What is Scientific Computing? Why study it?

Andreas Klöckner

November 2, 2026

A lot of CS: **'How can we put discrete math into a computer?'**

Meanwhile SC: **'How can we put continuous math into a computer?'**

A lot of CS: **'How can we put discrete math into a computer?'**

Meanwhile SC: **'How can we put continuous math into a computer?'**

`https://github.com/theabbie/DoublePendulum`

What continuous math?

- $\mathbb{R}$
- linear algebra
- integrals
- derivatives
- differential equations
  - ordinary
  - partial

And: What is lost as we put these things into a computer?
('Error')

Methods, analysis, implementation

But why? (i.e., Applications)

What could you do if you could do calc. and lin. algebra **at scale?**

- Computational geometry, computer graphics
- Economics, HFT
- Design optimization
- Robotics
- ML
- Predictive/engineering simulation
- ...

But why? (i.e., Applications)

What could you do if you could do calc. and lin. algebra **at scale?**

- Computational geometry, computer graphics
- Economics, HFT
- Design optimization
- Robotics
- ML
- Predictive/engineering simulation
- ...

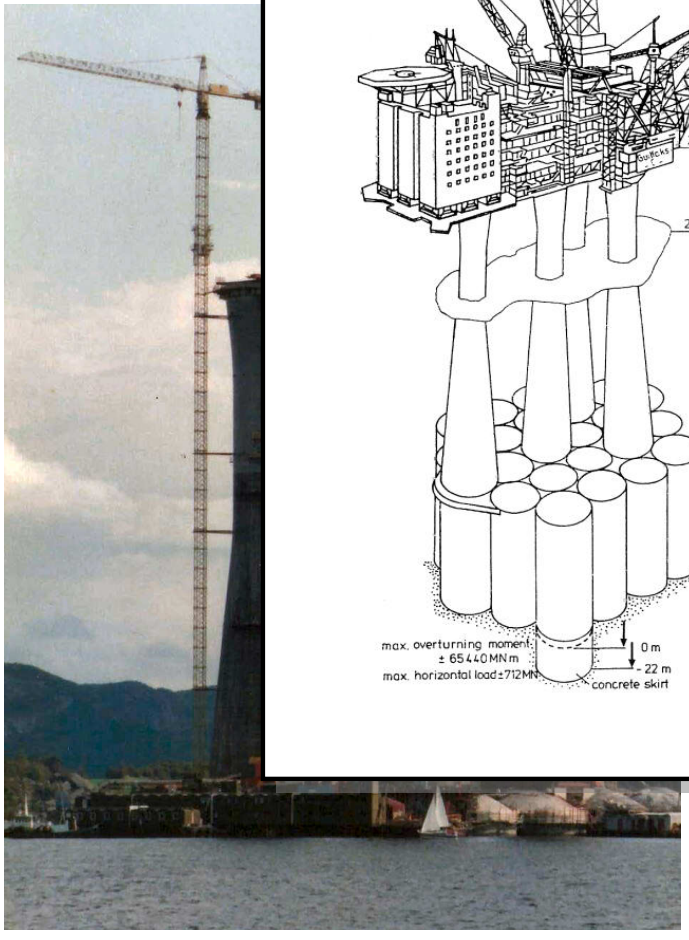
At a research level: focus on what's difficult: **really large scale**.

