Adventures in Scientific Computing: A Safari for Pure Mathematicians

Adam Larios

University of Nebraska Lincoln

6 June 2019

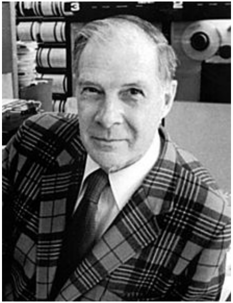
Western Washington University
Bellingham, WA, USA

Outline

Outline

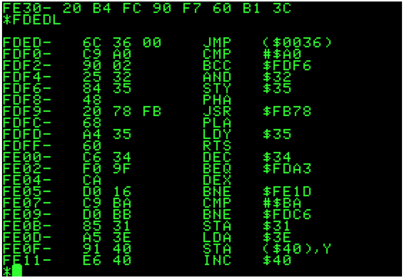
"The purpose of computing isn't numbers. It is insight."

-R.W. Hamming



- A reasonable question: Are computers beneath you?

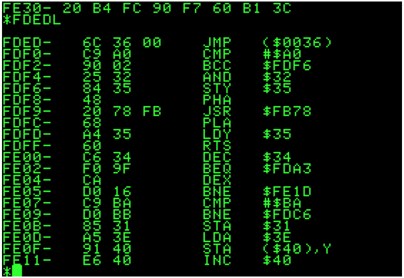
- A reasonable question: Are computers beneath you?



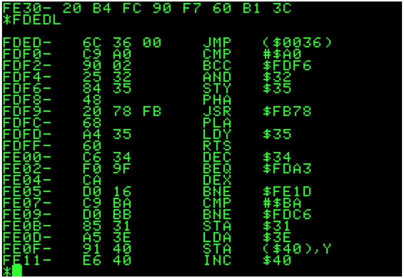
- A reasonable question: Are computers beneath you?

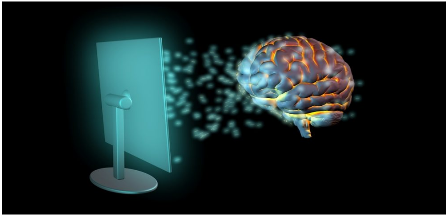
```
FE30-  20 B4 FC 90 F7 60 B1 3C
*FDEDL

FDED-    6C 36 00     JMP    ($0036)
FDF0-    C9 A0        CMP    #$A0
FDF2-    90 02        BCC    $FDF6
FDF4-    25 32        AND    $32
FDF6-    84 35        STY    $35
FDF8-    48           PHA
FDF9-    20 78 FB     JSR    $FB78
FDFC-    68           PLA
FDFD-    A4 35        LDY    $35
FDFF-    60           RTS
FE00-    C6 34        DEC    $34
FE02-    F0 9F        BEQ    $FDA3
FE04-    CA           DEX
FE05-    D0 16        BNE    $FE1D
FE07-    C9 BA        CMP    #$BA
FE09-    D0 BB        BNE    $FDC6
FE0B-    85 31        STA    $31
FE0D-    A5 3E        LDA    $3E
FE0F-    91 40        STA    ($40),Y
FE11-    E6 40        INC    $40
*
```

- A reasonable question: Are computers beneath you?



- A reasonable question: Are computers beneath you?





Question: Why learn about computers when one has mathematics?

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $$f'(x) = 1 + x$$

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $$f'(x) = 1 + x$$
- $$f'(x) = \frac{1}{1 + x}$$

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $$f'(x) = 1 + x$$
- $$f'(x) = \frac{1}{1+x}$$
- $$f'(x) = \frac{1}{1+x^2}$$

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $$f'(x) = 1 + x$$
- $$f'(x) = \frac{1}{1+x}$$
- $$f'(x) = \frac{1}{1+x^2}$$
- $$f'(x) = \frac{1}{1+x^{2.1}}$$

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $$f'(x) = 1 + x$$
- $$f'(x) = \frac{1}{1+x}$$
- $$f'(x) = \frac{1}{1+x^2}$$
- $$f'(x) = \frac{1}{1+x^{2.1}}$$

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $f'(x) = 1 + x$
- $f'(x) = \dfrac{1}{1+x}$
- $f'(x) = \dfrac{1}{1+x^2}$
- $f'(x) = \dfrac{1}{1+x^{2.1}}$

Exercise. ***Approximate***:

- $\displaystyle\int_0^t 1 + x\,dx$
- $\displaystyle\int_0^t \frac{1}{1+x}\,dx$
- $\displaystyle\int_0^t \frac{1}{1+x^2}\,dx$
- $\displaystyle\int_0^t \frac{1}{1+x^{2.1}}\,dx$

Question: Why learn about computers when one has mathematics?

Exercise. Find $f(x)$ such that:

- $f'(x) = 1 + x$
- $f'(x) = \dfrac{1}{1+x}$
- $f'(x) = \dfrac{1}{1+x^2}$
- $f'(x) = \dfrac{1}{1+x^{2.1}}$

Exercise. ***Approximate***:

- $\displaystyle\int_0^t 1 + x\,dx$
- $\displaystyle\int_0^t \frac{1}{1+x}\,dx$
- $\displaystyle\int_0^t \frac{1}{1+x^2}\,dx$
- $\displaystyle\int_0^t \frac{1}{1+x^{2.1}}\,dx$

Observation 1

- Exactness is a specialized tool.

Problem: Solve $x^2 - 2 = 0$.

Problem: Solve $x^2 - 2 = 0$.

Solution (?):

- $x = \pm\sqrt{2}$

Problem: Solve $x^2 - 2 = 0$.

Solution (?):

- $x = \pm\sqrt{2}$
- $x \approx \pm 1.414213562373\cdots$

Problem: Solve $x^2 - 2 = 0$.

Solution (?):

- $x = \pm\sqrt{2}$
- $x \approx \pm 1.414213562373\cdots$
- $x_0 = 1$ and then $x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n}$

Problem: Solve $x^2 - 2 = 0$.

Solution (?):

- $x = \pm\sqrt{2}$
- $x \approx \pm 1.414213562373 \cdots$
- $x_0 = 1$ and then $x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n}$

Problem: Solve $x^2 - 2 = 0$.

Solution (?):

- $x = \pm\sqrt{2}$
- $x \approx \pm 1.414213562373\cdots$
- $x_0 = 1$ and then $x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n}$

Observation 2

- Exactness is often a *placeholder* for an algorithm.

Problem: Solve $y'' - 4y' + 13y = 0$; $y(0) = y'(0) = 1$

Problem: Solve $y'' - 4y' + 13y = 0$; $y(0) = y'(0) = 1$

Solution:

- $y(t) = e^{-2t}\sin(3t) - e^{-2t}\cos(3t)$.

