Teaching CS&E
(why it is not enough what we do today)

Ulrich Rüde
Lehrstuhl für Simulation

Department of Computer Science
University of Erlangen

GAMM Workshop CS&E Education
Rostock, 14/15. Nov. 2013
Stipulation (from Max Gunzburger, FSU)

There are two kinds of computational scientists and engineers

1. those that are primarily interested in advancing a science or engineering discipline through computations (instead of theory or experiments)

2. those that want to develop new computational tools for advancing one or more science or engineering disciplines

The first type should be trained in a traditional science or engineering program

I am going to talk about the training of the second kind
What is CS&E

• part of a traditional discipline
  • mechanics?
  • computer science?
  • math?
• a union of the above?
  • an interdisciplinary endeavor, combining what is already existing?
  • it seems that most of our current programs implicitly use this definition
• a separate discipline?
How to organize CS&E?

• within an existing department, possibly „importing“ courses from other departments
  • this is what we do in Erlangen with limited success
  • difficult to give students an „identity“
  • curricular conflicts, permanent changes, internal competition
  • as soon as I stop fighting entropy, it starts loosing structure

• jointly by several departments, e.g. organized as a virtual faculty, DA?

• in a separate, independent unit, e.g. ICES Texas, sitting on an endowment of 120M, but even there, CS&E faculty must find a home in the classical schools
Designing genuine CS&E education

• We must start by asking: what collective knowledge is necessary to answer scientific question from engineering (physics, chemistry, biology, ...) on a computer?
Diagram from Max Gunzburger, FSU

Mathematical modeling

Analysis of mathematical models

Invention and development of new computational algorithms

Analysis of computational algorithms

Implementation of computational algorithms

Computational solution of applications problems

User-friendly presentation of solutions

Diagram from Max Gunzburger, FSU
CSE content

• From math: numerics, modeling, statistics, optimization, ...
• From CS: software engineering, visualization, image processing, data analysis, parallel algorithms, ...
• From a target discipline: deep theoretical knowledge at the heart of the discipline.

Is it realistic that a student can master all this in a single lifetime?
Let us be realistic!

• We must abandon the notion that a CSE student will know
  • all the math of a math major
  • all the CS of a CS major
  • all the physics (chemistry, engineering, ...) of a physics (chemistry, engineering, ...) major

• To earn a CS&E degree, a student has no time to study anything else ... but CS&E.

• How do we do this?
Can we teach CS&E from existing courses?

• Is CS&E just 30% math + 30% CS + 40% „target discipline“

• Such a student will belong to neither of the disciplines and I have increasing doubts that we can form a coherent new CS&E profile by just combining existing courses alone

• BTW - this is also true for research strategies, e.g. writing interdisciplinary CS&E proposals: interdisciplinary means more than piling up

• In a single program there is not enough time to teach a student enough out of the existing courses
  • she will not become a mathematician
  • she will not become a computer scientist
  • she will not become an engineer (physicist, chemist, etc).

• How do we position her in the academic marketplace?
Towards genuine CS&E

• Many programs already include courses specifically designed for CS&E

• In Erlangen:
  • Computational Engineering I+II in the first year (undergraduate)
  • Scientific Computing I+II in 3rd year
  • We do not have the resources to offer advanced additional courses in the Master program but have special „Bridging courses“ (also for students from other fields) entering into the CS&E master:
    • Elementary Numerical Methods
    • Advanced Programming Techniques
    • Functional Analysis for Engineers

• I believe that we must go beyond this
Towards genuine CS&E education

• **Too much:** We currently teach undergraduates efficient programming as part of an operating systems class. They become competent programmers this way, just as our CS students. But we waiste time, since learning about operating system kernels is only of little use in CS&E.

• **Too little:** We currently teach undergraduates math from the engineering curriculum and add a bit of numerics. They learn the concepts this way, but they fail to learn the rigor and abstraction they would need to pursue the mathematical analysis and mathematical design of algorithms.

• Other programs may do too much math and too little CS (as e.g. our colleagues in Delft and TUB)

• **Where is the right way in between?**
We must recognize CS&E as a separate discipline

- This means
  - acknowledging that CSE is made up of all the ellipses
  - designing courses that blend together the math, cs, and stats aspects of algorithms
  - designing applications courses each of which includes something about modeling, analysis, and computations
  - having qualifying exams that encompass more than one ellipse
  - having dissertations that reflect research in more than one ellipse
  - having committees consisting of experts in more than one ellipse