Sleipner A concrete deep-water structure



Wikipedia, `http://matdl.org/failurecases/`

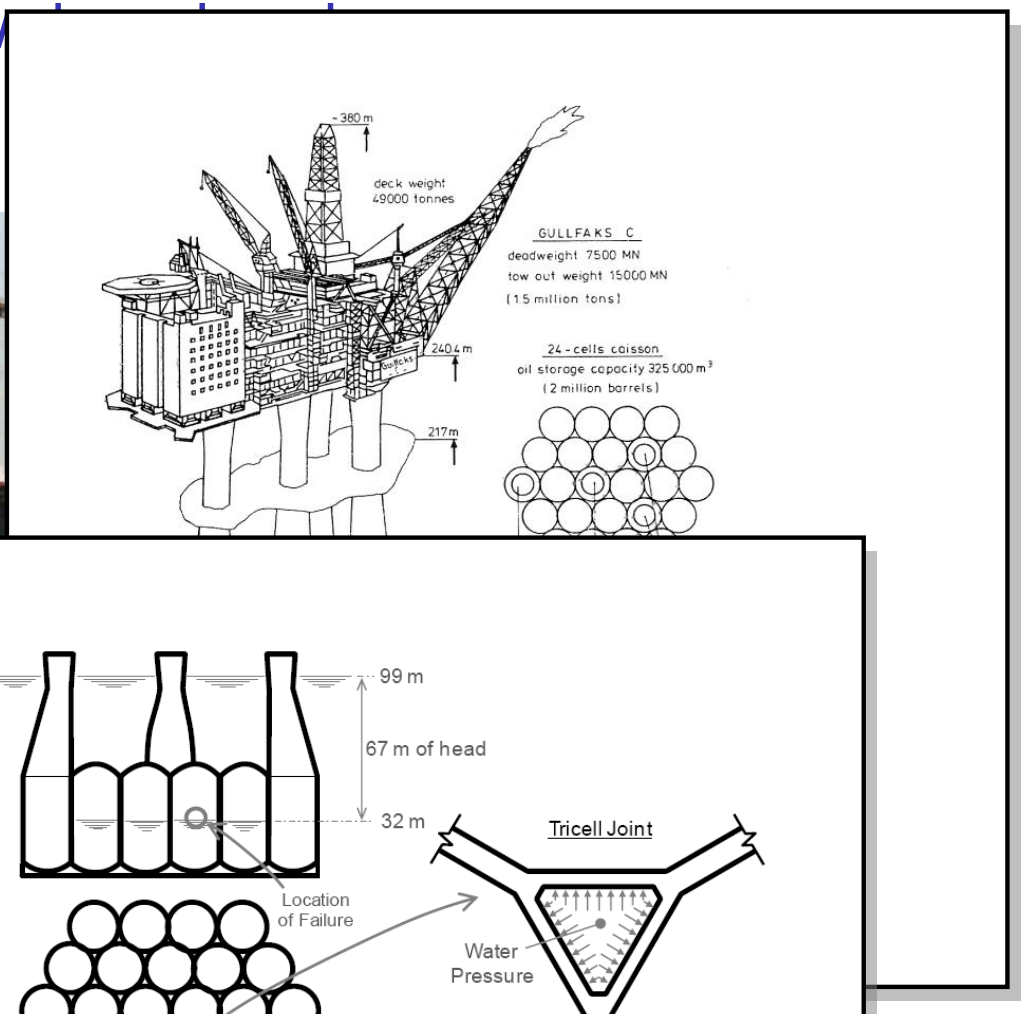
Sleipner A concrete deep-w



Wikipedia, `http://matdl.org/failurecases/`

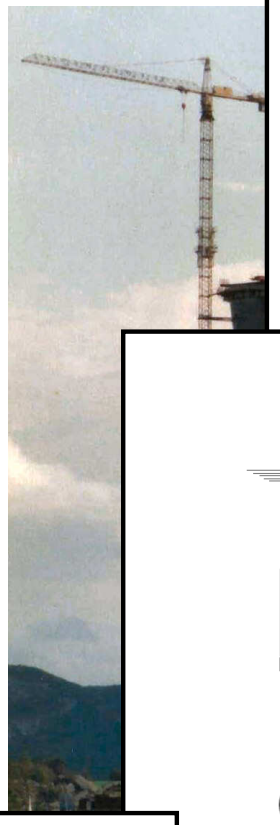
Sleipner A concrete deep-w

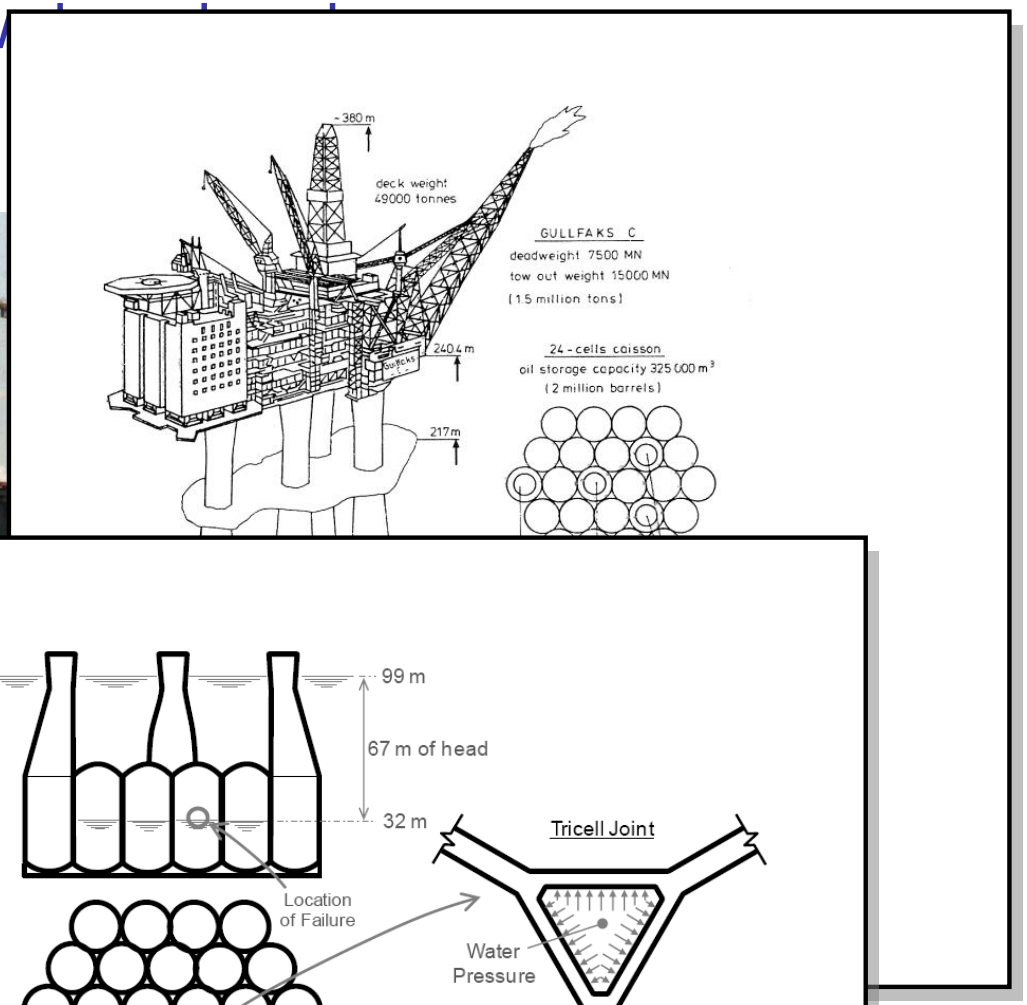




Wikipedia, `http://matdl.org/failurecases/`

Sleipner A concrete deep-w



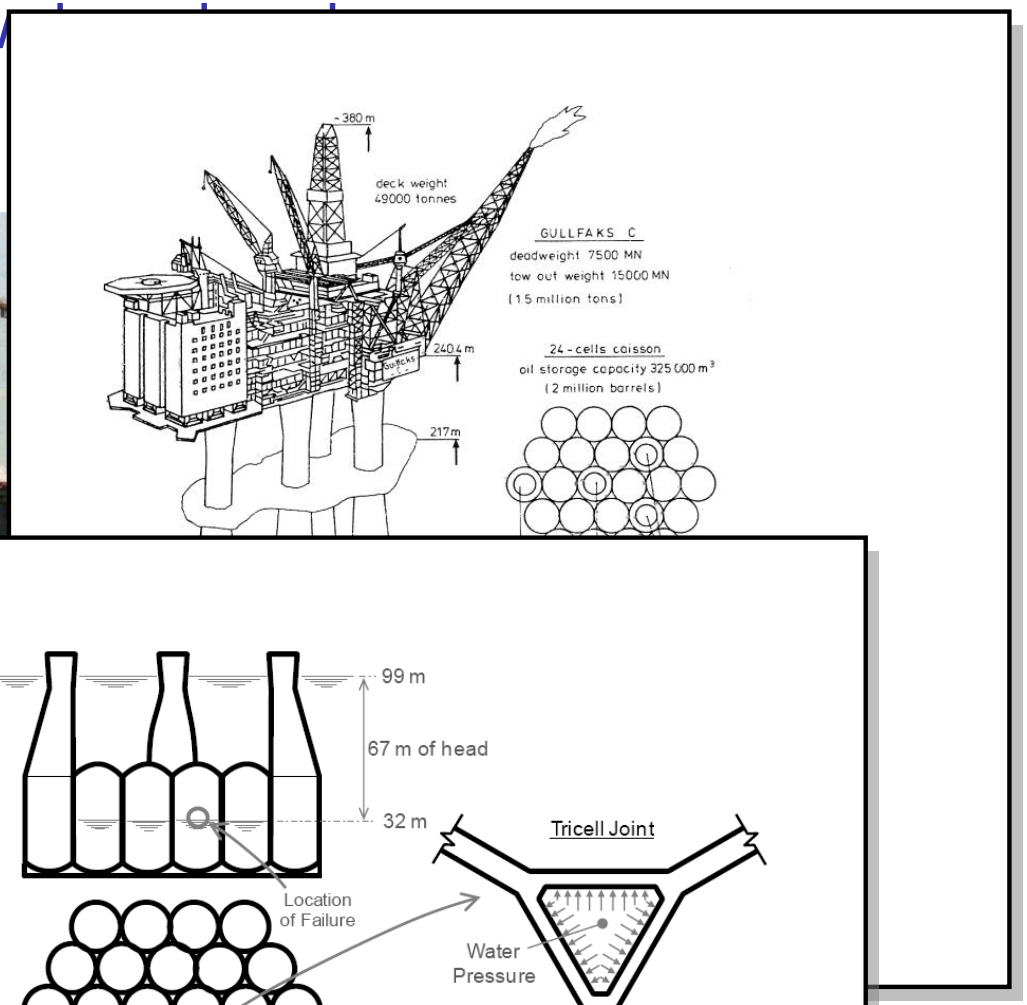


$1 billion

Wikipedia, `http://matdl.org/failurecases/`

Sleipner A concrete deep-w





$1 billion

↓

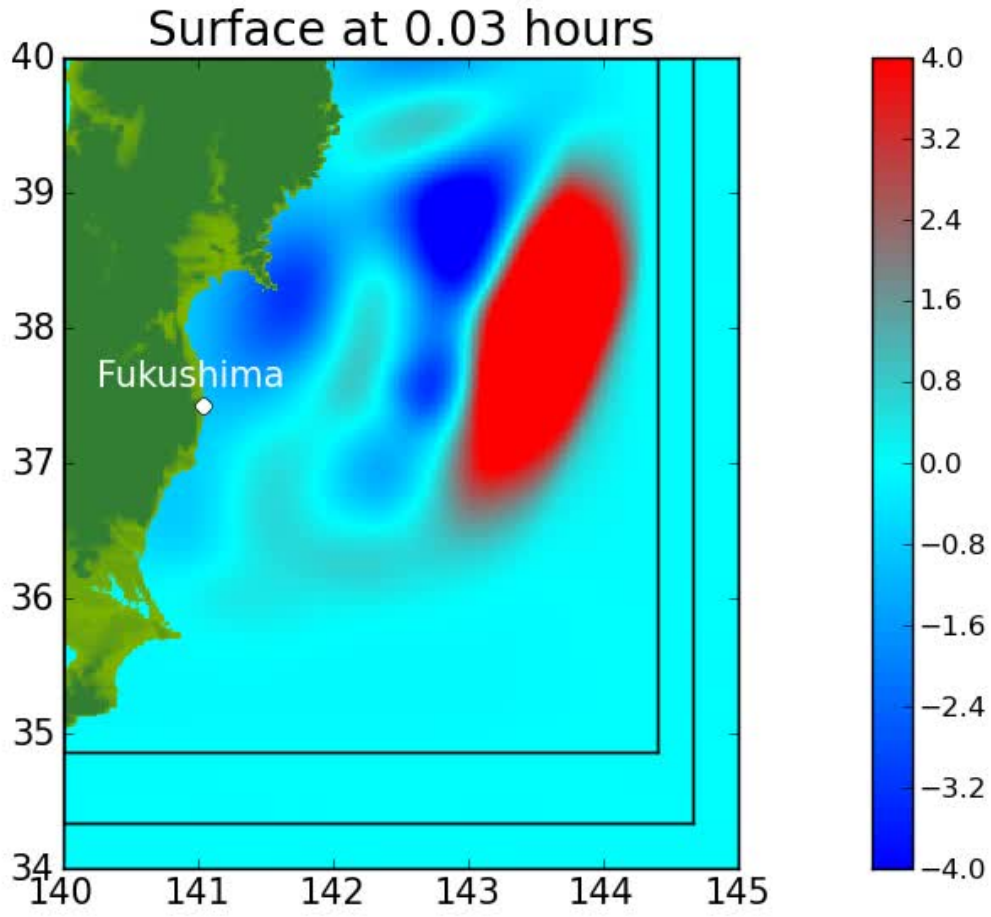
Wikipedia, `http://matdl.org/failurecases/`



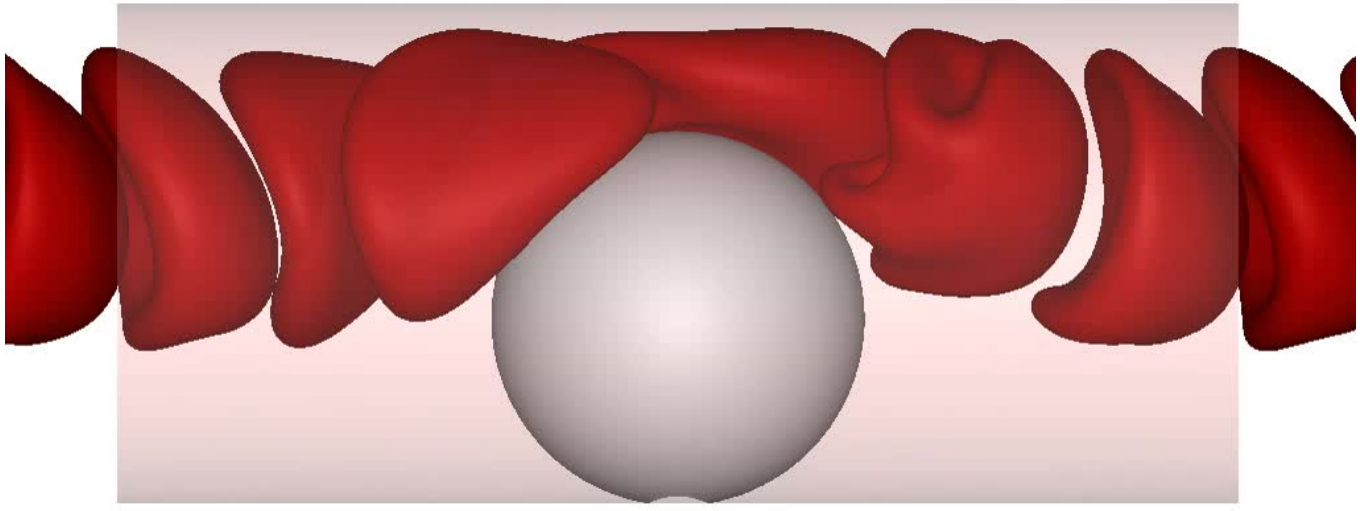
dailymail.co.uk



dailymail.co.uk



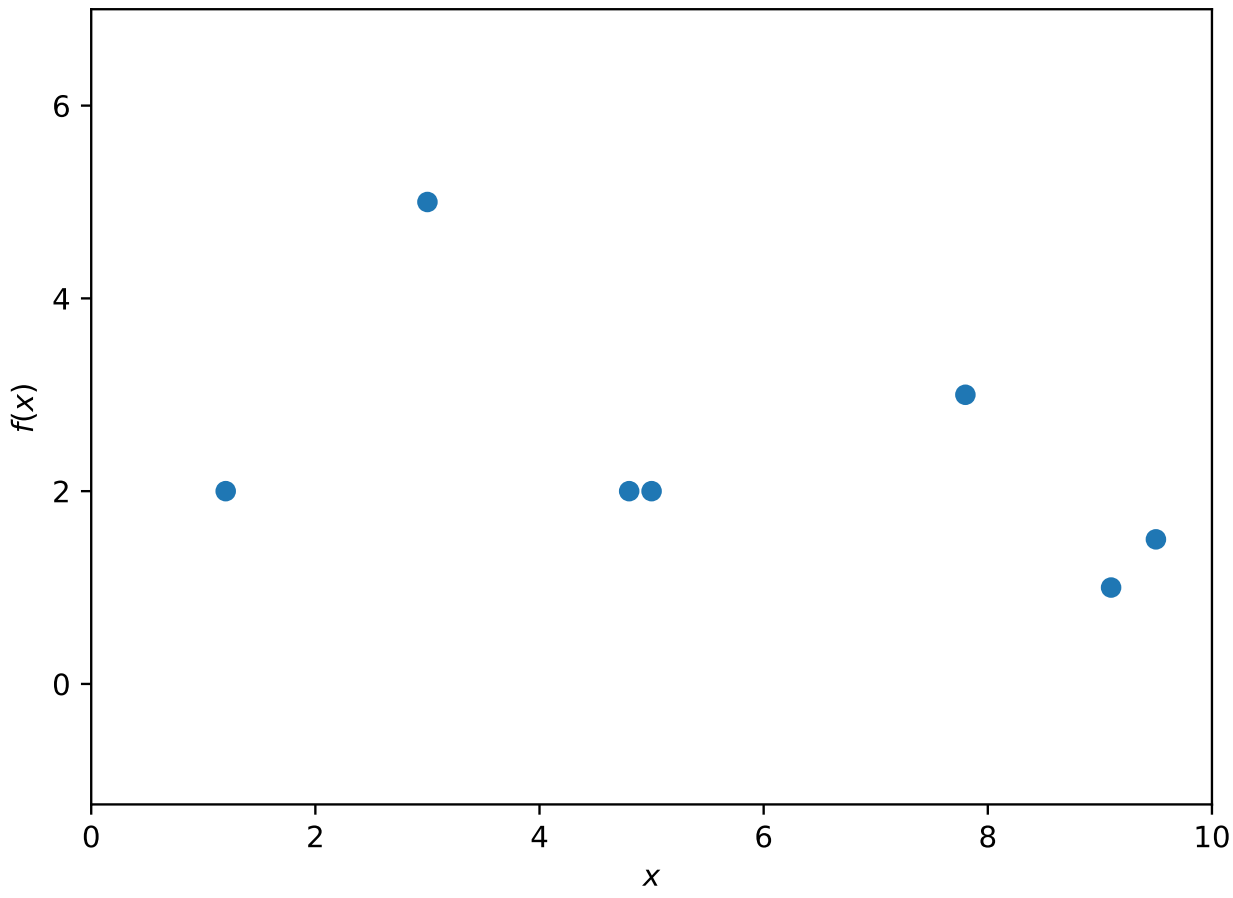
Randy LeVeque et al (U. Washington)



J. Freund et al (UIUC)

Have: Values $f(x)$ of function at a few points

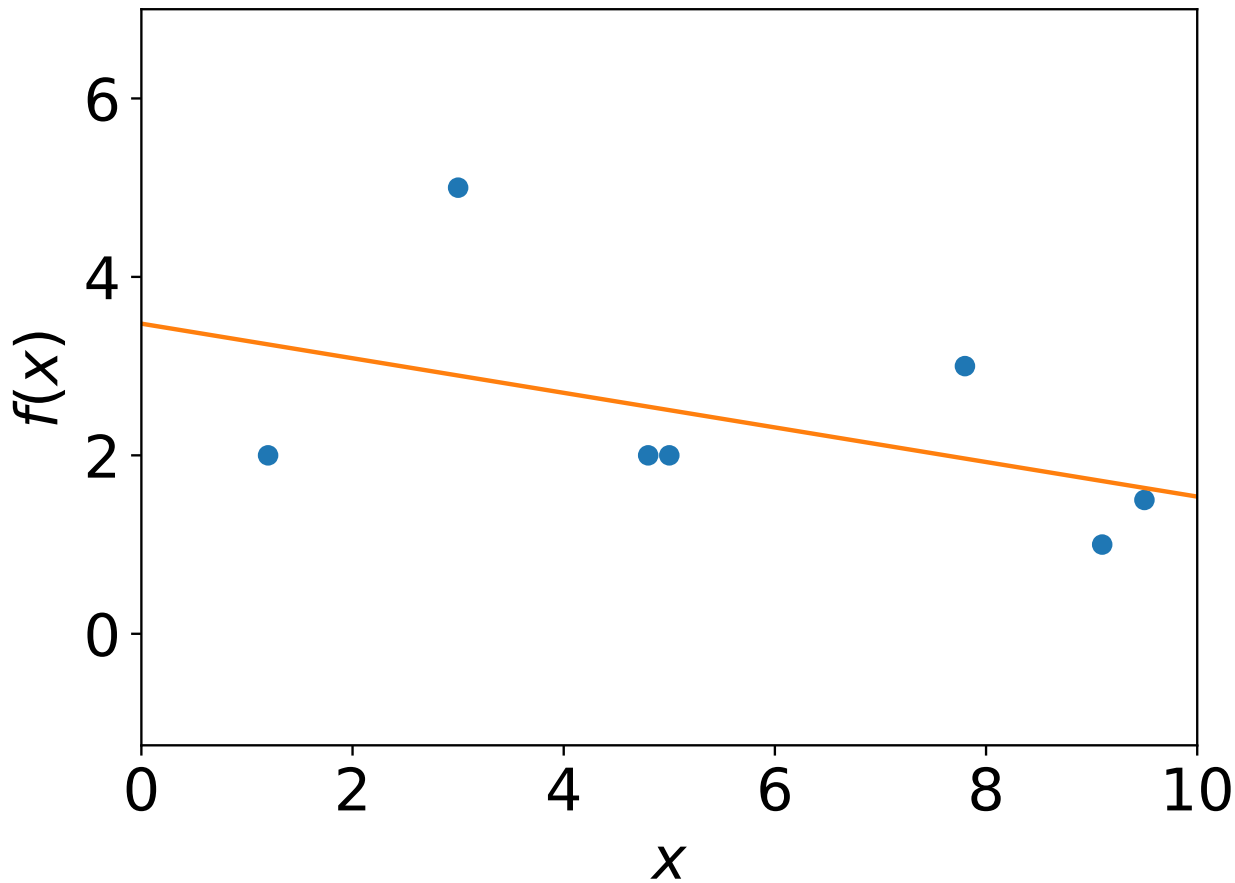
Want: Value of $f(x)$ everywhere


6
4
2
0
f(x)
0
2
4
6
8
10
x

Interpolation

- A first step in many methods
- Once you know the function, you can do business

Idea: $ax + b$


6
4
2
0
f(x)
0
2
4
6
8
10
x

Accurate?

