Integrating Multiphysics and Multiscale Modeling Environments Together

Is An Open Environment Possible?

D. Bindel

Department of Computer Science
Cornell University

24 May 2011

Outline

The CSE Picture

My Current Codes

The Dream

Partial Successes

The Road Forward

Outline

The CSE Picture

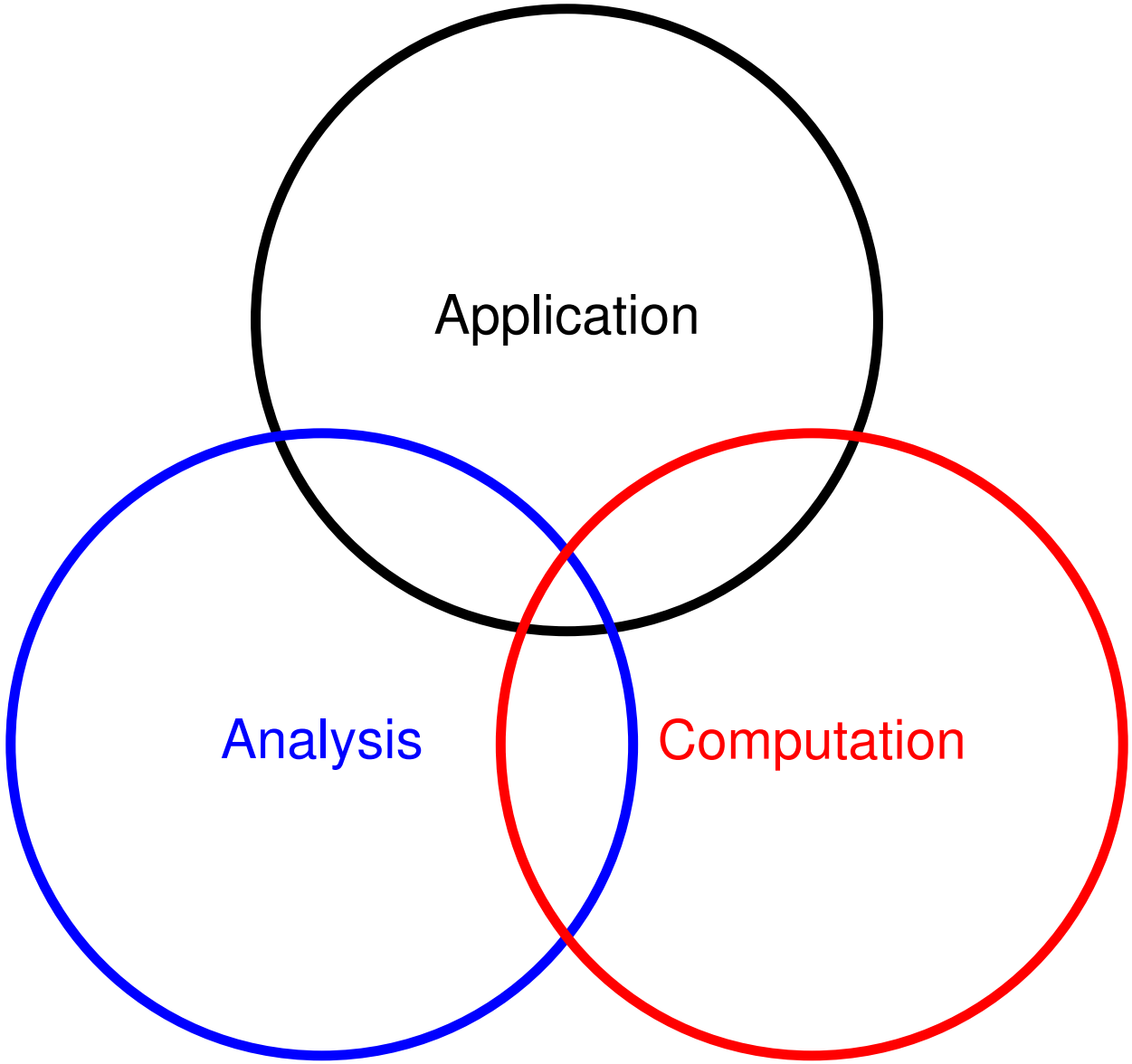
My Current Codes

The Dream