Problem: Solve $y'' - 4y' + 13y = 0$; $y(0) = y'(0) = 1$

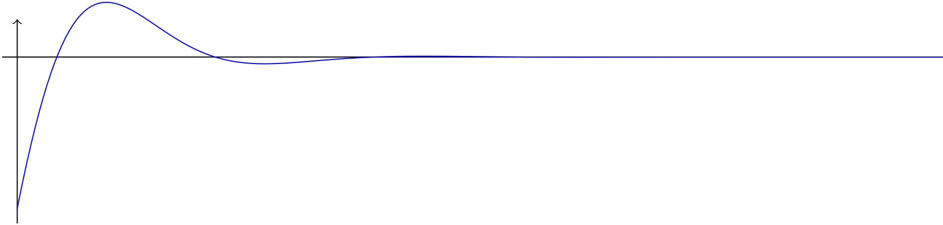
Solution:

- $y(t) = e^{-2t}\sin(3t) - e^{-2t}\cos(3t)$.

Problem: Solve $y'' - 4y' + 13y = 0$; $y(0) = y'(0) = 1$

Solution:

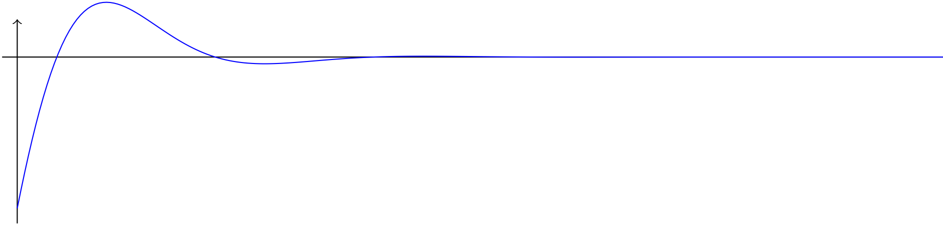
- $y(t) = e^{-2t}\sin(3t) - e^{-2t}\cos(3t)$.



Problem: Solve $y'' - 4y' + 13y = 0$; $y(0) = y'(0) = 1$

Solution:

- $y(t) = e^{-2t}\sin(3t) - e^{-2t}\cos(3t)$.



Observation 3

- Solutions are not always this simple...

The Incompressible Navier-Stokes Equations

Claude L.M.H. Navier

George G. Stokes

Momentum Equation

$$\underbrace{\frac{\partial}{\partial t}\mathbf{u}}_{Acceleration} + \underbrace{(\mathbf{u}\cdot\nabla)\mathbf{u}}_{Advection} = \underbrace{-\nabla p}_{\substack{Pressure\\Gradient}} + \underbrace{\nu\triangle\mathbf{u}}_{\substack{Viscous\\Diffusion}} + \underbrace{\mathbf{f}}_{\substack{Body\\Force}}$$

Continuity Equation

$$\nabla\cdot\mathbf{u} = 0$$

Unknowns

$\mathbf{u}$:= Velocity (vector)

p := Pressure (scalar)

Parameter

ν := Kinematic Viscosity

Problem (Leray 1933)

Existence and uniqueness of strong solutions in 3D for all time. ($1,000,000 Clay Millennium Prize Problem)

J. Fluid Mech. (1983), *vol.* 130, *pp.* 411–452
Printed in Great Britain

Small-scale structure of the Taylor–Green vortex

By MARC E. BRACHET†, DANIEL I. MEIRON,
STEVEN A. ORSZAG,
Massachusetts Institute of Technology, Cambridge, MA 02139

B. G. NICKEL,
University of Guelph, Guelph, Ontario

RUDOLF H. MORF
R.C.A. Laboratories, Zurich, Switzerland

AND URIEL FRISCH
CNRS, Observatoire de Nice, 06-Nice, France

(Received 5 February 1982 and in revised form 14 June 1982)

The dynamics of both the inviscid and viscous Taylor–Green (TG) three-dimensional vortex flows are investigated. This flow is perhaps the simplest system in which one can study the generation of small scales by three-dimensional vortex stretching and the resulting turbulence. The problem is studied by both direct spectral numerical solution of the Navier–Stokes equations (with up to 256^3 modes) and by power-series analvsis in time.

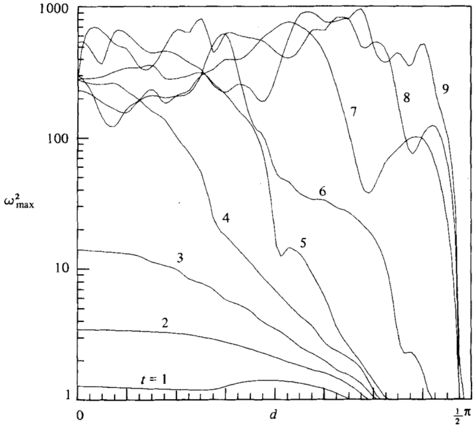


FIGURE 20. A plot of the maximum vorticity $\omega^2_{\max}$ as a function of the distance from the walls of the impermeable cube for $t = 1$–9 (1). Here we compute $\omega^2_{\max}$ over the faces of subcubes nested in such a way that the distance from the nearest wall of the impermeable cube is d. Thus $d = 0$ corresponds to the maximum vorticity on the faces of the impermeable cube while $d = \frac{1}{2}\pi$ corresponds to the vorticity at the centre of this cube. Observe that for early times the vorticity is concentrated near the walls of the impermeable cube. As t increases beyond 6, there is significant vorticity generation in the main body of the cube.

Commun. Math. Phys. 94, 61–66 (1984)

Communications in Mathematical Physics

Remarks on the Breakdown of Smooth Solutions for the 3-*D* Euler Equations

J. T. Beale[1]*, T. Kato[2]**, and A. Majda[2]***

1 Department of Mathematics, Duke University, Durham, NC 27701, USA
2 Department of Mathematics, University of California, Berkeley, CA 94720, USA

Abstract. The authors prove that the maximum norm of the vorticity controls the breakdown of smooth solutions of the 3-*D* Euler equations. In other words, if a solution of the Euler equations is initally smooth and loses its regularity at some later time, then the maximum vorticity necessarily grows without bound as the critical time approaches; equivalently, if the vorticity remains bounded, a smooth solution persists.

The motion of an ideal incompressible fluid is governed by a system of partial differential equations known as the Euler equations. For two-dimensional flow,

- Thus, computations can give mathematicians *insight*.

- Thus, computations can give mathematicians *insight*.
- Even if we don't want to use computations ourselves, it can benefit us to learn how they work (and learn to recognize when they *fail*.)

Outline

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Let's use a top-of-the-line Intel i7 processor:
$\approx 3 \times 10^{11}$ multiplications/divsions per second!

N	Time

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Let's use a top-of-the-line Intel i7 processor:
$\approx 3 \times 10^{11}$ multiplications/divsions per second!

N	**Time**
100	1μs

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Let's use a top-of-the-line Intel i7 processor:
$\approx 3 \times 10^{11}$ multiplications/divsions per second!

N	Time
100	1μs
1000	0.003s

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Let's use a top-of-the-line Intel i7 processor:
$\approx 3 \times 10^{11}$ multiplications/divsions per second!

