Adding Computational Modules into Required (Undergraduate) Courses at Illinois

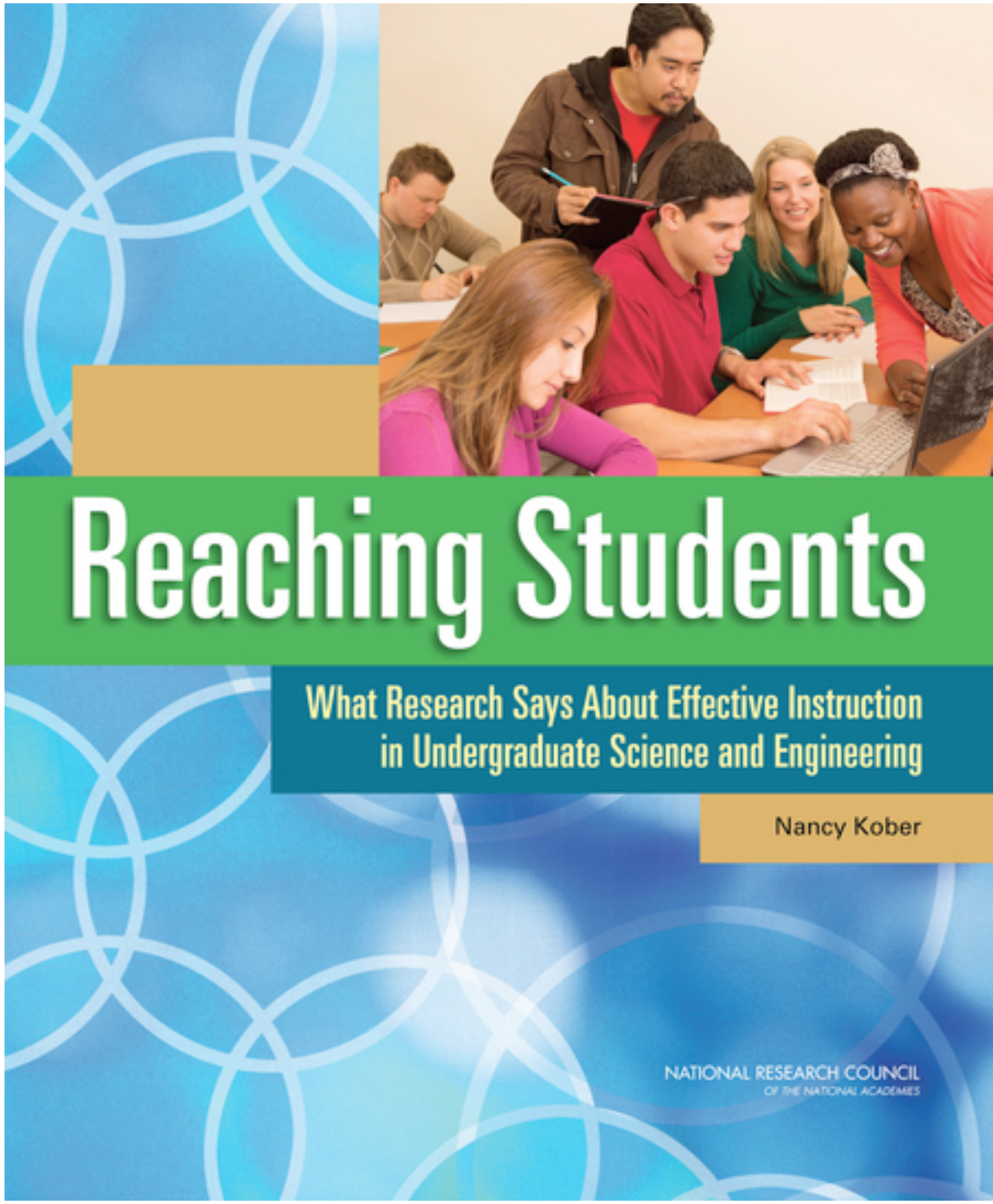
Dallas R. Trinkle, Andrew Ferguson, Cecília Leal, André Schleife

