Numerical Computing without Discretization Woes

Alex Townsend
Cornell University
townsend@cornell.edu

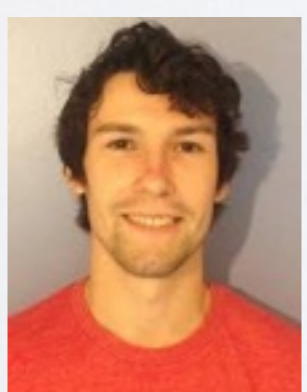
…give a ***visionary*** computational science talk…

Ben Zhang & Ricardo Baptista

Discretization alleviating colleagues:


Dan Fortunato

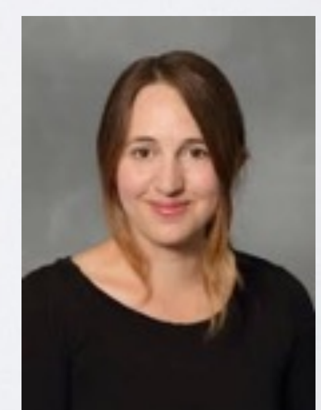

Marc Gilles

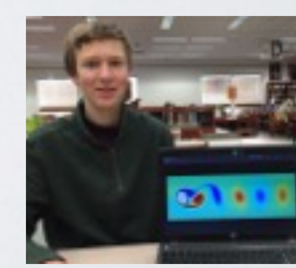

Andrew Horning


Sheehan Olver


Nick Trefethen

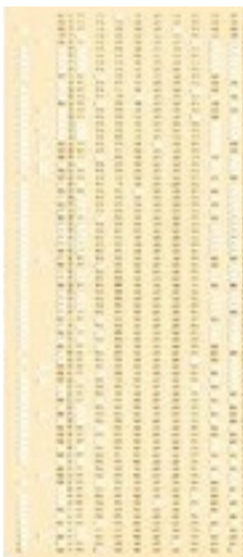

Grady Wright


Heather Wilber


Aaron Yeiser

Computing trends

Punchcards



DOS

Windows95

Device oblivious interaction

1940–1960

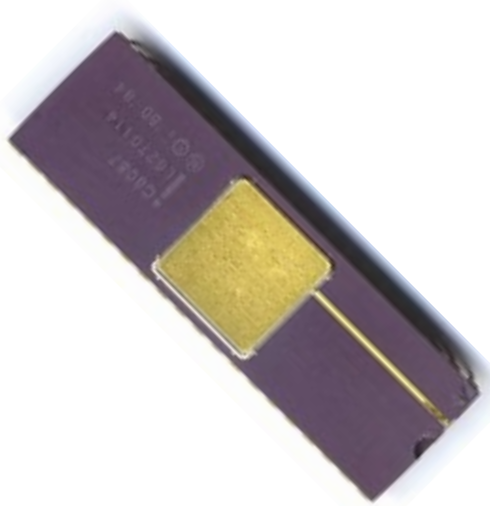
1980

2000

2050

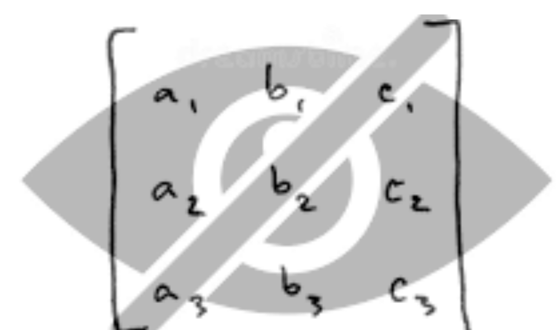
Tabulations

IEEE 754



Software

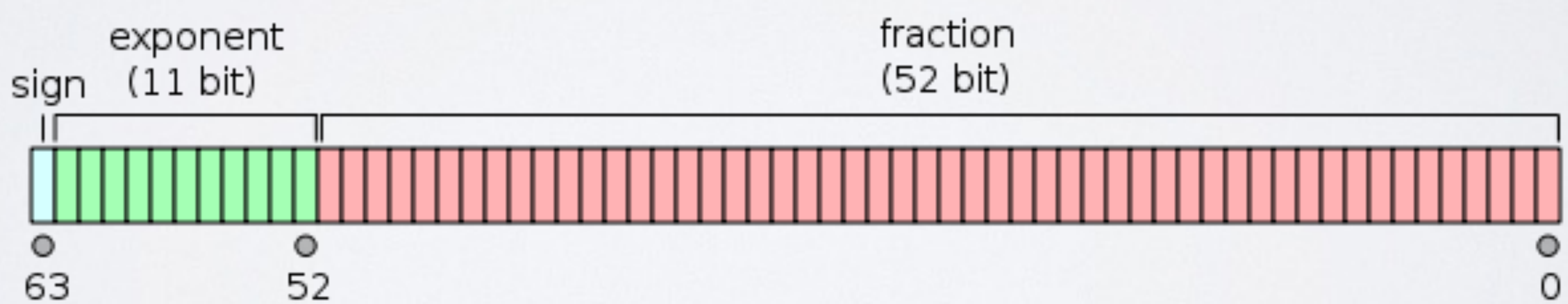
Discretization oblivious software



Floating point arithmetic

Prediction 1:

Floating point arithmetic will remain important.

$x \approx$ 

The IEEE 754-1984 format

William Kahan

"An adult's version of significant digits."

$$\pm\text{significant} \times \text{base}^{\text{exponent}}$$

Example: $x = 100\pi = 314.159265359$

$x \approx \hat{x} = 3.14 \times 10^2$

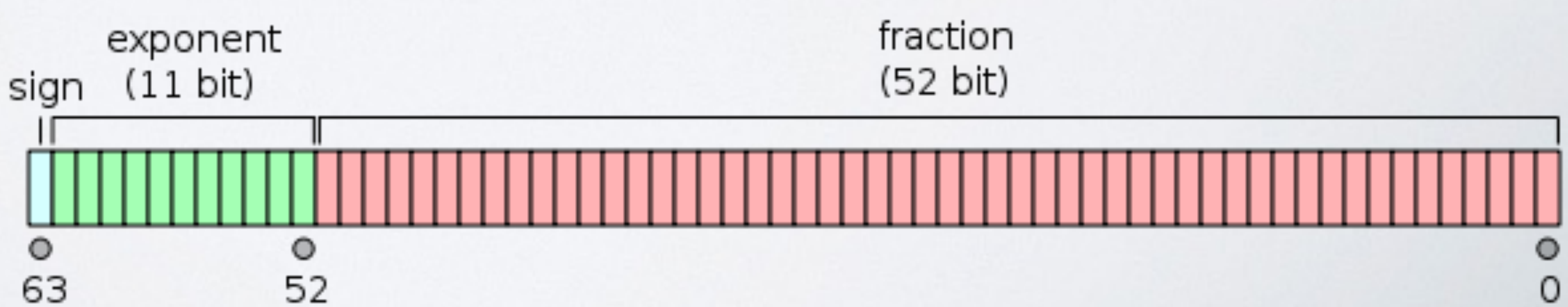
$\hat{x}^2 = 98596 \approx \widehat{98596} = 9.86 \times 10^2$

$\widehat{x^2} = 9.87 \times 10^4$

Round after every operation.