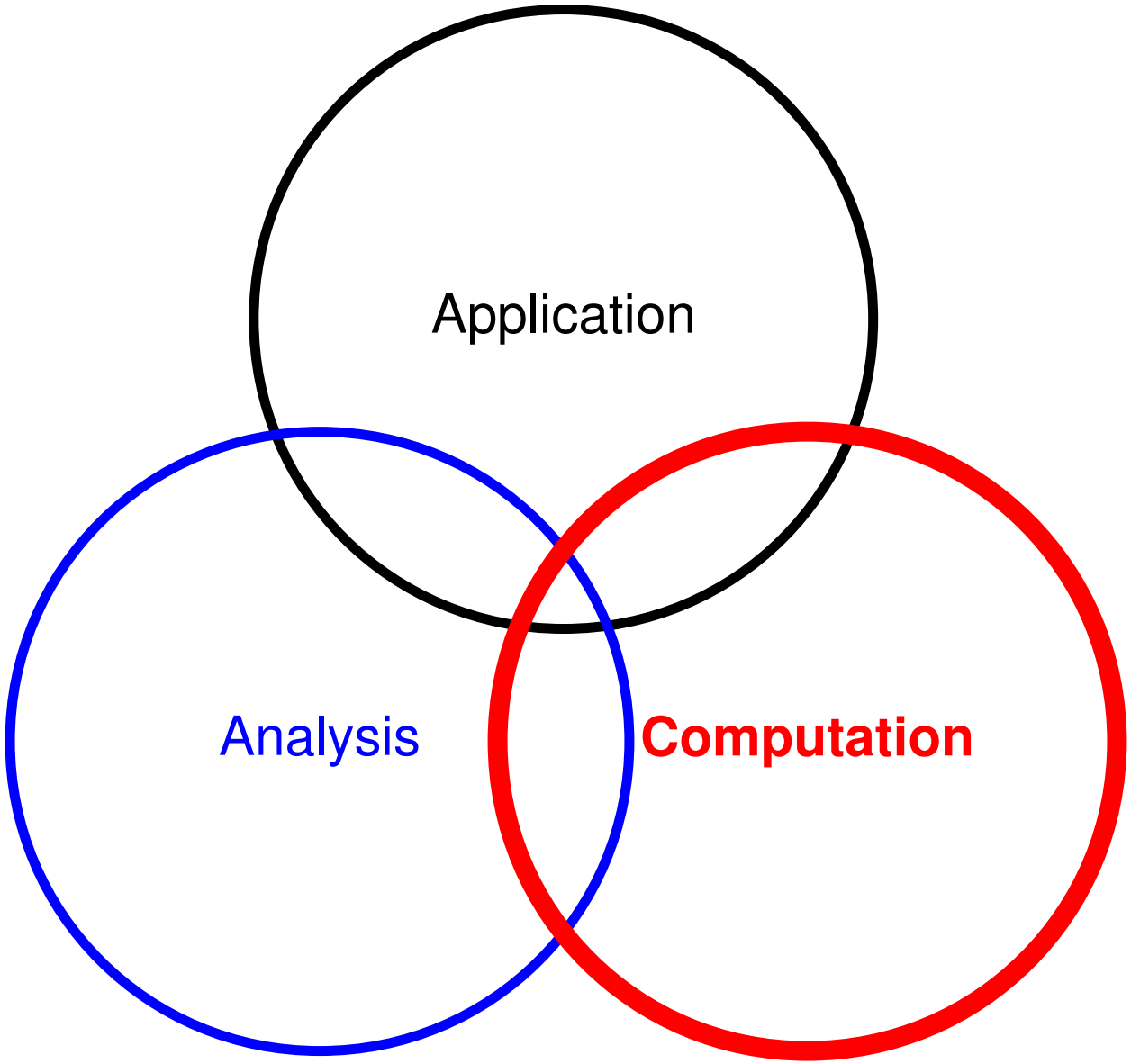
Partial Successes

The Road Forward

The Computational Science & Engineering Picture



What is this talk about?



The Goal

The difference between art and science is that science is what we understand well enough to explain to a computer. Art is everything else.

Donald Knuth

The purpose of computing is insight, not numbers.