- Goals:
  1. Teach students to view computation as a tool for problem-solving
  2. Explore complex course material for improved understanding
  3. Provide hands-on experience using real tools
- Approach:
  1. Integrating computational modeling into 20X-40X courses
  2. Developing a capstone integrated computational materials engineering senior course
  3. Leveraging new technology methods to enhance learning
- Targeted courses:
  1. MSE 201 (Phases and Phase Relations)
  2. MSE 206 (Statics and Mechanics of MatSE)
  3. MSE 304 (Electronic Properties of Materials)
  4. MSE 401 (Thermodynamics)
  5. MSE 406 (Thermal and Mechanical Behavior)
  6. MSE 498AF (Computational MatSE)

Adding Computational Modules into Required (Undergraduate) Courses at Illinois

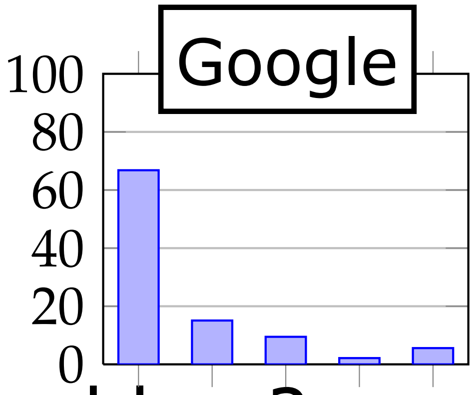
Dallas R. Trinkle, Andrew Ferguson, Cecília Leal, André Schleife

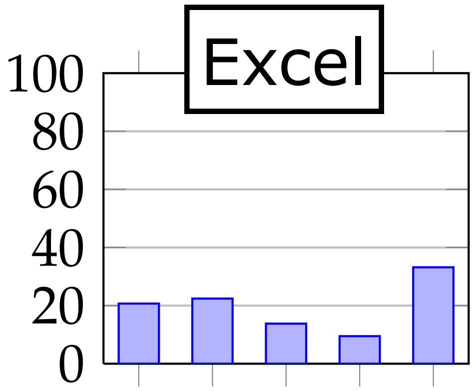
- **Strategic Instructional Initiatives Program**
  - Program through College of Engineering (2012–)
  - Originally targeting *large, introductory courses*
  - Improving and quantifying learning outcomes for students
- ***Reaching Students***
  - 2015 National Academy Press
  - Approaching teaching strategies with the same mentality we use for research



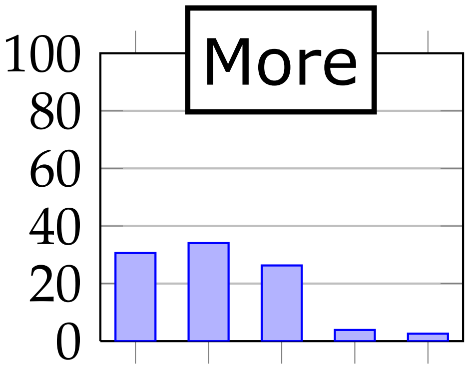
Where were MatSE students before?

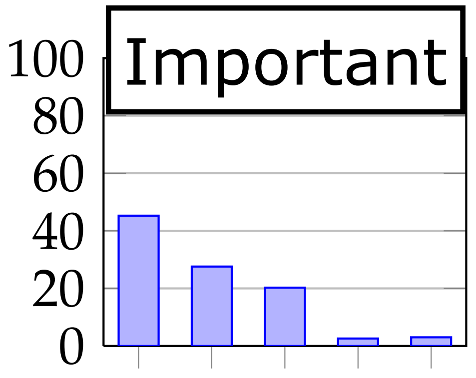
- Surveyed end of spring 2014 semester (after 201, 206, 304, 406)
- Using computer-based tools?
  - Search Engines (61% Very Often, 14% Often)
  - Excel (only 64% saying Never)
  - All others were >83% saying Never





- Using computer-based tools to solve a problem?
  - Appropriate tool: 80% Very Uncomfortable
  - Versus Analytic approaches: 46% Very Comf., 34% Comf.
- Want more computation in MatSE?
  - 31% Much More, 34% More
- Important for a post-graduation career?
  - 45% Strongly Agree, 28% Agree





What was our approach?

