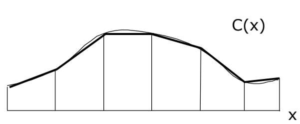
Outline

- Announcements
  - Add/drop today!
  - HWI due Friday
- Discrete Approximations
- Numerical Linear Algebra
- Solving ODE's
- Solving PDE's

Discrete Approximations

- The defining principle of numerical computing:
  Computers are finite
- This has several consequences:
  - Computers can only hold a finite amount of data (limited by memory)
  - Computers can represent integers exactly, but only over a finite range

Finite Memory Problem



- Because we can only store a finite amount of data, continuous curves or surfaces must be approximated by storing values at a finite number of points
- This leads to discretization errors
- Can improve the approximation by
  - adding more points
  - tracking higher-order properties (e.g. splines)

Finite Precision Problem

- Computers only work with integers
- To represent a real number, we use two integers:
  - $\pm m * b^p$
    - m="mantissa"
    - b=base, set by the system
    - p=exponent
  - Limited precision in both mantissa and exponent
  - Leads to roundoff errors

Finite Precision Problem

- Suppose we are working with base 10 numbers, and mantissa and exponent have 2 digits:
  - $\pm$ xx 10^{yy}
  - smallest number: $1*10^{-99}$
    - $0.5* 1*10^{-99}$ = ??? --Underflow
  - largest number: $99*10^{99}$
    - $2* 99*10^{99}$ = ??? --Overflow
  - Only 99 numbers in each decade
  - Only 200*99-1=19,799 numbers!

Finite Precision Problem

Precision	Bytes	m(bits)	eps	p(bits)	range
Single	4	24	1e-7	8	$10^{\pm 38}$
Double	8	53	1e-16	11	$10^{\pm 308}$

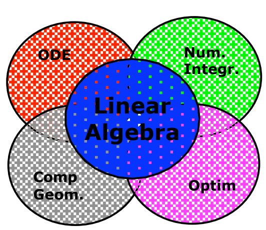
Numerical Analysis

- The study of algorithms for mathematical problems
- concerned with
  - accuracy
  - stability
  - performance

Numerical Analysis

- Three big areas (i.e. physics)
  - Linear algebra
  - ODE's/PDE's
  - Optimization problems
- Other topics
  - Computational geometry
  - Numerical integration

Numerical Linear Algebra



Numerical Linear Algebra

- Linear Systems
- Matrix Factorizations
- Eigenproblems

Solving Linear Systems

$$\begin{aligned} 2x + 3y &= a \\ 1x + 5y &= b \end{aligned}$$

$$\begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Solving Linear Systems

$$\begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ 0 & 7/2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b - a/2 \end{bmatrix}$$

Gaussian Elimination

- This procedure is known as "Gaussian Elimination"
- for j=1:m-1
  1. Divide row j by its jth entry
  2. Subtract row j from rows j+1 through m

Gaussian Elimination

- GE is also known as "LU factorization"
  - A=LU where L is lower triangular, U is upper triangular
  - Ax=b
  - LUx=b
  - Solve Ly=b for y, then Ux=y for x

$$L = \begin{bmatrix} 1 & 0 \\ 1/2 & 1 \end{bmatrix}$$
$$U = \begin{bmatrix} 2 & 3 \\ 0 & 7/2 \end{bmatrix}$$

A Problem with GE

$$\begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 10^{20} \\ 0 & 1 - 10^{20} \end{bmatrix}$$

- $1\text{-}10^{20}$--Big number-small number
  - difference is less then EPS
  - round to -10^{20}
  - could lead to large errors
- Solution: pivot rows

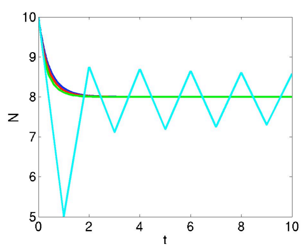
ODE/PDE

$$
\begin{aligned}
\frac{dN}{dt} &= f(N(t),t) \\
\frac{N(t+\Delta t)-N(t)}{\Delta t} &= f(N(t),t) \\
N(t+\Delta t)-N(t) &= \Delta t f(N(t),t)
\end{aligned}
$$

Either $\begin{cases} N(t+\Delta t) = N(t)+\Delta t f(N(t),t) \\ N(t+\Delta t) = N(t)+\Delta t f(N(t+\Delta t),t+\Delta t) \end{cases}$

- Bigger Δt means we can get solution in fewer steps, but
- If Δt is too big, then solution will be inaccurate

ODE/PDE



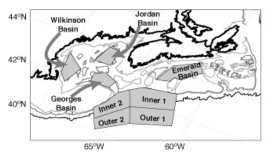
$$
\begin{aligned}
\frac{dN}{dt} &= 2N\left(\frac{8-N}{8}\right) \\
N(0) &= 10
\end{aligned}
$$

ODE/PDE

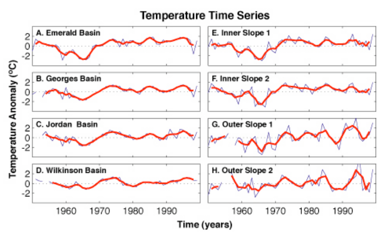
- Solutions involve a trade-off between
  - simple computation/small Δt
  - expensive computation/big Δt
    - includes implicit methods, which involve solving linear systems

Principal Components

- Example: Temperature in NW Atlantic

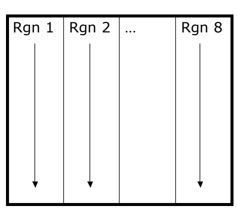


Principal Components



Principal Components

C =



$Cov=1/(m-1)*C^T*C$

Principal Components