Floating point is similar, but in base 2:

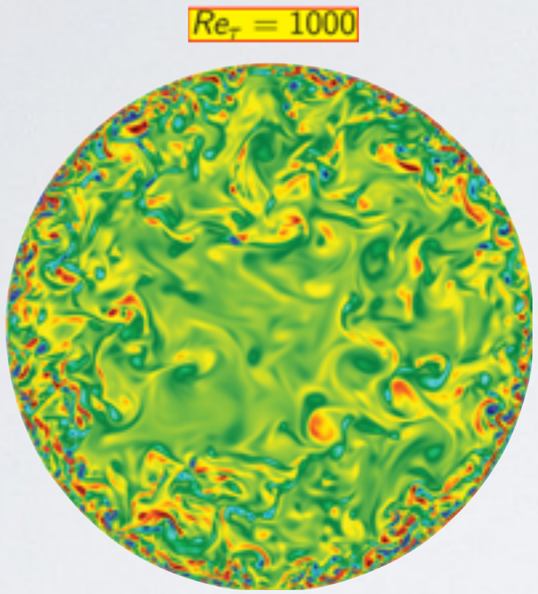
64-bit IEEE 784 float:



Floating point numbers

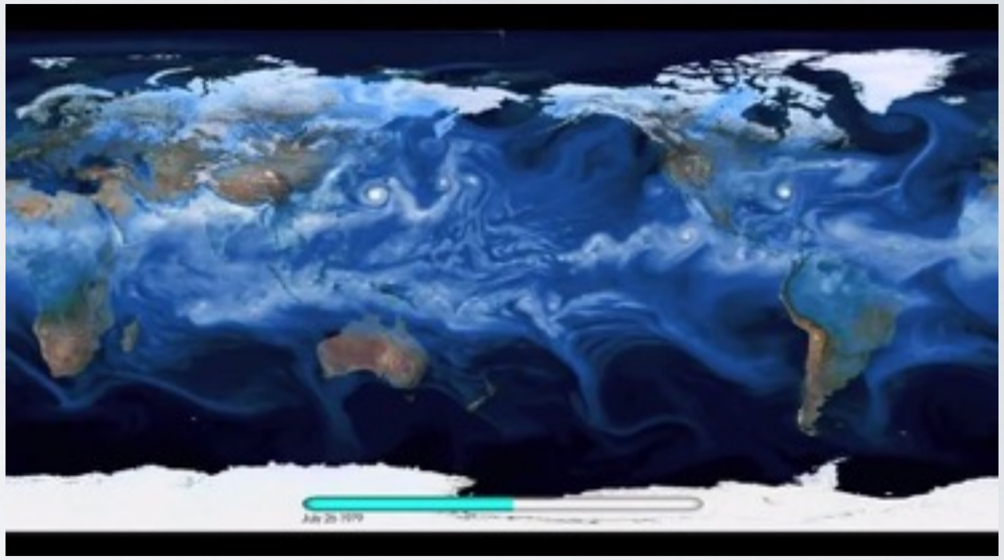
Most of time, we do not worry about floating point arithmetic.

Pipeflow:



nek5000

Weather:



Lawrence Berkeley National Laboratory, 2014

Except… in the rare cases when it matters…

How high can a computer count?

for $k = 1, 2, \ldots,$

  print k

end

```
>> (2^53+1)-2^53
ans =
     0
```

PBS infinite series

Kelsey Houston-Edwards

Numerical computing without discretization woes

(1) Functions

chebfun

RKToolbox

ApproxFun

(2) Differential equations

FEniCS Project

Dedalus Project

(3) Geometry

gmsh

OpenFOAM

Triangle

Goal: Develop discretization oblivious software for (1)-(3).

Functions

Prediction 2:

Computing with functions will be more adaptive.

$$f(x)$$

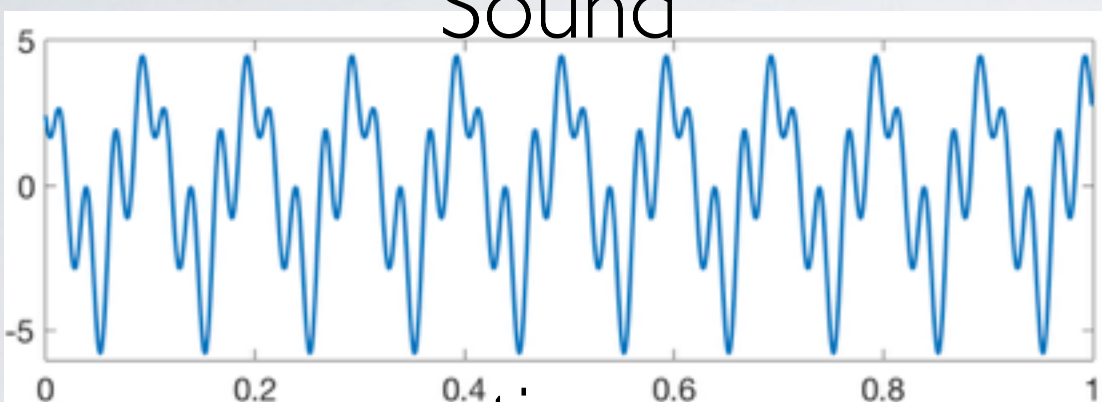
Adaptively computing with functions

A simple example:

sound

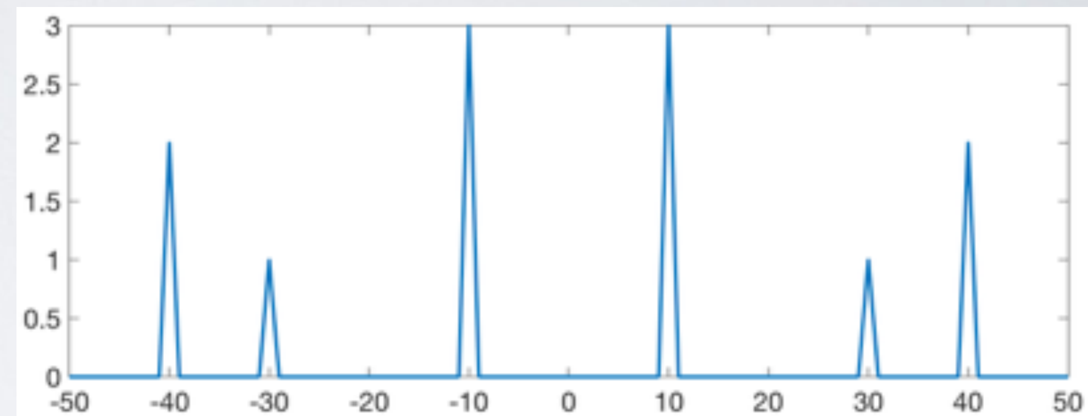
|Frequencies|

Sound



time

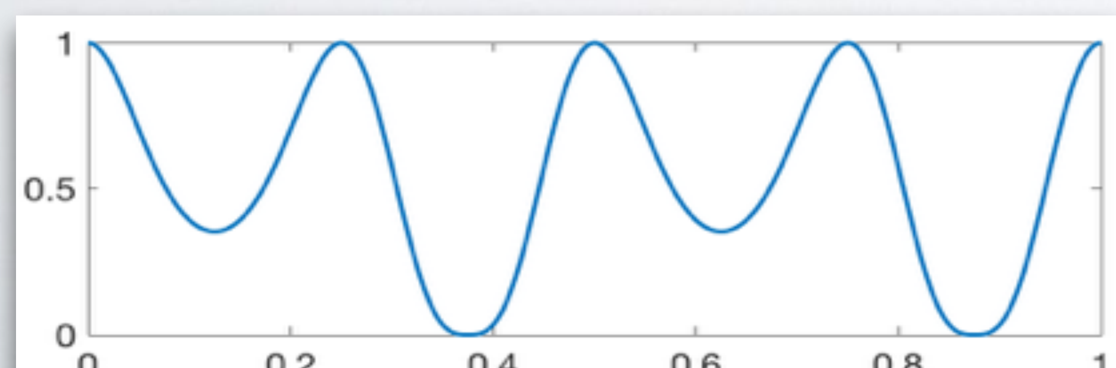
FFT



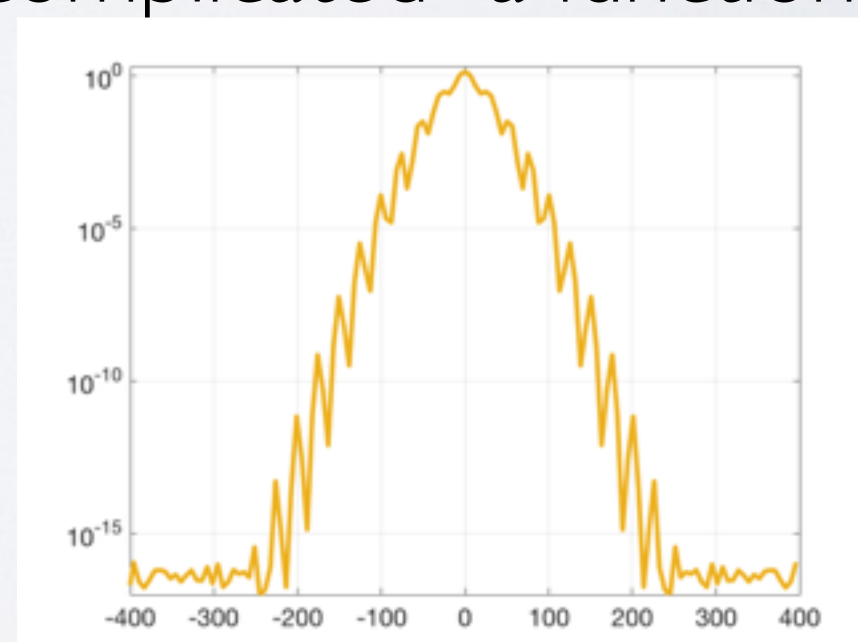
Hz

$$\mathrm{sound}(t) = 3\cos(2\pi 10t + 0.2) + \cos(2\pi 30t - 0.3) + 2\cos(2\pi 40t + 2.4)$$

An automatic way to tell us how "complicated" a function is.



FFT

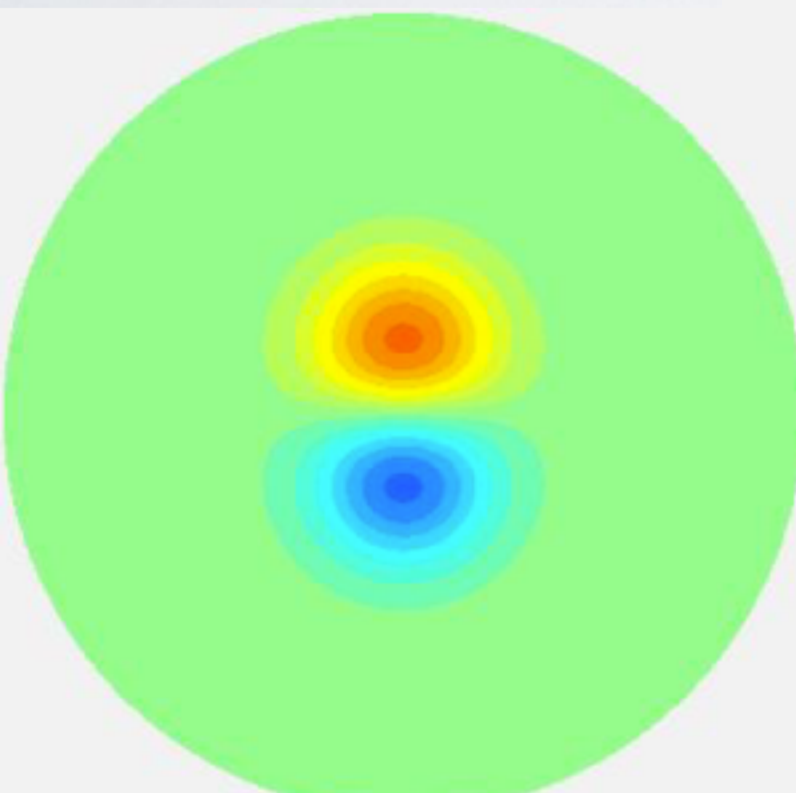


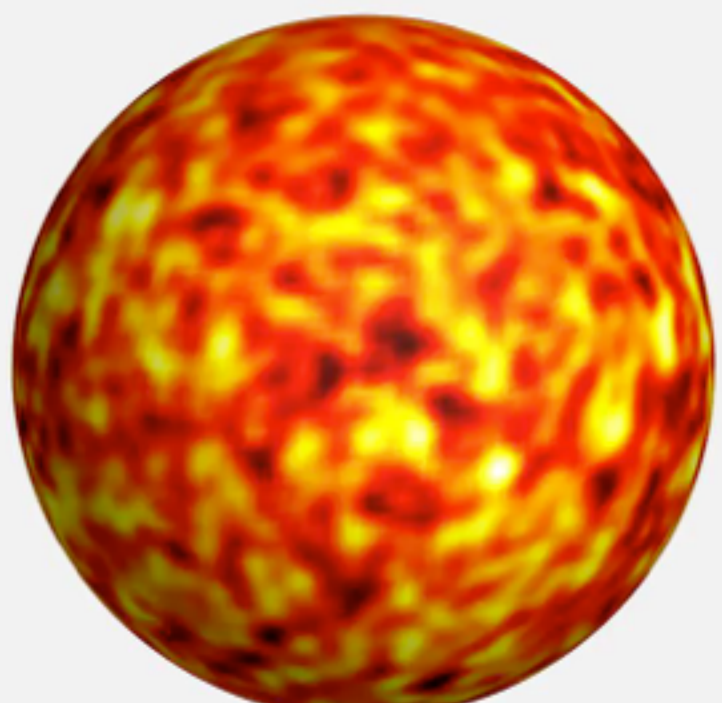
Computing with functions without discretization woes

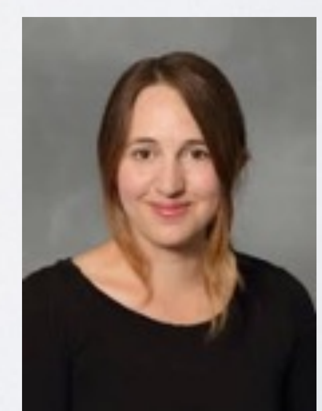
```
>> f = chebfun(@(x) abs(4*cos(3*pi*x))./(x-.2));
   chebfun column (8 smooth pieces)
       interval         length      endpoint values    endpoint exponents
[       -1,    -0.83]       13        -3.3 -7.6e-16          [0        0]
[    -0.83,     -0.5]       16    -4.8e-15 -1.6e-14          [0        0]
[     -0.5,    -0.17]       20     1.3e-14    6e-16          [0        0]
[    -0.17,     0.17]       59     2.9e-14 -2.6e-14          [0        0]
[     0.17,      0.2]        9    -2.6e-13     -Inf          [0       -1]
[      0.2,      0.5]       15         Inf -4.8e-14          [-1       0]
[      0.5,     0.83]       21     6.3e-14  1.4e-14          [0        0]
[     0.83,        1]       13    -1.3e-14        5          [0        0]
vertical scale = Inf    Total length = 166
```

Rodrigo Platte

Ricardo Pachon







Heather Wilber

Grady Wright

Differential equations

Prediction 3:

Differential equations may not be discretized by matrices.

$$\mathcal{L}u = f$$

Discretize-then-solve

"Discretize-then-solve"

