium on the opposite side. The total size of a gap is an accumulated value. The parameter \(cs_{\text{ngap}}\) specifies how much of the gap is effective – \(cs_{\text{ngap}} = 0\) causes the gap to be ignored completely: the pile is considered to be always in contact with the medium. \(cs_{\text{ngap}} = 1\) causes 100% of the gap to close before the pile will reload the medium. The selection of an appropriate value for \(cs_{\text{ngap}}\) requires knowledge of the actual problem conditions.

It is also possible for users to input their own force-displacement law for the normal coupling springs. This is accomplished with a \textit{FISH} function that can be accessed via the pile property keyword \texttt{cs\_nfunc}. The procedure to implement user-defined force-displacement laws is described in Section 1.10.4.

### 1.5.3 Commands Associated with Pile Elements

All the commands associated with pile elements are listed in Table 1.5. This includes the commands associated with the generation of piles and those required to monitor histories, plot and print pile-element variables. See Section 1.3 in the \textbf{Command Reference} for a detailed explanation of these commands.
<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
<th>keyword</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>pile</td>
<td>begin</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>grid</td>
<td>ij</td>
<td></td>
</tr>
<tr>
<td></td>
<td>node</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>end</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>grid</td>
<td>ij</td>
<td></td>
</tr>
<tr>
<td></td>
<td>node</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prop</td>
<td></td>
<td>np</td>
<td></td>
</tr>
<tr>
<td>segment</td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td></td>
<td>&lt;n1 n2&gt;</td>
<td></td>
</tr>
<tr>
<td>node</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>x y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>node</td>
<td>n*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>fix</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>free</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>initial</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xdis</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xvel</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ydis</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yvel</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rvel</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>load</td>
<td>fx fy m</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pin</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>slave</td>
<td>&lt;x&gt; &lt;y&gt; m</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unslave</td>
<td>&lt;x&gt; &lt;y&gt;</td>
<td></td>
</tr>
<tr>
<td>prop</td>
<td></td>
<td>np</td>
<td></td>
</tr>
<tr>
<td></td>
<td>area</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_ncoh</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_nfric</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_nfunc</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_ngap</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_nstiff</td>
<td>value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_nten</td>
<td>value</td>
<td></td>
</tr>
</tbody>
</table>

* For the keywords **fix**, **free**, **initial**, **load** and **pin**, a range of nodes can be specified with the phrase **range n1 n2**.
Table 1.5  Commands associated with pile elements (continued)

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>prop</th>
<th>np</th>
<th>keyword</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>cs_scoh</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>cs_sfric</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>cs_sstiff</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>density</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>e</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>height</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>i</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>perimeter</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>pmom</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>radius</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>spacing</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>width</td>
<td>value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>chprop</th>
<th>nel</th>
<th>np</th>
<th>range</th>
<th>nel1 nel2</th>
</tr>
</thead>
<tbody>
<tr>
<td>hinge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HISTORY</th>
<th>keyword</th>
<th>n</th>
<th>keyword</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td></td>
<td></td>
<td>aaccel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>adisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>avel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nbond</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ndisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nforce</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sbond</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sdisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sforce</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>xaccel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>xdisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>xvel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>yaccel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ydisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>yvel</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>element</th>
<th>nel</th>
<th>keyword</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>axial</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>shear</td>
<td></td>
</tr>
<tr>
<td>Table 1.5</td>
<td>Commands associated with pile elements (continued)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PLOT</strong></td>
<td>pile structure</td>
<td>&lt;pile&gt; keyword</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>adisp &lt;ng &lt;ng2*&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>avel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>axial &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_ndisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_nforce &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_sdisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_sforce &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>element load location &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>material moment &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>nbond node number sbond sdisp shear &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>strain &lt;ng &lt;ng2&gt; &gt; svelxdisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>xvel &lt;ng &lt;ng2&gt; &gt; ydisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>yvel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td><strong>PRINT</strong></td>
<td>structure keypord</td>
<td>pile hinge node property pile</td>
<td></td>
</tr>
</tbody>
</table>

* A range of group ID numbers can be specified for plotting by giving a beginning number ng and an ending number ng2. All groups within this range will be plotted.
1.5.4 Example Applications

Simple examples are provided to illustrate the behavior of pile elements in FLAC. These examples use FISH to access pile element variables for specifying specific loading and output conditions. Pile element variables are accessed via FISH using the “STR.FIN” file. See Section 1.10 for a description and examples for using FISH to access structural element data.

1.5.4.1 Axially Loaded Pile

Piles transfer axial loads to the ground via two mechanisms: skin friction and end bearing. Both are examined in the following examples.

Engineering solutions for axially loaded piles are commonly based on axisymmetric point-load solutions (e.g., the Boussinesq solution – see Geddes 1969). In FLAC, the pile is represented in plane-strain mode as a pile wall extending out of the plane of the cross section (in the z-direction). For comparison with FLAC in the following examples, the engineering solution is adapted to the plane-strain mode. Although the pile formulation in FLAC cannot be applied to simulate a single pile, the plane-strain mode can be used to represent equally spaced piles (see Section 1.9.4 for details on scaling).

Skin Friction – In this example, the vertical stresses in the ground (calculated by FLAC) that result from uniform skin friction resistance of an axially loaded pile, are compared to those estimated from a simple stress approximation. This approximation is based on integration of the Flamant solution for a point load on an elastic half-plane (adapted from the Terzaghi (1943) and Geddes (1969) approach to the plane-strain case).

The problem is analyzed in plane strain and half symmetry. A total axial load $2P$ is applied at the top of a vertical pile of length $D$ located along the symmetry plane. A system of Cartesian axes is defined with the $y$-axis pointing upwards along the pile, and in which the pile top is at elevation $y_0$ (see Figure 1.49).

![Figure 1.49 Friction pile loading conditions](image-url)
The load is transferred to the medium through uniform skin friction by selecting a constant shear spring stiffness, \( csstiff \), that is two orders of magnitude smaller than both the medium bulk modulus, \( K \), and the pile axial stiffness, \( E \). The shear cohesion, \( cscoh \), is set to a high value to prevent shear yielding.

For comparison to the stress-approximation solution, far-field stress boundary conditions are applied, which correspond to the Flamant solution for a point surface load of intensity \( 2P \), equal to that of the applied pile load. This solution has the form:

\[
\begin{align*}
\sigma_{xx} &= -\frac{4P}{\pi} \frac{(y_0 - y)x^2}{[(y_0 - y)^2 + x^2]^2} \\
\sigma_{yy} &= -\frac{4P}{\pi} \frac{(y_0 - y)^3}{[(y_0 - y)^2 + x^2]^2} \\
\sigma_{xy} &= +\frac{4P}{\pi} \frac{(y_0 - y)^2 x}{[(y_0 - y)^2 + x^2]^2}
\end{align*}
\] (1.38)

In the stress-approximation solution, the stresses due to uniform skin friction are estimated by assuming a load distribution of intensity \( 2P/(2D) \) over a vertical height \( D \) (see Figure 1.49). Neglecting the influence of the overburden at the elevation \( y = y_0 - h \), and using the Flamant solution, the vertical stress contribution caused by a load increment \( 2\,dP = \frac{P}{D}\,dh \) over the height \( dh \) may be approximated by

\[
d\sigma_{yy} = -\frac{2P}{\pi D} \frac{(y_0 - h - y)^3}{[(y_0 - h - y)^2 + x^2]^2} dh
\] (1.39)

Summing up incremental load contributions above the current elevation, we obtain

\[
\sigma_{yy} = -\frac{2P}{\pi D} \int_0^z \frac{(y_0 - h - y)^3}{[(y_0 - h - y)^2 + x^2]^2} dh
\] (1.40)

where \( z = y_0 - y \) for \( y \geq y_0 - D \), and \( z = D \) for \( y \leq y_0 - D \).

Upon integration, we obtain, for \( y \geq y_0 - D \),

\[
\sigma_{yy} = \frac{P}{\pi D} \left[ \ln \frac{x^2}{(y_0 - y)^2 + x^2} + \frac{(y_0 - y)^2}{(y_0 - y)^2 + x^2} \right]
\] (1.41)
and, for $y \leq y_0 - D$,

$$
\sigma_{yy} = \frac{P}{\pi D} \left[ \ln \frac{(y_0 - D - y)^2 + x^2}{(y_0 - y)^2 + x^2} + \frac{x^2}{(y_0 - D - y)^2 + x^2} - \frac{x^2}{(y_0 - y)^2 + x^2} \right] \quad (1.42)
$$

Vertical stresses calculated by FLAC are compared to those given by Eqs. (1.41) and (1.42). The FISH function `check_syy` calculates the vertical stress approximation at the location of zone centroids, and stores the values in `ex_1`. Pile element variables are accessed via FISH using the “STR.FIN” file (see Section 1.10).

Note that combined damping (SET `st_damp combined`), rather than local damping, is used for this example (see Section 1.9.3 for a discussion on damping mode). Example 1.19 contains the FLAC data file for this example.

**Example 1.19 Axially loaded pile – skin friction**

```plaintext
config ex 2
def ini_flam
  c_P = -1.e-3 ; half force on plane (symmetry)
  c_g = 3e2
  c_k = 5e2
  niz = 10
  nigp = niz+1
  njz = 10
  njgp = njz+1
end
ini_flam
grid niz njz
gen 0 0 0 10 10 10 10 0
model e
prop dens 2000 sh=c_g bu=c_k
; -- far field boundary conditions --
def coe_flam
  coe = 4.*c_P/pi
  c_yt = y(1,njgp)
end
coe_flam
def c_sxy
  c_x2 = c_x*c_x
  val = c_yt - c_y
  c_y2 = val*val
  c_dis2 = c_x2 + c_y2
  c_dis4 = c_dis2*c_dis2
  c_sxx = coe*val*c_x2/c_dis4
  c_syy = coe*val*c_y2/c_dis4
```

*FLAC Version 6.0*
c_sxy = -coe*c_x*c_y2/c_dis4
end
def ff_bc
  c_y = y(1,1)
jval = 1
loop ival (1,niz)
  ival1 = ival+1
  c_x = 0.5*(x(ival,jval)+x(ival1,jval))
  command
    apply sxy c_sxy i=ival,ival1 j=1
    apply syy c_syy i=ival,ival1 j=1
  end_command
end_loop
ival = nigp
c_x = x(nigp,1)
loop jval (1,njz)
  jval1 = jval+1
  c_y = 0.5*(y(ival,jval)+y(ival,jval1))
  command
    apply sxy c_sxy i=nigp j=jval,jval1
    apply sxx c_sxx i=nigp j=jval,jval1
  end_command
end_loop
end
; -- rigid medium-rigid pile-soft spring for constant tau --
stru pile beg 0,5 end 0,10. seg 10 prop 3001
stru pro 3001 e 5e2 a 0.5 cs_scoh 1e10 cs_sstiff 1. perim 1. ; soft spring
; -- boundary conditions --
fix x i=1
ff_bc
fix y i=1 j=1
stru node 11 load 0.,c_P,0.
; -- settings --
set st_damp struc combined
; -- histories --
hist unbal
hist ydisp i=6 j=11
hist yvel i=6 j=11
; -- test --
solve
save frict.sav
; comparison of numerical and analytical approximation (plane strain)
; ex.1: syy estimated by integration of Flamant solution
; The approximate solution was derived using the plane strain
; equivalent of the technique used by Geddes for the
; axisymmetric case (based on Boussinesq’s solution)
ca str.fin

def find_pd
    vall = 0.
    valf = 0.
    nn = 0
    nind = imem(str_pnt + $ksnode)

loop while nind # 0
    nn = nn + 1
    fs = fmem(nind + $kndfs)
    len = fmem(nind + $kndefl)
    vall = vall + len
    valf = valf + fs
    nind = imem(nind)
end_loop

c_pd = abs(valf)/vall
end

find_pd

print nn vall valf c_pd
pause

def check_syy
    yt = y(1,njgp)
    dd = 5.
    scoe = c_pd/pi

loop ii (1,niz)
    loop jj (1,njz)
        xx = (x(ii,jj)+x(ii+1,jj))*0.5
        yy = (y(ii,jj)+y(ii,jj+1))*0.5
        dis = (yt-yy)*(yt-yy)+xx*xx
        disd = (yt-yy-dd)*(yt-yy-dd)+xx*xx
        if yy < dd then
            ex_1(ii,jj)=scoe*(ln(disd/dis)+xx*xx/disd-xx*xx/dis)
        else
            ex_1(ii,jj)=scoe*(ln(xx*xx/dis)+1.-xx*xx/dis)
        end_if
    end_loop
end_loop
end

check_syy

plot hold syy int 5e-5 fill grid bl struct cs_sforce
plot hold ex_1 zone int 5e-5 fill grid bl struct cs_sforce
save frict_compare.sav

**Figure 1.50** shows contours of vertical stresses obtained numerically, and a diagram of forces in the pile shear-coupling springs. The contours in **Figure 1.51** correspond to the analytical approximation. As may be observed, the same order of magnitude for the stresses is obtained in the two solutions. (Discrepancies between the two responses may be attributed to the approximate character of the
analytical solution.) This example serves to demonstrate the capability of FLAC to transmit loads adequately to the medium by means of the coupling-spring connections.