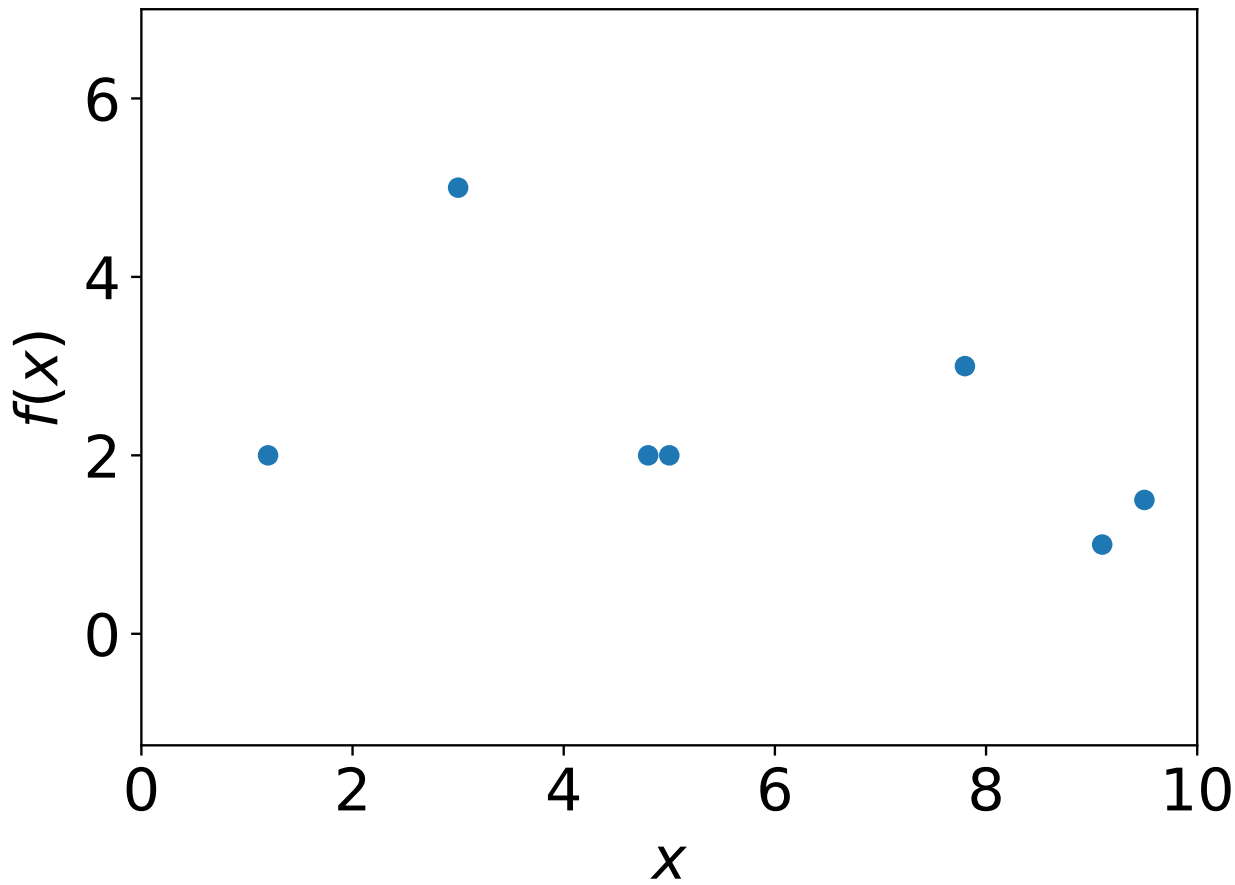
Small ‘error’

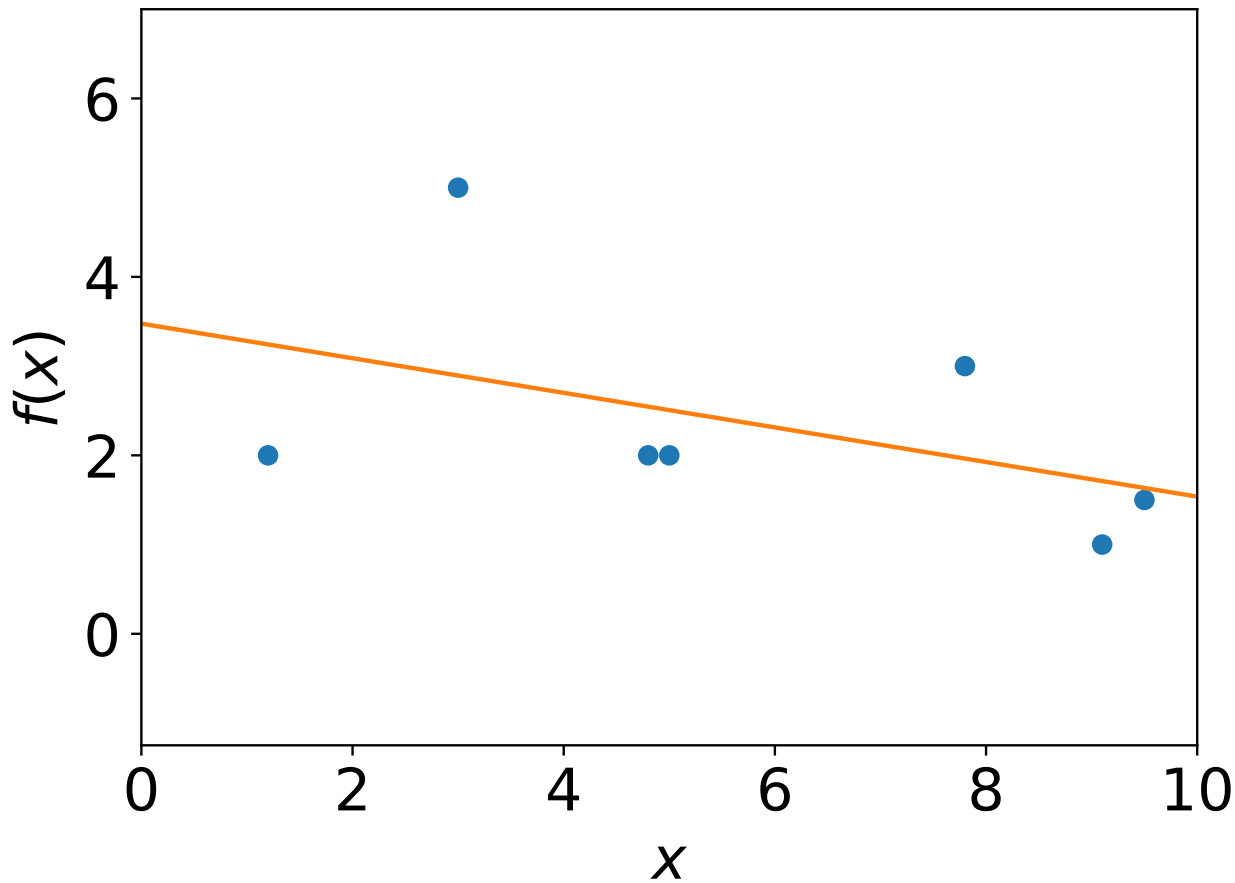
Accurate?

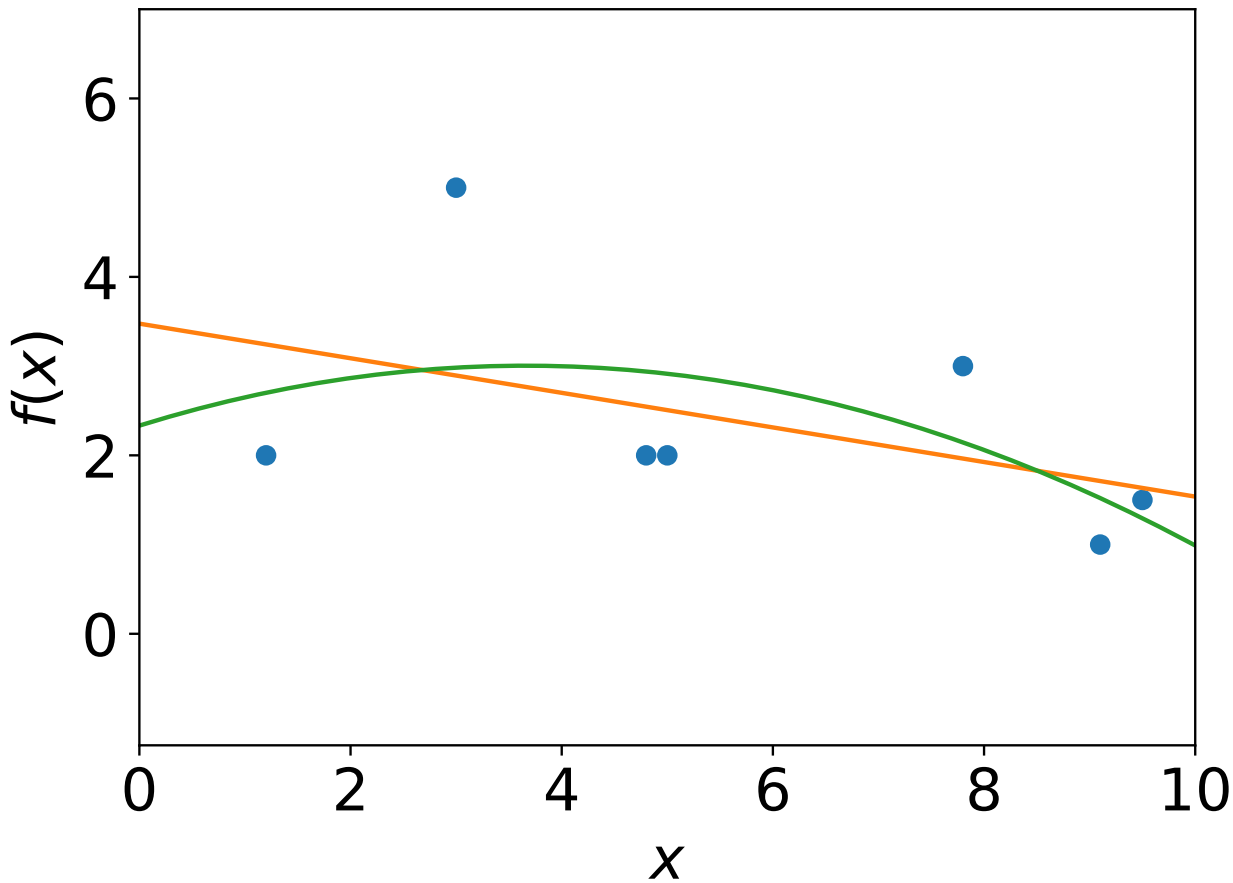
Small ‘error’