- Develop **computational modules** for hands-on student activities
  - Use real modeling tools (MD, FEM, DFT, CALPHAD)
  - Build modules tied to course content, and mirrored in MSE498
  - Focus on assignments where students use tools to answer questions from class
- Deploy in **required classes**
  - MSE 201: DFT for lattice constants, CALPHAD for phase diagrams
  - MSE 406: FEM for composites, fracture; MD for dislocations and strength, fracture
  - *New:* MSE 498: capstone course, covering all four modeling tools
- Provide support for students & instructors with **computational TA**
  - Separate office hours in computing lab for hands-on help
  - Reduces burden on faculty/course TA's with specialized instruction
- Use **technology** to **engage students** with better and more accessible content, and for more immediate feedback
  - Recorded lectures (posted online), in-class i>Clickers
  - Online homework (406), discussion forums (Piazza: 406)

Example Computational Report (MSE 406)

MSE 406 Computational Report 10

You will, in your assigned groups, use the molecular dynamics code LAMMPS and visualization code Ovito, available on the Ceramics 322 computers or for free download to your machine (see http://lammps.sandia.gov which includes a download page and the windows version is here: http://rpm.lammps.org/windows.html; and http://www.ovito.org/), to visualize the stress concentration in front of a crack tip, and compare with the continuum (finite element) version of the same, using OOF2.

You've previously used both of these tools on different problems, so now you will see to compare the results from atomic-scale simulations with continuum-scale simulations. In these cases, you will visualize the stress fields in front of a crack tip. You should save your visualizations, and comment on the similarities and differences between the two simulation results and the geometries. Also discuss how your results reflect what we discussed in class about stress intensity, and the distribution of stresses in front of a crack. The instructions below explain how to simulate the different cases.