**Figure 1.50** Shear force at pile/grid interface and FLAC vertical stress contours

**Figure 1.51** Shear force at pile/grid interface and estimated vertical stress contours
**End Bearing** – One way to model end-bearing capacity in FLAC is to neglect shear friction along
the bottom segment of the pile and, instead, adjust the properties of the corresponding shear spring
to account for the pile-bearing capacity. The approach of assigning a limit-load value on input
is justified by considering that the indentation mechanism responsible for the limit load may not
develop naturally in the approximate FLAC model adopted for the pile analysis (because the pile
model does not account for cavity expansion).

In this example, the end-bearing spring is assigned a limit load, evaluated using an engineering
bearing capacity formula (e.g., see Coduto 1994). The limit load can also be derived from a cavity
expansion theory (e.g., see Bishop et al. 1945, Johnson 1970, and Teh and Houlsby 1991) or from
information derived from full-scale tests.

To illustrate the effect of end-bearing capacity, we first consider the case corresponding to a friction
pile in sand (see the data file in Example 1.20). The pile and grid geometries are similar to those
of the previous example. Here, the property \(cs\_scoh\) is set to zero and the pile friction to 10\(^\circ\), so
that the pile shear strength is generated by confining pressure according to Eq. (1.34). A downward
velocity is applied at the pile top, and reaction force and vertical displacement are monitored there
for a total of 20,000 steps.

As may be observed from the pile top force-displacement history in Figure 1.52, the pile capacity
has been reached by the end of the test. Figure 1.53 shows the distribution of shear forces and yield
indicator in the shear coupling springs after 20,000 steps. (The slight irregularity in the shear force
profile is caused by the averaging process used to evaluate mean confining stress for definition of
the shear strength property.)

The data file Example 1.20 is then modified to simulate end bearing. This is accomplished by
assigning a limit-load capacity to the structural node at the base of the pile. The structural node
at the pile base (node 11) is the end-bearing node. The shear cohesion and shear stiffness of the
coupling spring associated with node 11 are adjusted to represent the effect of an end-bearing
condition.

The stiffness, \(K_b\), of the bearing spring (at node 11) is adjusted to reflect the pile axial properties
using

\[
K_b = \frac{EA}{L_e^2} \tag{1.43}
\]

where \(E\) is the pile elastic modulus, \(A\) is the cross-sectional area, and \(L_e\) is the effective length
assigned internally to the bearing spring. The value for \(K_b\) (16 \(\times\)10\(^9\) N/m/m) is assigned via
\texttt{cs\_sstiff}. Conceptually, the bearing spring acts as a pile extension, and the spring force may be
interpreted as the axial force at the pile base.

The limit load, \(F_s^{\text{max}}\), in the bearing spring is evaluated from an engineering formula (e.g., see
Cernica 1995, p. 395 and p. 120):

\[
F_s^{\text{max}} = A(cN_c + \gamma LN_q) \tag{1.44}
\]
where $c$ is the medium cohesion, $N_c = 2\sqrt{K_p(K_p + 1)}$, $K_p = (1 + \sin\phi)/(1 - \sin\phi)$, $\phi$ is the soil friction and $N_q = K_p^2$, $\gamma$ is the unit weight of the medium, and $L$ is the height of the overburden at the pile-base horizon (buried pile length). The input value of $\text{cs\_scoh}$ for the bearing spring (at node 11) is calculated by dividing $F_{s}^{\text{max}}$ by the spring effective length $L_e (F_{s}^{\text{max}}/L_e = 0.82 \times 10^6 \text{ N/m})$, and the property of $\text{cs\_sfric}$ is set to zero.

In order to change the properties of the bottom node, a different property number is first assigned to the bottom pile element. (Property number 3001 is assigned to element 10, and property number 3002 is assigned to elements 1 through 9.) Because only pile element segments can have property numbers changed with the STRUCT chprop command, it is necessary to use a FISH function to re-assign the structural element property number of the structural node and coupling spring, associated with the node adjacent to the bottom node. Using FISH function endb\_prop, structural node 10 is assigned property number 3002, while node 11 (the bottom node) is assigned property number 3001. Note that the FISH function is invoked after one calculation step, because property numbers are only assigned to nodes when the calculation begins. Use the PRINT struct node info command to check property number assignment to nodes.

The modified data file is listed in Example 1.21. (Pile element variables are accessed via FISH using the “STR.FIN” file (see Section 1.10).) The model is run with the same applied velocity and number of steps as in the previous example. In this case, as may be seen from the plot of load versus pile-top settlement in Figure 1.54, the pile capacity has increased. Full capacity is not reached by the end of the simulation, although the shear capacity is attained as indicated in Figure 1.55; all shear springs are at yield, but the bearing spring at node 11 is still intact.
Example 1.20 Axial loading of a friction pile in sand

```plaintext
config ex 2

def ini_flam
    c_g = 25.6e6
    c_k = 30.30e6
    niz = 10
    nigp = niz+1
    njz = 10
    njgp = njz+1
end
ini_flam
grid niz njz
gen 0 0 0 10 10 10 10 0
model mo
prop dens 1980 sh=c_g bu=c_k fr = 20.
stru pile beg 0,10 end 0,5 seg 10 pro 3002
stru pro 3002 e 2e9 a 0.5 cs_scoh 0.0 cs_sstiff 1e8 cs_sfric 10 perim 1.
    ; -- boundary conditions --
    fix x i=1
    fix y j=1
    fix x i=nigp
    ; -- initial conditions --
    set st_damp struct combined
    set grav 10
    ini syy -1.98e5 var 0 1.98e5
    ini sxx -1.10e5 var 0 1.10e5
    ini szz -1.10e5 var 0 1.10e5
    ; -- histories --
    history 1 element 1 axial
    history 2 node 1 ydisplace
    ; -- test --
    stru node 1 fix y
    stru node 1 ini yvel -1e-7
    history 999 unbalanced
cycle 20000
save fbear.sav
plot hold his 2 max 6e4 vs -3
plot hold grid struc cs_sforce struc sb red
```
Example 1.21 Axial loading of an end bearing pile in sand

```plaintext
config ex 2
def ini_flam
  c_g = 25.6e6
  c_k = 30.30e6
  niz = 10
  nigp = niz+1
  njz = 10
  njgp = njz+1
end
ini_flam
grid niz njz
gen 0 0 0 10 10 10 10 0
model mo
prop dens 1980 sh=c_g bu=c_k fr = 20.
; -- fish functions --
ca str.fin
def endb_prop
; assign property block of node 1 to node 10
  nind = imem(str_pnt + $ksnode)
loop while nind # 0
  id = imem(nind + $kndid)
  if id = 1 then
    addr = imem(nind + $kndtad)
    topn = nind
    nind = 0
  end_if
  if nind # 0 then
    nind = imem(nind)
  end_if
end_loop
nind = imem(str_pnt + $ksnode)
loop while nind # 0
  id = imem(nind + $kndid)
  if id = 10 then
    imem(nind + $kndtad) = addr
    nind = 0
  end_if
  if nind # 0 then
    nind = imem(nind)
  end_if
end_loop
end
; -- shear strength in prop 3002, end bearing in prop 3001 --
```

FLAC Version 6.0
stru pile beg 0,10 end 0,5 seg 10 pro 3001
stru pro 3001 e 2e9 a 0.5 cs_scoh 0.82e6 cs_sstiff 16e9 cs_sfric 0. per 1.
stru pro 3002 e 2e9 a 0.5 cs_scoh 0.0 cs_sstiff 1e8 cs_sfric 10 per 1.
stru chprop 3002 range 1 9
; -- boundary conditions --
fix x i=1
fix y j=1
fix x i=nigp
; -- initial conditions --
set st_damp struc combined
set grav 10
ini syy -1.98e5 var 0 1.98e5
ini sxx -1.10e5 var 0 1.10e5
ini szz -1.10e5 var 0 1.10e5
; -- histories --
history 1 element 1 axial
history 2 node 1 ydisplace
; -- test --
stru node 1 fix y
stru node 1 ini yvel -1e-7
hist 999 unbal
step 1
endb_prop
step 19999
save ebear.sav
plot hold his 2 max 6e4 vs -3
plot hold grid struc cs_sforce struc sb red
**Figure 1.52**  Friction pile: top force-displacement history

**Figure 1.53**  Friction pile: forces and yield indicators in shear springs
Figure 1.54  End bearing pile: top force-displacement history

Figure 1.55  End bearing pile: forces and yield indicators in shear springs
1.5.4.2 Laterally Loaded Pile

In this example, a vertical pile is subjected to a lateral displacement at the top of the pile. The pile is pushed in one direction. The loading is then reversed and the pile is pushed in the opposite direction; the loading is reversed one more time, and the pile is pushed in the original direction. This loading cycle is performed for the case of no normal gap present, and then for 100% of the gap effective. The results demonstrate the response of the pile-element model to lateral loading with and without the effect of a gap between the pile and the medium.

The data file for this example is in Example 1.22. The horizontal movement and the shear load at the pile top are monitored. For this problem, the cohesion of the normal coupling spring is specified as 0.01 MN/m, and the friction is set to zero. Also note that the default (local) damping is used for this example; combined damping is not required because velocity sign-changes occur.

**Example 1.22 Laterally loaded pile**

```plaintext
config extra 3
ca str.fin
;
; calculate mean stresses
def mean_stress
  loop ii (1, izones)
    loop jj (1, jzones)
      ex_1(ii,jj) = (sxx(ii,jj) + szz(ii,jj))/2.0
      ex_2(ii,jj) = 0.0
      ex_3(ii,jj) = 0.0
    endloop
  endloop
end
;
grid 11 11
mo el
prop bulk 5e9 shear 1e9 dens 2000
;
; boundary conditions
fix y j 1
fix x i 1
fix x i 12
;
; initial stress state
ini syy -2.2000E+05 var 0 2.2000E+05
ini sxx -1.3200E+05 var 0 1.3200E+05
ini szz -0.8800E+05 var 0 0.8800E+05
set grav 10
;
; 1 m diameter friction pile
```
stru pile begin 5.5 12.0 end 5.5 4.0 prop 3001 segment 8
struc prop 3001 e 8e10 radius = 0.5 perimeter = 3.14
struc prop 3001 cs_sstif = 1.3e11 cs_scoh = 1e10 cs_sfric = 30
struc prop 3001 cs_nstif = 1.3e9 cs_ncoh = 1e4 cs_nfric = 0

save sh_p0.sav
; set cs_ngap = 0 no gap (default : cs_ngap = 0)
struc prop 3001 cs_ngap = 0
;
; histories
hist unbal
hist node 1 xdisp
hist elem 1 shear
;
; apply horizontal loading (velocity)
stru node 1 fix x ini xvel -1e-7
set large
step 4000
mean_stress
save sh_p1.sav
; reverse horizontal loading
stru node 1 ini xvel 1e-7
step 8000
mean_stress
save sh_p2.sav
; reverse horizontal loading again
stru node 1 ini xvel -1e-7
step 8000
mean_stress
plot hol bou struc cs_nfor fil ex_1,2,3 zone alias 'Mean Stress' pile lmag
plot hol hist -3 v -2
save sh_p3.sav
;
restore sh_p0.sav
; set cs_ngap = 1 for total gap effective (default : cs_ngap = 0)
struc prop 3001 cs_ngap = 1
;
; histories
hist unbal
hist node 1 xdisp
hist elem 1 shear
;
; apply horizontal loading (velocity)
stru node 1 fix x ini xvel -1e-7
set large
step 4000
mean_stress
save sh_p4.sav
; reverse horizontal loading
stru node 1 ini xvel 1e-7
step 8000
mean_stress
save sh_p5.sav
; reverse horizontal loading again
stru node 1 ini xvel -1e-7
step 8000
mean_stress
plot hol bou struc cs_nfor fil ex_1,2,3 zone alias 'Mean Stress' pile lmag
plot hol hist -3 v -2
save sh_p6.sav

Figure 1.56 shows the normal loading at the pile/grid interface at the end of the first loading increment. Figure 1.57 shows the same results after the loading is reversed. In the first plot, the mean stress is increased to the left of the pile; in the second plot, the mean stress is increased to the right.