N	Time
100	1μs
1000	0.003s
10^6	12 days

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Let's use a top-of-the-line Intel i7 processor:
$\approx 3 \times 10^{11}$ multiplications/divsions per second!

N	**Time**
100	1μs
1000	0.003s
10^6	12 days
10^7	35 years

Problem: Solve $A\vec{x} = \vec{b}$, where A is an invertible matrix.

(Note: We don't bother to find or store A^{-1}.)

Gaussian Elimination

- Works for any matrix!
- Algorithm is "unstable": tiny round-off errors are strongly amplified.
- For an $N \times N$ matrix, cost $\approx \frac{1}{3}N^3$ multiplications/divisions.

Let's use a top-of-the-line Intel i7 processor:
$\approx 3 \times 10^{11}$ multiplications/divsions per second!

N	Time
100	1μs
1000	0.003s
10^6	12 days
10^7	35 years
10^9	35 million years!

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{cases} 5x + 6y + 4z = 3 \\ 7x + 9y + 3z = 1 \\ 2x + 1y + 8z = 2 \end{cases}$$

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{cases} 5x + 6y + 4z = 3 \\ 7x + 9y + 3z = 1 \\ 2x + 1y + 8z = 2 \end{cases}$$

$$\begin{cases} 5x = 3 \qquad\quad - 6y - 4z \\ 9y = 1 - 7x \qquad\quad - 3z \\ 8z = 2 - 2x - 1y \end{cases}$$

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{cases} 5x + 6y + 4z = 3 \\ 7x + 9y + 3z = 1 \\ 2x + 1y + 8z = 2 \end{cases}$$

$$\begin{cases} 5x = 3 \qquad - 6y - 4z \\ 9y = 1 - 7x \qquad - 3z \\ 8z = 2 - 2x - 1y \end{cases}$$

$$\begin{cases} x = \frac{1}{5}(3 - 6y - 4z) \\ y = \frac{1}{9}(1 - 7x - 3z) \\ z = \frac{1}{8}(2 - 2x - 1y) \end{cases}$$

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{cases} 5x + 6y + 4z = 3 \\ 7x + 9y + 3z = 1 \\ 2x + 1y + 8z = 2 \end{cases}$$

$$\begin{cases} 5x = 3 \quad\quad - 6y - 4z \\ 9y = 1 - 7x \quad\quad - 3z \\ 8z = 2 - 2x - 1y \end{cases}$$

$$\begin{cases} x = \frac{1}{5}(3 - 6y - 4z) \\ y = \frac{1}{9}(1 - 7x - 3z) \\ z = \frac{1}{8}(2 - 2x - 1y) \end{cases}$$

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\begin{cases} 5x + 6y + 4z = 3 \\ 7x + 9y + 3z = 1 \\ 2x + 1y + 8z = 2 \end{cases}$$

$$\begin{cases} 5x = 3 \quad - 6y - 4z \\ 9y = 1 - 7x \quad - 3z \\ 8z = 2 - 2x - 1y \end{cases}$$

$$\begin{cases} x = \frac{1}{5}(3 - 6y - 4z) \\ y = \frac{1}{9}(1 - 7x - 3z) \\ z = \frac{1}{8}(2 - 2x - 1y) \end{cases}$$

$$\begin{cases} x_{n+1} = \frac{1}{5}(3 - 6y_n - 4z_n) \\ y_{n+1} = \frac{1}{9}(1 - 7x_n - 3z_n) \\ z_{n+1} = \frac{1}{8}(2 - 2x_n - 1y_n) \end{cases}$$

One alternative to Gaussian Elimination

$$\begin{bmatrix} 5 & 6 & 4 \\ 7 & 9 & 3 \\ 2 & 1 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

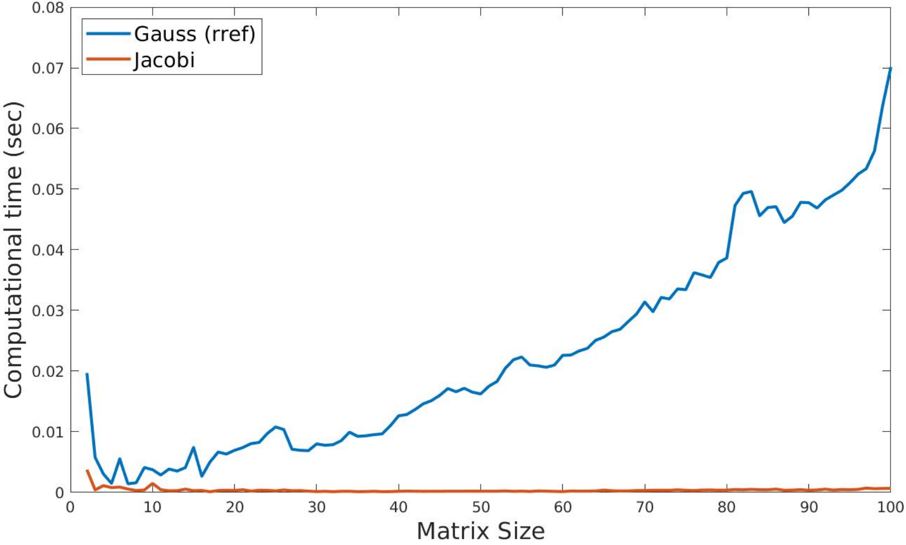
$$\begin{cases} 5x + 6y + 4z = 3 \\ 7x + 9y + 3z = 1 \\ 2x + 1y + 8z = 2 \end{cases}$$

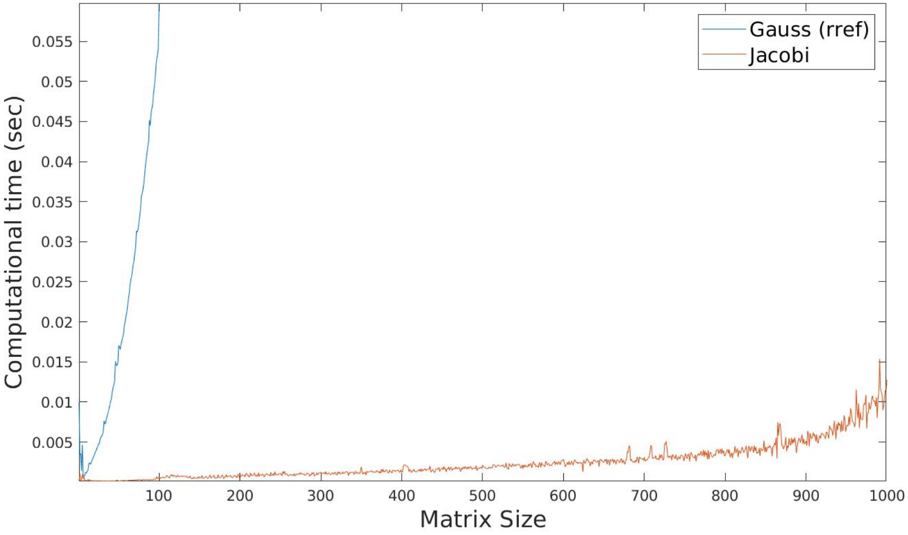
$$\begin{cases} 5x = 3 \qquad - 6y - 4z \\ 9y = 1 - 7x \qquad - 3z \\ 8z = 2 - 2x - 1y \end{cases}$$

$$\begin{cases} x = \frac{1}{5}(3 - 6y - 4z) \\ y = \frac{1}{9}(1 - 7x - 3z) \\ z = \frac{1}{8}(2 - 2x - 1y) \end{cases}$$

$$\begin{cases} x_{n+1} = \frac{1}{5}(3 - 6y_n - 4z_n) \\ y_{n+1} = \frac{1}{9}(1 - 7x_n - 3z_n) \\ z_{n+1} = \frac{1}{8}(2 - 2x_n - 1y_n) \end{cases}$$

This is Jacobi's iteration method.