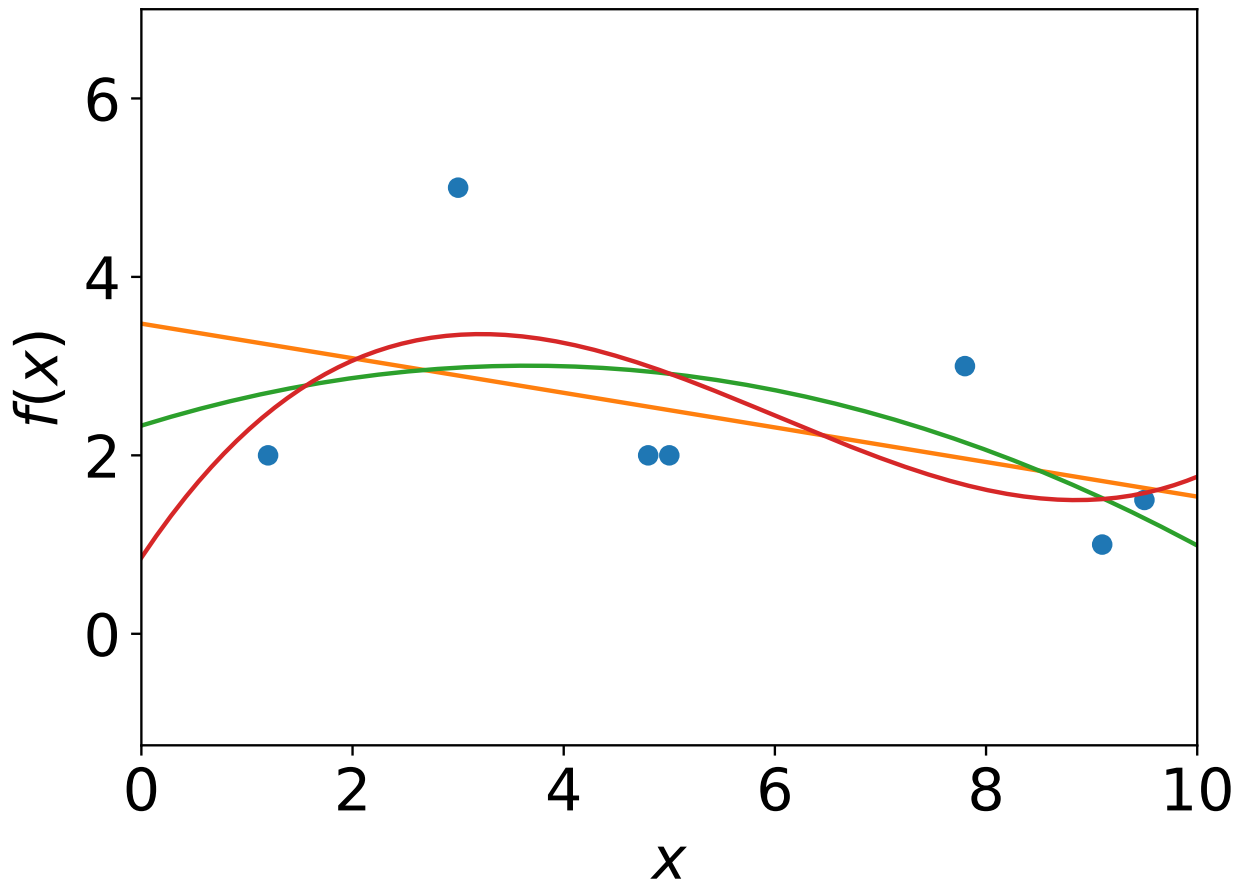
What is ‘error’ here?