Richard Hamming

The ideal(ized?) state:

- Human explores, gains insight into good design
- Computer manages computational details

Outline

The CSE Picture

My Current Codes

The Dream

Partial Successes

The Road Forward

HiQLab

I wanted a code to explore quality factors in MEMS, but:

- Existing codes do not compute quality factors
- ... and awkward to prototype new solvers
- ... and awkward to programmatically define meshes
- So I wrote a new finite element code: HiQLab

HiQLab Desiderata

What I needed:

- Flexible infrastructure for multiphysics FEA
- Electrical, mechanical, and coupling element libraries
- Interfaces to good existing solver libraries
- Scripting interface for fast algorithm prototyping
- Scriptable mesh generation for parameter studies
- Some control over code base and build processes

Didn't worry about:

- User interface design (much)
- Automatic mesh refinement
- Integration with commercial codes

Heritage of HiQLab

SUGAR: SPICE for the MEMS world

- System-level simulation using modified nodal analysis
- Flexible device description language
- C core with MATLAB interfaces and numerical routines
- Briefly existed as web service (M&MEMS)

FEAPMEX/MATFEAP: MATLAB + a finite element code

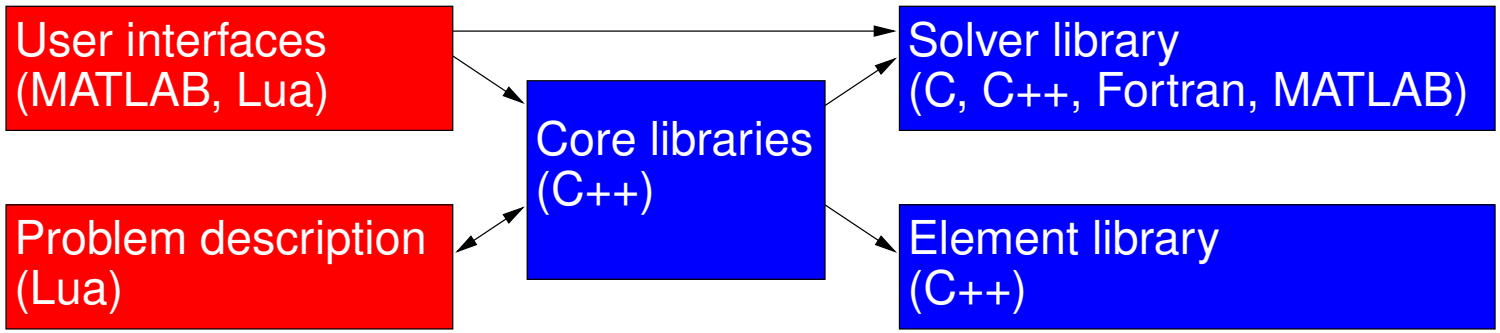
- MATLAB interfaces for steering, testing solvers, running parameter studies
- Time-tested finite element architecture
- But old F77, brittle in places

Other Ingredients

"Lesser artists borrow. Great artists steal."
– Picasso, Dali, Stravinsky?

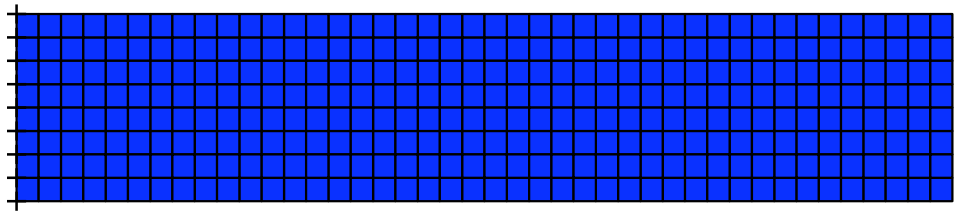
- Lua: www.lua.org
  - Evolved from simulator data languages (DEL and SOL)
  - Pascal-like syntax fits on one page; complete language description is 21 pages
  - Fast, freely available, widely used in game design
- MATLAB: www.mathworks.com
  - "The Language of Technical Computing"
  - OCTAVE also works well
- Standard numerical libraries: ARPACK, UMFPACK
- MATEXPR, MWRAP, and other utilities