Gauss (rref)
Jacobi
Computational time (sec)
Matrix Size
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0
10
20
30
40
50
60
70
80
90
100


Gauss (rref)
Jacobi
Computational time (sec)
0.055
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
100
200
300
400
500
600
700
800
900
1000
Matrix Size

Jacobi Iteration In Detail

Problem: Solve $A\vec{x} = \vec{b}$ via

$$\text{diag}(A)\vec{x}_{n+1} + (A - \text{diag}(A))\vec{x}_n = \vec{b}.$$

Rewrite as:

$$\begin{aligned}
\vec{x}_{n+1} &= (\text{diag}(A))^{-1}(\vec{b} - (A - \text{diag}(A))\vec{x}_n) \\
&= (\text{diag}(A))^{-1}(\vec{b} - A\vec{x}_n) + \vec{x}_n \\
&= \vec{x}_n + (\text{diag}(A))^{-1}(\vec{b} - A\vec{x}_n)
\end{aligned}$$

Jacobi Iteration In Detail

Problem: Solve $A\vec{x} = \vec{b}$ via

$$\text{diag}(A)\vec{x}_{n+1} + (A - \text{diag}(A))\vec{x}_n = \vec{b}.$$

Rewrite as:

$$\begin{aligned}
\vec{x}_{n+1} &= (\text{diag}(A))^{-1}(\vec{b} - (A - \text{diag}(A))\vec{x}_n) \\
&= (\text{diag}(A))^{-1}(\vec{b} - A\vec{x}_n) + \vec{x}_n \\
&= \vec{x}_n + (\text{diag}(A))^{-1}(\vec{b} - A\vec{x}_n)
\end{aligned}$$

Write $B = (\text{diag}(A))^{-1}$. Then

$$\begin{aligned}
\vec{x}_{n+1} &= \vec{x}_n + B(\vec{b} - A\vec{x}_n) \\
&= (I - BA)\vec{x}_n + B\vec{b}
\end{aligned}$$

Jacobi Iteration In Detail

Problem: Solve $A\vec{x} = \vec{b}$ via

$$\text{diag}(A)\vec{x}_{n+1} + (A - \text{diag}(A))\vec{x}_n = \vec{b}.$$

Rewrite as:

$$\begin{aligned}
\vec{x}_{n+1} &= (\text{diag}(A))^{-1}(\vec{b} - (A - \text{diag}(A))\vec{x}_n) \\
&= (\text{diag}(A))^{-1}(\vec{b} - A\vec{x}_n) + \vec{x}_n \\
&= \vec{x}_n + (\text{diag}(A))^{-1}(\vec{b} - A\vec{x}_n)
\end{aligned}$$

Write $B = (\text{diag}(A))^{-1}$. Then

$$\begin{aligned}
\vec{x}_{n+1} &= \vec{x}_n + B(\vec{b} - A\vec{x}_n) \\
&= (I - BA)\vec{x}_n + B\vec{b}
\end{aligned}$$

To generalize, write $Q = B^{-1} = \text{diag}(A)$ and rearrange:

$$Q\vec{x}_{n+1} = (Q - A)\vec{x}_n + \vec{b}$$

Convergence for Iterative Methods

Problem: Solve $A\vec{x} = \vec{b}$.

Algorithm: Choose an invertible, easy-to-compute *preconditioner* matrix B.
Iterate with $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

Convergence for Iterative Methods

Problem: Solve $A\vec{x} = \vec{b}$.

Algorithm: Choose an invertible, easy-to-compute *preconditioner* matrix B.
Iterate with $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

- We can now consider other choices for $B = Q^{-1}$.
- Q is called the *splitting* of A.
- $B = Q^{-1}$ is called the *preconditioner* of A.

Convergence for Iterative Methods

Problem: Solve $A\vec{x} = \vec{b}$.

Algorithm: Choose an invertible, easy-to-compute *preconditioner* matrix B.
Iterate with $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

- We can now consider other choices for $B = Q^{-1}$.
- Q is called the *splitting* of A.
- $B = Q^{-1}$ is called the *preconditioner* of A.

Idea: *If* $\vec{x}_n \to \vec{x}$ for some $\vec{x}$, then:

$$\vec{x} = (I - BA)\vec{x} + B\vec{b}$$
$$\Rightarrow \vec{x} = \vec{x} - BA\vec{x} + B\vec{b}$$
$$\Rightarrow BA\vec{x} = B\vec{b}$$
$$\Rightarrow A\vec{x} = \vec{b}$$

So $\mathbf{x}$ is the solution to our problem!

Algorithm: $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

Q: When does the iteration converge?

Algorithm: $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

Q: When does the iteration converge?

Idea: Let error $= \vec{e}_n = \vec{x}_n - \vec{x}$. A little algebra shows,

$$e_{n+1} = (I - BA)e_n.$$

Algorithm: $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

Q: When does the iteration converge?

Idea: Let error $= \vec{e}_n = \vec{x}_n - \vec{x}$. A little algebra shows,

$$e_{n+1} = (I - BA)e_n.$$

Thus,

$$\vec{e}_{n+1} = (I - BA)\vec{e}_n = (I - BA)^2\vec{e}_{n-1} = \dots (I - BA)^n\vec{e}_0.$$

Algorithm: $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

Q: When does the iteration converge?

Idea: Let error $= \vec{e}_n = \vec{x}_n - \vec{x}$. A little algebra shows,

$$e_{n+1} = (I - BA)e_n.$$

Thus,

$$\vec{e}_{n+1} = (I - BA)\vec{e}_n = (I - BA)^2\vec{e}_{n-1} = \dots (I - BA)^n\vec{e}_0.$$

- Worst-case scenario: When $\vec{e}_0$ is an eigenvector of $I - BA$ with $|\lambda| > 1$.
- Strategy: Show spectral radius of $I - BA$, $\rho(I - BA) < 1$.

Algorithm: $\vec{x}_{n+1} = (I - BA)\vec{x}_n + B\vec{b}$.

Q: When does the iteration converge?