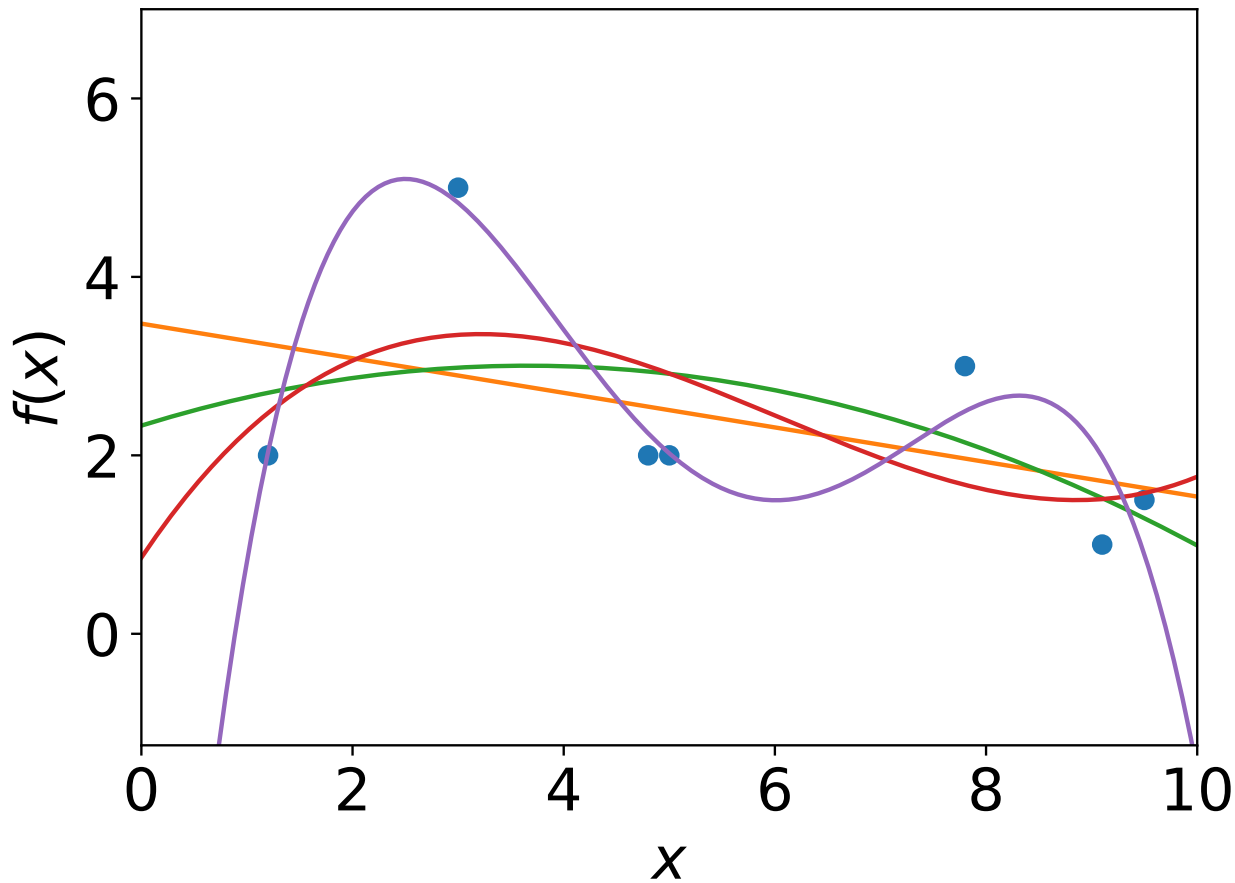
Idea: $ax^2 + bx + c$

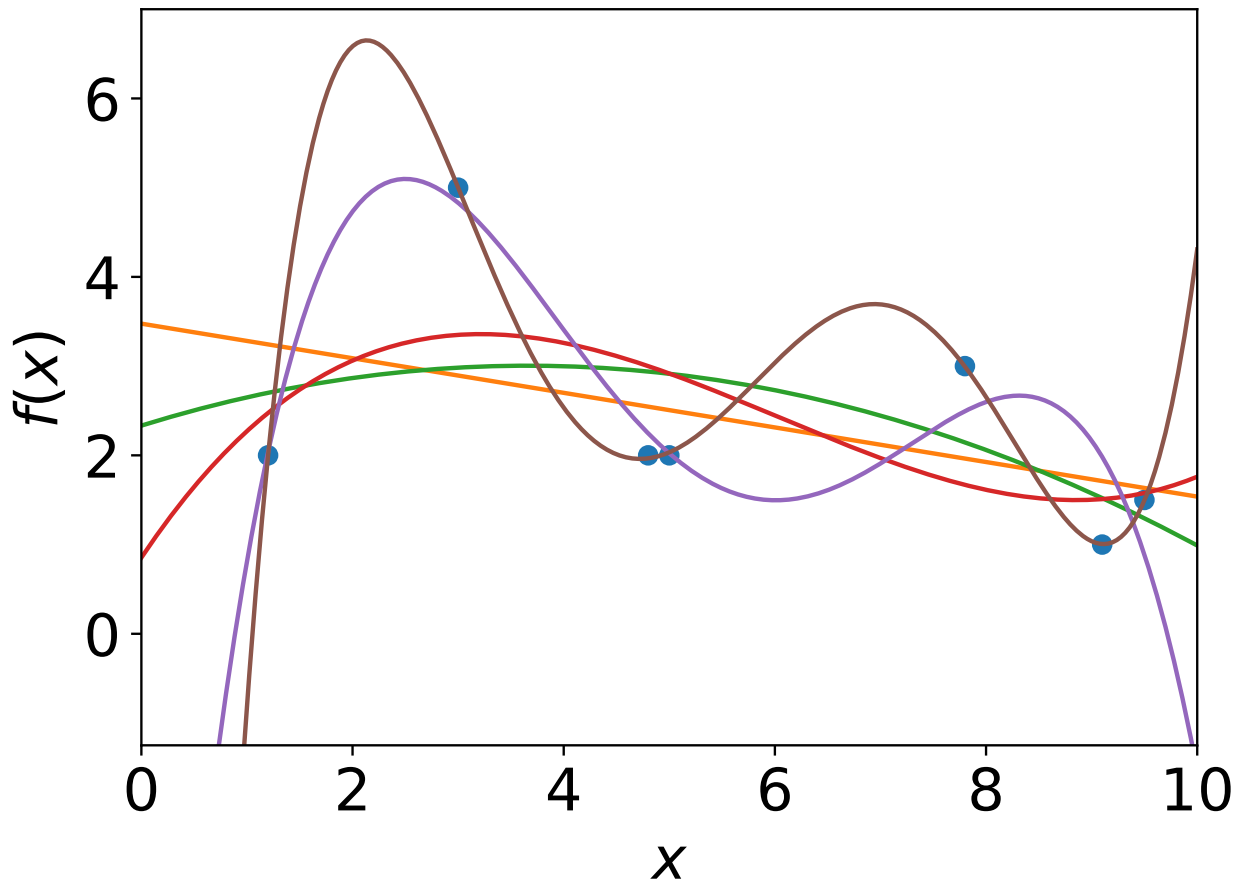

6
4
2
0
f(x)
0
2
4
6
8
10
x

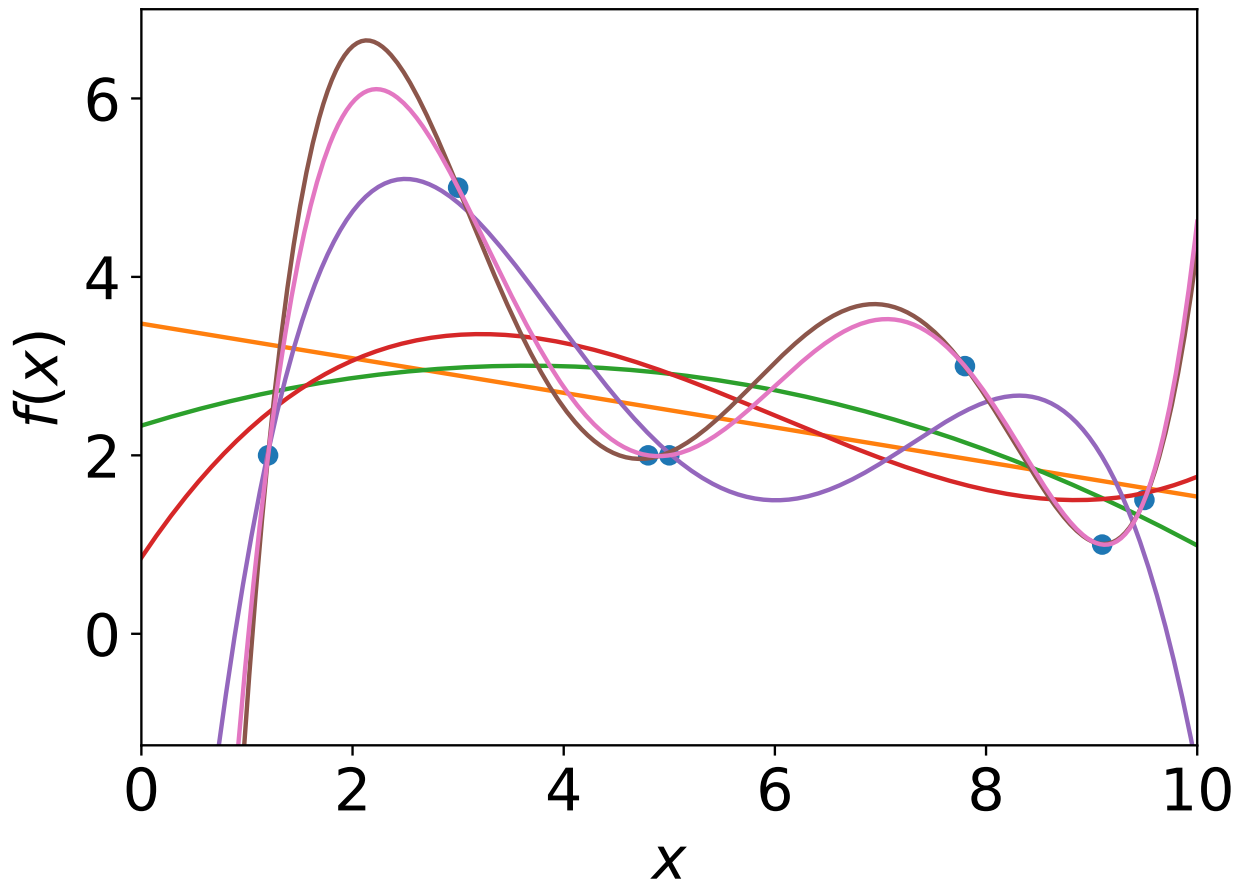

6
4
2
0
f(x)
0
2
4
6
8
10
x


6
4
2
0
f(x)
0
2
4
6
8
10
x


6
4
2
0
f(x)
0
2
4
6
8
10
x

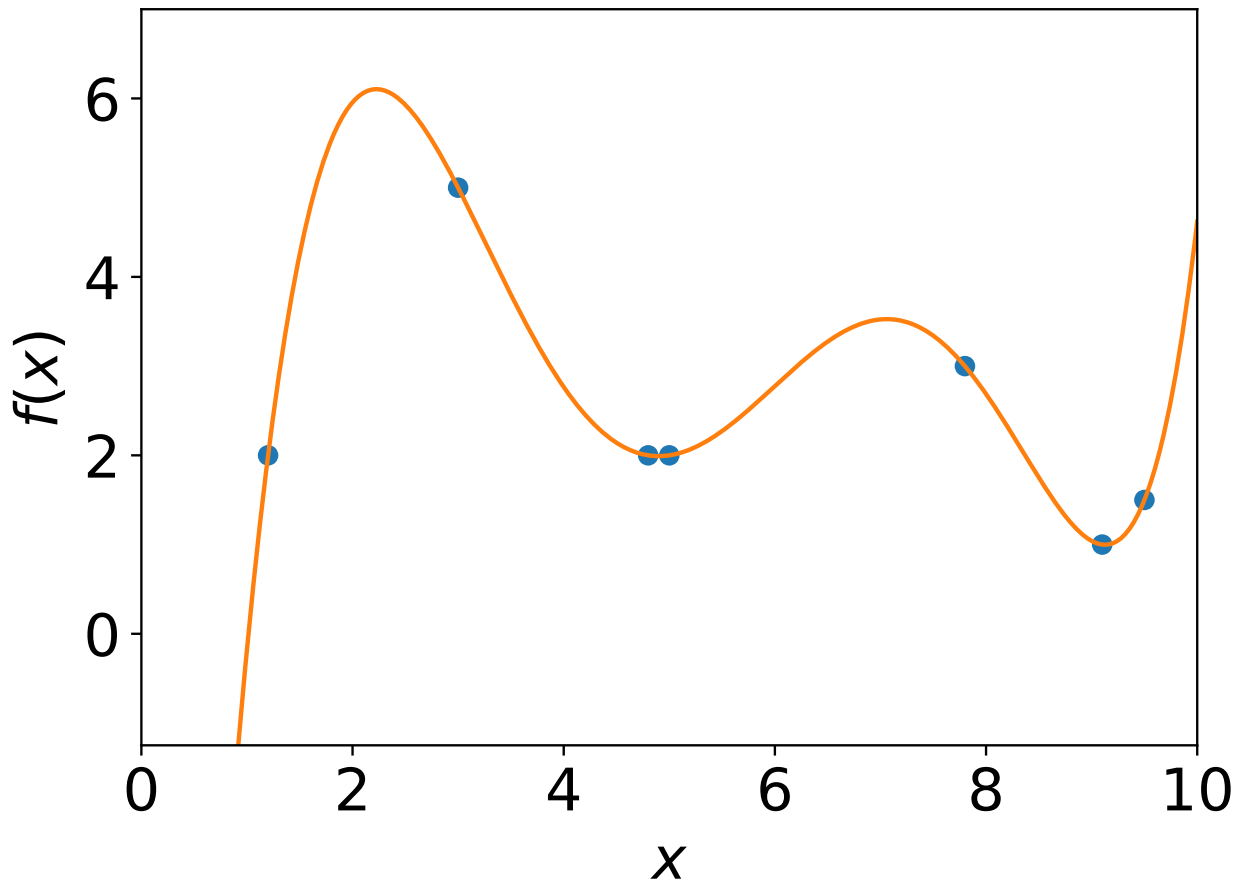

6
4
2
0
f(x)
0
2
4
6
8
10
x

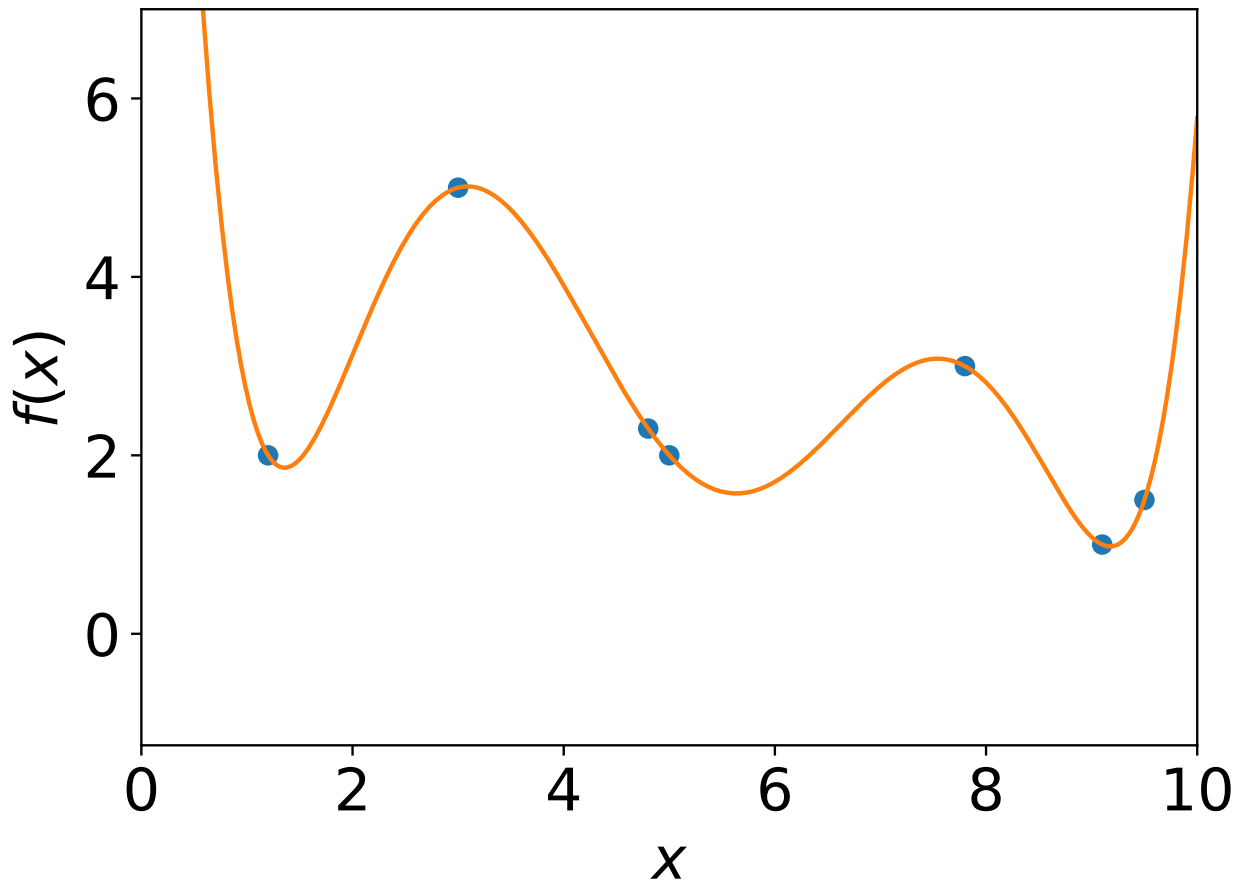

6
4
2
0
f(x)
0
2
4
6
8
10
x


6
4
2
0
f(x)
0
2
4
6
8
10
x

Stable?

Small change in input $\rightarrow$ small change in output

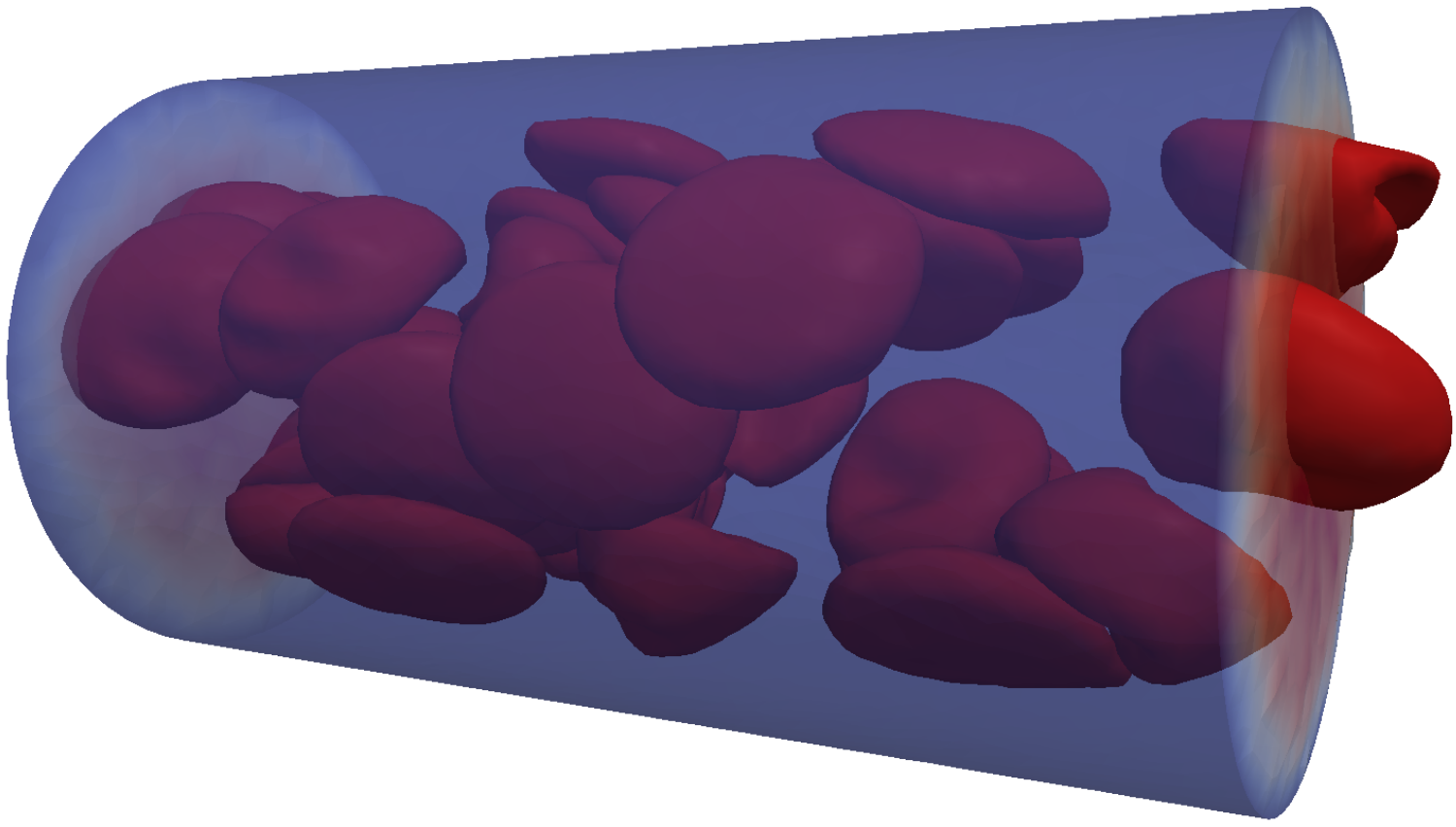

6
4
2
0
f(x)
0
2
4
6
8
10
x


6
4
2
0
f(x)
0
2
4
6
8
10
x

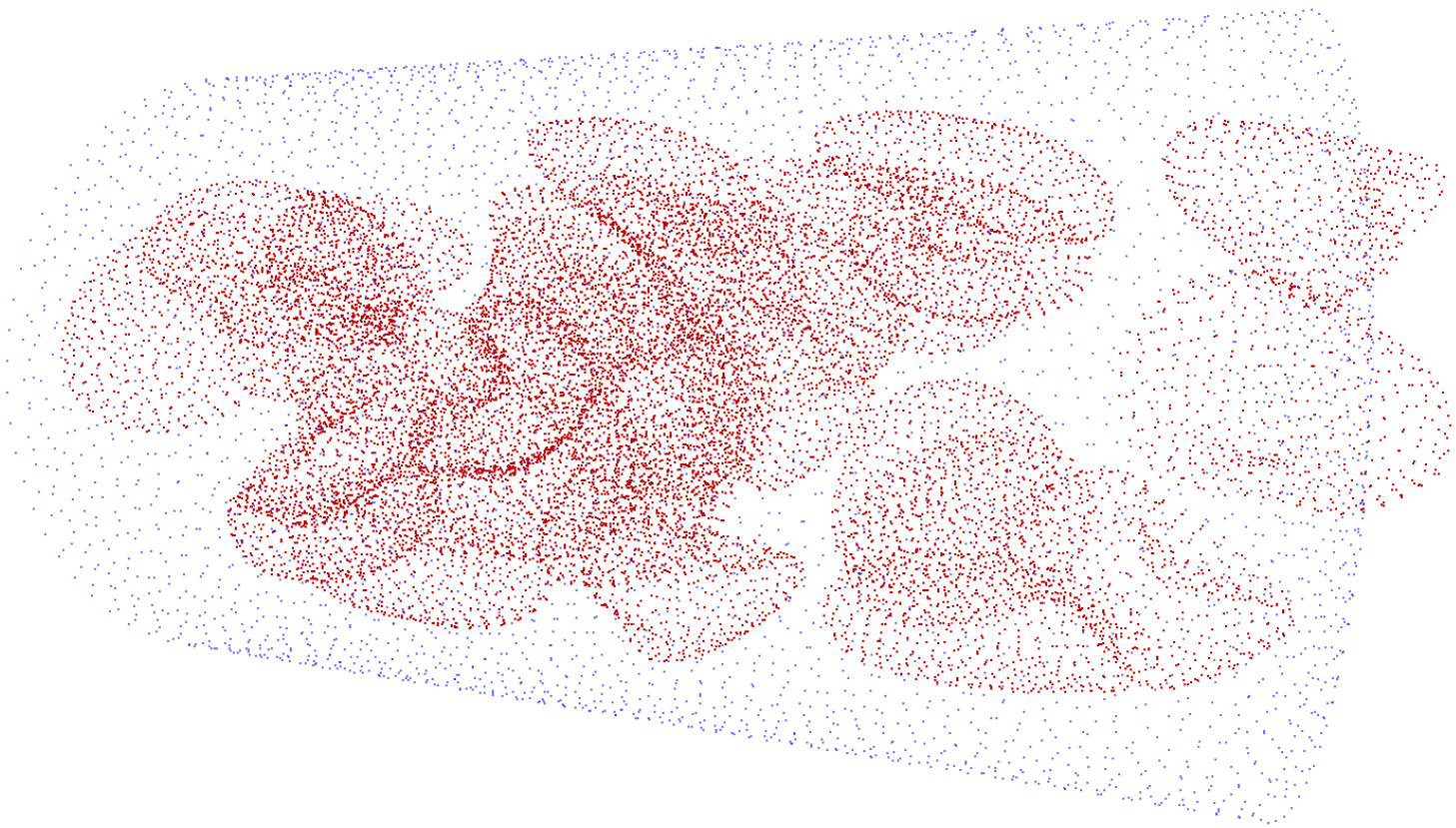
Stable + Accurate

$\Rightarrow$ error can be made arbitrarily small

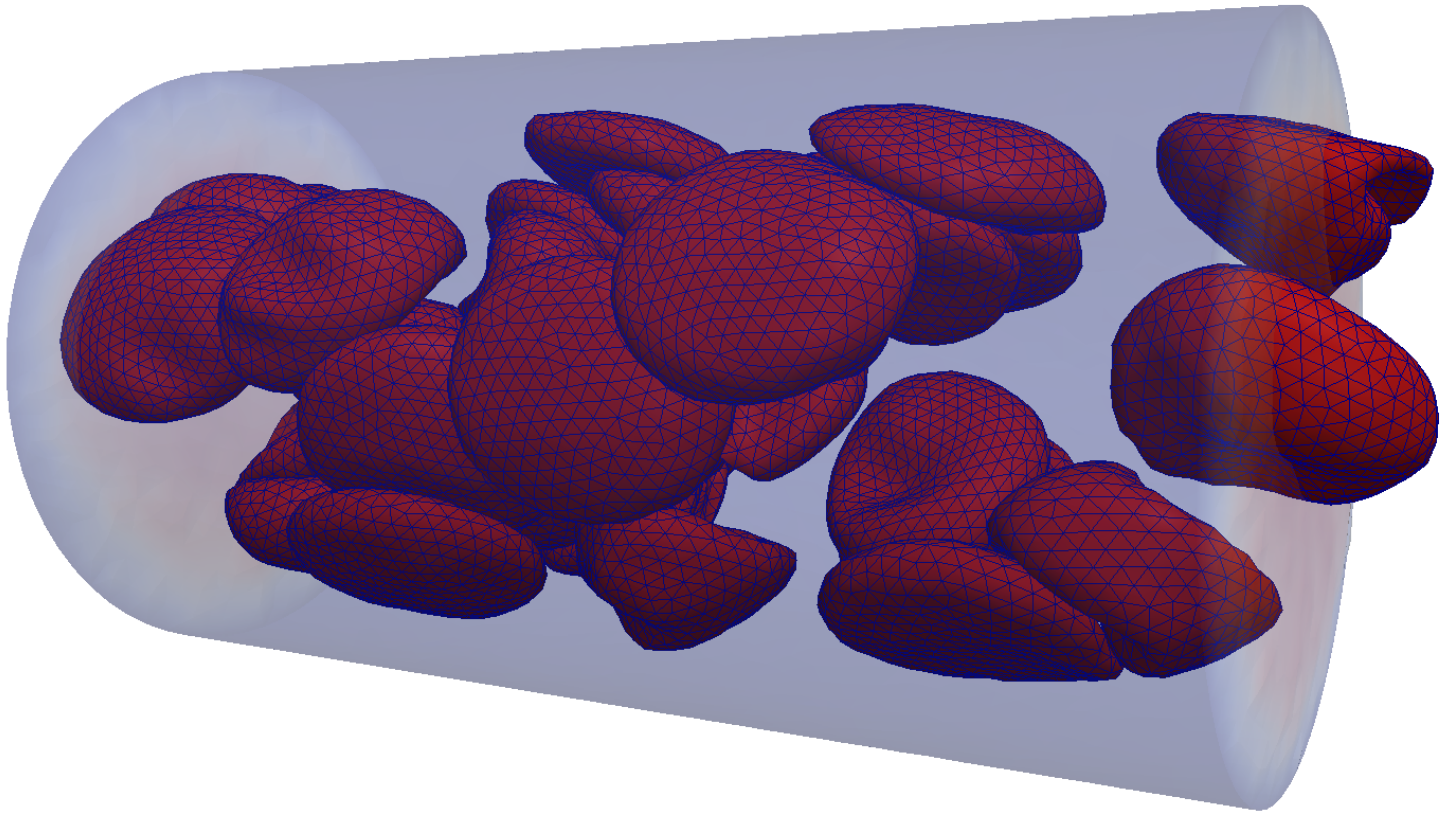
'Convergent'