HiQLab Structure



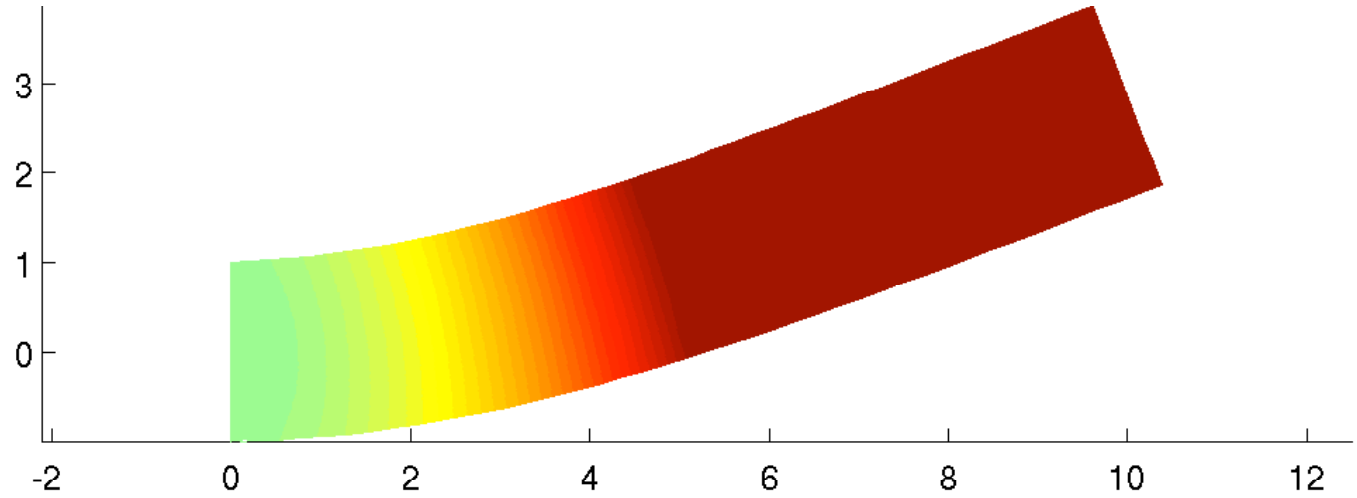
- Standard finite element structures + some new ideas
- Full scripting language for mesh input
- Callbacks for boundary conditions, material properties
- MATLAB interface for quick algorithm prototyping
- Cross-language bindings are automatically generated

HiQLab's Hello World



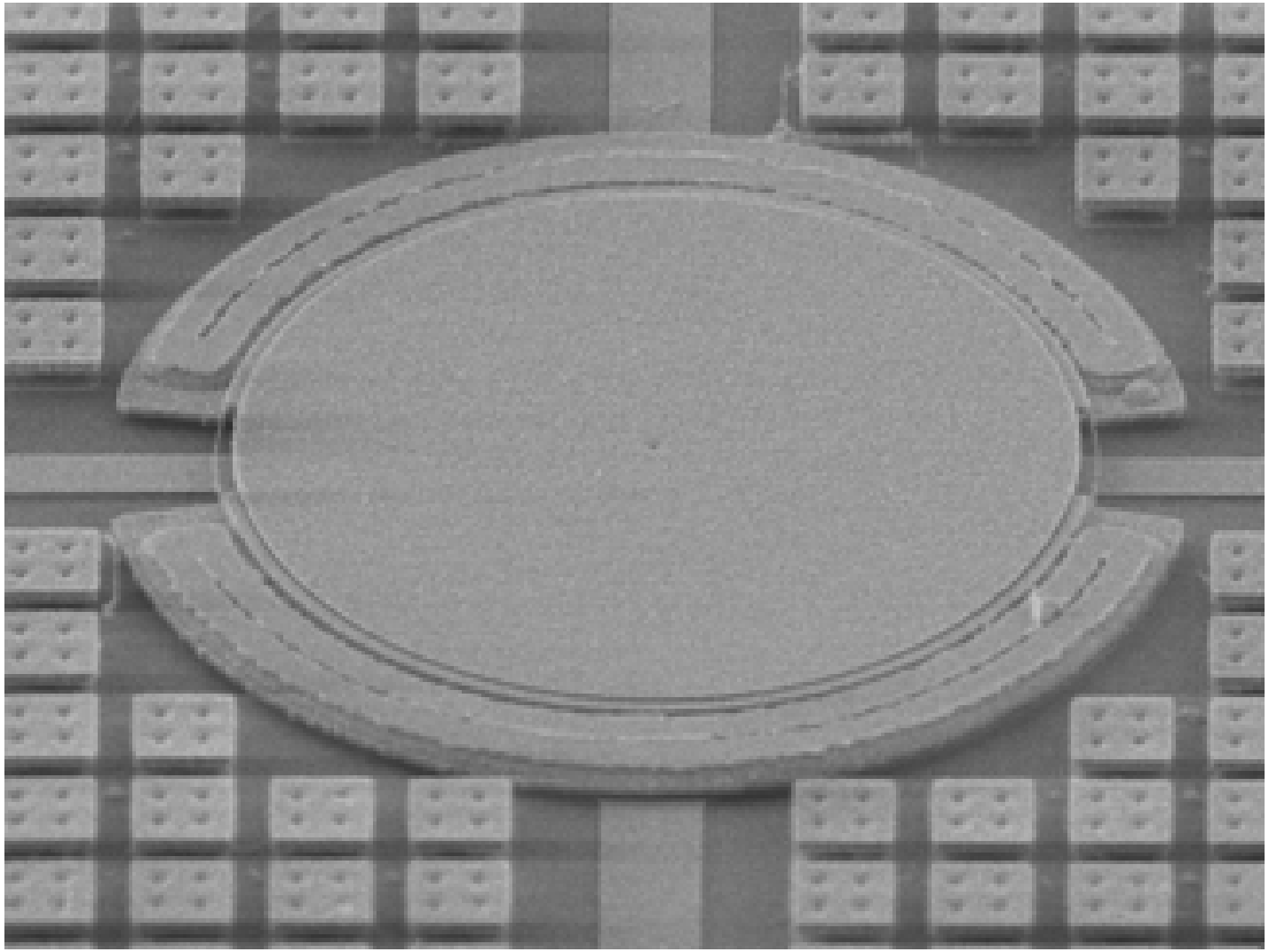
```
mesh  = Mesh:new(2)
mat = make_material('silicon2', 'planestrain')
mesh:blocks2d( { 0, l }, { -w/2.0, w/2.0 },
               mat )

mesh:set_bc(function(x,y)
  if x == 0 then return 'uu', 0, 0; end
end)
```