In both the case without a gap and that with a gap, the limiting normal force is the same. However, the normal load versus displacement histories are different. Figures 1.58 and 1.59 show the different results that are calculated when the gap is not present and when it is fully effective.
Figure 1.56  Normal force at pile/grid interface and mean stress in grid at 4000 steps

Figure 1.57  Normal force at pile/grid interface and mean stress in grid at 12,000 steps for full gap
Figure 1.58  Shear load at top of pile versus horizontal displacement for no gap

Figure 1.59  Shear load at top of pile versus horizontal displacement for full gap
1.5.4.3 Softening Plastic Hinge

Once plastic rotation occurs at a particular location in a pile, this spot is weakened, and future deformation will tend to occur at the same location. By default, the limiting plastic moment does not change if the loading is reversed and the pile is reloaded in the opposite direction. It is possible to simulate a reduction in the limiting plastic moment that is related to softening of the plastic hinge. Softening of plastic hinge nodes is introduced by prescribing a softening relation through a \textit{FISH} function. The function is applied directly to the plastic moment via the \texttt{SET pmom_func} command.

A simple example is presented to illustrate this effect (see Example 1.23). The example conditions are similar to those for Section 1.5.4.2. In addition, a plastic moment of 49.6 kN-m is assigned, and plastic hinge conditions are set for nodes along the length of the pile. The pile is first pushed in one direction; the loading is then reversed and the pile is pushed in the opposite direction. For this example, the limiting plastic moment is reduced linearly from 49.6 kN-m to 10 kN-m as a function of the relative angular rotation (defined as $d\theta$ in function \texttt{pml} in Example 1.23). The updated value is passed to the pile property data using the special function \texttt{fc.arg} (see Section 2.5.5 in the \textit{FISH volume}). Note that the pointer to the node structure, the average axial force in element segments adjacent to the hinge node, the angular displacement, and the plastic moment at the node are all accessed using \texttt{fc.arg} when this function is invoked with the \texttt{SET pmom_func} command.

The result using the softening hinge is shown in Figures 1.60 and 1.61. The plastic rotation localizes at a single node, as shown in Figure 1.60. (Note that the pile geometry is magnified 80 times.) The limiting moment at this node reduces as the loading is reversed, as shown by the moment-displacement curve in Figure 1.61.

\textbf{Example 1.23 Softening plastic hinge}

\begin{verbatim}
config extra 3
;
; calculate mean stresses
def mean_stress
   loop ii (1, izones)
      loop jj (1, jzones)
         ex_1(ii,jj) = (sxx(ii,jj) + szz(ii,jj))/2.0
         ex_2(ii,jj) = 0.0
         ex_3(ii,jj) = 0.0
      endloop
   endloop
;
grid 11 11
mo el
prop bulk 5e9 shear 1e9 dens 2000
;
; boundary conditions
fix y j 1
\end{verbatim}
fix x i 1
fix x i 12
;
; initial stress state
ini syy -2.2000E+05 var 0 2.2000E+05
ini sxx -1.3200E+05 var 0 1.3200E+05
ini szz -0.8800E+05 var 0 0.8800E+05
set grav 10
;
; 1 m diameter friction pile
stru pile begin 5.5 12.0 end 5.5 4.0 prop 3002 segment 8
struc prop 3001 e 8e10 radius = 0.5 perimeter = 3.14
struc prop 3001 cs_sstif = 1.3e11 cs_scoh = 1e4 cs_sfric = 30
struc prop 3001 cs NSTIF = 1.3e10 cs_ncoh = 1e4 cs_nfric = 0
struc prop 3002 e 8e10 radius = 0.5 perimeter = 3.14
struc prop 3002 cs_sstif = 1.3e11 cs_scoh = 1e5 cs_sfric = 30
struc prop 3002 cs NSTIF = 1.3e11 cs_ncoh = 1e5 cs_nfric = 0
struc chprop 3001 range 1 7
save soft_p0.sav
call str.fin
;
def pm1max0
  pm1max = 4.96e4
  pm1min = 1e4
end
pm1max0
;
def pm1
; NOTE: Check STR.FIN file for availability of
; spare node offsets
; argument 1 : pointer to structure node
ipn = fc_arg(1)
; argument 2 : average axial force in adjacent elements
fax = fc_arg(2)
; argument 3 : hinge rotation
dth = fc_arg(3)
; argument 4 : plastic moment
pmom = fc_arg(4)
;
; rotation of node a (angular disp)
tha = fmem(ipn+$kndth)
; rotation of node b (angular disp)
thb = fmem(ipn+$kndhth)
; relative rotation
dth = tha-thb
; dthmax = 1.0e-2
;
if abs(dth) > 0.0
   sl = (pmlmax-pmlmin) / dthmax
   _del=abs( abs(dth)-abs(fmem(ipn+$kndsp8)) )
   if fmem(ipn+$kndsp9)= 0.0
      pmom = pmlmax - sl*_del
   else
      pmom = abs(fmem(ipn+$kndsp9)) - sl * _del
   endif
   if pmom < pmlmin
      pmom = pmlmin
   endif
   fmem(ipn+$kndsp9)=pmom
   fmem(ipn+$kndsp8)=dth
endif
;
fc_arg(4) = pmom
pmomh = pmom
end
; set pmom limit
struc prop 3001 pmom 4.96e4
struc prop 3002 pmom 4.96e4
set pmom_func pml
struc hinge 1 8
;
; histories
hist unbal
hist node 1 xdisp
hist elem 1 shear
hist elem 4 moment1
;
; apply horizontal loading (velocity)
stru node 1 fix x ini xvel -5e-8
set large
step 60000
mean_stress
save soft_p1.sav

; reverse horizontal loading
stru node 1 ini xvel 5e-8
step 120000
mean_stress
save soft_p2.sav
; reverse horizontal loading
stru node 1 ini xvel -5e-8
step 110000
mean_stress
save soft_p3.sav

; reverse horizontal loading
stru node 1 ini xvel 5e-8
step 100000
mean_stress
save soft_p4.sav
Figure 1.60 Magnified plot of pile geometry after load reversal in lateral direction for softening plastic hinges (magnification factor = 80)

Figure 1.61 Moment versus x-displacement at softening node
1.5.4.4 Determining Coupling-Spring Pile Properties

In many cases, properties needed to characterize the response of pile/soil interaction for piles are not available. However, basic site-specific soil properties are usually provided from standard laboratory and in-situ testing. It is possible to estimate pile/soil interaction properties from basic soil properties; this example illustrates one method to derive a set of coupling-spring parameters.

An analysis is required for lateral-loading of a row of piles as a result of soil movement from an actively failing slope. The piles are 0.02 m thick, 0.762 m diameter steel pipes, and are spaced at 3.66 m. Before performing this analysis we need to determine the representative shear and normal coupling-spring properties for an individual pile.*

In this example, the soil is a sand-clay mixture, and failure associated with the soil/structure response is assumed to occur within the soil (i.e., there is a rough interface between the pile and the soil). The soil is assumed to behave as a Mohr-Coulomb material with a friction angle of 22°, a cohesion of 24 kPa, a dilation angle of zero and tensile strength of zero. The elastic behavior of the soil is defined by a bulk modulus of 75 MPa and a shear modulus of 12.6 MPa.

The shear response of the soil/structure interface can be estimated by the following relations:†

a. shear coupling-spring frictional resistance \( (c_{s}\text{fric}) \) can be taken as the internal friction angle of the soil (i.e., \( c_{s}\text{fric} = 22^\circ \)); and

b. shear coupling-spring cohesive strength \( (c_{s}\text{coh}) \) can be taken as the cohesion of the soil times the perimeter of the pile (e.g., for a circular pile, \( 2\pi \) times the radius) – i.e., \( c_{s}\text{coh} = 57.45 \text{ kN/m} \).

A plane-strain FLAC model that simulates the process of pushing a row of piles laterally into the soil can be used to estimate the normal coupling-spring properties. Figure 1.62 shows the concept of this model. The symmetry conditions for this model represent a row of piles with a uniform spacing. The pile is represented by beam elements located along the pile periphery and connected to the grid via interface elements. In this way, the pile/soil interaction can be included in the analysis. Figure 1.63 displays the FLAC model. The pile is pushed into the soil by applying a constant negative y-velocity at the beam nodes. As the pile is pushed into the soil, we monitor the lateral forces exerted on the pile and the relative displacement of the pile through the soil. The maximum lateral force depends on the initial confining stress in the soil. If the stiffness and strength properties

* Note that the individual pile properties are scaled to represent a row of piles in the FLAC analysis for this problem by specifying \( \text{spacing} = 3.66 \). See Section 1.9.4 for the scaling relations.

† The values selected for \( c_{s}\text{fric} \) and \( c_{s}\text{coh} \) reflect the roughness of the pile surface. These values should be reduced for a smooth pile surface; a reduction factor of 2/3 is often selected for smooth piles.
of the soil are constant with respect to the stress and deformation of the soil, then only two models, at different initial confining stresses, are required to estimate the normal coupling-spring parameters. *

Figure 1.62  Conceptual model to estimate normal coupling-spring properties

* This procedure is still valid if the soil properties are nonlinear, but several models may be required to cover the expected confining stress regime. The normal coupling-spring behavior may then be implemented via a FISH function by using the cs_nfunc keyword.
In this model, the in-plane stresses, $\sigma_{xx}$ and $\sigma_{yy}$, define the horizontal stresses, and the out-of-plane stress, $\sigma_{zz}$, defines the vertical stress for the initial stress state. Two conditions for the initial stress state are analyzed in this example:

soil stresses at 6 m depth:

\[
\begin{align*}
\sigma_{xx} &= -0.0912 \text{ MPa} \\
\sigma_{yy} &= -0.0912 \text{ MPa} \\
\sigma_{zz} &= -0.126 \text{ MPa}
\end{align*}
\]

soil stresses at 12 m depth:

\[
\begin{align*}
\sigma_{xx} &= -0.182 \text{ MPa} \\
\sigma_{yy} &= -0.182 \text{ MPa} \\
\sigma_{zz} &= -0.252 \text{ MPa}
\end{align*}
\]

$FISH$ functions are used to monitor the maximum normal force per length of pile, $F_{n}^{\text{max}}/L$, and the average mean effective stress, $p'$, in the model times the exposed perimeter. $p'$ is calculated as the mean value of $0.5 (\sigma_{xx} + \sigma_{yy})$ for all zones in the model. The exposed perimeter in this example is considered to be the lower half of the pile boundary that is applying a velocity into the grid.
The relative displacement of the pile through the soil is calculated as the difference between the
y-displacement of the gridpoint at the pile centerline \((i = 1, j = 44)\) and the y-displacement of the
gridpoint at the midpoint between piles \((i = 21, j = 51)\). Example 1.24 lists the data file including
the *FISH* functions for this example.

### Example 1.24 Determining normal coupling-spring properties

```plaintext
; *** File used to determine normal capacity of a single row
; *** of 0.762 m (2.5 ft) diameter piles
; *** spaced 3.66 m (12 feet) c/c in a sandy clay ...
; *** The file can be modified for different initial soil
; *** stress conditions ...
;
grid 20 100
model mohr
gen 0 -10 0 1.83 0 1.83 -10 rat 1.08 0.95 j 1,51
gen 0 0 0 10 1.83 10 1.83 0 rat 1.08 1.0526 j 51,101
gen cir 0 0 0.381
m n reg 1 50
def follow_mark
  nbeam = 0
  iprev = 0
  jprev = 0
  i = istart
  j = jstart
loop n (1,1000)
  section
    i1 = i - 1
    j1 = j
    if isnext = 1
      exit section
    endif
    i1 = i
    j1 = j - 1
    if isnext = 1
      exit section
    endif
    i1 = i + 1
    j1 = j
    if isnext = 1
      exit section
    endif
    i1 = i
    j1 = j + 1
    if isnext = 1
      exit section
```

*FLAC Version 6.0*
endif
n_nodes = nbeam + 1
exit
endSection
x0 = x(i,j)
y0 = y(i,j)
x1 = x(i1,j1)
y1 = y(i1,j1)
nbeam = nbeam + 1
command
    struct beam beg x0,y0 end x1,y1 seg=1 prop 1001
endCommand
iprev = i
jprev = j
i    = i1
j    = j1
endLoop
end

def isnext
isnext = 0
if i1 < 1
    exit
endif
if i1 > igp
    exit
endif
if j1 < 1
    exit
endif
if j1 > jgp
    exit
endif
if i1 = iprev
    if j1 = jprev
        exit
    endif
endif
if and(flags(i1,j1),128) # 0
    isnext = 1
endif
end

set istart=1 jstart=44
follow_mark
struc prop 1001 e=1 a 1 i 1
struc node range 1 n_nodes fix x y r
struc node range 1 n_nodes ini yvel -1e-6

FLAC Version 6.0
int 1 as from node n_nodes to node 1 bs from 1,44 to 1 58
int 1 kn 1e10 ks 1e10 fric 45
;
prop b 7.5e7 s 1.26e7 d 2100 coh 23.95e3 fri 22.0 ten 0
;
fix x i 1
fix x i 21
fix y j 1
;
def p_prime
  r_pile = 0.381
  sum = 0.0
  loop i (1,izones)
    loop j (1,jzones)
      if model(i,j) # 1
        sum = sum + sxx(i,j) + syy(i,j)
      endif
    endLoop
  endLoop
  p_prime = (r_pile * pi) * sum / (2.0 * izones * jzones)
end
call str.fin

def F_n
  sum = 0.0
  pnt = imem(str_pnt+$ksnode)
  loop while pnt # 0
    sum = sum + fmem(pnt+$kndf2c)
    pnt = imem(pnt)
  endLoop
  F_n = 2.0 * sum
end
;
def rel_disp
  y_pile = ydisp(1,44)
  y_soil = ydisp(21,51)
  rel_disp = abs(y_pile - y_soil)
end
his 1 p_prime
his 2 F_n
hist 3 rel_disp
his 4 node 1 ydisp
;
set large
;
save se_01_22_in1.sav
;
The value for $F_{n}^{\text{max}}/L$ at the 6 m depth is found to be 1.06 MN/m (as indicated by the plot of force versus relative displacement in Figure 1.64). The mean effective confining stress times the exposed perimeter is found to be 0.26 MN/m ($FISH$ variable $p_{\text{prime}}$, which is also shown in Figure 1.64).