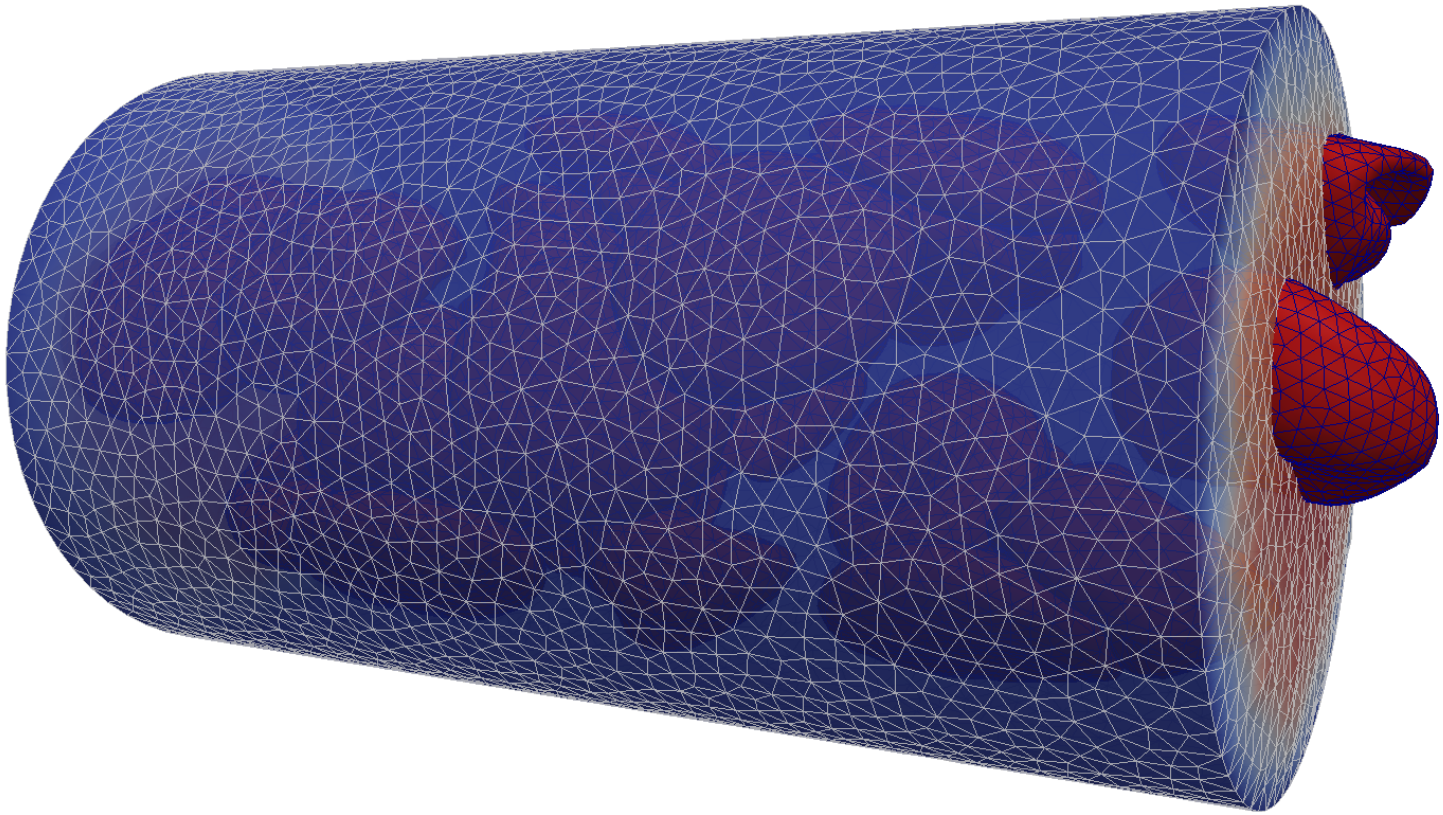
ANL, Karniadakis, Paraview



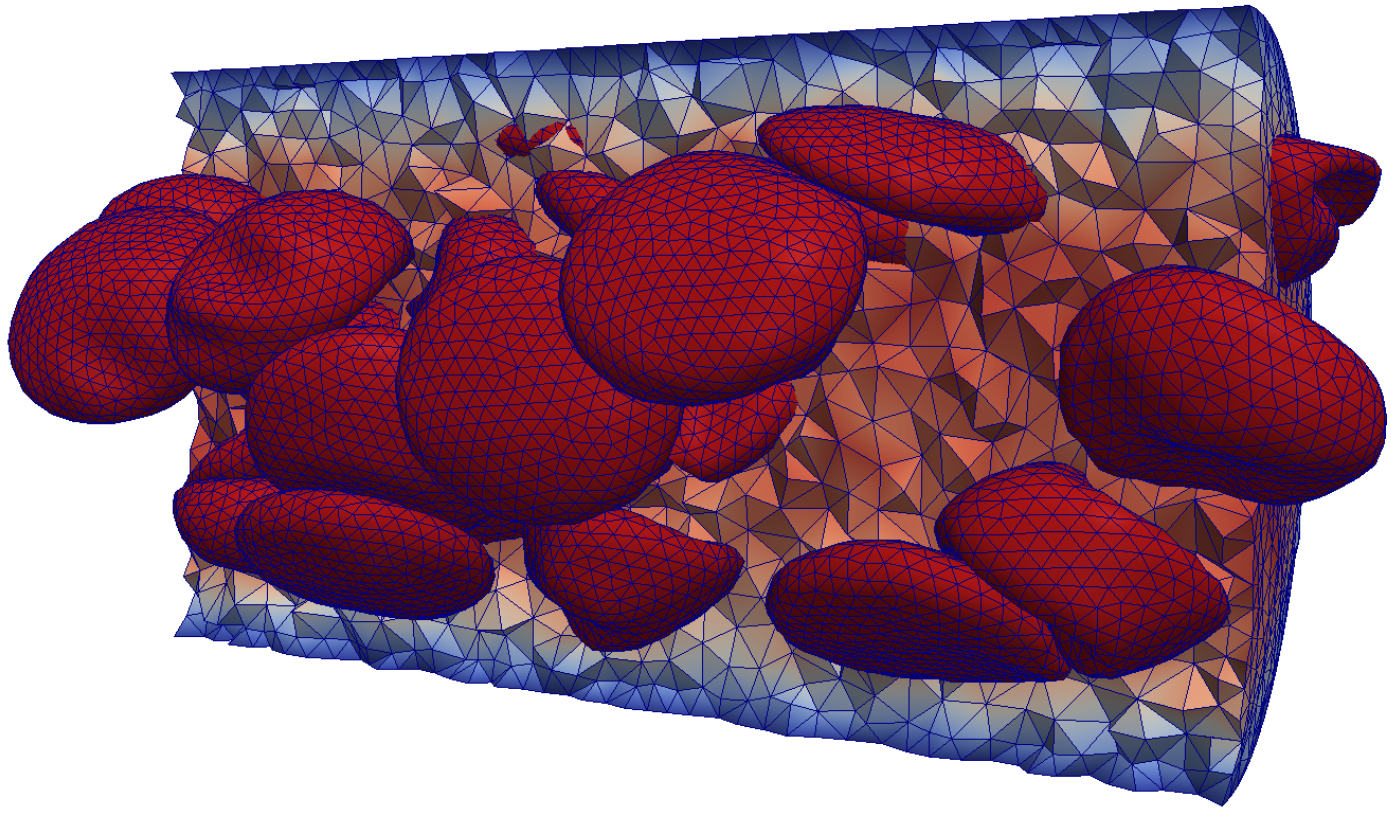
ANL, Karniadakis, Paraview



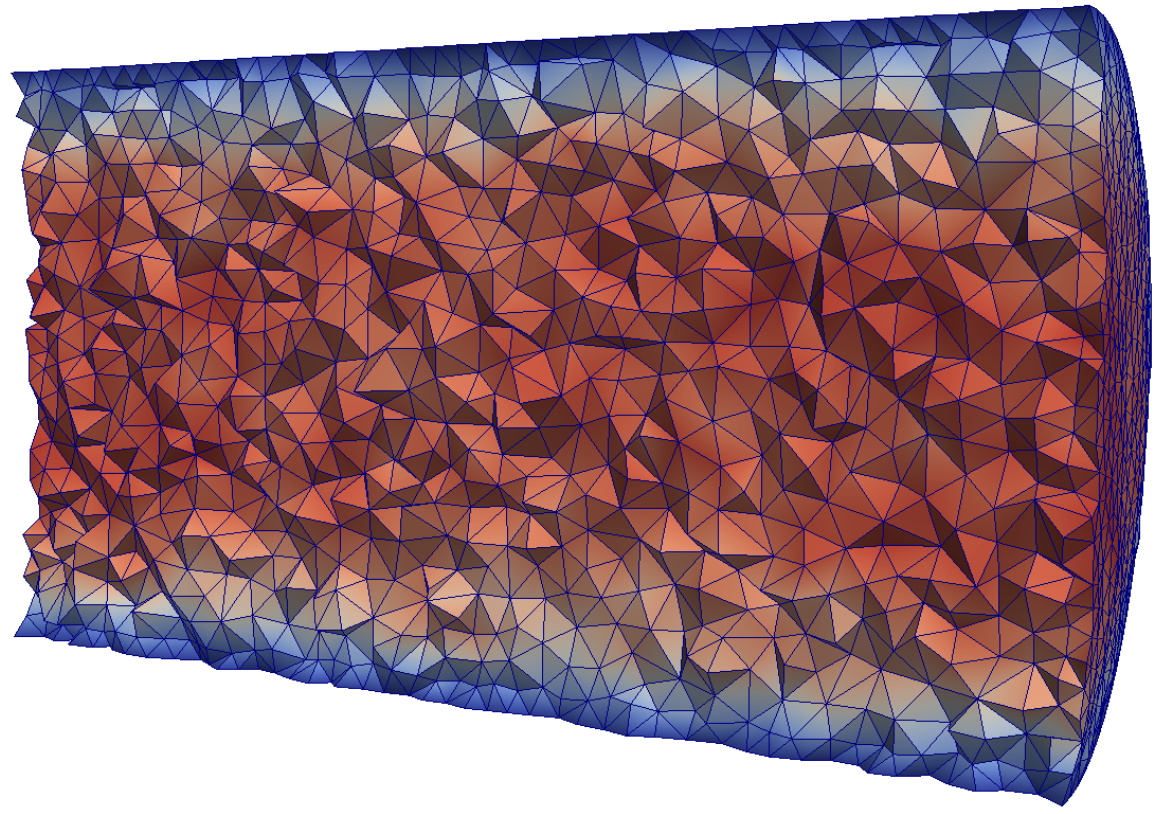
ANL, Karniadakis, Paraview



ANL, Karniadakis, Paraview



ANL, Karniadakis, Paraview



ANL, Karniadakis, Paraview

Features of **Sci. Comp.**

- Naturally interdisciplinary
- Varied work
  - Methods
  - Analysis
  - Algorithms
  - High-performance implementation
- Overlap with:
  - Engineering
  - Math
  - Algorithms
  - Parallel/HPC

Why is it interesting?

- Broadly applicable
- Provides motivation for many branches of CS
- Collaborative
- Jobs!