HiQLab's Hello World

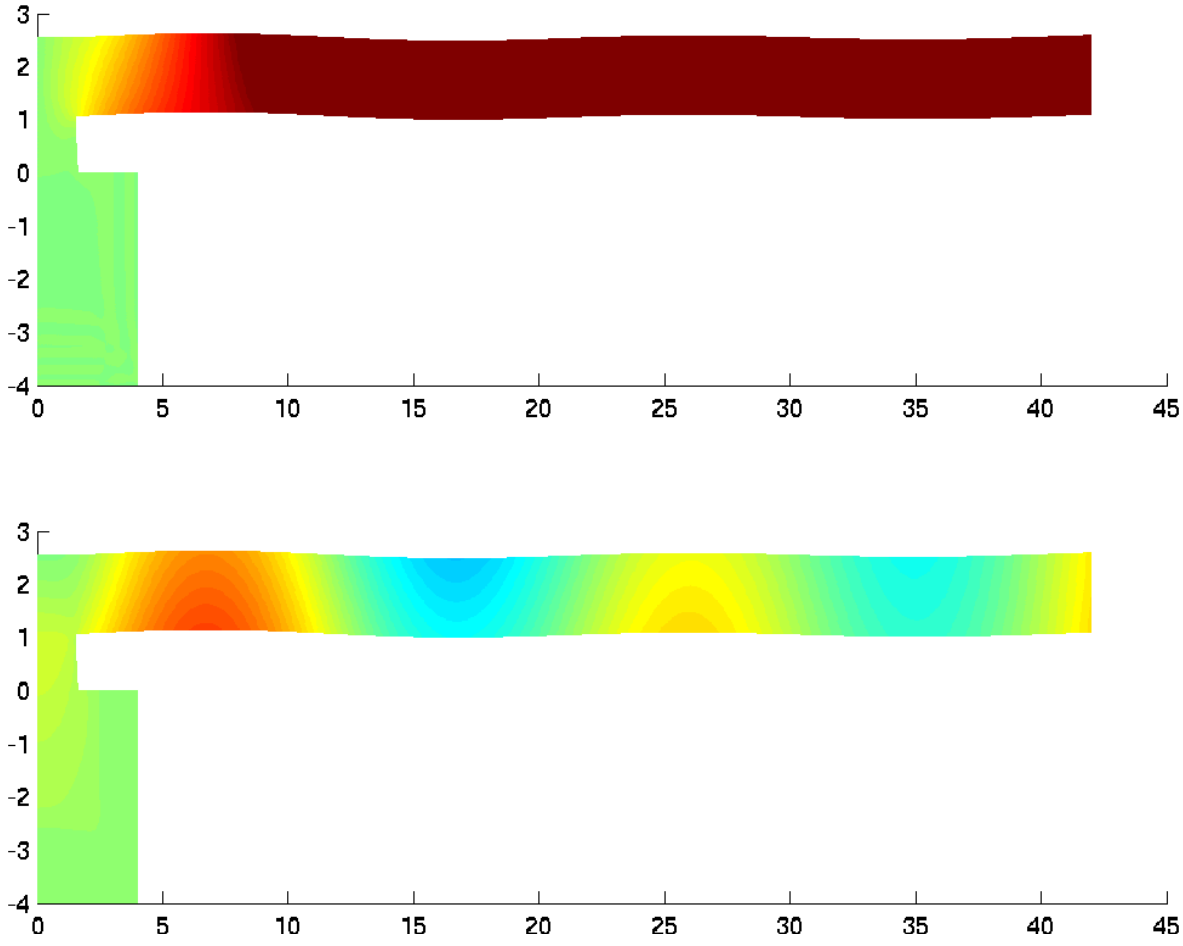



```
>> mesh = Mesh_load('beammesh.lua');
>> [M,K] = Mesh_assemble_mk(mesh);
>> [V,D] = eigs(K,M, 5, 'sm');
>> opt.axequal = 1; opt.deform = 1;
>> Mesh_scale_u(mesh, V(:,1), 2, 1e-6);
>> plotfield2d(mesh, opt);
```