The slope of the plot of $F_{n}^{\text{max}}/L$ versus relative displacement in Figure 1.64 varies from approximately 88.9 MN/m/m at initial loading to approximately 8.2 MN/m/m before reaching the limit load. A secant modulus, measured from the initial load to the limit load, is roughly 17.2 MN/m/m.

At 12 m depth, the value for $F_{n}^{\text{max}}/L$ is calculated to be 1.78 MN/m, the mean effective confining stress times the exposed perimeter is 0.46 MN/m, and the secant modulus measured from the initial load to the limit load is approximately 15.7 MN/m/m.

The strength results can now be plotted to determine the coupling-spring parameters, $cs_{\text{nfric}}$ and $cs_{\text{ncoh}}$, as indicated in Figure 1.47(a). The value for $cs_{\text{nfric}}$ is determined from the two stress level tests to be 74.5°, and the value for $cs_{\text{ncoh}}$ is 124 kN/m. The value for $cs_{\text{nstiff}}$ is estimated to be 16.5 MN/m/m, which is the average of the secant moduli measured from the two tests.

The zone of yielding at the maximum normal force is shown in Figure 1.65. Note that the failure extends to the right boundary, which indicates that there is an influence of the pile spacing on the pile/soil response. Also, a gap forms between the pile and grid, as shown in the figure.
The effect of pile spacing can be evaluated by adjusting the width of the model. The interaction of the piles may be expected to affect the calculated values for the normal coupling-spring properties. As the width of the model is increased, the effect of the interaction should diminish.

The values calculated for $cs_{nf\text{ric}}$, $cs_{ncoh}$ and $cs_{nstiff}$ from these tests reflect the behavior both at the pile/soil interface and within a local volume of soil, as a result of the pile movement. This includes the nonlinear deformation due to local failure of the soil material.

The coupling-spring properties derived in the scheme described above are similar to the $p - y$-curves employed in empirical schemes for pile analysis. However, the p-y approach is intended to capture the response of the entire soil mass (in an approximate way), whereas FLAC’s pile properties reflect only the local interaction of pile and soil (within a distance in the order of the pile spacing). The long-range effects (e.g., plastic slip surfaces that develop deep within the soil mass) are modeled in a realistic way by the complete FLAC grid. Therefore, the zones in a region adjacent to the pile elements, and extending from the pile a distance equal to the pile spacing, should be given an elastic model if the pile properties are derived by a scheme similar to that described above. Otherwise, the local pile-soil behavior would be represented twice.

Figure 1.64  Pile lateral force ($F_n/L$) and ($p' \times$ perimeter) versus relative displacement
Figure 1.65  Displacement of pile (beams) and zone of shear and tensile yielding at the maximum normal load
1.6 Rockbolt Elements

1.6.1 Formulation

The rockbolt element is based on the pile element, with axial and bending behavior.* The connection to the grid, in both the normal and shear directions, is via coupling springs, as described in Sections 1.5.1.1 and 1.5.1.2. The following additional behavior is provided for rockbolts:

1. The rockbolt element may yield in the axial direction in both tension and compression (yield and ycomp).

2. Rockbolt breakage is simulated based upon a user-defined tensile failure strain limit (tfstrain). A strain measure, based on adding the axial and bending plastic strains, is evaluated at each rockbolt node. The axial plastic strain, $\varepsilon_{pl}^{ax}$, is accumulated based on the average strain of rockbolt element segments using the node. The bending plastic strain is averaged over the rockbolt and then accumulated. The total plastic tensile strain, $\varepsilon_{pl}$, is then calculated by

$$\varepsilon_{pl} = \Sigma \varepsilon_{pl}^{ax} + \Sigma \frac{d}{2L} \frac{\theta_{pl}}{L}$$  \hspace{1cm} (1.45)

where:

- $d$ = rockbolt diameter;
- $L$ = rockbolt segment length; and
- $\theta$ = average angular rotation over the rockbolt.

If this strain exceeds the limit tfstrain, the forces and moment in this rockbolt segment are set to zero, and the rockbolt is assumed to have failed.

3. The effective confining stress acting on the rockbolt is based on the change in stress since installation. Stresses in the grid around the rockbolt are stored when the element is installed, and as calculation progresses, the effective confining stress around the element is calculated as the change in stress from the installation state. (For the pile element, the effective confining stress is based on the current stress state in the zones surrounding the pile.)

4. A user-defined table (cs_cftable) can be specified to give a correction factor for the effective confining stress, in cases of non-isotropic stress, as a function of a deviatoric stress ratio. By default, the confining stress acting on piles is given by Eq. (1.35). By specifying a table with cs_cftable, factors are applied to the value of $\sigma_m$ to account for non-isotropic stresses.

---

* The rockbolt model was developed in collaboration with Geocontrol S.A., Madrid, Spain for application to analyses in which nonlinear effects of confinement, grout or resin bonding, or tensile rupture are important.
5. Softening as a function of shear displacement for the shear coupling-spring cohesion and friction angle properties can be prescribed via the user-defined tables \texttt{cs\_sctable} and \texttt{cs\_sftable}.

\subsection{1.6.2 Rockbolt-Element Properties}

The rockbolt elements in \textit{FLAC} require the following input parameters (rockbolt property keywords are shown in parentheses):

1. cross-sectional area (\texttt{area} or \texttt{radius}) [length$^2$] of the rockbolt;
2. second moment of area (i) [length$^4$] (commonly referred to as the moment of inertia) of the rockbolt;
3. density (\texttt{density}) [mass/volume] of the rockbolt (optional – used for dynamic analysis and gravity loading);
4. elastic modulus (\texttt{e}) [stress] of the rockbolt;
5. spacing (\texttt{spacing}) [length] (optional – if not specified, rockbolts are considered to be continuous in the out-of-plane direction);
6. plastic moment (\texttt{pmom}) [force-length] (optional – if not specified, the moment capacity is assumed to be infinite);
7. tensile yield strength (\texttt{yield}) [force] of the rockbolt (if not specified, the tensile yield strength is zero);
8. compressive yield strength (\texttt{ycomp}) [force] of the rockbolt (if not specified, the compressive yield strength is zero);
9. tensile failure strain limit of the rockbolt (\texttt{tfstrain});
10. exposed perimeter (\texttt{perimeter}) [length] of the rockbolt (i.e., the length of the rockbolt surface that is in contact with the medium);
11. stiffness of shear coupling spring (\texttt{cs\_sstiff}) [force/rockbolt length/displ.];
12. cohesive strength of shear coupling spring (\texttt{cs\_scoh}) [force/rockbolt length];
13. frictional resistance of the shear coupling spring (\texttt{cs\_sfric}) [degrees];
14. number of table relating cohesion of shear coupling spring to relative shear displacement (\texttt{cs\_sctable});
15. number of table relating friction angle of shear coupling spring to relative shear displacement (\texttt{cs\_sftable});
16. number of table relating confining stress factor to deviatoric stress (\texttt{cs\_cftable});
(17) stiffness of normal coupling spring \((cs\_nstiff)\) [force/rockbolt length/displ.];

(18) cohesive (and tensile) strength of normal coupling spring \((cs\_ncoh)\) [force/rockbolt length]; and

(19) frictional resistance of the normal coupling spring \((cs\_nfric)\) [degrees].

The radius of the rockbolt element cross-section can also be prescribed instead of the area and moment of inertia. The area and moment of inertia will then be calculated automatically.

Rockbolt element properties are determined in a fashion similar to that used for beam elements. (See Section 1.2.2.)

A limiting plastic moment and plastic hinge condition can be prescribed for rockbolt nodes. See Section 1.1.7 for details. Softening relations for plastic hinges can also be defined by the user.

The exposed perimeter of a rockbolt element and the properties of the coupling springs should be chosen to represent the behavior of the rockbolt/medium interface commensurate with the problem being analyzed. The rockbolt/rock interaction can be expressed in terms of a shear response along the length of the bolt as a result of axial loading and/or in terms of a normal response when the direction of loading is perpendicular to the rockbolt axis.

1.6.3 Commands Associated with Rockbolt Elements

All of the commands associated with rockbolt elements are listed in Table 1.6. This includes the commands associated with the generation of rockbolts, and those required to monitor histories, plot and print rockbolt-element variables. See Section 1.3 in the Command Reference for a detailed explanation of these commands.
**Table 1.6  Commands associated with rockbolt elements**

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>rockbolt</td>
<td>keyword</td>
</tr>
<tr>
<td>begin</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>ij</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>ij</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>segment</td>
<td>ns</td>
</tr>
<tr>
<td>delete</td>
<td>&lt;n1 n2&gt;</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>node</td>
<td>n*</td>
</tr>
<tr>
<td>fix</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>free</td>
<td>&lt;x&gt; &lt;y&gt; &lt;r&gt;</td>
</tr>
<tr>
<td>initial</td>
<td>keyword</td>
</tr>
<tr>
<td>xdis</td>
<td>value</td>
</tr>
<tr>
<td>xvel</td>
<td>value</td>
</tr>
<tr>
<td>ydis</td>
<td>value</td>
</tr>
<tr>
<td>yvel</td>
<td>value</td>
</tr>
<tr>
<td>rvel</td>
<td>value</td>
</tr>
<tr>
<td>load</td>
<td>fx fy m</td>
</tr>
<tr>
<td>pin</td>
<td></td>
</tr>
<tr>
<td>slave</td>
<td>&lt;x&gt; &lt;y&gt; m</td>
</tr>
<tr>
<td>unslave</td>
<td>&lt;x&gt; &lt;y&gt;</td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>area</td>
<td>value</td>
</tr>
<tr>
<td>cs_ncoh</td>
<td>value</td>
</tr>
<tr>
<td>cs_nfric</td>
<td>value</td>
</tr>
<tr>
<td>cs_nstiff</td>
<td>value</td>
</tr>
</tbody>
</table>

* For the keywords *fix, free, initial, load* and *pin*, a range of nodes can be specified with the phrase *range n1 n2*. 
### Table 1.6 Commands associated with rockbolt elements (continued)

<table>
<thead>
<tr>
<th>STRUCTURE prop</th>
<th>np</th>
<th>keyword</th>
<th>value</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>cs_scoh</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_sfric</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_sstiff</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_setable</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_sftable</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs_cftable</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>density</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>e</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>i</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>perimeter</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmom</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>radius</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>spacing</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tfstrain</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ycomp</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>yield</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmom</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment1</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment2</td>
<td>value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>shear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HISTORY node</th>
<th>n</th>
<th>keyword</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>aaccel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>adisp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>avel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>nbond</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ndisp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>nforce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sbond</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdisp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sforce</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>xaccel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>xdisp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>xvel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>yaccel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ydisp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>yvel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>element nel</th>
<th></th>
<th>keyword</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>axial</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>moment2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>shear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1.6  Commands associated with rockbolt elements (continued)

<table>
<thead>
<tr>
<th>PLOT</th>
<th>rockbolt</th>
<th>structure &lt;rockbolt&gt; keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>adisp &lt;ng &lt;ng2*&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>avel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>axial &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_ndisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_nforce &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_sdisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cs_sforce &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>element</td>
<td></td>
</tr>
<tr>
<td></td>
<td>location &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>material</td>
<td></td>
</tr>
<tr>
<td></td>
<td>moment &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>nbond</td>
<td></td>
</tr>
<tr>
<td></td>
<td>node</td>
<td></td>
</tr>
<tr>
<td></td>
<td>number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sbond</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sdisp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shear &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>strain &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>svel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xdisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xvel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ydisp &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>yvel &lt;ng &lt;ng2&gt; &gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRINT</th>
<th>structure</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rockbolt</td>
<td>rockbolt</td>
</tr>
<tr>
<td></td>
<td>node</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hinge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>property</td>
<td></td>
</tr>
</tbody>
</table>

* A range of group ID numbers can be specified for plotting by giving a beginning number *ng* and an ending number *ng2*. All groups within this range will be plotted.
1.6.4 Example Applications

Simple examples are provided to illustrate the behavior of rockbolt elements in FLAC. Note that combined damping (**SET st_damp combined**), rather than local damping, is used for these examples (see Section 1.9.3 for a discussion on damping mode).

1.6.4.1 Rockbolt Pullout Tests

The most common method for determination of rockbolt properties is to perform pullout tests on small segments of rockbolts in the field. Typically, segments of 50 cm in length or longer are grouted into boreholes. The ends of these segments are pulled with a jack mounted to the surface of the tunnel and connected to the rockbolt via a barrel-and-wedge type anchor. The force applied to the rockbolt and the deformation of the rockbolt are plotted to produce an axial force-deflection curve. From this curve, the peak shear strength of the grout bond is determined. The results of simulated pullout on one-half meter segments are illustrated in this example.

The data file in Example 1.25 contains several variations of a single rockbolt pull-test. The rockbolt end node is pulled at a small, constant y-direction velocity, as indicated in Figure 1.66. A FISH function **ff** is used to sum the reaction forces and monitor nodal displacement generated during the pull-tests.

![Figure 1.66 Rockbolt element in grid; velocity applied at end node](image-url)
Example 1.25 Rockbolt pullout tests

g 4 6
mo el
gen 0.0,0.0 0.0,0.6 0.4,0.6 0.4,0.0
pro bulk 5e9 she 3e9 den 2000
fix y j 7
set large
set st_damp struc combined
; -- Rockbolt installation --
stru rockbolt beg .2 0.1 end .2 .7 seg 12 prop 4001
stru pro 4001 e 200e9 a 5e-4 cs_scoh 1.00e5 cs_sstiff 2.00e7 per 0.08
stru pro 4001 yield 2.25e5 ; ult. tens. strength (450 MPa)*area=Force
stru pro 4001 i=2e-8 ; 0.25*pi*r^4
; -- Fish functions --
; ff : Pull force in bolt
; dd : Displacement of rockbolt end
def ff
   sum = 0.0
   loop i (1,igp)
      sum = sum+yforce(i,7)
   end_loop
   ff=sum
   dd=step*1e-6
end
; -- Histories --
his nstep 100
hist ff
hist dd
hist unbal
stru node 13 fix y ini yvel 1e-6
save pull0.sav
; -- Pull out tests - single 25mm rockbolt(20 mm deformation) --
; 1. Default behavior
step 20000
save pull1.sav
;
; 2. Cohesion softening
rest pull0.sav
stru pro 4001 cs_sctable=100
table 100 0 1e5 0.01 1e4 ;change in cohesion with relative shear displ.
step 20000
save pull2.sav
;
; 3. Confinement = 5 MPa

FLAC Version 6.0
rest pull0.sav
stru pro 4001 e 200e9 a 5e-4 cs_sstiff 2.00e7 per 0.08
stru pro 4001 yield 2.25e5 ; ult. tens. strength (450 MPa)*area=Force
stru pro 4001 i=2e-8 ; 0.25*pi*r ^ 4
stru pro 4001 cs_sfric=45
stru pro 4001 cs_scoh=0.0
step 1
; -- Fish functions --
def con_p
  cpm = - cp
  command
    ini sxx cpm szz cpm
    app pr cp i 1
    app pr cp i 5
  end_command
end
; confining pressure = 5e6
set cp = 5e6
con_p
step 20000
save pull3.sav
;
; 4. Confinement = 5 MPa with cohesion table
rest pull0.sav
stru pro 4001 e 200e9 a 5e-4 cs_sstiff 2.00e7 per 0.08
stru pro 4001 yield 2.25e5 ; ult. tens. strength (450 MPa) * area = Force
stru pro 4001 i=2e-8 ; 0.25*pi*r ^ 4
stru pro 4001 cs_sfric=45
stru pro 4001 cs_scoh=0.0
; define table for confining stress correction factor
table 1 0.5 0.3 0.48 0.5 0.45 0.6 0.39 0.68 0.36
struct prop 4001 cs_cftable 1
;
; note : (snn-szz)/(snn+szz) is 1 , so cfac=0.36
step 1
; -- Fish functions --
def con_p
  cpm = - cp
  command
    ini sxx cpm
    app pr cp i 1
    app pr cp i 5
  end_command
end
; confining pressure = 5e6
set cp = 5e6

FLAC Version 6.0
con_p
step 20000
save pull4.sav
;
; 5. Tensile rupture
rest pull10.sav
stru pro 4001 e 200e9 a 5e-4 cs_scoh 1.00e5 cs_sstiff 2.00e7 per 0.08
stru pro 4001 yield 1.0e5 ; ult. tens. strength (200 MPa) * area = Force
stru pro 4001 i=2e-8 ; 0.25*pi*r^4
stru pro 4001 cs_sfri=45
stru pro 4001 tfs = 1e-2
step 1
; -- Fish functions --
def con_p
cpm = - cp
    command
        ini sxx cpm szz cpm
        app pr cp i 1
        app pr cp i 5
    end_command
end
; confining pressure = 5e6
set cp = 5e6
con_p
step 30000
save pull15.sav
In the first test, confining stress dependence on the rockbolt shear bond strength is neglected. The resulting axial force-deflection plot is shown in Figure 1.67. The peak force is approximately 50 kN.

**Figure 1.67** Rockbolt pull force (N) versus rockbolt axial displacement (meters) for a single 25 mm grouted rockbolt
In the second test, displacement weakening of the shear bond strength is introduced using the \texttt{cs\_sctable} property. The displacement weakening relation to shear displacement is defined in table 100. The results are shown in Figure 1.68:
The rockbolt shear bond strength will, in general, increase with increasing effective pressure acting on the rockbolt. A linear law is implemented in FLAC, whereby the rockbolt shear strength is defined as a constant ($c_s$ cohesion) plus the effective pressure on the rockbolt multiplied by the rockbolt perimeter ($perimeter$) times the tangent of the friction angle ($c_s$ friction). The pressure dependence is activated automatically by issuing the rockbolt properties $perimeter$ and $c_s$ friction.

In the third test, a 5 MPa confining stress is applied after the rockbolt is installed. Note that one calculational step is taken in order to assign the rockbolt properties before the confining stress is applied. The results are shown in Figure 1.69:

Figure 1.69 Rockbolt pull force (N) versus rockbolt axial displacement (meters) for a single 25 mm grouted rockbolt – with uniform 5 MPa confinement
In the fourth test, the property `cs_cftable` is used to define the confining stress applied to the rockbolt, accounting for the reduced affect of the out-of-plane stress and the in-plane stress normal to the bolt. Table 1 is used to apply the reduction factor. The results are shown in Figure 1.70. Note that the pullout resistance is greatly reduced compared to the previous case (compare Figure 1.70 to Figure 1.69).

**Figure 1.70**  Rockbolt pull force (N) versus rockbolt axial displacement (meters) for a single 25 mm grouted rockbolt – with 5 MPa in-plane confinement and zero out-of-plane confinement
In the fifth test, **yield** is used to define the limiting axial yield force (100 kN) of the bolt, and **tfstrain** is used to define the plastic strain (0.01) at which the bolt ruptures. The results are shown in **Figure 1.71**:

**Figure 1.71**  Rockbolt pull force (N) versus rockbolt axial displacement (meters) for a single 25 mm grouted rockbolt – with tensile rupture
1.6.4.2 Rockbolt Shear Tests

Two shear tests are performed in this example. The tests use the same model as the pullout tests. In this case, though, a horizontal velocity is applied to the top rockbolt node. The data file is listed in Example 1.26. Note that normal coupling spring properties are now included.

Example 1.26 Rockbolt shear tests

```plaintext
g 3 6
mo el
  gen 0.0,0.0 0.0,0.6 0.3,0.6 0.3,0.0
  pro bulk 5e10 she 3e10 den 2000
  fix x y j 1
  fix x i 1
  fix x i 4
set large
set st damp struc combined
; -- Rockbolt installation --
stru rockbolt beg .15 0.1 end .15 .625 seg 25 prop 4001
stru pro 4001 e 200e9 a 5e-4 cs.scoh 1.00e5 cs.sstiff 2.00e7 per 0.08
stru pro 4001 yield 2.25e5 ; ult. tens. strength (450 MPa) * area = Force
stru pro 4001 i=2e-8 ; 0.25*pi*r^4
stru prop 4001 cs.nstiff 1e10 cs.ncoh 2e6 cs.nfric=45
; -- Fish functions --
; ff : Pull force in bolt
; dd : Displacement of rockbolt end
def ff
  sum = 0.0
  loop i (1,igp)
    loop j (1,jgp)
      sum = sum+xforce(i,j)
    end
  end
  end
  ff=sum
  dd=step*1e-6
end
; -- Histories --
his nstep 100
hist ff
hist dd
hist unbal
; -- Shear test --
stru node 26 fix x ini xvel 1e-6
save shear0.sav
step 30000
save shear1.sav
```
Figure 1.72 shows the plot of shear force versus shear displacement for a non-yielding bolt. Figure 1.73 shows the rockbolt geometry at the end of the test. The large displacement of the rockbolt near the rock surface is a result of the failure of the normal coupling springs, which simulates the crushing of the rock.

In the second test, `pmom` is specified to define a limiting moment (5000 N-m) of the bolt, and `tfstrains` is set to define a limiting plastic strain (0.01) at which the bolt ruptures. The results are shown in Figures 1.74 and 1.75.
Figure 1.72  Rockbolt shear force (N) versus rockbolt shear displacement (meters) for a single 25 mm grouted rockbolt

Figure 1.73  Deformed shape of 25 mm diameter rockbolt at end of shear test
Figure 1.74  Rockbolt shear force (N) versus rockbolt shear displacement (meters) for a single 25 mm grouted rockbolt – with tensile rupture

Figure 1.75  Deformed shape of 25 mm diameter rockbolt following rupture at end of shear test
1.7 Strip Elements

1.7.1 Formulation

The strip element is a type of structural element specifically designed to simulate the behavior of thin, flat reinforcing strips placed in layers within a soil embankment to provide support.* Figure 1.76 shows a typical reinforced earth retaining wall containing layers of strip reinforcement.

The strip element has characteristics similar to the rockbolt element and the cable element. The strip can yield in compression and tension, and a rupture limit can be defined, which is similar to rockbolt behavior. Strips provide shear resistance but cannot sustain bending moments, which is similar to cable behavior. In addition, the shear behavior at the strip/soil interface is defined by a nonlinear shear failure envelope that varies as a function of a user-defined transition confining pressure.

* The strip model was developed in collaboration with Terre Armée/Reinforced Earth Company, Soiltech R & D Division, Nozay, France. The model was developed to represent the behavior of the Terre Armée reinforcing strips.
The strip element has the following characteristics:

1. The reinforcing strips are prescribed by the number of strips (\textit{nstrips}) per calculation width (\textit{calwidth}), measured out-of-plane. The individual strip thickness (\textit{strthickness}) and strip width (\textit{strwidth}) are also input.

2. The elastic stiffness of the strip is defined by the cross-sectional area of the strip per calculation width (out-of-plane) and the Young’s modulus (\textit{E}) of the strip material.

3. The strip may yield in tension (defined by the strip tensile yield-force limit, \textit{stryield}) and in compression (defined by the strip compressive yield-force limit, \textit{strcomp}).

4. Strip breakage is simulated with a user-specified tensile failure strain limit (\textit{tfstrain}). The strain measure is based on the accumulated plastic strain calculated at each strip segment along the length of the strip. The strip breakage formulation is similar to that used for rockbolts (see Eq. (1.45)), except that bending strain is not included in the strip breakage calculation. If the plastic strain at a segment exceeds the tensile failure strain limit, the strip segment is assumed to have failed, the forces in the strip segment are set to zero, and the segment is separated into two segments.