http://scicomp.cs.illinois.edu/


Paul Fischer


Bill Gropp

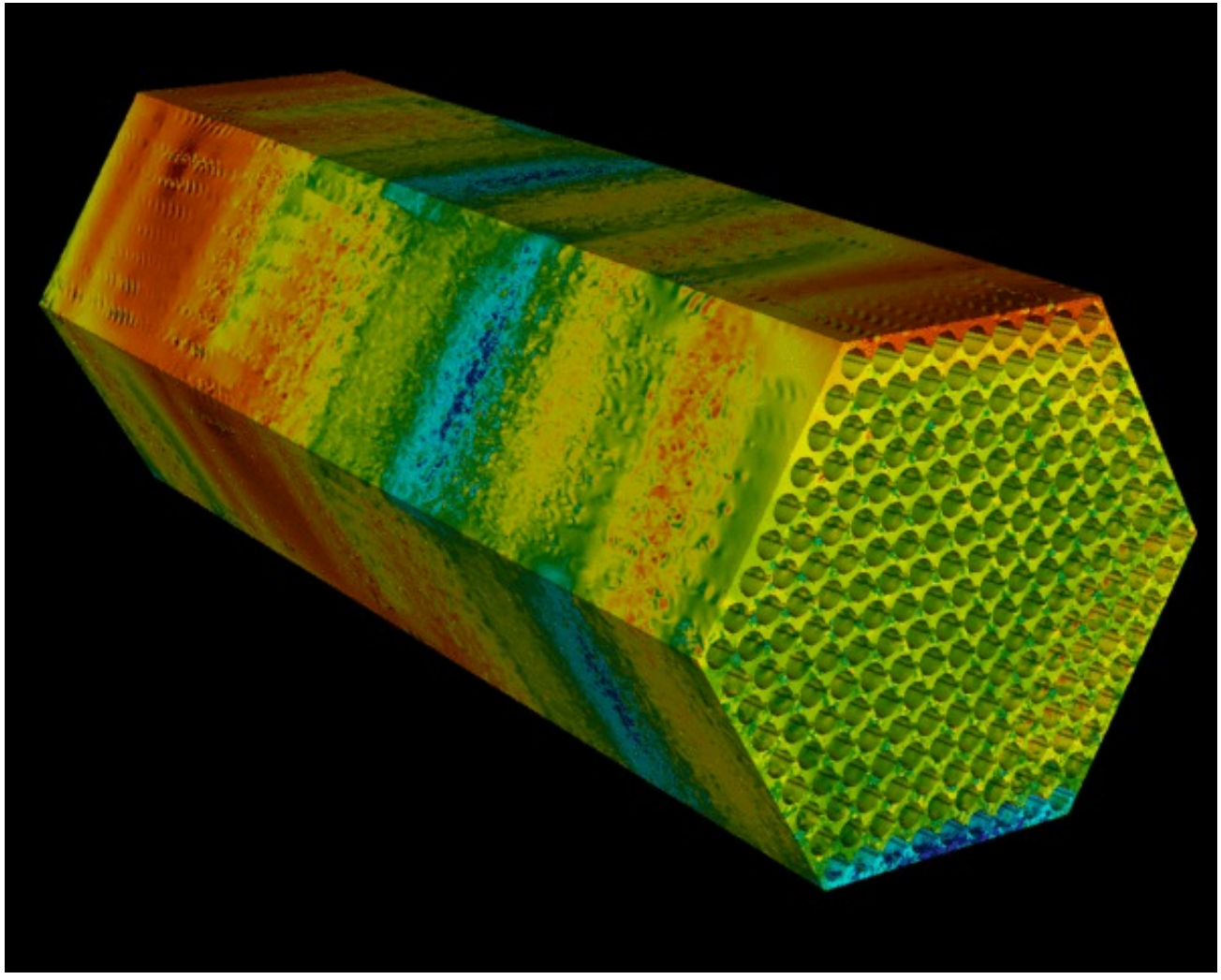

Andreas Kloeckner


Luke Olson

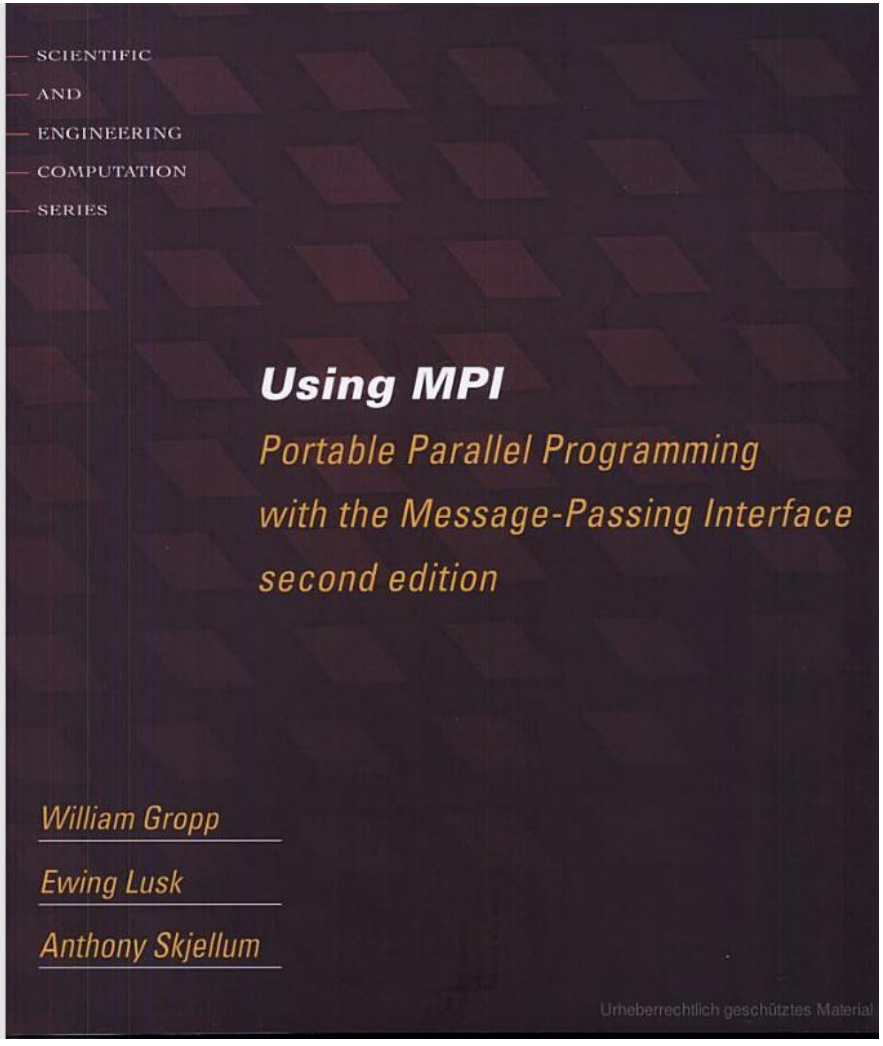

Edgar Solomonik

and about 15-20 grad students

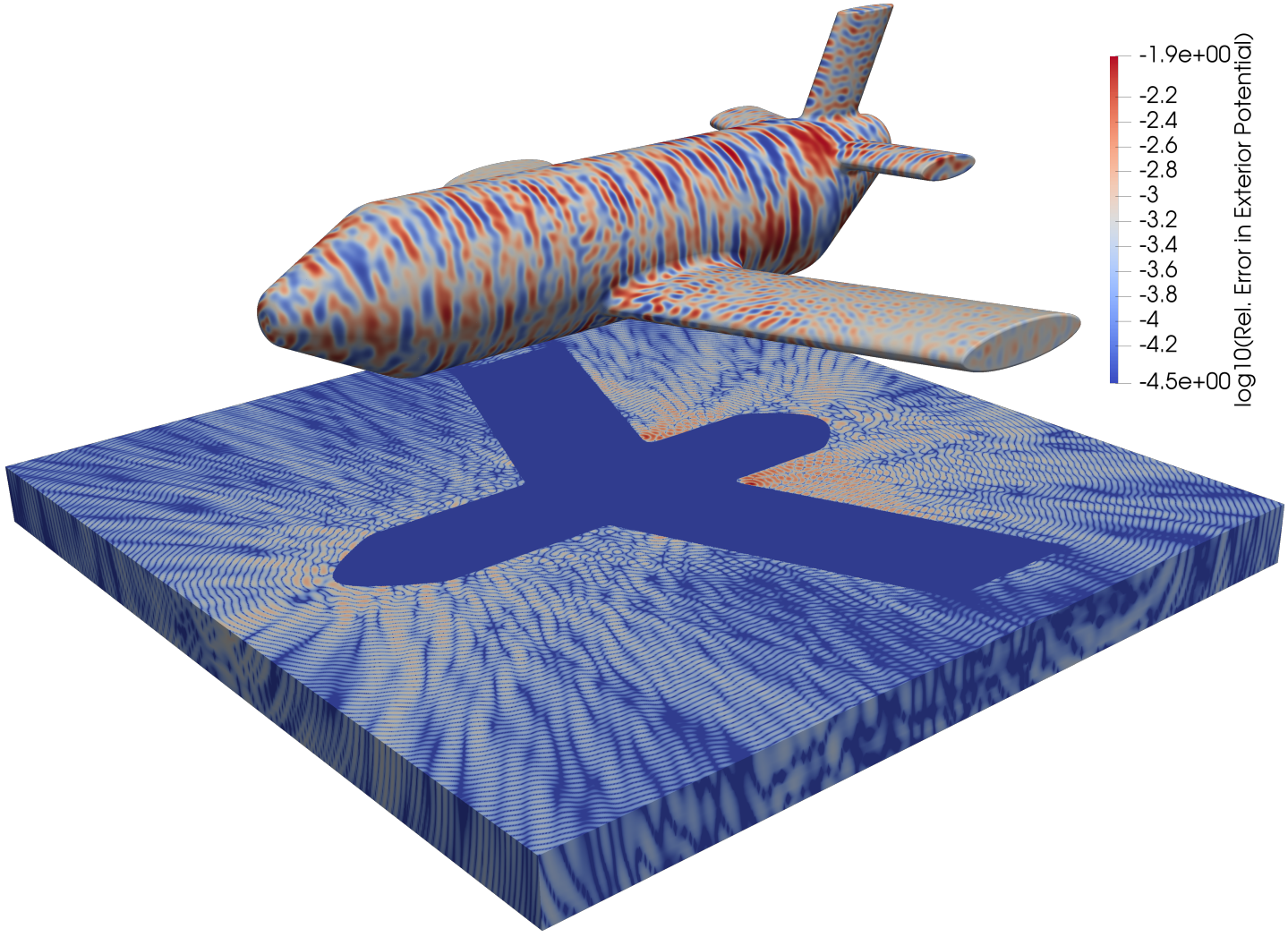
Paul Fischer



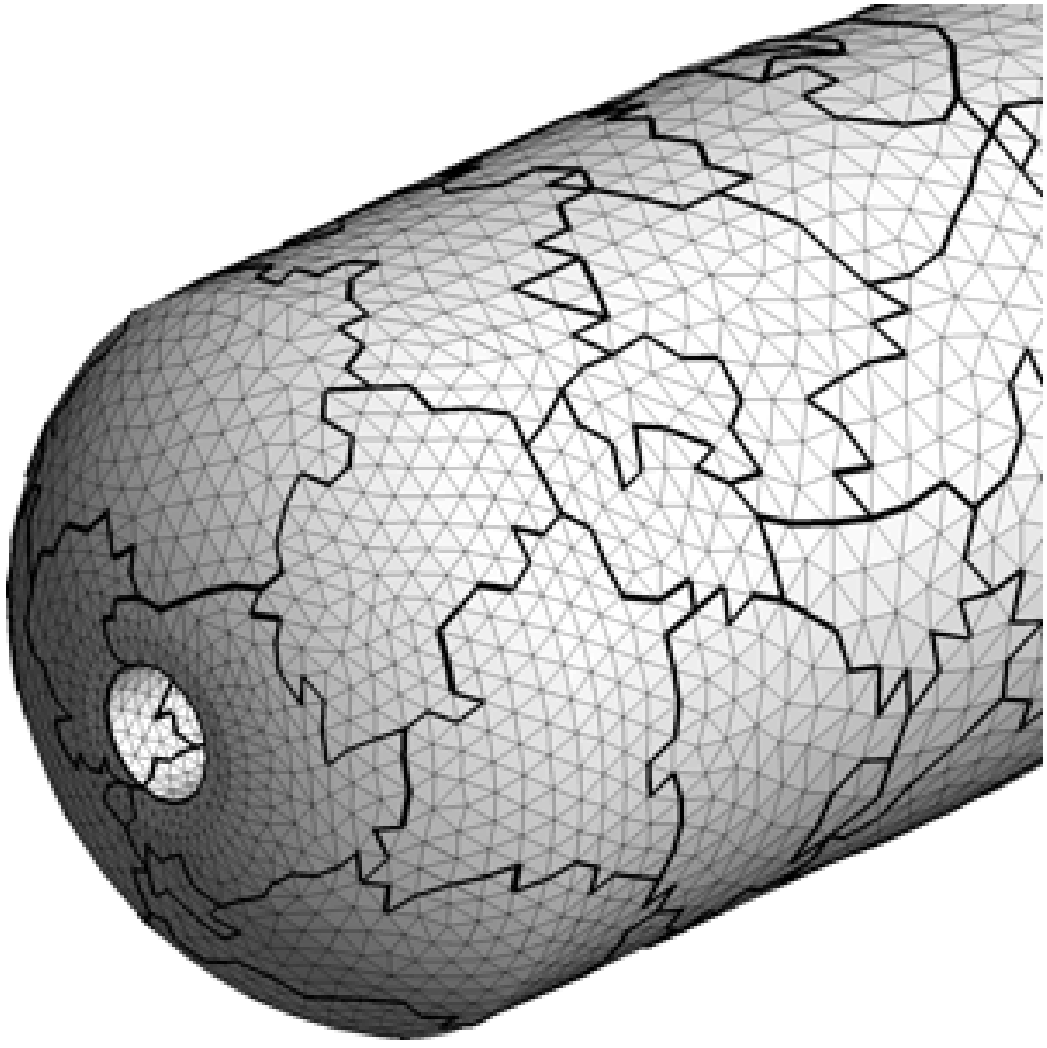
Bill Gropp


SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES
Using MPI
Portable Parallel Programming
with the Message-Passing Interface
second edition
William Gropp
Ewing Lusk
Anthony Skjellum
Urheberrechtlich geschütztes Material

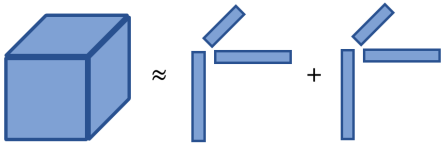
Andreas Kloeckner

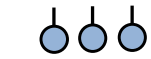


Luke Olson

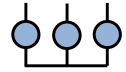


Edgar Solomonik

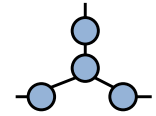




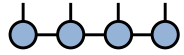
Kronecker product



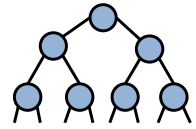
canonical polyadic (CP)



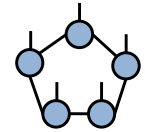
Tucker



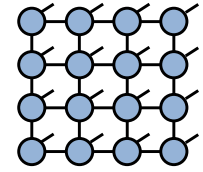
matrix product state (MPS)



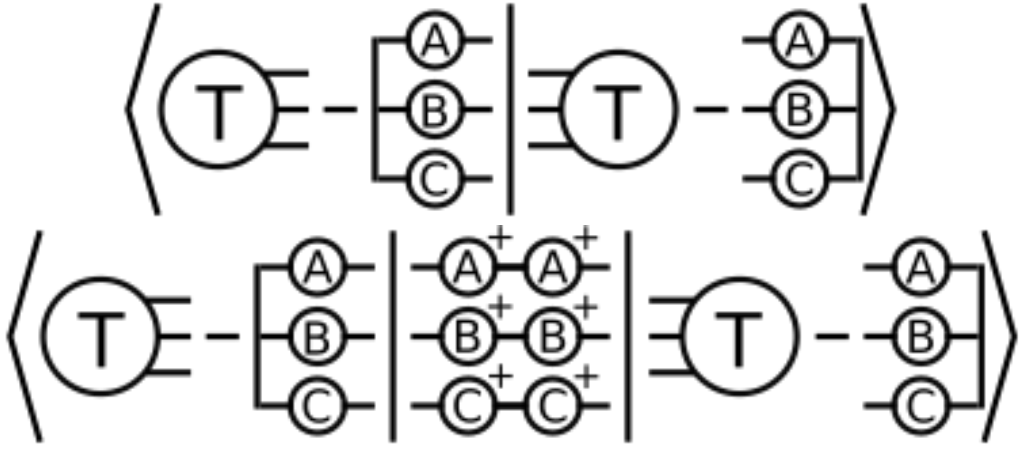
hierarchical Tucker



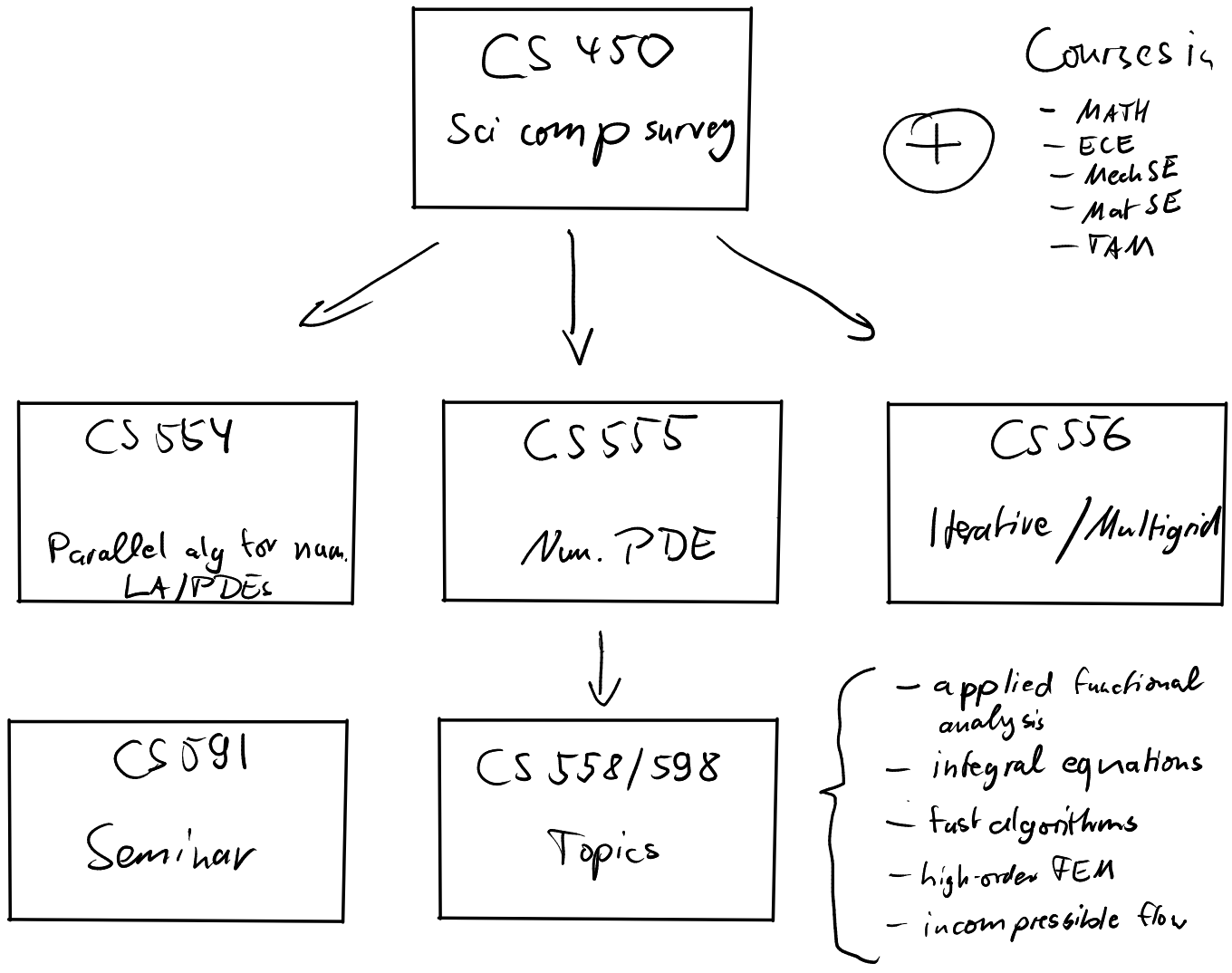
tensor ring



projected entangled pair states (PEPS)



Classes


CS 450
Sci comp survey
Courses in
- MATH
- ECE
- MechSE
- MatSE
- TAM
CS 554
Parallel alg for num. LA/PDEs
CS 555
Num. PDE
CS 556
Iterative / Multigrid
CS 591
Seminar
CS 558/598
Topics
- applied functional analysis
- integral equations
- fast algorithms
- high-order FEM
- incompressible flow

Qualifying Exam

- Format: 90 minute oral exam on a survey of topics in numerical analysis and scientific computing
- High level prep:
  - CS450 essential
  - CS555, CS556 help round out advanced numerical methods
  - Some MATH course (real analysis) and HPC (like CS484)
- Advice: take CS450, set up *mock* qualifying exams with students in the area
- General topics: errors and floating point, linear systems, eigenvalue problems, root finding and optimization, linear/nonlinear least-squares, integration/differentiation/interpolation, numerical ODEs and PDEs.

How to get involved?

- Check out/visit our courses; talk to the faculty
  `http://scicomp.cs.illinois.edu/`
- SIAM student chapter
- CS591MH Sci comp seminar (Wed 11am)
- Scicomp mailing list;
  `https://lists.illinois.edu/lists/info/scicomp`
- Computational Science and Engineering Concentration
  `https://cse.illinois.edu/`
  - 2 core courses (e.g. CS450) and 2 electives (very broadly "computational science")
  - Central place for seminars, news
  - Hands-on workshops on Python, git, scientific workflows, etc

http://scicomp.cs.illinois.edu/


Paul Fischer


Bill Gropp


Andreas Kloeckner


Luke Olson


Edgar Solomonik