Linear, elliptic, 2nd order, PDE

$$\mathcal{L}u = f \text{ on } \Omega, \quad u|_{\partial\Omega} = g$$

$\downarrow$

Matrix

$$Ax = b$$

$\downarrow$

Discrete solution

$$x$$

$\downarrow$

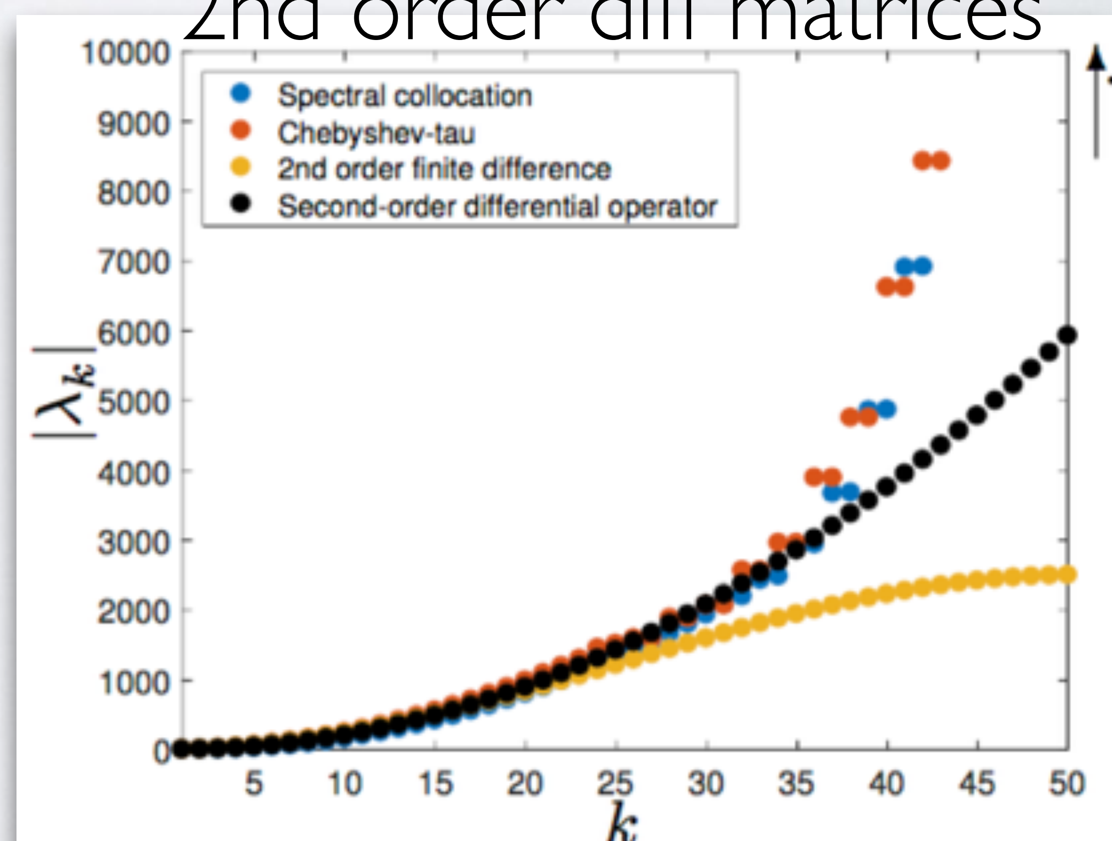
Approx. solution

$$\hat{u}$$

Finite diff & pseudospectral:

- $\mathcal{L}$ is an unbounded operator
- A is a bounded operator
- Want A to be well-conditioned
- Want A to capture $\mathcal{L}$

2nd order diff matrices



Revisiting Krylov subspace

Task: Solve $Ax = b$ for x

A is large
fast mat-vecs

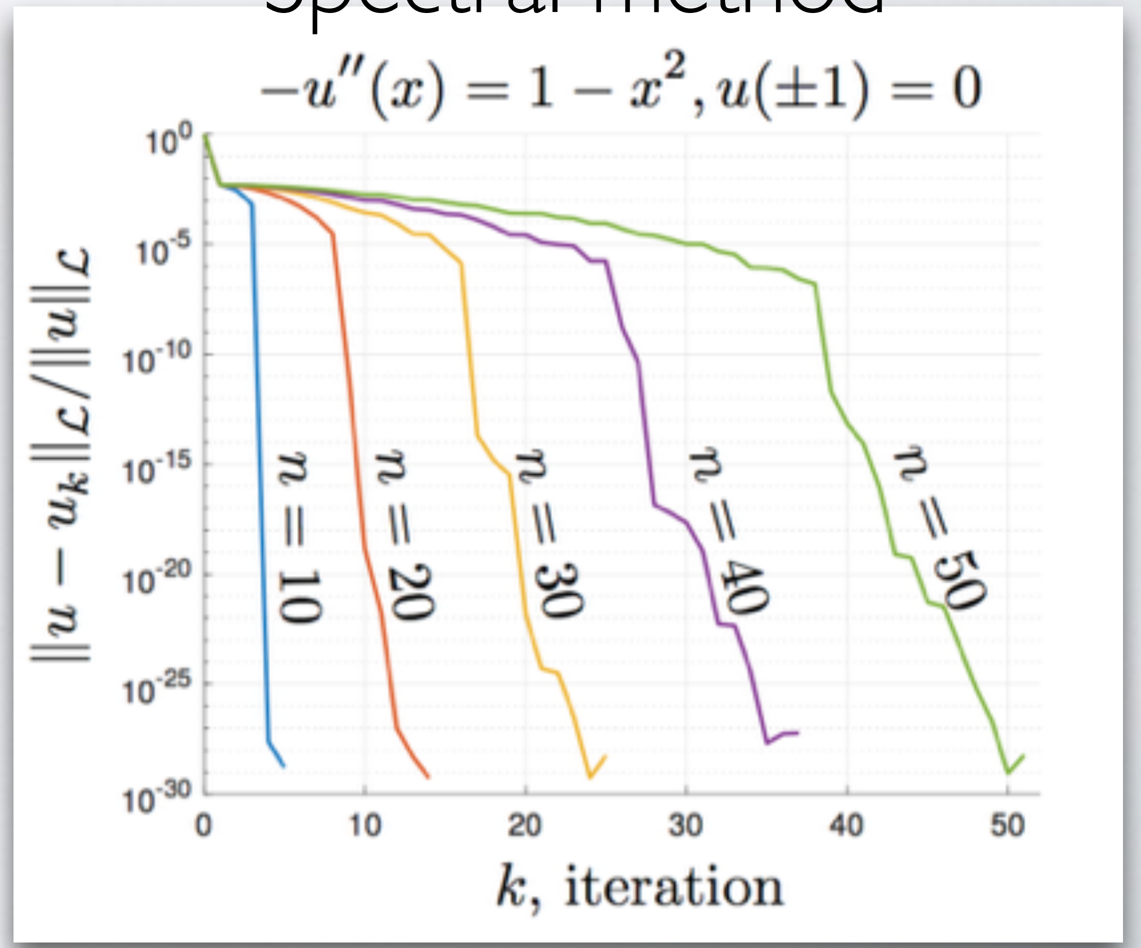
Krylov subspace methods are iterative solvers, computing iterates from:

$$\mathcal{K}_k(A, b) = \mathrm{Span}\left\{b, Ab, A^2b, \ldots, A^{k-1}b\right\}$$

- Preconditioning $PAx = Pb$ depends on both the discretization and PDE.

- If n is inadequate to resolve solution, then one needs to rediscretize the PDE.

Spectral method



Operator analogue of Krylov methods

Task: Solve $\mathcal{L}u = f$ for u

$\mathcal{L}$ is 2nd order, elliptic

$$\mathcal{K}_k(\mathcal{L}, b) = \text{Span}\left\{f, \mathcal{L}f, \mathcal{L}^2 f, \ldots, \mathcal{L}^{k-1} f\right\}$$

Example: $-u_{xx} = 1 - x^2 \quad u(\pm 1) = 0$

$$\mathcal{K}_k(\mathcal{L}, f) = \text{Span}\left\{1 - x^2, 2\right\}$$

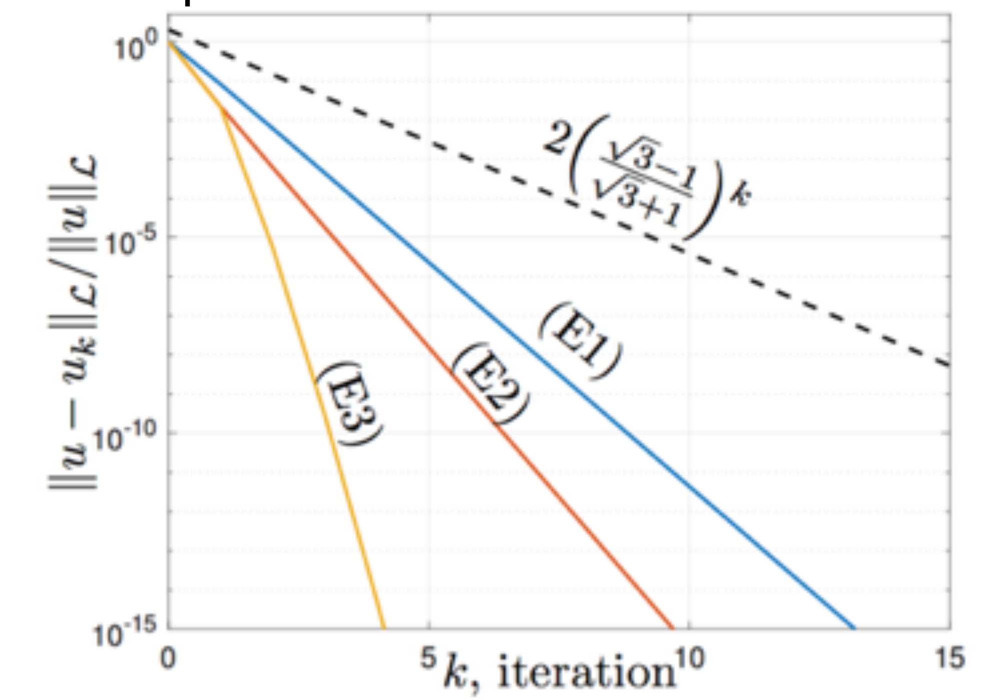
Does not contain the solution.

