An interactive approach to teaching computational physics

Thomas Vojta

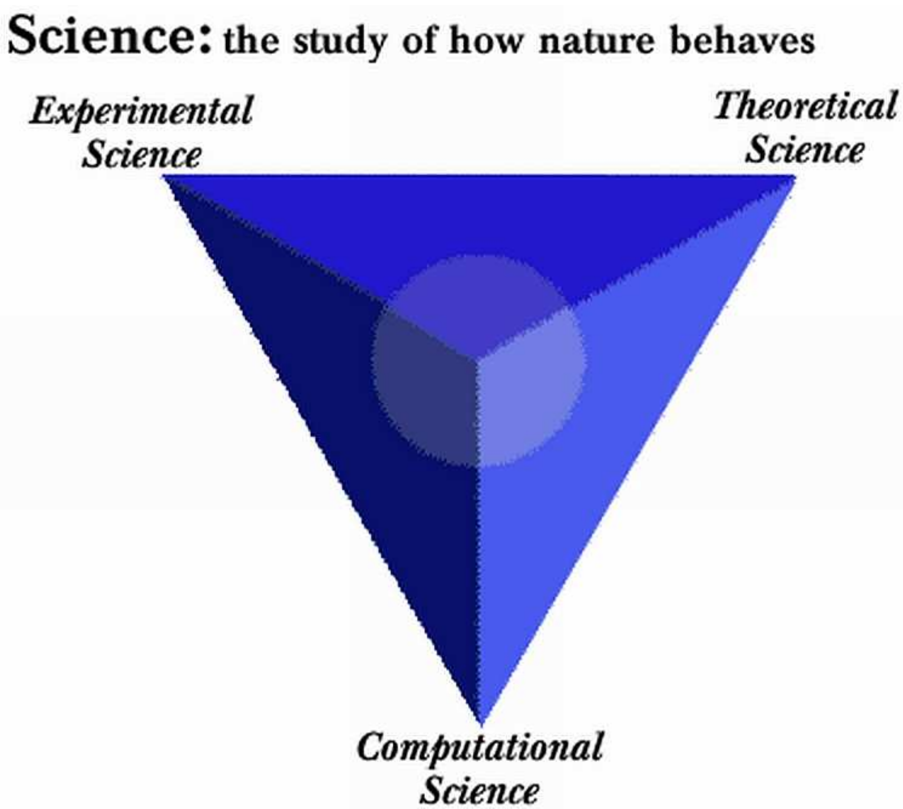
Department of Physics, University of Missouri-Rolla

UMR

- Computational physics education: Why and how?
- Project-based computational physics courses at UMR
- Experiences, challenges, and evaluations

Computational Science

Application of computational and numerical techniques to solve large and complex science problems