5. The shear behavior of the strip/soil interface is defined by a nonlinear shear failure envelope that varies as a function of confining pressure. The maximum shear force \( F_{s}^{\text{max}} \) is determined from the following equations:

\[
\frac{F_{s}^{\text{max}}}{L} = S_{\text{bond}} \quad \text{if } \sigma_{c}' < 0 \quad (1.46)
\]

\[
\frac{F_{s}^{\text{max}}}{L} = S_{\text{bond}} + \sigma_{c}' \times f^\ast \times \text{perimeter} \quad \text{if } \sigma_{c}' \geq 0 \quad (1.47)
\]

where:

\[
f^\ast = f_{0}^\ast - (f_{0}^\ast - f_{1}^\ast) \times \frac{\sigma_{c}'}{\sigma_{c0}'} \quad \text{if } 0 \leq \sigma_{c}' < \sigma_{c0}' \quad (1.48)
\]

\[
f^\ast = f_{1}^\ast \quad \text{if } \sigma_{c}' \geq \sigma_{c0}' \quad (1.49)
\]
and:

\[ L \quad = \quad \text{strip element length;} \]
\[ S_{\text{bond}} \quad = \quad \text{strip/interface cohesion;} \]
\[ \sigma_c' \quad = \quad \text{effective confining stress normal to the strip;} \]
\[ \text{perimeter} \quad = \quad \text{perimeter of strip;} \]
\[ f_0^* \quad = \quad \text{initial apparent friction coefficient;} \]
\[ f_1^* \quad = \quad \text{minimum apparent friction coefficient;} \]
\[ \sigma_{c0}' \quad = \quad \text{transition confining pressure.} \]

The effective confining pressure acting normal to the flat strip is

\[ \sigma_c' = -\sigma_{nn} - p \quad (1.50) \]

where: \( p \quad = \quad \text{pore pressure;} \)

\[ \sigma_{nn} = \sigma_{xx} n_1^2 + \sigma_{yy} n_2^2 + 2 \sigma_{xy} n_1 n_2; \quad \text{and} \]
\[ n_i = \text{unit vector normal to the strip.} \]

6. Softening of the strip/interface strength as a function of shear displacement for the interface cohesion and apparent friction can be prescribed via user-defined tables, \texttt{strsctable} (for cohesion) and \texttt{strsftable} (for apparent friction).

Note that forces calculated for strip elements are “scaled” forces (i.e., they are forces per unit model thickness out-of-plane). Actual forces in a strip can be derived from the scaled forces, the calculation width, \texttt{calwidth}, and the number of strips per width, \texttt{nstrips}.

1.7.2 Strip-Element Properties

The strip elements used in FLAC require the following input properties (strip property keywords are shown in parentheses):

1. calculation width (\texttt{calwidth}) [length];
2. density of the strip (\texttt{density}) [mass/volume] (optional – used for dynamic analysis and gravity loading);
3. elastic modulus (\texttt{e}) [stress] of the strip;
4. initial apparent friction coefficient at the strip/interface (\( f_0^* \)) (\texttt{fstar0});
5. minimum apparent friction coefficient at the strip/interface (\( f_1^* \)) (\texttt{fstar1});
6. number of strips per calculation width (\texttt{nstrips});
7. transition confining pressure (\( \sigma_{c0}' \)) (\texttt{sigc0}) [stress];
(8) strip/interface shear stiffness \((\text{strkbond})\) [force/strip length/displ.];

(9) strip/interface cohesion \((\text{strsbond})\) [force/strip length];

(10) number of table relating strip/interface cohesion to plastic relative shear displacement \((\text{strsctable})\);

(11) number of table relating strip/interface apparent friction angle to plastic relative shear displacement \((\text{strsftable})\);

(12) strip thickness \((\text{strthickness})\) [length];

(13) strip width \((\text{strwidth})\) [length];

(14) strip compressive yield-force limit \((\text{strycomp})\) [force];

(15) strip tensile yield-force limit \((\text{stryield})\) [force]; and

(16) tensile failure strain limit of strip \((\text{tfstrain})\).

The perimeter of a strip element is calculated from the strip width \((\text{strwidth})\), the number of strips \((\text{nstrips})\), and the calculation width \((\text{calwidth})\):

\[
\text{perimeter} = 2 \times \frac{\text{strwidth} \times \text{nstrips}}{\text{calwidth}} \quad (1.51)
\]

The cohesion, \(S_{\text{bond}}\), at the strip/interface is calculated from the cohesion of the individual strip \((\text{strbond})\), the number of strips \((\text{nstrips})\), and the calculation width \((\text{calwidth})\):

\[
S_{\text{bond}} = \frac{\text{strbond} \times \text{nstrips}}{\text{calwidth}} \quad (1.52)
\]

1.7.3 Commands Associated with Strip Elements

All of the commands associated with strip elements are listed in Table 1.7. This includes the commands associated with the generation of rockbolts, and those required to monitor histories, and plot and print rockbolt-element variables. See Section 1.3 in the Command Reference for a detailed explanation of these commands.
Table 1.7  Commands associated with strip elements

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip</td>
<td>keyword</td>
</tr>
<tr>
<td>begin</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>keyword</td>
</tr>
<tr>
<td>grid</td>
<td>i j</td>
</tr>
<tr>
<td>node</td>
<td>n</td>
</tr>
<tr>
<td>x y</td>
<td></td>
</tr>
<tr>
<td>prop</td>
<td>np</td>
</tr>
<tr>
<td>segment</td>
<td>ns</td>
</tr>
<tr>
<td>delete</td>
<td>&lt;n1 n2&gt;</td>
</tr>
</tbody>
</table>

| node n | x y |
| node n*| x y |
| node n | x y |
|        |     |

| prop np | keyword |
| calwidth| value   |
| density | value   |
| e       | value   |
| fstar0  | value   |

* For the keywords **fix**, **free**, **initial**, **load** and **pin**, a range of nodes can be specified with the phrase **range n1 n2**.
### Table 1.7  Commands associated with strip elements (continued)

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>prop</th>
<th>np</th>
<th>keyword</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>fstar1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nstrips</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sigc0</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strkbond</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strsbond</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strctable</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strfstable</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strthickness</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strwidth</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>strycomp</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>stryield</td>
<td>value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tfstrain</td>
<td>value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HISTORY</th>
<th>keyword</th>
<th>np</th>
<th>range</th>
<th>nel1 nel2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>node</td>
<td>n</td>
<td></td>
<td>nel1 nel2</td>
</tr>
<tr>
<td></td>
<td>sbond</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sdisp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sforce</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xaccel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xdisp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xvel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>yaccel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ydisp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>yvel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>axial</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Table 1.7  Commands associated with strip elements (continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLOT</strong> strip structure</td>
</tr>
<tr>
<td>&lt;strip&gt; keyword</td>
</tr>
<tr>
<td>axial &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>cs_sdisp &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>cs_sforce &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>element load location &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>material node number sbond sdisp strain &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>svel xdisp &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>xvel &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>ydisp &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td>yvel &lt;ng &lt;ng2&gt; &gt;</td>
</tr>
<tr>
<td><strong>PRINT</strong> structure</td>
</tr>
<tr>
<td>keyword strip node property strip</td>
</tr>
</tbody>
</table>

* A range of group ID numbers can be specified for plotting by giving a beginning number *ng* and an ending number *ng2*. All groups within this range will be plotted.
1.7.4 Example Applications

Simple examples are provided to illustrate the behavior of strip elements in FLAC. Note that combined damping (SET st_damp combined), rather than local damping, is used for these examples (see Section 1.9.3 for a discussion on damping mode).

1.7.4.1 Strip Pullout Test – No Confinement

In this test, the effect of confining stress is neglected. A single cohesive strip is pulled at a small, constant velocity, as indicated in Figure 1.77. A FISH function, ff, is used to sum the reaction forces and monitor the nodal displacement generated during the test. The file in Example 1.27 lists the FLAC commands for this test. Three models are evaluated.

In the first case, shear failure at the strip/interface is simulated as a function of the strip/interface cohesion. The peak force is approximately 48 kN. The force versus displacement history is shown in Figure 1.78.

In the second case, displacement weakening of the strip/interface cohesion is introduced using the strstetable property. The results are shown in Figure 1.79.

In the third case, tensile rupture of the strip is simulated. tfstrain is used to define the limiting plastic tensile strain. A low tensile yield force limit (stryield) for the element and high strip/interface cohesion (strsbond) are specified to produce tensile failure in the strip during the pull test. The results are shown in Figure 1.80.

Example 1.27 Strip pullout test – no confinement

```
grid 6,4
gen (0.0,0.0) (0.0,0.4) (0.6,0.4) (0.6,0.0) ratio 1.0,1.0 i=1,7 j=1,5
model elastic
group ‘elastic‘ notnull
model elastic notnull group ‘elastic‘
prop density=2000.0 bulk=5E9 shear=3E9 notnull group ‘elastic‘
fix x i 1
set large
set st_damping struct=combined 0.8
struct node 1 -0.1,0.2
struct node 2 0.5,0.2
struct strip begin node 1 end node 2 seg 12 prop 7001
struct prop 7001
struct prop 7001 e 2E11 calwidth 1.0 nstrips 1.0 strwidth 0.04 &
strthickness 0.0125 stryield 225000.0 &
strkbond 2.0E7 strsbond 100000.0
def ff
    sum = 0.0
    loop jj (1,jgp)
        sum = sum + xforce(1,jj)
```

FLAC Version 6.0
endloop
 ff = sum
 dd = step * 1e-6
end
ff
history nstep 100
history 2 ff
history 3 dd
struct node 1 fix x initial xvel=-1.0E-6
save stripl_0.sav
;*** constant strength ****
history 999 unbalanced
cycle 20000
save stripl_1.sav
;*** displ. weakening ****
restore stripl_0.sav
struct prop 7001 strctable 100
table 100 delete
table 100 0 100000 0.009999 10000
history 999 unbalanced
cycle 20000
save stripl_2.sav
;*** tensile rupture ****
restore stripl_0.sav
struct prop 7001 stryield 100000.0 strsbond 1.0E7 tfstrain 0.08
struct node 1 initial xvel=-5.0E-7
history 999 unbalanced
cycle 50000
save stripl_3.sav
Figure 1.77  Strip element in grid: x-velocity applied at end node

Figure 1.78  Strip pull force versus axial displacement – strip/interface shear failure
Figure 1.79 Strip pull force versus axial displacement – displacement weakening

Figure 1.80 Strip pull force versus axial displacement – tensile rupture
1.7.4.2 Strip Pullout Test – with Confinement

The effect of confining stress is evaluated in this test. A single horizontal strip is placed within a grid, as shown in Figure 1.81. The strip/interface has an initial apparent friction coefficient of 1.5, and a minimum apparent friction coefficient of 0.727. The grid is fixed in the $x$- and $y$-directions at the base, and in the $x$-direction along the sides. A uniform, vertical confining pressure of 80 kPa is applied to the top of the model. After the model is brought to equilibrium for the specified confining stress, the strip is pulled in the negative $x$-direction by applying a small constant velocity to the left-end node of the strip. The data file is listed in Example 1.28.

The axial force in the left-end segment of the strip is monitored and plotted versus the relative $x$-displacement of the left-end node. Figure 1.82 shows the results. Note that, for this case, the transition confining pressure ($\sigma_{c0}$) is 120 kPa.

For comparison, a second case is run with the transition confining pressure set to 70 kPa. The resulting axial force/displacement plot is shown in Figure 1.83. In this case, a lower peak force is calculated than in the first case.

**Example 1.28 Strip pullout test – with confinement**

```plaintext
grid 22,12
gen (0.0,0.0) (0.0,0.4) (1.0,0.4) (1.0,0.0) i=1,23 j=1,13
model elastic
prop density=2000.0 bulk=1.0E8 shear=3.0E7
fix x y j 1
fix x i 23
fix x i 1
apply pressure 80000.0 from 1,13 to 23,13
struct node 1 0.0,0.2
struct node 2 1.0,0.2
struct strip begin node 1 end node 2 seg 22 prop 7001
struct prop 7001
struct prop 7001 e 2.1E11 calwidth 1.0 nstrips 1.0 strwidth 0.05 &
strthickness 0.0040 styyield 52000.0 strycomp 52000.0 strkbond 1.0E9 &
frstar 1.5 fstar1 0.727 sigc0 120e3
history 999 unbalanced
solve
save strip_0.sav
;*** conf. press 120 kPa ****
struct node 1 fix x initial xvel=-1.0E-8 yvel=0.0
history 1 element 1 axial
history 2 node 1 xdisplace
set st_damp struc combined
cycle 20000
save strip_1.sav
;*** conf. press 70 kPa ****
```

FLAC Version 6.0
restore strip_0.sav
struct node 1 fix x initial xvel=-1.0E-8 yvel=0.0
history 1 element 1 axial
history 2 node 1 xdisplace
struct prop 7001 sigc0 70000.0
set st_damp struct combined
cycle 20000
save strip_2.sav

Figure 1.81 Strip element in grid: vertical confining pressure and x-velocity applied at end node
Figure 1.82 Strip axial force versus axial displacement – $\sigma_{c0} = 120$ kPa

Figure 1.83 Strip axial force versus axial displacement – $\sigma_{c0} = 70$ kPa
1.8 Support Members

1.8.1 Formulation

The formulation for support members is not incremental. Rather, the force in the support member is related to the total displacement of the member in the tangential (axial) and normal (transverse) directions accumulated since its creation. When a support member has non-zero width (i.e., it is divided into sub-members), the force in each sub-member is computed in one of two ways:

\[ F = \frac{k_t}{n+1} \]

or

\[ F = \frac{f(u)}{n+1} \]  

where:
- \( n \) = number of sub-members;
- \( u \) = displacement of a sub-member;
- \( f(u) \) = table look-up function; and
- \( k_t \) = tangential (axial) stiffness.