Idea: Let error $= \vec{e}_n = \vec{x}_n - \vec{x}$. A little algebra shows,

$$e_{n+1} = (I - BA)e_n.$$

Thus,

$$\vec{e}_{n+1} = (I - BA)\vec{e}_n = (I - BA)^2\vec{e}_{n-1} = \dots (I - BA)^n\vec{e}_0.$$

- Worst-case scenario: When $\vec{e}_0$ is an eigenvector of $I - BA$ with $|\lambda| > 1$.
- Strategy: Show spectral radius of $I - BA$, $\rho(I - BA) < 1$.
  Requires spectral analysis!

Partial Answer: Gershgoin Discs

For any matix A with eigenvalue λ, choose eigenvector $\mathbf{x}$ so that $x_i = 1$ for some i and $|x_j| \le 1$ for all j. Rewrite $A\mathbf{x} = \lambda\mathbf{x}$ as

$$\sum_j a_{ij}x_j = \lambda x_i = \lambda.$$

Then, since again $x_i = 1$,

$$\sum_{j\neq i} a_{ij}x_j + a_{ii} = \lambda.$$

Thus,

$$|\lambda - a_{ii}| = \left|\sum_{j\neq i} a_{ij}x_j\right| \le \sum_{j\neq i} |a_{ij}| =: R_i.$$

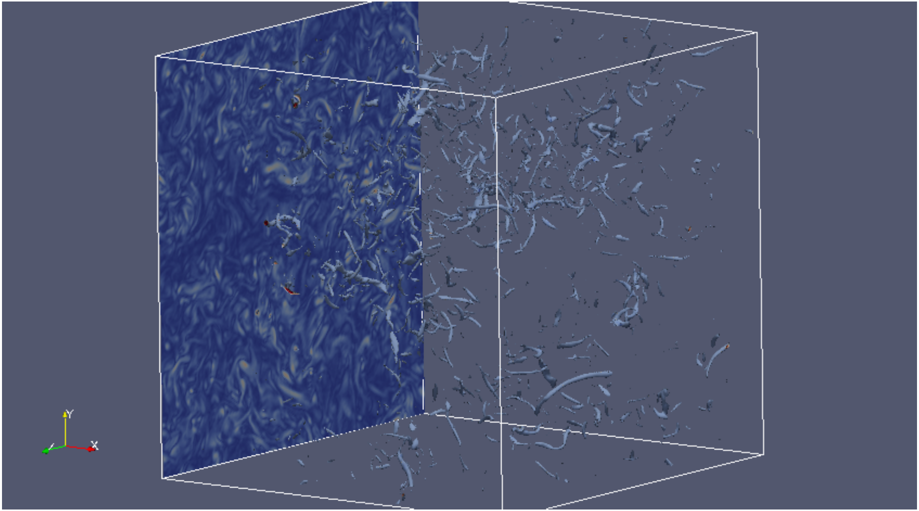
Thus, every eigenvalue must lie in one of the Gershgorin discs $D_I := D(a_{ii}, R_i)$!

Theorem: The Jacobi method is convergent for any strictly diagonally dominant matrix A, i.e., $|a_{ii}| > \sum_{j\neq i} |a_{ij}|$.
Proof.
Jacobi matrix is

$$I - BA = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & -\frac{a_{i1}}{a_{ii}} & \cdots & 0 & \cdots & -\frac{a_{in}}{a_{ii}} \\ \vdots & \vdots & \vdots & & \vdots & \end{bmatrix}$$

Gershgorin Discs: $D_i = \left\{ z \in \mathbb{C} : |z - 0| \leq \sum_{j\neq i} \left|\frac{a_{ij}}{a_{ii}}\right| \right\} \subset B(0,1)$. Thus, $\rho(I - BA) < 1$!

3D Vortex Structures



Mappings

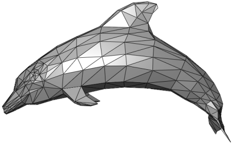
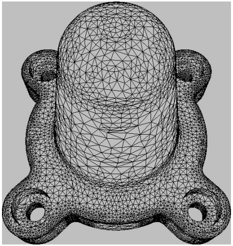
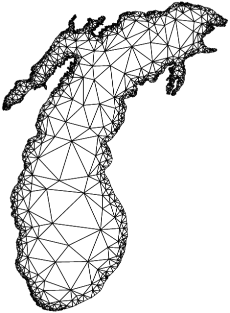
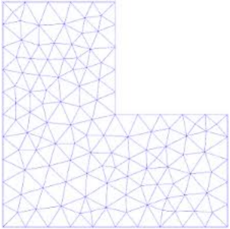
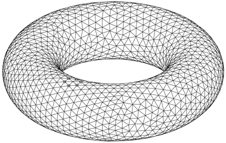
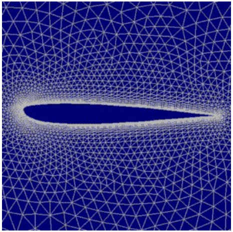
Mesh generation	$\longleftarrow$	Graph Theory
Iterative Methods ($A\mathbf{x} = \mathbf{b}$)	$\longleftarrow$	Spectral Theory
Elliptic Curve Cryptography	$\longleftarrow$	Number Theory
Finite Element Methods	$\longleftarrow$	PDE Theory
Visualization	$\longleftarrow$	Geometry
Parallel Computing	$\longleftarrow$	Scheduling and Graph Theory
Large matrices	$\longleftarrow$	Operator Theory

Mappings

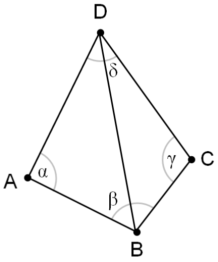
Mesh generation	$\longleftrightarrow$	Graph Theory
Iterative Methods ($A\mathbf{x} = \mathbf{b}$)	$\longleftrightarrow$	Spectral Theory
Elliptic Curve Cryptography	$\longleftrightarrow$	Number Theory
Finite Element Methods	$\longleftrightarrow$	PDE Theory
Visualization	$\longleftrightarrow$	Geometry
Parallel Computing	$\longleftrightarrow$	Scheduling and Graph Theory
Large matrices	$\longleftrightarrow$	Operator Theory

Outline

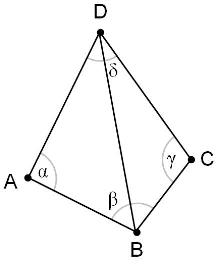
Meshes

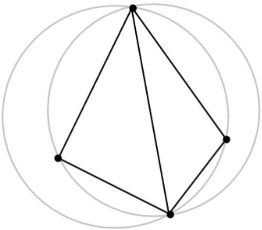







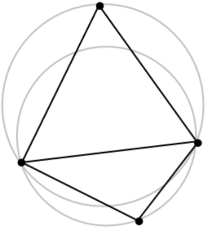
Delaunay Mesh



Delaunay Mesh



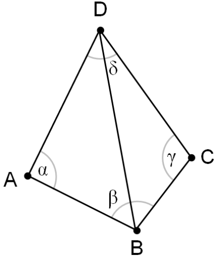


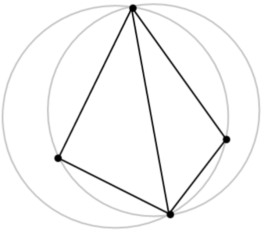


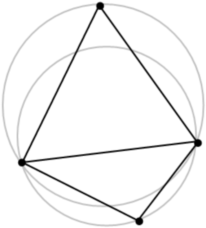
Delaunay condition

- Three points p, q, r, in S form a Delaunay triangle if and only if the circumcircle of these points contains no other point of S.