Part 1: Molecular Dynamics

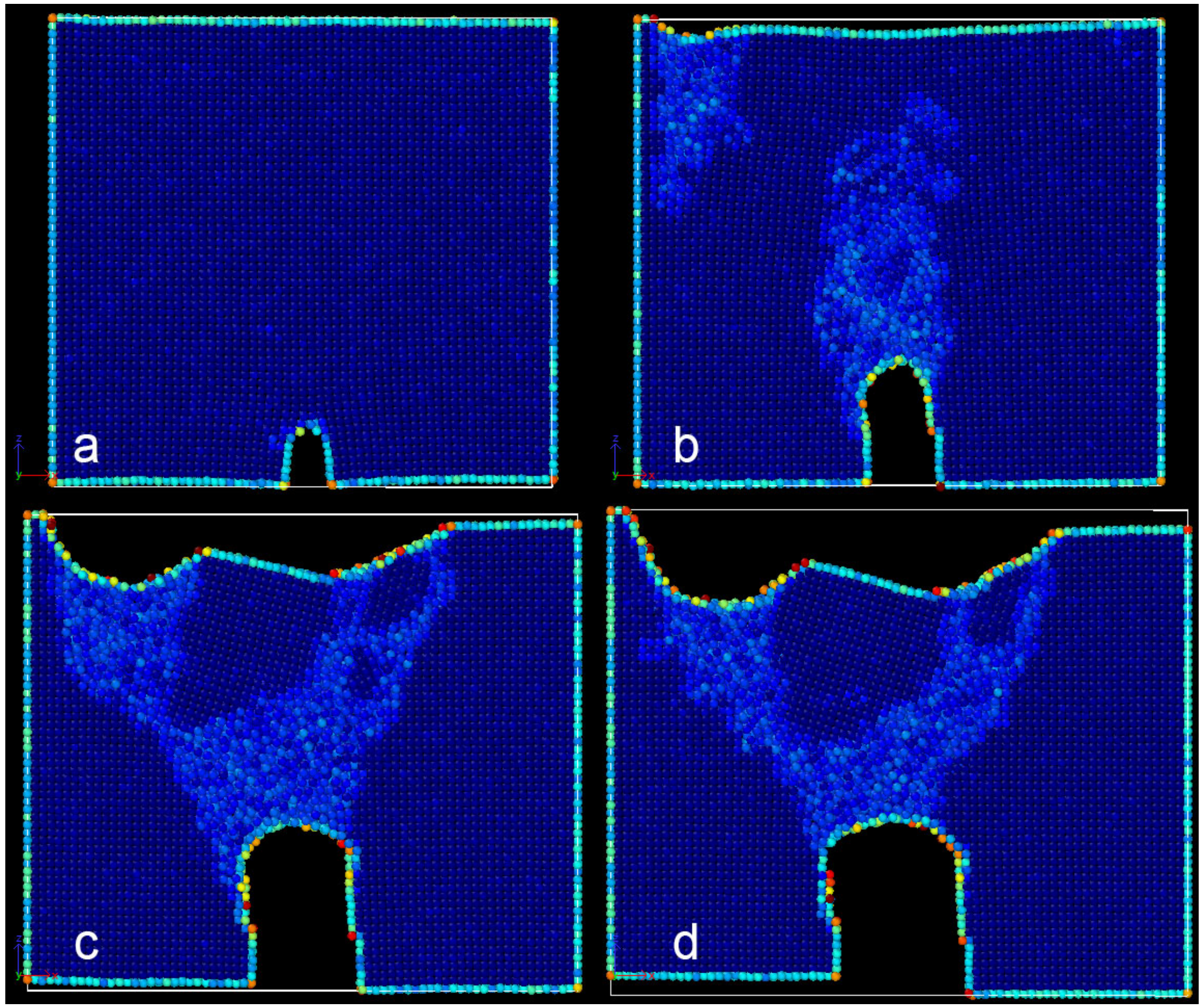
If you are using the computers in Ceramics 322, it is easiest to save your files in your local User Desktop. To do this, from the Start menu select "Computer." Double-click on "Hard Disk Drives." Then double-click on "Users" and go into the folder named with your netid. Now, double-click "Desktop" and copy the appropriate files to this folder. (Please make sure these files are saved to this Desktop, not the network Desktop)

1. Now take a look at the input file you will be using named `Al_crack.in`. Open the input

Example Computational Report (MSE 406)

1. LAMMPS Crack Simulation

We were given a bulk Al material with an initial crack to run a simulation of crack propagation when put under tensile stress. The crack was expected to propagate forwards and also widen, creating a region of plastic deformation before the crack tip.