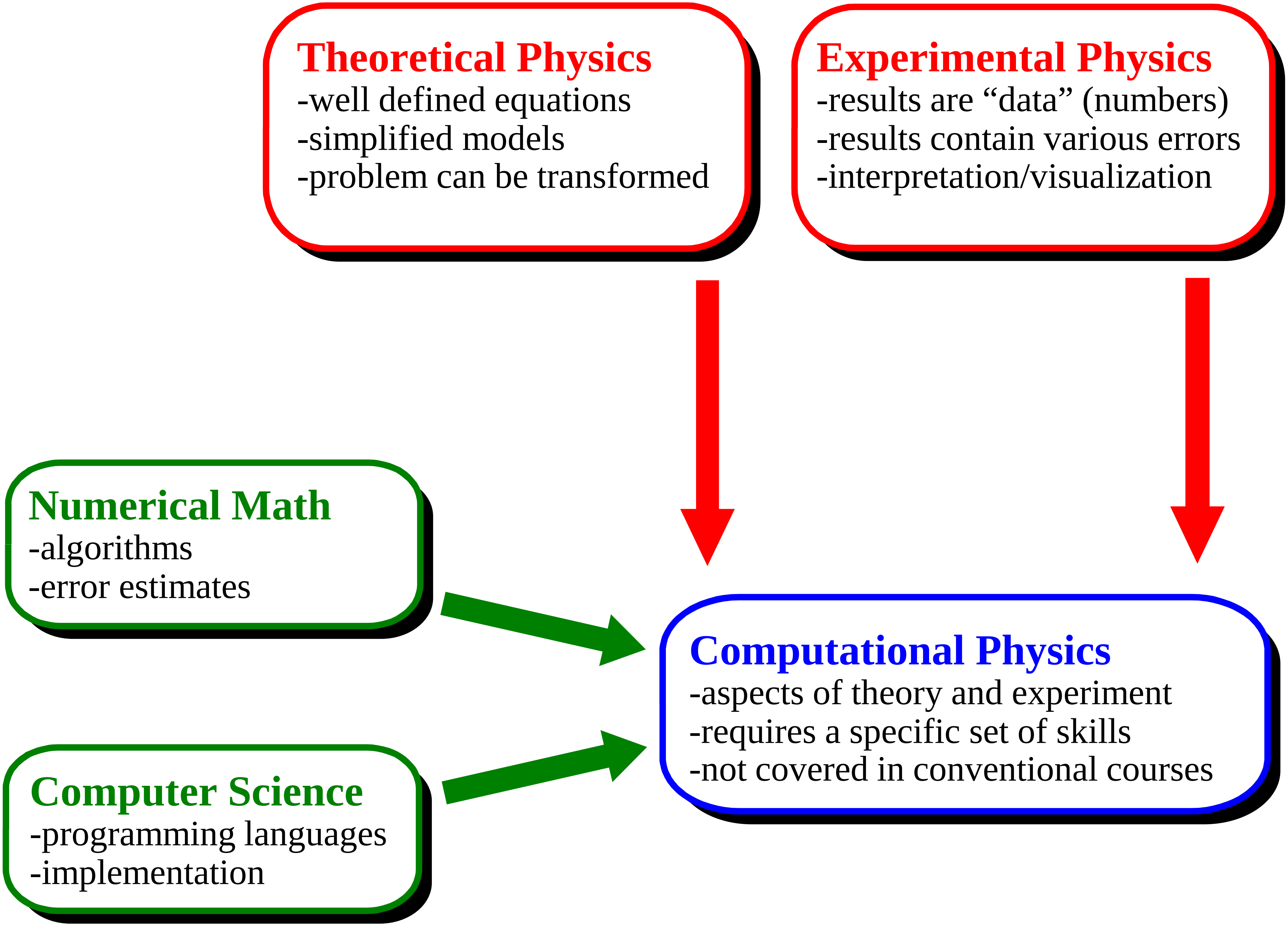
3rd independent scientific methodology

has arisen over the last 20 years or so

shares characteristics with both theory and experiment

requires interdisciplinary skills in science, mathematics, computer science

Computational Science $\neq$ Computer Science


Theoretical Physics
-well defined equations
-simplified models
-problem can be transformed
Experimental Physics
-results are “data” (numbers)
-results contain various errors
-interpretation/visualization
Numerical Math
-algorithms
-error estimates
Computer Science
-programming languages
-implementation
Computational Physics
-aspects of theory and experiment
-requires a specific set of skills
-not covered in conventional courses

Goals of teaching computational physics

What do we want to achieve?

Students learn how to **solve a physics problem** using computer simulations.

This includes diverse tasks:

- formulating the physics problem in a way suitable for simulations
- choosing an efficient computational algorithm
- writing and testing computer code
- running the simulations and collecting numerical data
- analyzing and visualizing the data obtained
- extracting the solution of the physics problem

How to teach computational physics?

"Traditional" sources of computational physics skills:

- **Regular physics courses**
  Problem: many students do not have background in computation, computer use restricted mostly to computer demos
- **Computer science and applied mathematics courses**
  Problem: often focus on formal and technical aspects of computation, do not address how to use it for real-world science problems

Clear need for Computational Physics courses that:

- focus on the science aspects of computational physics
- give students hands-on experience in designing and running computer simulations
- play a role similar to that of laboratory courses for experiment

- Computational physics education: Why and how?
- **Project-based computational physics courses at UMR**
- Experiences, challenges, and evaluations

Project-based computational physics course

- aimed at upper-level undergraduates and beginning graduate students
- intended **not only for physics students** but also students from the other sciences and engineering
- prerequisites: Modern physics, Differential equations, Programming class
- course is **project-based**

In each project, students:

- start from a specific self-contained physics problem
- design a computer simulation
- write and debug computer code
- run the simulation and collect numerical data
- analyze and visualize these data
- write a project report on their solution of the physics problem

Structure of the course

Lecture component

3 one-hour classes per week

Lectures cover:
- physics background of the projects
- introduction to numerical algorithms
- basics of data analysis

follows Tao Pang: "An introduction to Computational Physics"

fairly conventional in style

Laboratory component

1 three-hour lab session per week

Students:
- develop computer code
- run simulations
- analyze and visualize data

Instructor:
- guides setup of simulations
- helps with coding and debugging
- provides input for data analysis

What it is not . . .

Course focuses on how to **use computation to solve a physics problem**

It is therefore:

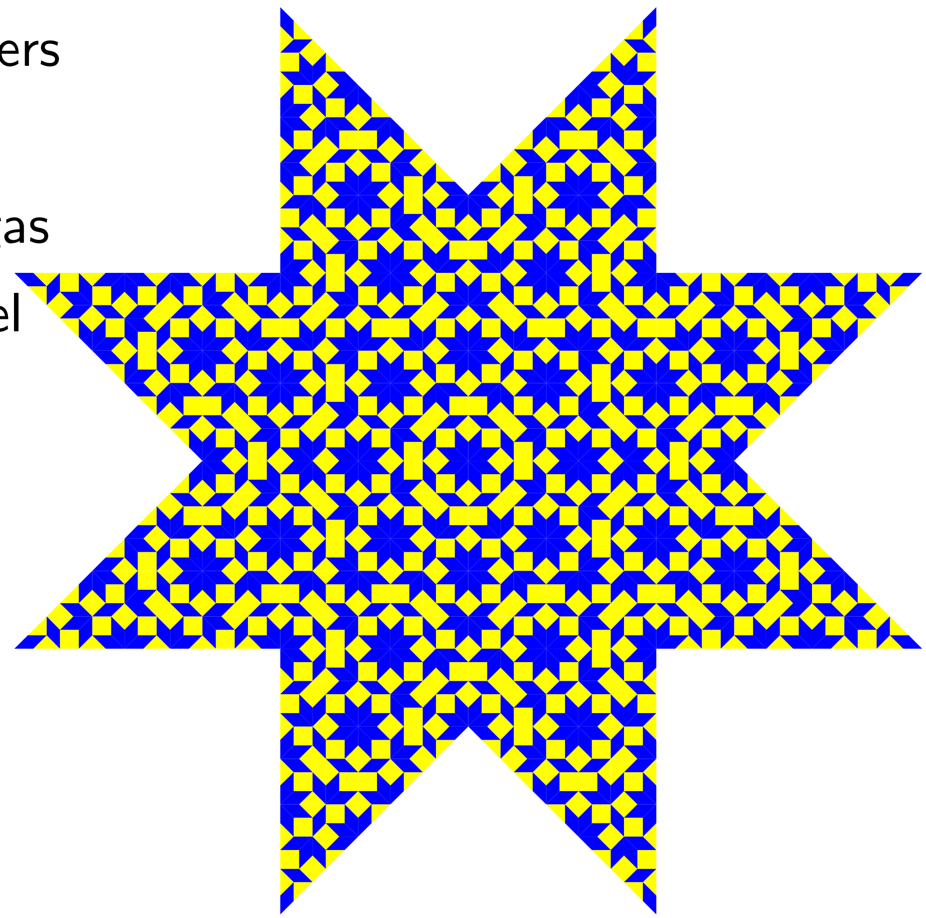
- not a replacement for a conventional physics course;
  physics background discussions in the lectures are not intended to be systematic introductions to the field
- not a course in numerical mathematics;
  algorithms are introduced ad-hoc, not in a mathematical rigorous way
- not a programming course;
  students are expected to know how to program (but programming and debugging help is provided in the lab)

Implementation details

- students develop programs **from scratch**
- no code is provided except for a few subroutines, e.g., an eigenproblem solver
- **FORTRAN 90/95** and **C/C++** are supported in the course
  (i.e., appropriate compilers are installed in the computer classroom, and programming help is available)
- for plotting and visualization the course supports **GNUPlot**
  (students are free to use other plotting programs if they do not need support)

Example projects

- Radioactive decay
- Chaotic dynamics of a damped driven pendulum
- Eigenstates of a particle in a quantum well
- Geometric structure of multi-charge clusters
- Quasiperiodic systems and quasicrystals
- Molecular dynamics of a Lennard-Jones gas
- Monte-Carlo simulation of the Ising model



- Computational physics education: Why and how?
- Project-based computational physics courses at UMR
- **Experiences, challenges, and evaluations**

Experiences and challenges

- so far, taught class three times
  (experimental class in SS2003 and FS2004, regular class since SS2006)
- ratio of undergraduates to graduate students about 1:1
- about 2/3 of the students are from physics

Observations:

- laboratory sessions are the **heart** of the course because students are **actively working** on the material!
- most students will **need help finding programming bugs**, otherwise their projects never reach the data collection and analysis stage
  (very time consuming for instructor - TA may be necessary)
- even the better students have difficulties thinking physically about
  (i) how to **choose parameters** for the simulations
  (ii) how to **plot numerical data** (data ranges, axes, etc.)

Student feedback and evaluation of the class

Student input/feedback

- entrance questionnaire (prerequisites, expectations)
- mid-term discussion on pace, workload, etc
- official teaching evaluations and student comments
- big end-of-term discussion on the class (**project hit list**)

Evaluation results (average over SS2003, FS2004, SS2006)

- Overall: 3.5/4 (UMR average 3.0)
- Educational value: 3.9/4 (UMR average 2.8)
- Assignments: 3.6/4 (UMR average 2.8)

Students like the contents and the project-based character of the course.

From ratemyprofessor.com

- Though the work is long and difficult, I've probably learned more in this class than I have in any other. Dr. Vojta is a great professor, and constantly makes himself available if you need assistance.
- Maybe the most educational course I've taken yet. He is a great instructor. He'll challenge you and if you rise to it he helps you out any way possible. Highest possible recommendation.

Students like to be challenged! If they are motivated, and take charge of their own work, they will succeed in a difficult class.

The future

Expand course into sequence of two courses:

- **Computational Physics I**: similar to present course, maybe at a somewhat lower level physics-wise (no modern physics) to attract broader audience
- **Computational Physics II**: sequel to the first course, aimed more at physics students (undergraduate and graduate) covering research-level methods including parallel and cluster computing