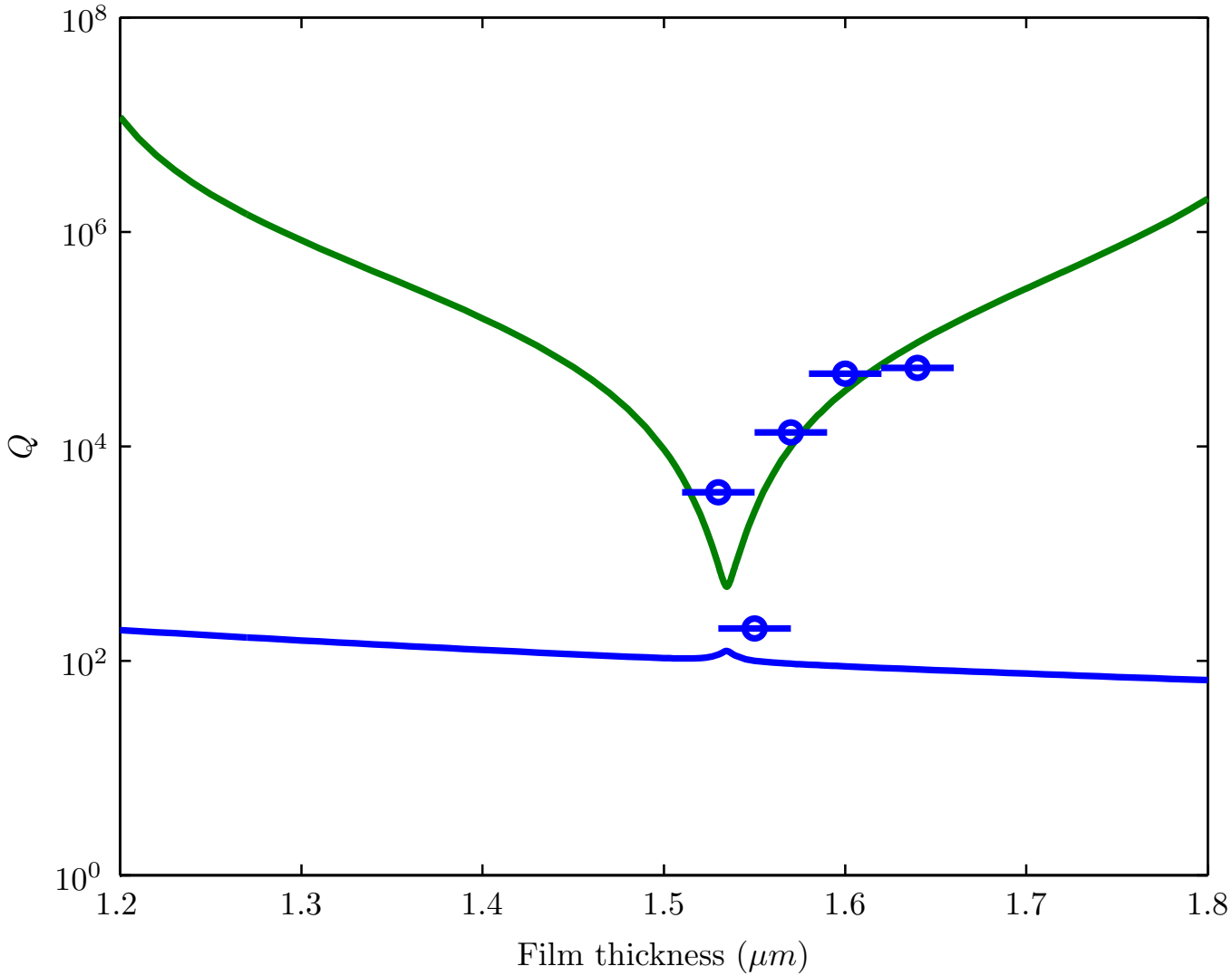
Disk Resonator Simulations



Response of the Disk Resonator

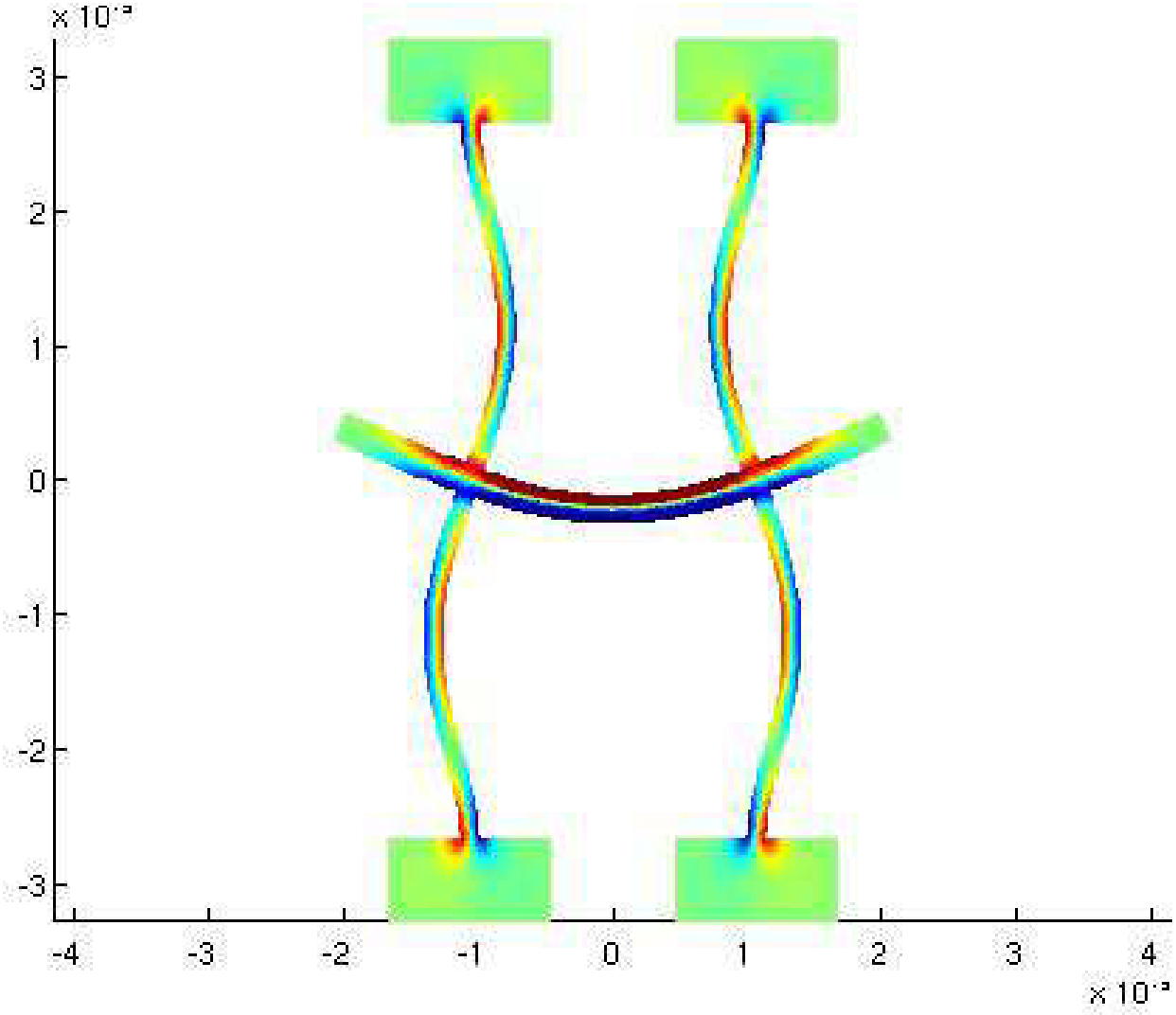


Variation in Quality of Resonance



Simulation and lab measurements vs. disk thickness

Thermoelastic Damping (TED)



HiQLab now?