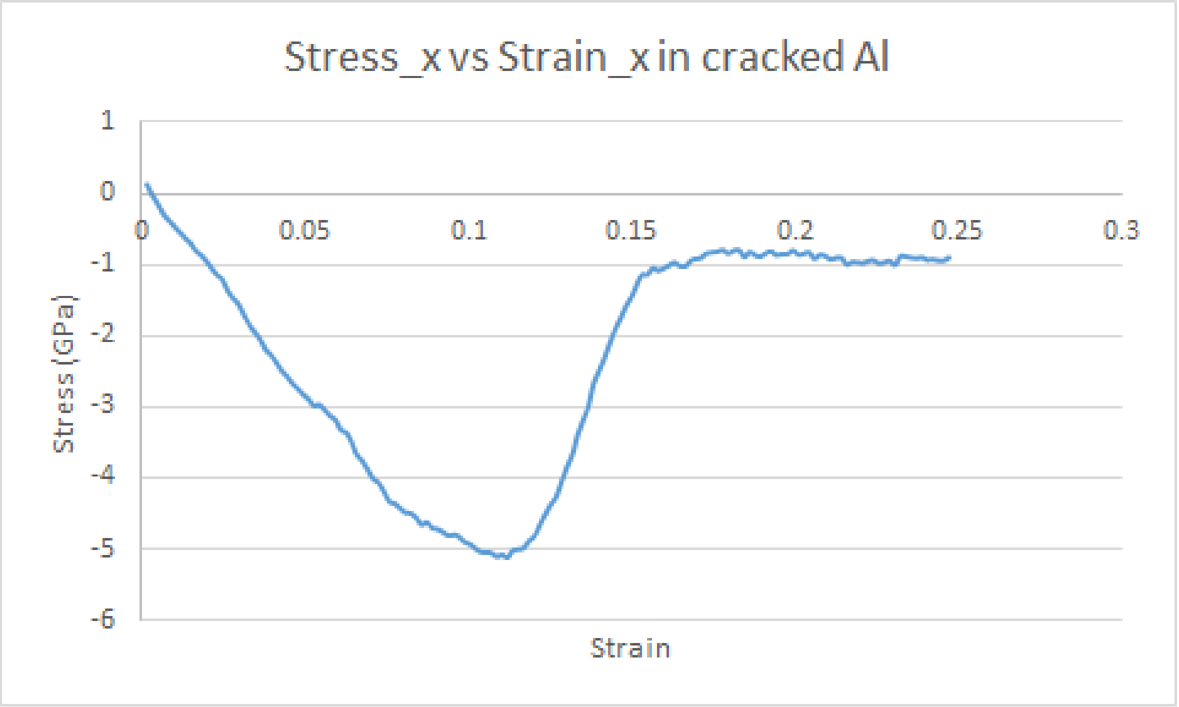


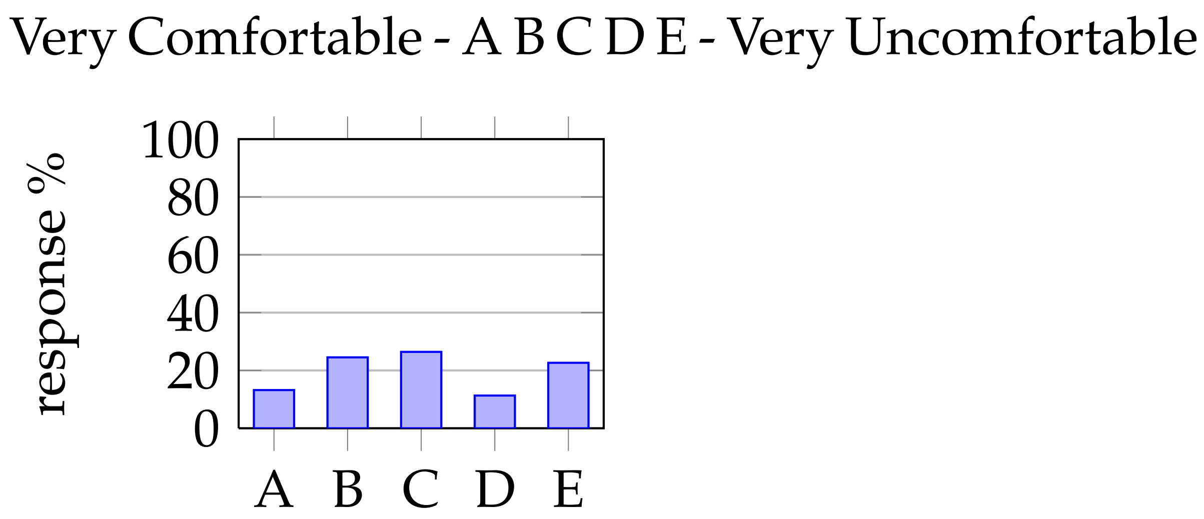
Figure 1: Shows the crack propagating through the material in the LAMMPS simulation.