Delaunay Mesh







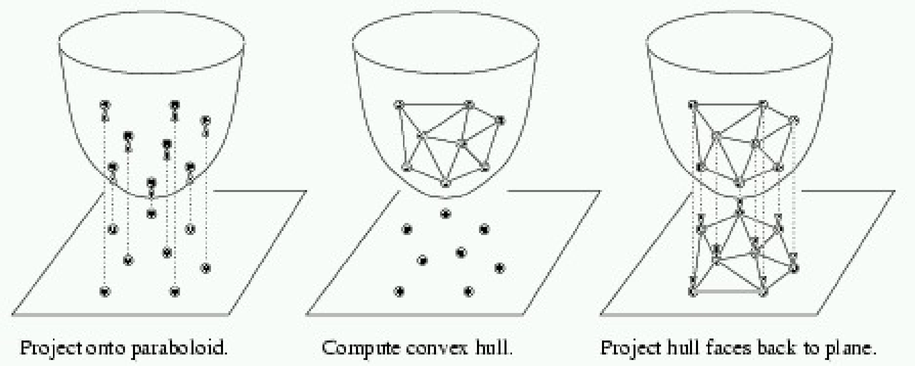
Delaunay condition

- Three points p, q, r, in S form a Delaunay triangle if and only if the circumcircle of these points contains no other point of S.

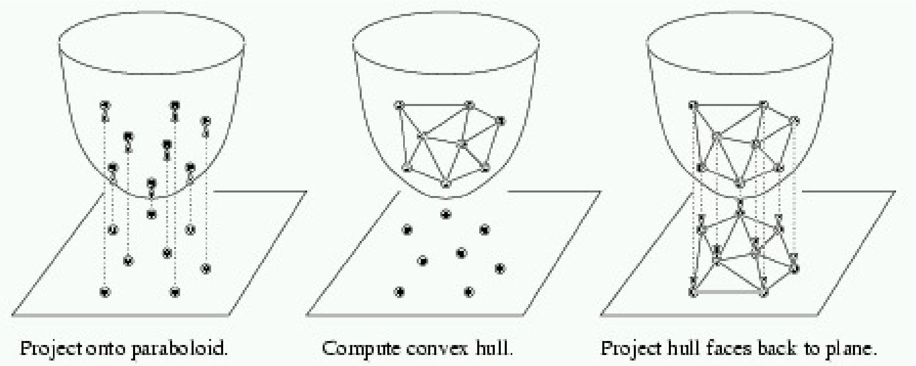
Convex hull condition

- Three points p_0, q_0, r_0 in S_0 form a face of the convex hull of S_0 if and only if the plane passing through p_0, q_0, r_0 has all the points of S_0 lying to one side.

Efficient Delaunay Computation



Efficient Delaunay Computation

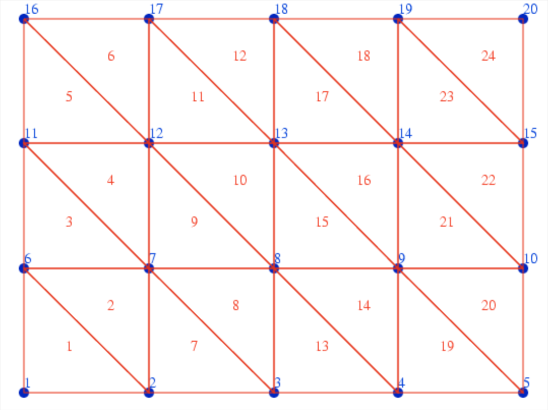


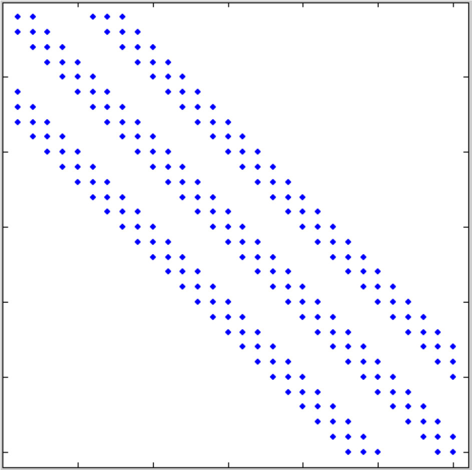
- Computational problems can lead to new ideas!

Mesh Labeling

Given $f = f(x, y)$, find $u = u(x, y)$ such that $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f$.
Write $u_{i.j} = u(x_i, y_j)$.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}}{\Delta x \Delta y}$$

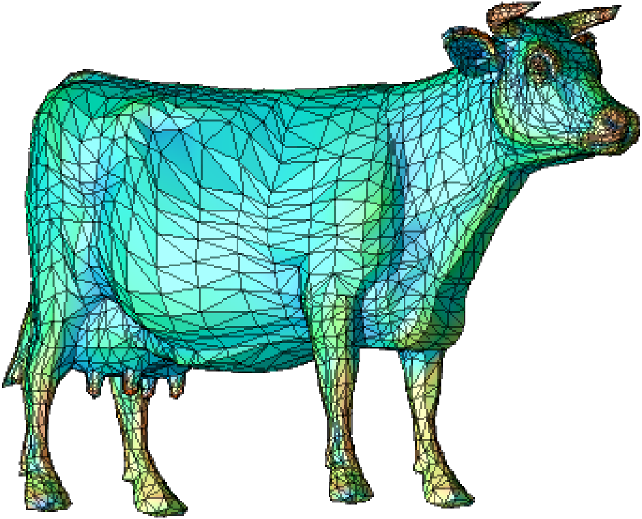


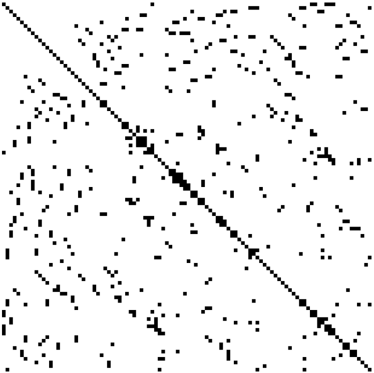


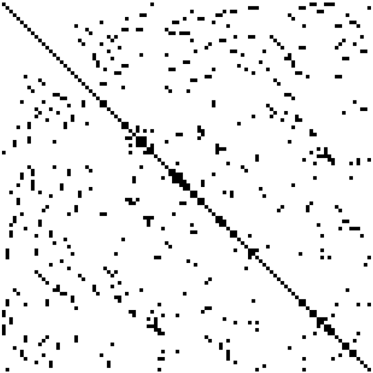
Note

- Tightly "banded" structure is more efficient for computation.