- Good workbench for my *numerical* work
  - on new eigensolvers and model reduction methods
  - on various types of sensitivity analysis
  - on loss modeling
  - on parallel solvers
- Useful starting point for other codes
- But a mixed success for conveying results to designers
- Lack time for a rewrite!

Outline

The CSE Picture

My Current Codes

The Dream

Partial Successes

The Road Forward

The Designer's Dream

Ideally, would like

- Simple models for behavioral simulation
- Parameterized for design optimization
- Including all relevant physics
- With reasonably fast and accurate set-up

We aren't there yet.

The Dream: Problem Solving Environments

A PSE is a computer system that provides all the computational facilities needed to solve a target class of problems. These features include advanced solution methods, automatic and semiautomatic selection of solution methods, and ways to easily incorporate novel solution methods. Moreover, PSEs use the language of the target class of problems, so users can run them without specialized knowledge of the underlying computer hardware or software. ... Overall, they create a framework that is all things to all people: they solve simple or complex problems, support rapid prototyping or detailed analysis, and can be used in introductory education or at the frontiers of science.

Gallopoulos, Houstis, and Rice, IEEE CSE 1994.

The Dream Continues

The Modelica modeling language and technology is being warmly received by the world community in modeling and simulation with major applications in virtual prototyping. It is bringing about a revolution in this area, based on its ease of use, visual design of models with combination of lego-like predefined model building blocks, its ability to define model libraries with reusable components, its support for modeling and simulation of complex applications involving parts from several application domains, and many more useful facilities. To draw an analogy, Modelica is currently in a similar phase as Java early on, before the language became well known, but for virtual prototyping instead of Internet programming.

Fritzon, Modelica Tutorial, 2006.

The Skeptic Intrudes

Skepticism is not pessimism, however. Although we see no startling breakthroughs–and indeed, I believe such to be inconsistent with the nature of software–many encouraging innovations are under way. A disciplined, consistent effort to develop, propagate, and exploit these innovations should indeed yield an order-of-magnitude improvement. There is no royal road, but there is a road.

Fred Brooks, "No Silver Bullet", 1986

- Rice started writing about PSEs in the early 1960s
- Modelica started in 1996
- Where is my MovieOS CAD system?

Outline

The CSE Picture

My Current Codes

The Dream

Partial Successes

The Road Forward

Some Successes

- Numerical programming environments
  - MATLAB, SciPy, ROOT, StarP
  - Integrated libraries as a big selling point
- Circuit systems
  (SPICE, Cadence, Saber; VHDL, MAST, Verilog)
- System modeling (Saber, Modelica, Simulink)
- Commercial FEA (COMSOL, ANSYS, ABAQUS)
- MEMS CAD environments (Coventor, Intellisuite)

Ingredients

- High-level domain-specific languages (declarative?)
  - VHDL, Verilog, and company for circuit design
  - Block diagrams in Simulink, Modelica, LabView
  - Variational forms in Sundance, FENiCS, related projects
  - AMPL for optimization problems
  - Numerical scripting in MATLAB and Python
- Backed by reasonable numerics (models, solvers)
- Using common building blocks (libraries, languages, legacy codes)
- Targeted application domain (can't solve everything!)
- Documentation, education, and advertising

Outline

The CSE Picture

My Current Codes

The Dream

Partial Successes

The Road Forward

Desiderata Revisited

I want CAD with

- Good mechanisms for domain-specific abstractions
- Flexible solvers balancing speed and robustness
- “Software bus” connections between different simulations (or maybe integration at a lower level?)
- Design checks, debugging support, support for V&V
- Modern, professionally designed interfaces (MovieOS?)

Tech Challenges for Next-Gen CAD

We want to glue together:

- System models and device models (ROM)
- Geometric CAD and discretizations (IGA)
- Different types of discretizations (mortars)
- Multiscale (specialized methods even in simple cases)
- Multiphysics (integrating engines, specialized solvers)
- Different engineering domains (circuits, solids, fluids)
- Different codes (legacy or new, commercial or open)
- Different representations (text or diagrams? implicit or explicit? black-box or with metadata?)
- Flexibility (for openness) and control (for reproducibility)

Nontech Challenges

- Most of us are terrible UI designers
- Intuition is a function of education
- Need to leverage both industry and academia