Figure 2: Shows the stress vs. the strain in the sample in the LAMMPS simulation, both in the x direction.

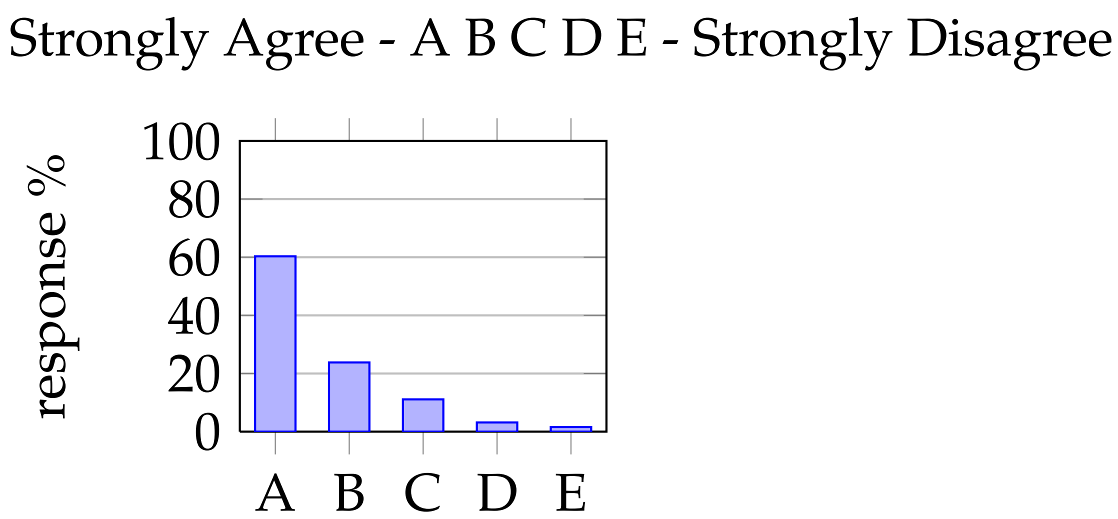
Outcomes from required courses

- Surveys: Mid-class and end-of-class feedback

201 Calphad for a phase diagram:



406 comfortable plotting data:



- Additional comments:
  - Too "cookbook": following set of directions (need to be more open-ended)
  - Need more in-depth introduction into different tools
  - Liked working on computational HW in lab and outside
  - More computation
  - Focus on applications, learning skills to use the tools

Outcomes from capstone course

- Surveys: Mid-class and end-of-class feedback
  - 498: 72% Very useful, 28% useful
  - 498: 4% Very confident, 48% confident using packages independently
  - Ranking software, most useful:
    - bash/MatLab 40%
    - Quantum Espresso, OOF2 20%
    - ThermoCalc 12%
    - LAMMPS 8%
  - Difficulty: 88% about right   Pace: 96% about right
  - Many comments liking hands-on experience with codes
- Increased enrollment in 498
  - 9 in 2013 (first taught)
  - 26 in 2014 (capped due to limited number of workstations)

Lessons learned?

- Incorporating computational can be done in “non-CS” classes
  - Some web-based tools can do the job (e.g.: nanohub)
  - Students are capable of using the computing labs, but will need some guidance (a good case for online discussion forums)
  - Making open-ended exploration for students is a challenge, but even something that gets students doing *something* is worthwhile
- Worth considering computation throughout curriculum
  - One-off in a class is less useful; building competency in multiple courses should pay dividends (now in second year)
  - Can allow more “realistic” material to be considered: simple models in class, more complex models with computation
  - Using computation to investigate phenomena is an additional approach to learning
  - With some support, can be done with non-computational faculty
    - Team of faculty working together with stable assignments
    - Trained computational TA across courses