Modify the diff. operator: $\mathcal{T} = \Pi^* \mathcal{R}^* \mathcal{L} \mathcal{R} \Pi$

$\Pi =$ projection operator, imposes bc

$\mathcal{R} =$ operator preconditioner

Operator CG method



Marc Gilles

Jörg Liesen

Josef Málek

Zdenek Strakoš

[Málek & Strakoš, 2014] [Gilles & T., 2018]

Differential eigenproblems

"Discretize-then-solve"

Differential eigenvalue problem

$$\mathcal{L}u = \lambda u \text{ on } \Omega \qquad u|_{\partial\Omega} = 0$$

$\downarrow$

Eigenvalue problem

$$Ax = \hat{\lambda}x$$

$\downarrow$

Eigenvalues, eigenvectors

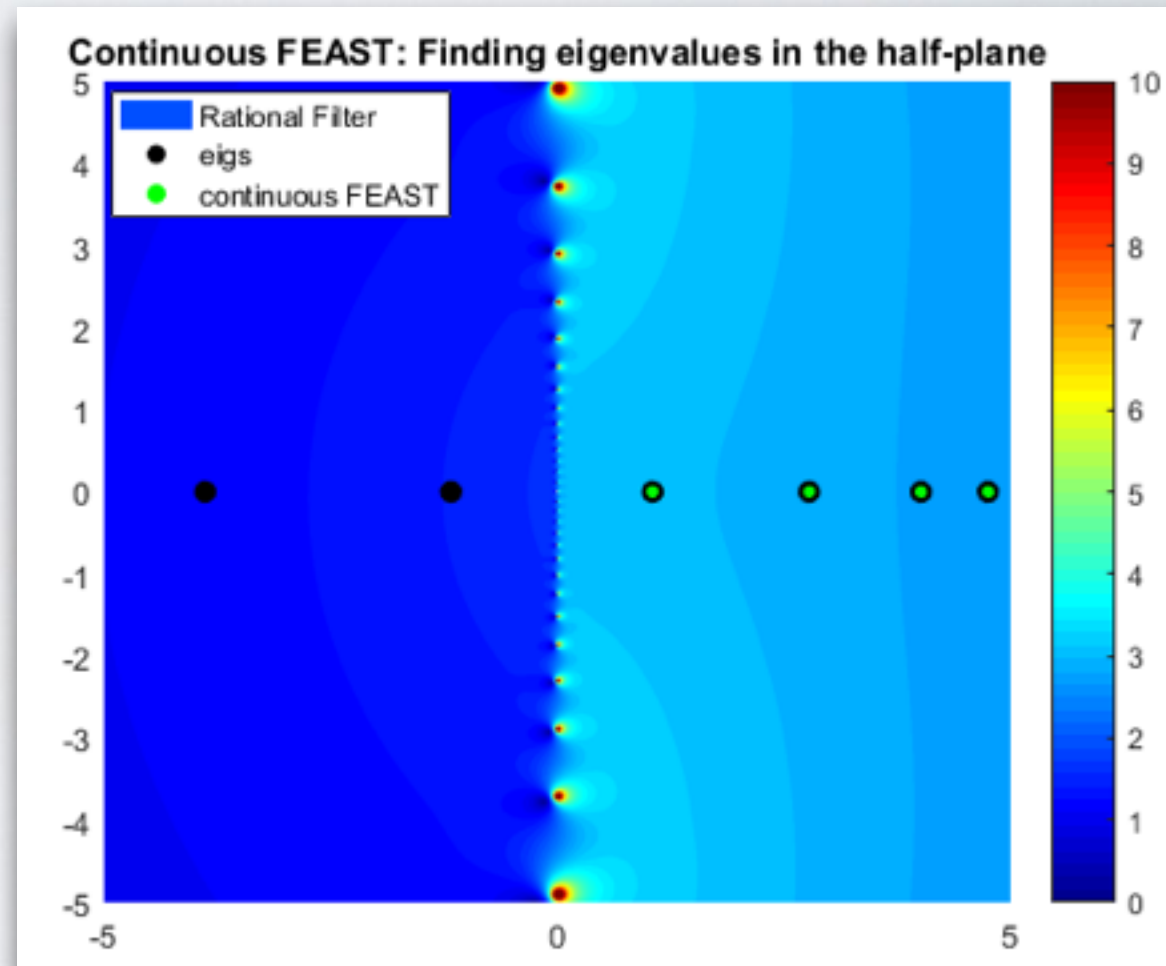
$$\hat{\lambda}, x$$

$\downarrow$

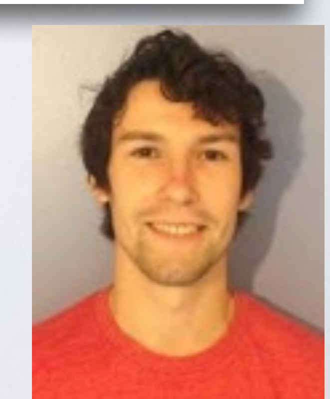
Eigenvalues, eigenfunctions

$$\hat{\lambda}, \hat{u}$$

- Discretizing can increase the sensitivity of eigenvalues.



Anthony Austin



Andrew Horning

Geometry

Prediction 4:

Element methods will be oblivious to domain discretizations.