The total force exerted by the member is the sum of the sub-member forces.

1.8.2 Support-Member Properties

The support elements in FLAC require the following input parameters (support property keywords are shown in parentheses):

1. axial stiffness (\( k_n \)) of the support member (force/displacement);
2. compressive yield strength (\( y_{prop} \)) (force) of the support member;
3. spacing (\( \text{spacing} \)) [length] (optional – if not specified, supports are considered to be continuous in the out-of-plane direction).

If the support member contains sub-elements, then the axial stiffness and yield strength are for the group of sub-members.

Alternatively, the relation between axial force and axial displacement can be specified by a look-up table. However, a table should not be used if the support is subjected to unloading. Also, if a look-up table is specified, spacing does not apply.
1.8.3 Commands Associated with Support Elements

All the commands associated with support elements are listed in Table 1.8, below. See Section 1 in the Command Reference for a detailed explanation of these commands.

<table>
<thead>
<tr>
<th>STRUCTURE</th>
<th>keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>support</td>
<td>keyword</td>
</tr>
<tr>
<td></td>
<td>x y</td>
</tr>
<tr>
<td></td>
<td>angle</td>
</tr>
<tr>
<td></td>
<td>delete</td>
</tr>
<tr>
<td></td>
<td>prop np</td>
</tr>
<tr>
<td></td>
<td>remove</td>
</tr>
<tr>
<td></td>
<td>segment</td>
</tr>
<tr>
<td></td>
<td>width</td>
</tr>
<tr>
<td>prop np</td>
<td>keyword</td>
</tr>
<tr>
<td></td>
<td>kn</td>
</tr>
<tr>
<td></td>
<td>kn table</td>
</tr>
<tr>
<td></td>
<td>spacing</td>
</tr>
<tr>
<td></td>
<td>yprop</td>
</tr>
<tr>
<td>PLOT</td>
<td>support</td>
</tr>
<tr>
<td>PRINT</td>
<td>structure support</td>
</tr>
</tbody>
</table>
1.8.4 Example Application

Support elements are commonly used to simulate props in an underground excavation.

1.8.4.1 Support of Faulted Ground

This example problem illustrates the use of support members in faulted ground. Example 1.29 contains the commands for this model.

**Example 1.29 Support of faulted ground**

```plaintext
grid 9 6
model mohr
prop d 1000 s .5e8 b 1.5e8 coh 2.5e5
; create interfaces
model null j 2
model null i 5 j 3,6
gen 0,0 0,1 8,1 8,0 j 1,2 ; bottom layer
gen 0,3 0,7 4,7 4,3 j 3,7 i 1,5 ; l.h. top
gen 4,3 4,7 8,7 8,3 j 3,7 i 6,10 ; r.h. top
int 1 aside from 6,3 to 6,7 bside from 5,3 to 5,7
int 1 ks 1e8 kn 1e8 fric 5
int 2 a from 1,2 to 5,2 b from 5,3 to 1,3
int 3 a from 6,2 to 10,2 b from 10,3 to 6,3
int 2 ks 1e8 kn 1e8 fric 5
int 3 ks 1e8 kn 1e8 fric 5
; boundary conditions
fix x y j 1
fix x i 1
fix x i 10
fix x y j 7
; initial compressive stress to load fault
ini sxx -1e5 j 3,6
set large
; specify 20 supports centered at x=4.0 y=2.0 with width=3.0
struc supp 4,2 wid 3.0 seg 20 prop 6001
; specify force displacement relation for support
struct prop 6001 kn table 5
table 5 0,0 0,2,0 0,4,0.4e7 10,0,0.4e7
; displace top of model downward
ini yv -2e-3 i 6 10 j 7
step 500
plot hold bou supp
```

FLAC Version 6.0
Figure 1.84 shows the location of the vertical fault and support members. The support members behave according to the force-displacement relation shown in Figure 1.85. The specified support “yields” at 4 MN, as shown.

Figure 1.86 shows that the deformed position of the supports after the upper surface on the right side of the problem has displaced downward 1 m.

*Figure 1.84 Support members before loading*
Figure 1.85  Force-displacement relation specified for support in example problem

Figure 1.86  Support members after loading
1.9 Modeling Considerations

1.9.1 Limitations

The present formulation for structural elements has the following limitations:

1. The structural element formulation is a plane-stress formulation. If the beam elements or pile elements are representing a structure that is continuous in the direction perpendicular to the plane of analysis, then the elastic modulus of the element, $E$, should be divided by $(1 - \nu^2)$ to account for plane-strain conditions.

2. None of the formulations will work when the problem is configured for axisymmetry (CONFIG axi).

3. The tensile and compressive yield criterion for beams is based on axial thrust, and for liners on axial thrust and bending stresses. Pile elements do not include a tensile or compressive yield criterion. The yield criterion for beams and liners does not consider shear failure.

4. Beam, pile and rockbolt elements allow specification of a maximum moment by the pmom (plastic moment) property, and a plastic hinge can develop at nodes that have been previously identified as potential plastic hinge nodes, with the STRUCTURE hinge command. The limiting plastic moment is calculated automatically from the yield criterion for liner elements.

5. Each structural element is assumed to have constant cross-sectional area and properties.

1.9.2 Symmetry Conditions

Beam, liner, cable, pile or rockbolt element nodes that lie on a line of symmetry should be assigned full properties for modulus and grout or coupling-spring stiffness. Property values for cross-sectional area, yield strength, and grout or coupling spring cohesive strength should be reduced by 50% compared to the same property values for elements not on the symmetry line. Loads applied to structural elements on symmetry lines should also be reduced by 50% compared to the same loads applied away from symmetry lines. As mentioned in Section 1.1.5, beam, liner, pile and rockbolt elements that terminate at a line of symmetry should have their nodal rotations fixed at the symmetry line.
1.9.3 Equilibrium Conditions

As explained in Section 2.6.4 in the User’s Guide, the user must decide when the model has reached equilibrium. Equilibrium for problems involving structural elements can be determined by all the usual criteria (e.g., histories, velocity fields). However, if beam, liner, pile or rockbolt elements are used, an additional equilibrium criterion is available. At equilibrium, beam, liner, pile or rockbolt element segments that share a common node will have equal and opposite moments. This can be confirmed with the PRINT struct beam, PRINT struct liner, PRINT struct pile or PRINT struct rockbolt command.

For certain types of structural-element problems (e.g., axial loading of piles or pull-tests on cables) a significant portion of the model region may develop non-zero components of velocity at the final state of solution. The default mechanical damping algorithm in FLAC can have difficulty damping this motion properly, because the mass-adjustment process requires velocity sign-changes (see Section 1.3.4 in Theory and Background). An alternative form of damping is available for this type of problem. This damping, known as combined damping or “creep-type” damping, is also described in Section 1.3.4 in Theory and Background. Combined damping is invoked for the FLAC grid with the command SET st_damp combined, and for the structural elements with the command SET st_damp struct combined. See Sections 1.5.4 and 1.6.4 and Section 9 in the Examples volume for example problems in which combined damping should be used.

1.9.4 2D/3D Equivalence – Property Scaling

Reducing 3D problems (with regularly spaced beams, liners, cables, piles, rockbolts or supports) to 2D problems involves averaging the effect in 3D over the distance between the elements. Donovan et al. (1984) suggest that linear scaling of material properties is a simple and convenient way of distributing the discrete effect of elements over the distance between elements in a regularly spaced pattern.

The relation between actual properties and scaled properties can be demonstrated by considering the strength properties for regularly spaced piles. The actual maximum normal force per length of the pile is defined by Eq. (1.37). Internally, FLAC uses the expression

\[
\frac{(F_{n}^{\text{max}})^{s}}{L} = (c_{n\text{coh}})^{s} + p' \times \tan(c_{n\text{fric}}) \times \text{(perimeter)}^{s}
\]  

where \((F_{n}^{\text{max}})^{s}\) is the (scaled) maximum normal force per unit model thickness calculated by FLAC. (The superscript \(s\) does not denote a power.) We want the total force calculated by FLAC over a spacing, \(S\), to be the same as the actual force. The actual maximum normal force is then

\[
F_{n}^{\text{max}} = (F_{n}^{\text{max}})^{s} \times S
\]

and the actual normal force is
The relation between the actual force and the FLAC force can be satisfied by substituting Eq. (1.55) and the following relations into Eq. (1.54):

\[
(F_n)^s = (F_n) \times S 
\]

\[
(c_{s_{ncoh}})^s = \frac{c_{s_{ncoh}}}{S} 
\]

\[
(\text{perimeter})^s = \frac{\text{perimeter}}{S} 
\]

The actual normal stress on the pile, \(\sigma_n\), is calculated by dividing the actual force by the actual effective area (perimeter \(\times L\)):

\[
\sigma_n = \frac{(F_n)^s \times S}{\text{perimeter} \times L} 
\]

Note that the choice to scale perimeter is arbitrary, because only the product \(\tan(c_{s_{nfric}}) \times \text{perimeter}\) is relevant. Alternatively, the friction term could be scaled.

It is important to remember that the forces (and moments) for structural elements that are calculated by FLAC are scaled forces (and moments). The actual forces and moments can be calculated by multiplying the FLAC forces and moments by \(S\). FISH access to FLAC values for forces and moments access scaled values, and thus should be multiplied by the appropriate spacing value to determine the actual values.

The \text{spacing} property is provided with beams, liners, cables, piles, rockbolts and supports to scale properties, and account for a spaced pattern of these structural elements.* When \text{spacing} is specified in the \text{STRUCT prop} command, the actual properties of the structural elements are input. The scaled properties are then calculated automatically by dividing the actual properties by the spacing, \(S\). When the calculation is complete, the actual forces and moments in the spaced structural elements are then determined automatically (by multiplying by the spacing) for presentation in output results (i.e., using the \text{PRINT} or \text{PLOT} command).†

The following lists summarize the structural element properties that are scaled when \text{spacing} is specified to simulate regularly spaced structural elements.

* The spacing for strip elements is directly accounted for with the \text{calwidth} and \text{nstrips} properties.

† Note that this is a change from previous versions of FLAC, in which only the scaled values are printed or plotted.
For beam elements, the following properties are scaled:

1. elastic modulus;
2. plastic moment;
3. tensile yield strength;
4. residual tensile yield strength; and
5. compressive yield strength.

For liner elements, the following properties are scaled:

1. elastic modulus;
2. tensile yield strength;
3. residual tensile yield strength; and
4. compressive yield strength.

For cable elements, the following properties are scaled:

1. elastic modulus of the cable;
2. tensile yield strength of the cable;
3. compressive yield strength of the cable;
4. stiffness of the grout;
5. cohesive strength of the grout; and
6. exposed perimeter of the cable.

For pile elements, the following properties are scaled:

1. elastic modulus of the pile;
2. plastic moment of the pile;
3. stiffness of the shear coupling spring;
4. cohesive strength of the shear coupling spring;
5. stiffness of the normal coupling spring;
6. cohesive (and tensile) strength of the normal coupling spring; and
7. exposed perimeter of the pile.
Structural Elements

For rockbolt elements, the following properties are scaled:

1. elastic modulus of the rockbolt;
2. plastic moment of the rockbolt;
3. tensile yield strength of the rockbolt;
4. compressive yield strength of the rockbolt;
5. stiffness of the shear coupling spring;
6. cohesive strength of the shear coupling spring;
7. stiffness of the normal coupling spring;
8. cohesive strength of the normal coupling spring; and
9. exposed perimeter of the rockbolt.

For support elements, the following properties are scaled:

1. axial stiffness of the support member; and
2. compressive yield strength of the support member.

In addition to the above properties, the \texttt{spacing} keyword also applies to gravity loads, which are calculated using the true cross-sectional area and the scaled structure density. Also, any pretensioning that is applied to cable elements (i.e., using the \texttt{tension} keyword) is scaled when \texttt{spacing} is given. Note that if loading is applied using the \texttt{STRUCT node n load} command (e.g., pre-loaded struts), these loads are \textit{not} scaled when \texttt{spacing} is provided. The loads should be scaled by dividing by \textit{S}.

Finally, actual stresses within structural elements are obtained from actual forces and moments using the real cross-sectional area and moment of inertia. The \textit{FISH} function “PRSTRUC.FIS” is provided in Section 3 in the \textit{FISH volume} to demonstrate the procedure for calculating actual axial stresses in regularly spaced beams subjected to bending.

The following example illustrates the simulation of regularly spaced structural elements. In this case, vertical piles at an equal spacing of 2 m are subjected to axial loading. The actual elastic modulus of the pile is 10 GPa, and the actual stiffness of the shear coupling spring is 1 GN/m/m. The cohesive strength of the shear coupling spring is set to a high value to prevent shear failure for this simple example. A vertical axial loading of 2 MN is applied at the top of the pile, and the pile spacing is set to 2 m. \texttt{Example 1.30} lists the commands for this example.

Results are shown for both the case in which \texttt{spacing} is specified, and the case in which it is not. In the second case, the input values for elastic modulus and shear coupling spring stiffness are scaled (by dividing by 2). Note that for both cases, the applied vertical load is scaled (\texttt{STRUCT node 1 load 0.0,-1000000.0,0.0}).
Figure 1.87 displays the result for the first case. When spacing is given, the actual axial forces are displayed in the pile axial force plot. Figure 1.88 shows the result for the second case. When spacing is not given, but the input properties are scaled, the axial force plot displays the scaled values for axial force. The axial forces in Figure 1.88 must be multiplied by 2 to obtain the actual values.

**Example 1.30 Axial loading of piles at 2 m spacing**

```
grid 5,5
model elastic
group 'soil' notnull
model elastic notnull group 'soil'
prop density=1000.0 bulk=1E8 shear=3E7 notnull group 'soil'
struct node 1 2.5,5.0
struct node 2 2.5,2.5
struct pile begin node 1 end node 2 seg 3 prop 3001
struct prop 3001
; using spacing keyword
struct prop 3001 e 1e10 cs_sstiff 1e9 cs_scoh 1e20
struct prop 3001 area 1.0 spac 2.0
; without spacing keyword
; struct prop 3001 e 5e9 cs_sstiff 5e8 cs_scoh 1e20
; struct prop 3001 area 1.0
fix x y j 1
fix x i 6
fix x i 1
struct node 1 load 0.0,-1000000.0, 0.0
history 999 unbalanced
set sratio 1e-4
solve
save spacing.sav
```

See Sections 6 and 11 in the Examples volume for additional illustrations using spacing with cables and beams that represent regularly spaced support.
Figure 1.87  Actual axial forces in vertically loaded piles at 2 m spacing (spacing given)

Figure 1.88  Scaled axial forces in vertically loaded piles at 2 m spacing (spacing not given)
1.9.5 Sign Convention

Axial forces in structural elements are positive in compression. Shear forces follow the opposite sign convention as that given for zone shear stress, illustrated in Figure 2.42 in the User’s Guide. Moments at the ends of beam elements are positive in the counterclockwise direction.

Translational displacements at nodes are positive in the direction of the positive coordinate axes, and angular displacements are positive in the counterclockwise direction.

The shear force and shear displacement at a cable/grout interface-spring node, or a pile shear coupling-spring node, are positive if the node displacement is in the direction of the specification of the cable or pile (i.e., begin -> end).

The normal force and normal displacement at a pile normal coupling-spring node are positive if the coupling spring is in compression. See Figure 1.48.

1.9.6 Numerical Stability

The numerical stability of the structural-element solution depends on the structural timestep determined automatically by FLAC. As described previously, in Section 1.3.5 in Theory and Background, structural-element inertial masses are set equal to the effective stiffness connected to the node in the coordinate directions. The stiffness (i.e., inertial masses) in the x-, y- and rotational directions are required for the timestep calculation, as well as for the application of the equations of motion to the structural masses. The stiffness is found by a unit displacement method by alternately fixing \( u_{a} \) and \( u_{b} \) (Figure 1.2) and calculating the values of the stiffnesses \( k_{x} \) and \( k_{y} \).

From Eq. (1.5):

\[
F_{i} = F^{t} t_{i} + F^{n} n_{i}
\]

\[
= \left( \frac{EA}{L} u_{j} t_{j} \right) t_{i} + \left( \frac{12EI}{L^{3}} u_{j} n_{j} \right) n_{i}
\]

So, the stiffnesses \( k_{x} \) and \( k_{y} \) are:

\[
k_{x} = \frac{F_{1}}{u_{1}}
\]

\[
= \frac{EA}{L} t_{1}^{2} + \frac{12EI}{L^{3}} n_{1}^{2}
\]
and the stiffness, $k_r$, for rotation is

$$k_r = \left| \frac{M}{\theta} \right| = \frac{4EI}{L}$$

The above values ($k_x$, $k_y$ and $k_r$) are local stiffness values for the beam or liner elements, which are connected to gridpoints of the finite-difference zone. The cable element stiffness, $k^c$, is taken to be equal to the product of $(K_{\text{bond}} \cdot L)$, where $L$ is the length of the cable element. The stiffness for the pile or rockbolt elements will correspond to that for beam or liner elements if the pile or rockbolt node is connected to the grid, or to the stiffness for cable elements if not connected to the grid.

The weighted inertial mass is set equal to the sum of stiffnesses for each node (recall that $\Delta t = 1.0$ – see Section 1.3.5 in \textit{Theory and Background}). For the case of a structural node connected to a gridpoint, the inertial mass becomes:

$$m_x = m^G + 4.0(k_x + k^c)$$
$$m_y = m^G + 4.0(k_y + k^c)$$

and

$$m_r = 4.0 \cdot (k_r)$$

where: $m_{x,y}$ = inertial mass due to translational stiffness;

$m_r$ = inertial mass due to rotational stiffness; and

$m^G$ = gridpoint mass.

The multiplier of 4.0 is described in Section 1.3.5 in \textit{Theory and Background}.

If a structural node is not connected to the grid, then:

$$m_x = k_x$$
$$m_y = k_y$$

$$m_r = k_r$$

(1.62)

(1.63)

(1.64)

(1.65)
where $k_x, k_y$ are the sums of the local translational stiffness contributions, and $k_r$ is the sum of the local rotational stiffness contributions from each connected beam. These masses are now used in the motion equation for the beam, liner, cable, pile and rockbolt element nodes.

For beam, liner, cable, pile and rockbolt elements, experience with FLAC has shown that the structural-element formulation is stable and converges to the steady-state solution for nearly all cases. Numerical instability has been observed in the case of an end-loaded column with a lower pin joint subjected to a velocity load on the free end. This is equivalent to Euler buckling under dynamic loading.

For support members, the stiffness is not taken into account in the consideration of numerical stability, mainly because it is difficult to estimate the stiffness in advance for table look-ups. If the support is stiffer than the rock, this may lead to numerical instability, but the reverse is likely to be the case in most problems.
1.10  *FISH* and Structural Elements

1.10.1 Introduction

*FISH* functions have access to the linked-list data structure for structural elements. The typical application is to read and manipulate data from the lists to provide histories or controls. This discussion covers all the structural elements, although individual data entries may be pertinent to specific structures.

Access is provided via pointers stored as *FISH* scalar variables. The variables are provided symbolic names in files that have the extension “.FIN” (for *FISH* Include) – see Section 4 in the *FISH* volume. The “FIN” file for structural elements is “STR.FIN” (in the “\FISH\4-ProgramGuide” sub-directory, which also contains documentation for the meaning of each variable). This file should be called from a data file so that the pointers can be identified by name in the data file. **str.pnt** is the name of the pointer to the control block containing a list of pointers to individual areas within the structural elements. Figure 1.89 shows the linkage:

![Figure 1.89 The linkages](image)

The structural element lists are broken down into:

1. node data;
2. element data;
3. property data;
4. master/slave data; and
5. support element data.
The property data list can be accessed directly from the node element and support lists, to obtain properties for that particular entity, or the property list can be accessed from the control block.

1.10.2 The Address

The address of a particular entity is a computer index for the location in RAM containing the specific item of data. Identification (ID) numbers are assigned for nodes, element segments, etc. However, these numbers are different from addresses. An ID number is one of the items of data stored in the list. It is necessary to first obtain the correct address for a particular entity (for example, the address of node number 13) before other data about node 13 can be extracted.

1.10.3 Obtaining and Using Addresses

Example 1.31 shows a \textit{FISH} function, \texttt{get\_node\_addr}, that allows the user to obtain the address for any given structural element node.

\textbf{Example 1.31} \textit{FISH} function used to obtain the address of a given structural node

```
call str.fin
def get_node_addr
    ip=imem(str_pnt + $ksnode) ; top of node list
    loop while ip #0
        id_num = imem(ip + $kndid) ; id number of the node
        if id_num = node_num then
            get_node_addr = ip
            exit
        endif
    ip=imem(ip)
endloop
end
```

This \textit{FISH} function returns the address \texttt{get\_node\_addr} for the specified node number \texttt{node\_num}. The address is then used to collect a specific item of data associated with that structural node.

For example, assume that we want to monitor the \textit{y}-reaction force that develops in pile node number 1 when a velocity is applied and fixed for that node. The \textit{FISH} function from Example 1.31 is named \texttt{node\_addr\_fis} and then called into a data file and applied, as shown in Example 1.32, to access the reaction force and store it as a history.
Example 1.32 Using `get_node_addr` to monitor histories

```
grid 10 10
gen 0 0 0 10 10 10 0
model mohr
prop dens 1620 sh=5.77e6 bu=1.25e7 fric 32
stru pile beg 0,10 end 0,5 seg 10 pro 3001
stru pro 3001 e 4e9 rad 0.1524 per 0.976 cs_scoh 1.35e5 cs_sstiff 1e9
fix x i=1
fix y j=1
fix x i=11
set st_damp struc combined
set grav 10
ini syy -1.62e5 var 0 1.62e5
ini sxx -1.10e5 var 0 1.10e5
ini szz -1.10e5 var 0 1.10e5
call node_addr.fis
def y_reaction
  y_reaction = fmem(n2addr + $kndf2c)
end
set node_num=1 n2addr = get_node_addr
hist y_reaction
stru node 1 fix y
stru node 1 ini yvel -1e-7
step 10000
save monitor.sav
plot hold hist 1
```

1.10.4 FISH-Controlled Force-Displacement Relations

_FISH_ can be used to implement user-defined force-displacement relations for the normal coupling springs of pile elements. A _FISH_ function can be accessed directly via the pile property keyword _cs_nfunc_, as described previously in Section 1.5.2. The argument passed to the _FISH_ function is a pointer to the pile node. This argument is communicated to the function by using the special function _fc_arg_ (see Section 2.5.5 in the _FISH volume_). The user can then access all variables related to the node, including coupling-spring variables, by using “STR.FIN.” Given the relative normal displacement (at offset _$kndua_ or the relative normal displacement increment (at offset _$kndunr_), a _FISH_ function can be written to calculate the force in the spring, which is then stored at offset _$kndfn_. To calculate the force in the spring, the normal force per unit pile length must be multiplied by the effective length (offset _$kndefl_).

To demonstrate the application of this feature, Example 1.33 presents a _FISH_ function that calculates the normal coupling-spring behavior as a function of normal stiffness and normal cohesion.
Example 1.33  *FISH function p_y to define the behavior of a normal coupling spring*

```frr
def p_y
    ; pointer to node structure
    ipn = fc_arg(1)
    fndis = fmem(ipn+$kndua) ;relative displ.
    dunrel = fmem(ipn+$kndunr) ;relative displ. increment
    ; Calculate normal force
    it = imem(ipn+$kndtad) ;pointer to property list
    stiff = nor_stiff ;KN
    eflen = fmem(ipn+$kndefl) ;effective length
    fn = fmem(ipn+$kndfn) + stiff * eflen * dunrel
    ; Check for yield
    fnmax = nor_cohes * eflen
    if abs(fn) > fnmax then
        fn = sgn(fn) * fnmax
    end_if
    ; Store normal force
    fmem(ipn+$kndfn) = fn
end
```

This function is associated with a particular property number when the pile-element property command `cs_nfunc` is given. For this example, the normal spring stiffness and cohesion can be specified with the `SET` command. The `p_y` function is implemented, for example, with the following commands:

```frr
struct prop 1 cs_nfunc = p_y
set nor_stiff = 1e9  nor_cohes = 1e4
```

`p_y` will then be used, instead of the built-in force-displacement relation, to update the normal coupling spring forces at every pile node of property number 1 while cycling.
1.11 References