A more challenging mesh

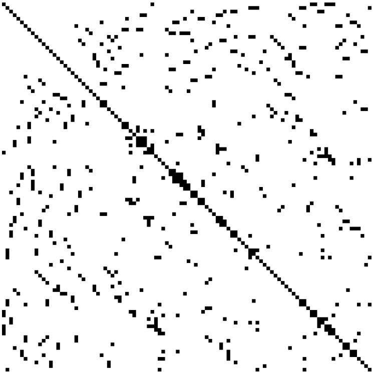






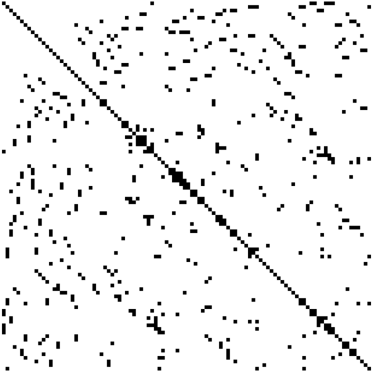
New ideas are needed!

- Efficient preconditioners.



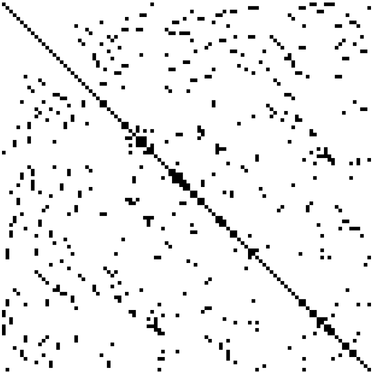
New ideas are needed!

- Efficient preconditioners.
- New iteration methods.



New ideas are needed!

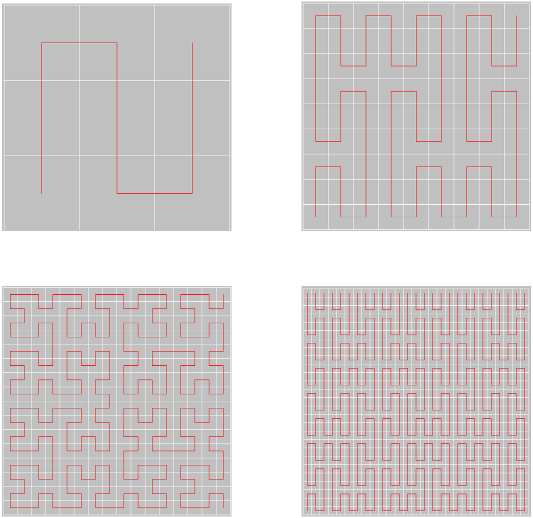
- Efficient preconditioners.
- New iteration methods.
- Ideas that exploit “sparse” structure.



New ideas are needed!

- Efficient preconditioners.
- New iteration methods.
- Ideas that exploit “sparse” structure.
- Mesh Relabeling. . .

Space Filling Curves



Example Problem: Linear Systems

Outline

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

- Wrong way: $\mathbf{x} = A^{-1}\mathbf{b}$.

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

- Wrong way: $\mathbf{x} = A^{-1}\mathbf{b}$.
- Better way:

$$
\begin{aligned}
Q\mathbf{x}_{k+1} &= (Q - A)\mathbf{x}_k + \mathbf{b} \\
\mathbf{x}_{k+1} &= (I - Q^{-1}A)\mathbf{x}_k + Q^{-1}\mathbf{b}
\end{aligned}
$$

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

- Wrong way: $\mathbf{x} = A^{-1}\mathbf{b}$.
- Better way:

$$
\begin{aligned}
Q\mathbf{x}_{k+1} &= (Q - A)\mathbf{x}_k + \mathbf{b} \\
\mathbf{x}_{k+1} &= (I - Q^{-1}A)\mathbf{x}_k + Q^{-1}\mathbf{b}
\end{aligned}
$$

- Fundamental Theorem of Iterative Methods:
  Iteration will converge if and only if spectral radius of $(I - Q^{-1}A)$ is < 1.

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

- Wrong way: $\mathbf{x} = A^{-1}\mathbf{b}$.
- Better way:

$$
\begin{aligned}
Q\mathbf{x}_{k+1} &= (Q - A)\mathbf{x}_k + \mathbf{b} \\
\mathbf{x}_{k+1} &= (I - Q^{-1}A)\mathbf{x}_k + Q^{-1}\mathbf{b}
\end{aligned}
$$

- Fundamental Theorem of Iterative Methods:
  Iteration will converge if and only if spectral radius of $(I - Q^{-1}A)$ is < 1.

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

- Wrong way: $\mathbf{x} = A^{-1}\mathbf{b}$.
- Better way:

$$\begin{aligned} Q\mathbf{x}_{k+1} &= (Q - A)\mathbf{x}_k + \mathbf{b} \\ \mathbf{x}_{k+1} &= (I - Q^{-1}A)\mathbf{x}_k + Q^{-1}\mathbf{b} \end{aligned}$$

- Fundamental Theorem of Iterative Methods:
  Iteration will converge if and only if spectral radius of $(I - Q^{-1}A)$ is < 1.

Problem 2

- Design Q with $\rho(I - Q^{-1}A) < 1$ so that the iteration converges fast.

Iterative Methods

Problem

- Solve $A\mathbf{x} = \mathbf{b}$.

- Wrong way: $\mathbf{x} = A^{-1}\mathbf{b}$.
- Better way:

$$\begin{aligned} Q\mathbf{x}_{k+1} &= (Q - A)\mathbf{x}_k + \mathbf{b} \\ \mathbf{x}_{k+1} &= (I - Q^{-1}A)\mathbf{x}_k + Q^{-1}\mathbf{b} \end{aligned}$$

- Fundamental Theorem of Iterative Methods:
  Iteration will converge if and only if spectral radius of $(I - Q^{-1}A)$ is < 1.

Problem 2

- Design Q with $\rho(I - Q^{-1}A) < 1$ so that the iteration converges fast.
- Now we get to play with spectral theory!

Variation/Minimization Methods

Problem (Again)

- Solve $A\mathbf{x} = \mathbf{b}$.

Variation/Minimization Methods

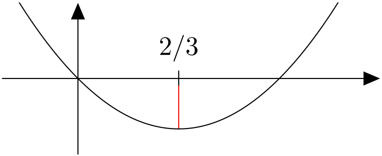
Problem (Again)

- Solve $A\mathbf{x} = \mathbf{b}$.

Easy 1D version

- Solve $3x = 2$.

- Let $f(x) = \frac{3}{2}x^2 - 2x$.



Variation/Minimization Methods

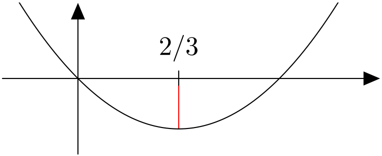
Problem (Again)

- Solve $A\mathbf{x} = \mathbf{b}$.

Easy 1D version

- Solve $3x = 2$.

- Let $f(x) = \frac{3}{2}x^2 - 2x$.



- Then $f'(x) = 3x - 2$.

Variation/Minimization Methods

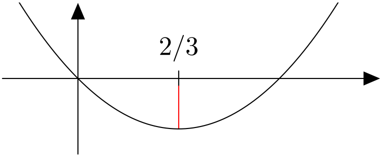
Problem (Again)

- Solve $A\mathbf{x} = \mathbf{b}$.

Easy 1D version

- Solve $3x = 2$.

- Let $f(x) = \frac{3}{2}x^2 - 2x$.



- Then $f'(x) = 3x - 2$.
- Minimum occurs when $0 = f'(x) = 3x - 2$, i.e., $3x = 2$.

Variation/Minimization Methods

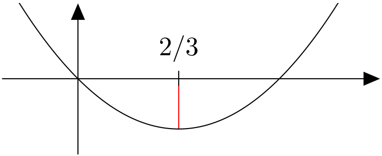
Problem (Again)

- Solve $A\mathbf{x} = \mathbf{b}$.

Easy 1D version

- Solve $3x = 2$.

- Let $f(x) = \frac{3}{2}x^2 - 2x$.



- Then $f'(x) = 3x - 2$.
- Minimum occurs when $0 = f'(x) = 3x - 2$, i.e., $3x = 2$.
- The **minimizer** is the **solution**!

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.

Convergence?

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!

Convergence?

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:

Convergence?

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:
  - Steepest Descent (Multivariable Calculus)

Convergence?

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:
  - Steepest Descent (Multivariable Calculus)
  - Conjugate Gradient (Krylov Subspaces)

Convergence?

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:
  - Steepest Descent (Multivariable Calculus)
  - Conjugate Gradient (Krylov Subspaces)

Convergence?

- Problem: Rate depends heavily on eigenvalue structure.

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:
  - Steepest Descent (Multivariable Calculus)
  - Conjugate Gradient (Krylov Subspaces)

Convergence?

- Problem: Rate depends heavily on eigenvalue structure.
- Idea: Solve $MA\mathbf{x} = M\mathbf{b}$ (M invertible) instead.

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:
  - Steepest Descent (Multivariable Calculus)
  - Conjugate Gradient (Krylov Subspaces)

Convergence?

- Problem: Rate depends heavily on eigenvalue structure.
- Idea: Solve $MA\mathbf{x} = M\mathbf{b}$ (M invertible) instead.
- New problem: Find M so that MA has a better eigenvalue structure.

Variation/Minimization Methods

Problem 1

- Solve $A\mathbf{x} = \mathbf{b}$.

Problem 2

- Minimize $\frac{1}{2}A\mathbf{x}\cdot\mathbf{x} - \mathbf{b}\cdot\mathbf{x}$.

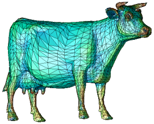
- This will work, so long as A is symmetric and positive-definitive.
- We can now have fun with optimization techniques!
- Examples:
  - Steepest Descent (Multivariable Calculus)
  - Conjugate Gradient (Krylov Subspaces)

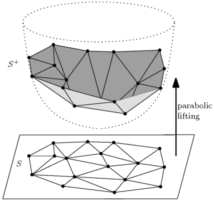
Convergence?

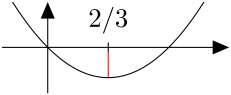
- Problem: Rate depends heavily on eigenvalue structure.
- Idea: Solve $MA\mathbf{x} = M\mathbf{b}$ (M invertible) instead.
- New problem: Find M so that MA has a better eigenvalue structure.
- **Again we can play with spectral theory!**

Conclusion

- $A\mathbf{x} = \mathbf{b}$
- $Q\mathbf{x}_{k+1} = (Q - A)\mathbf{x}_k + \mathbf{b}$





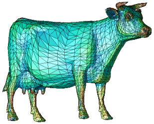


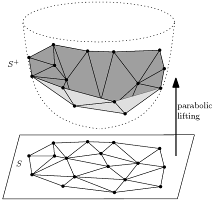
Observations

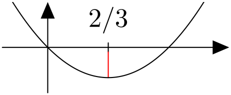
- Computers can do simple mathematics quickly.

Conclusion

- $A\mathbf{x} = \mathbf{b}$
- $Q\mathbf{x}_{k+1} = (Q-A)\mathbf{x}_k + \mathbf{b}$





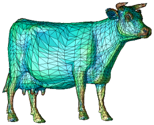


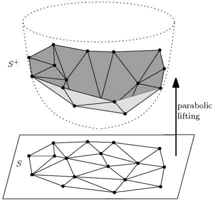
Observations

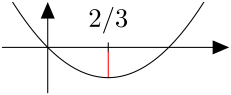
- Computers can do simple mathematics quickly.
- One must find creative solutions to make mathematics simple.

Conclusion

- $A\mathbf{x} = \mathbf{b}$
- $Q\mathbf{x}_{k+1} = (Q - A)\mathbf{x}_k + \mathbf{b}$





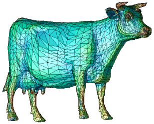


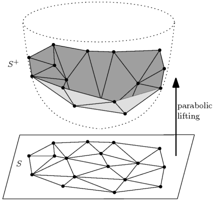
Observations

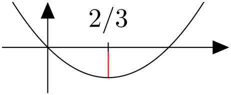
- Computers can do simple mathematics quickly.
- One must find creative solutions to make mathematics simple.
- Computational math presents new challenges. These problems can open doors to new mathematics!

Conclusion

- $A\mathbf{x} = \mathbf{b}$
- $Q\mathbf{x}_{k+1} = (Q - A)\mathbf{x}_k + \mathbf{b}$





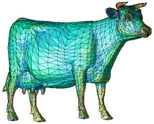


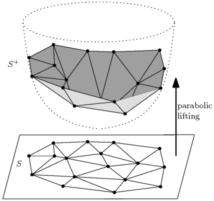
Observations

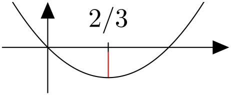
- Computers can do simple mathematics quickly.
- One must find creative solutions to make mathematics simple.
- Computational math presents new challenges. These problems can open doors to new mathematics!

Conclusion

- $A\mathbf{x} = \mathbf{b}$
- $Q\mathbf{x}_{k+1} = (Q-A)\mathbf{x}_k + \mathbf{b}$







Observations

- Computers can do simple mathematics quickly.
- One must find creative solutions to make mathematics simple.
- Computational math presents new challenges. These problems can open doors to new mathematics!

Thank you!