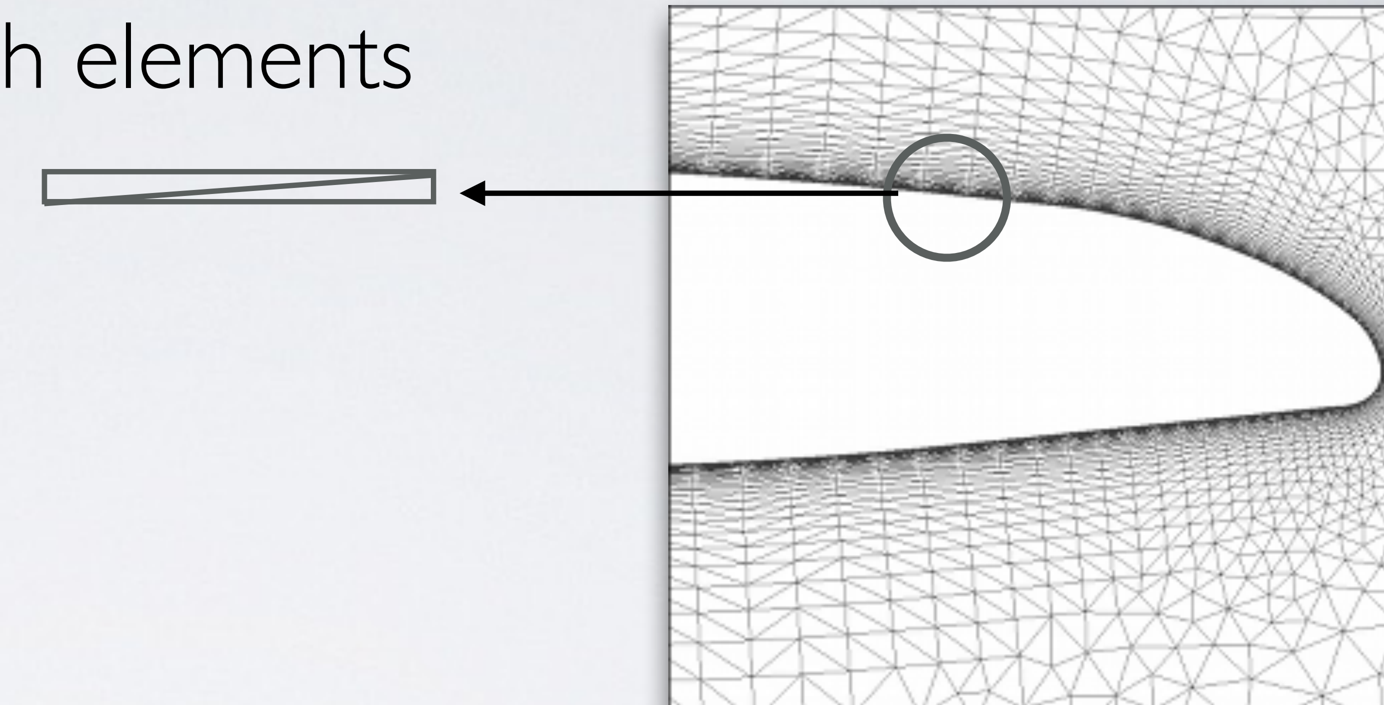
$$\Omega = \text{ domain}$$

Oblivious to mesh quality

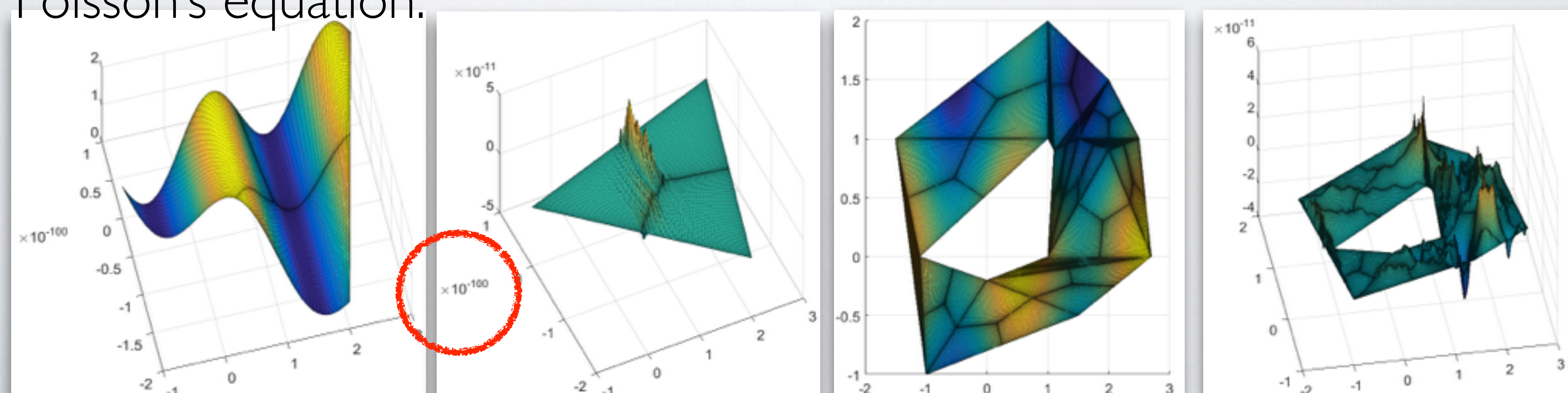
"FEMs are numerically unstable when a mesh has certain skinny triangles."
[Babuška & Aziz, 1978]

1. Instability with skinny mesh elements

A random mesh

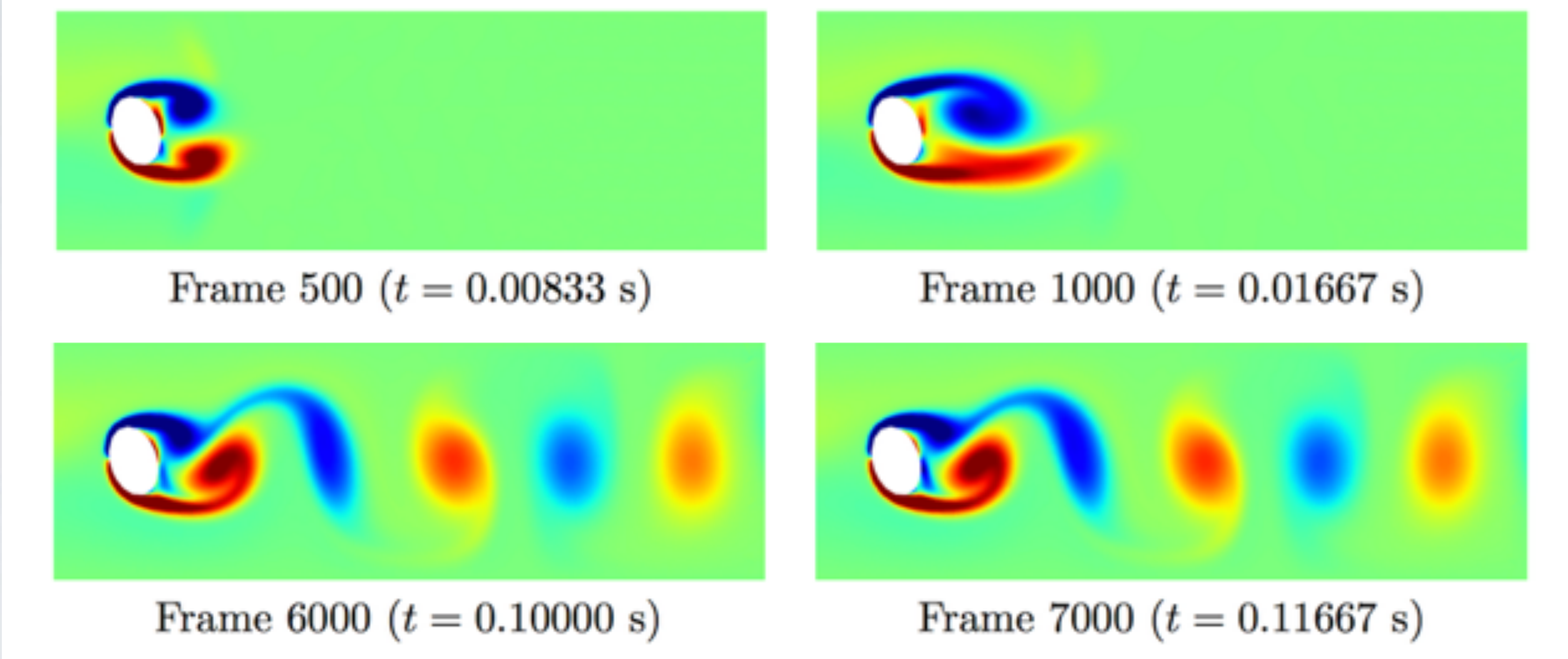


Poisson's equation:



Element methods oblivious to underlying mesh

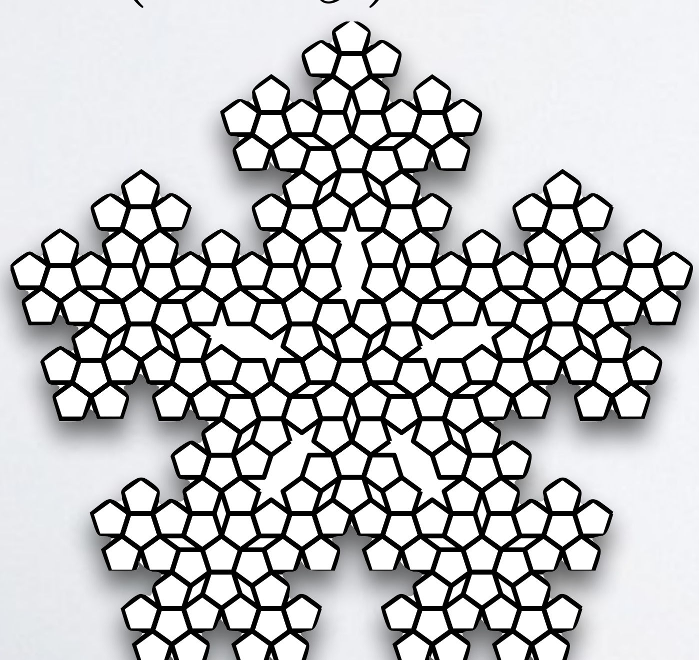
Navier-Stokes simulation

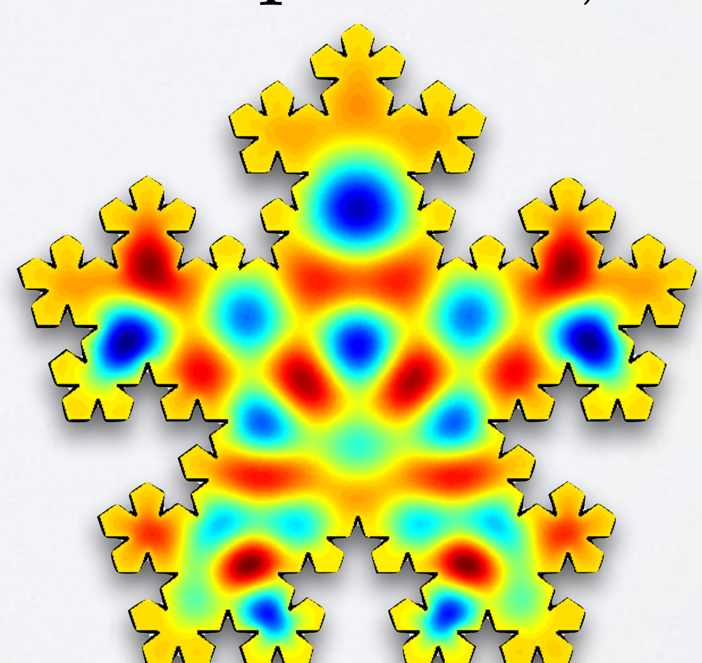


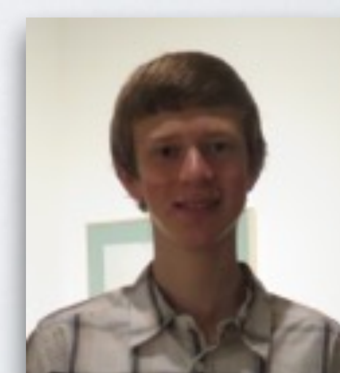
[Yeiser, 2018]

- Employ a mesh that reduces the CFL-like time-step restrictions.
- For high-Reynolds flows, we use skinny elements.

$$\nabla^2 u + 500(1-y)u = -1, \quad \Omega = \text{penrose}, \quad u|_{\partial\Omega} = 0$$





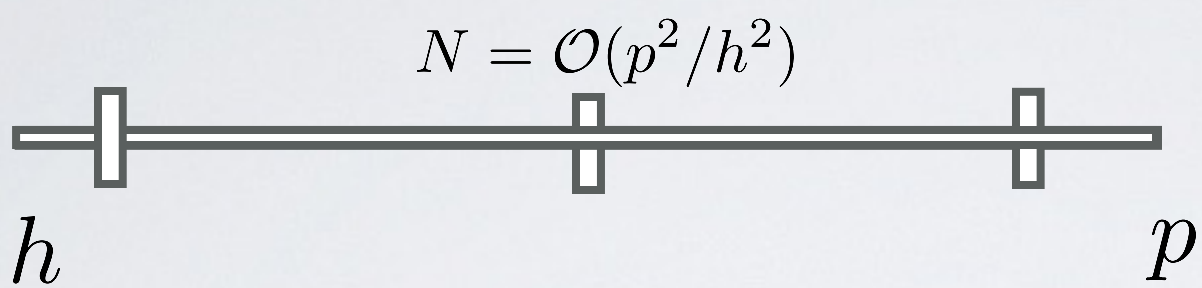


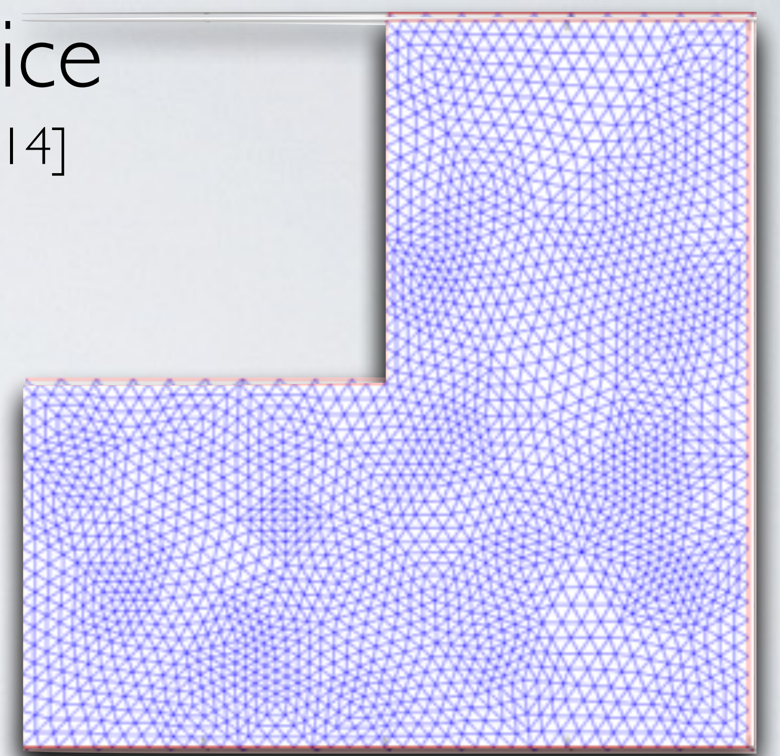
Aaron Yeiser

Oblivious to hp-adaptivity

In practice, hp-adaptivity means $p \lesssim 6$ in practice

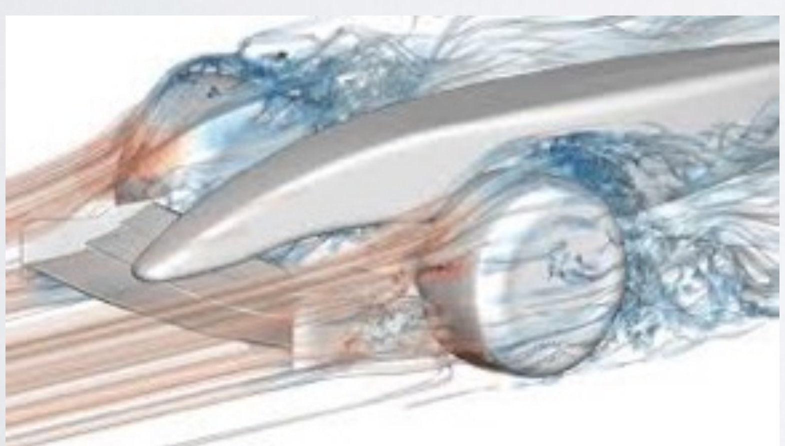
[Sherwin 2014]





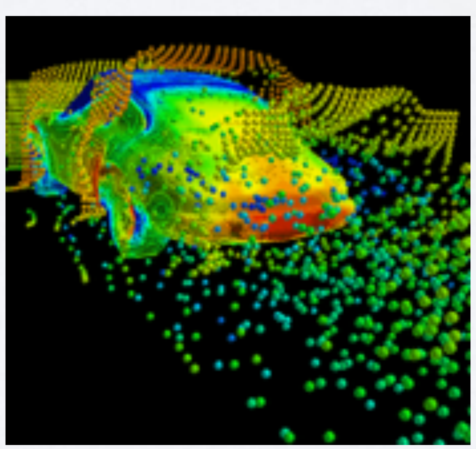
Complexity: $\mathcal{O}(p^6/h^2) = \mathcal{O}(p^4 N)$

High-p regime useful for advection-dominated flow simulations:

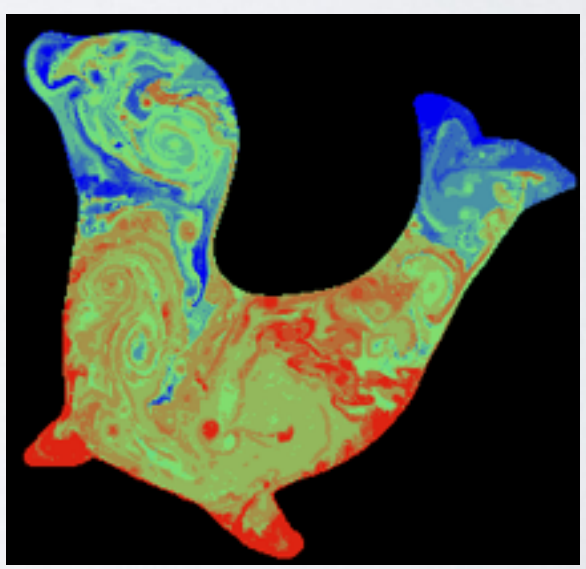


Nekar++

(simulation for Mclaren Racing Ltd)



Limiton Inc



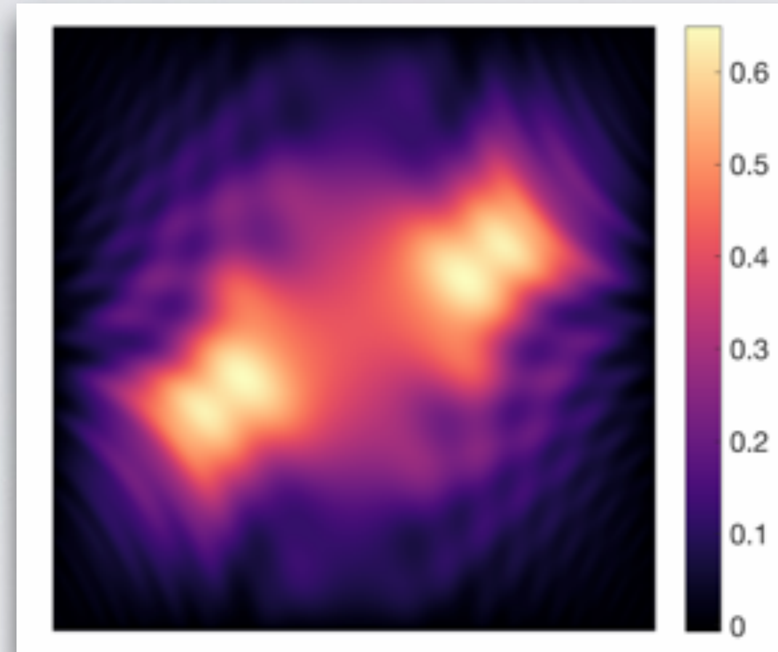
SEAL

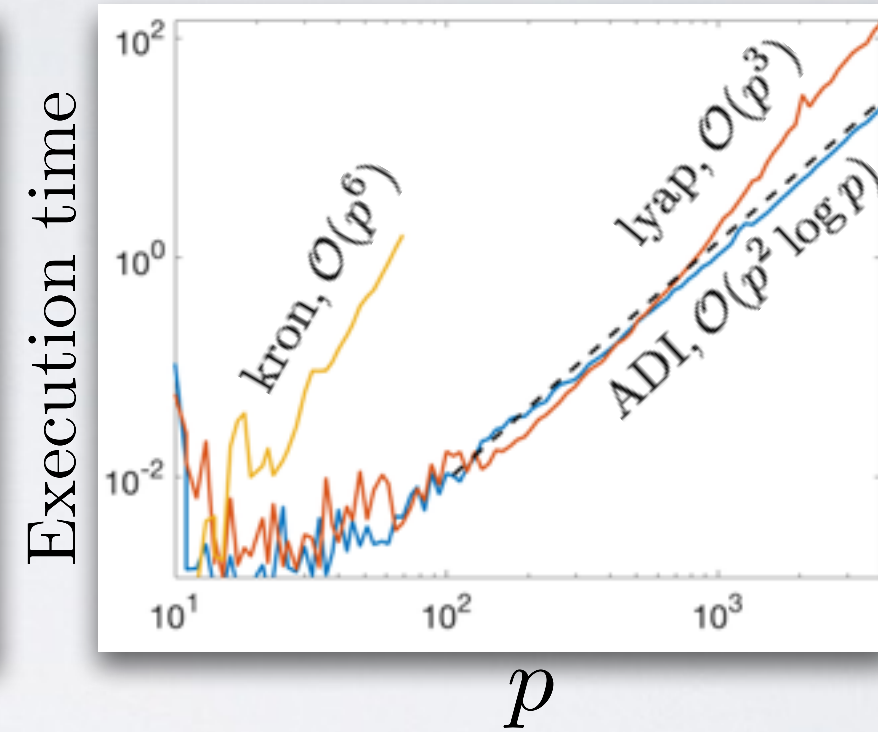
(spectral element analysis lab)

An optimal hp-adaptive FEM

Optimal complexity Poisson solver in p:

$$h = 2$$
$$p = 1{,}000$$

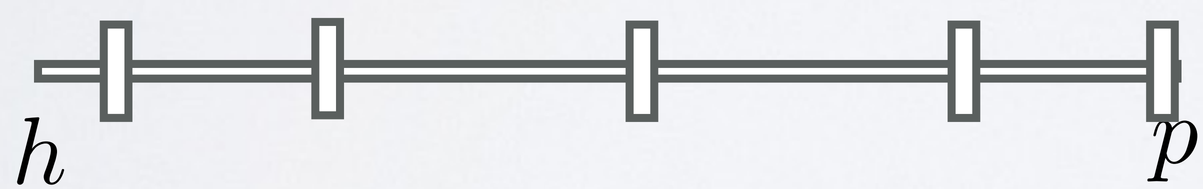




Dan Fortunato

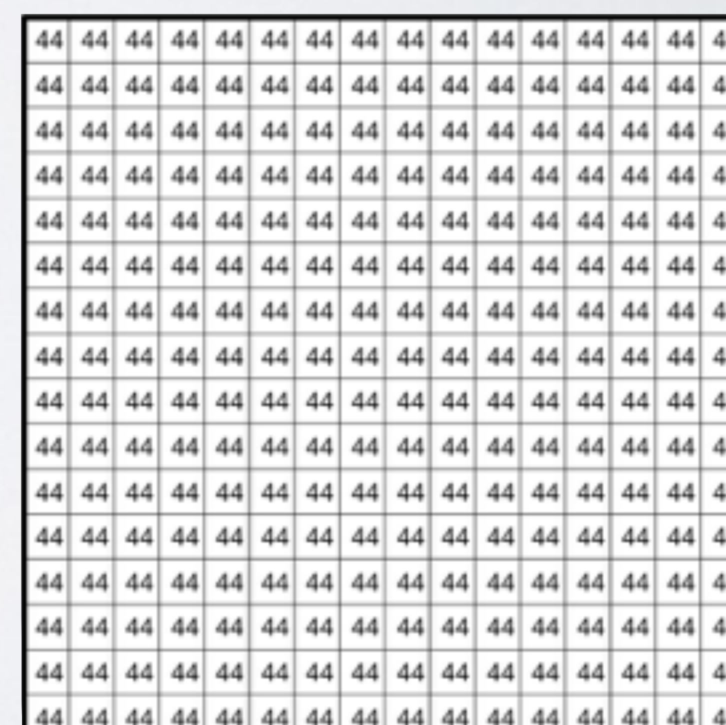
[Fortunato & T., 2018]

Challenge: Develop an optimal-complexity hp-adaptive Poisson solver:



Desired complexity:

$$\mathcal{O}(N \log p)$$



Predictions

1. Floating point arithmetic will remain important.
2. Computing with functions will be more adaptive.
3. Differential equations may not be discretized by matrices.
4. Element methods will be oblivious to domain discretizations.
5. We will be able to write down a PDE and have it solved.

Thank you

Research supported by: