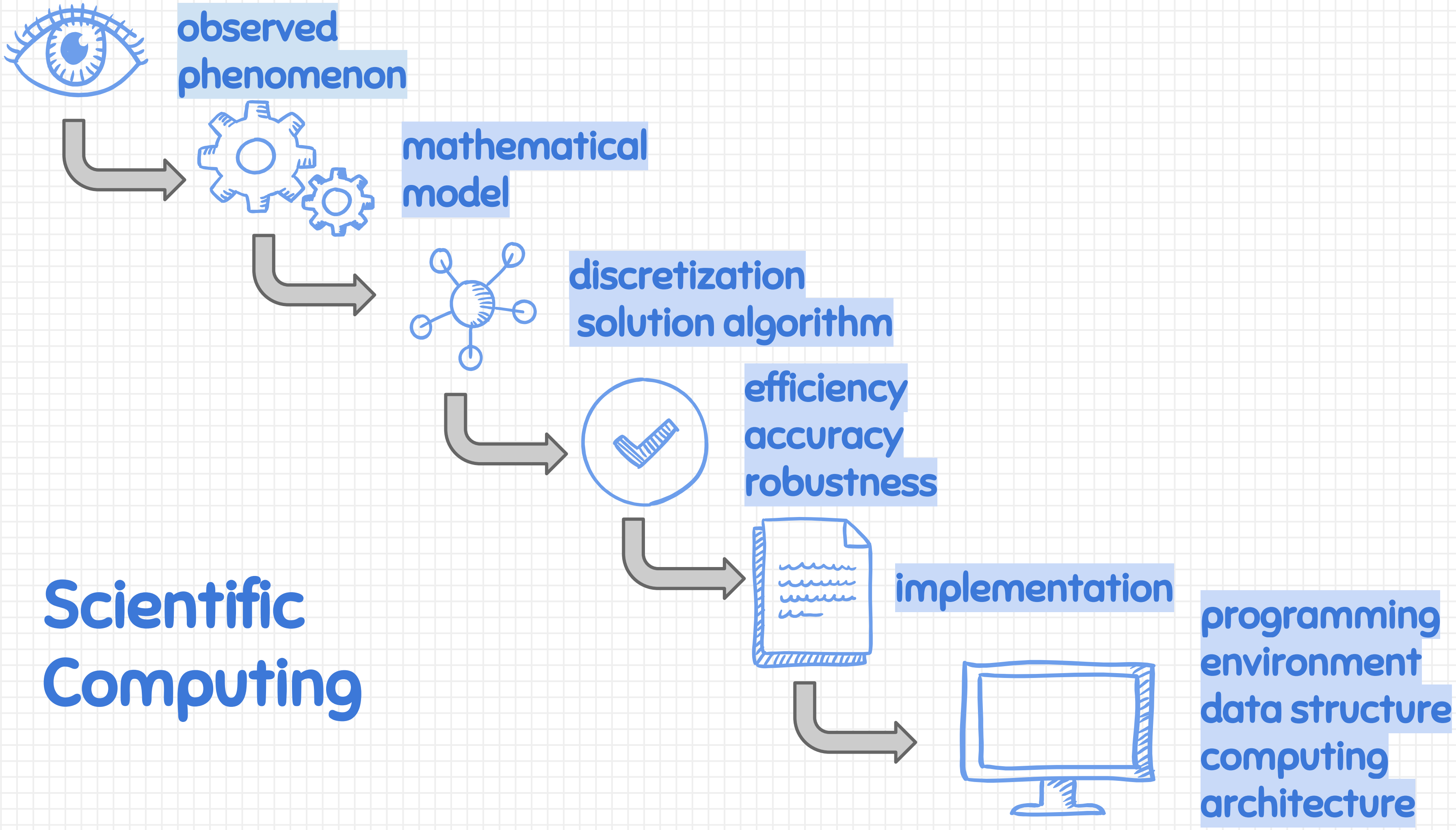
CS 6210 Fall 2016
Bei Wang

Review Lecture
What have we learnt in Scientific Computing?

Let's recall the scientific computing pipeline


observed phenomenon
mathematical model
discretization
solution algorithm
efficiency
accuracy
robustness
implementation
programming environment
data structure
computing architecture
Scientific Computing

We have accomplished our goal and leant a great deal!

Our Supercomputing Miniseries

Mark Kim (SCI): Fixed-Rate Compressed Floating-Point Arrays

Sidharth Kumar (SoC): Parallel I/O Library

Arnab Das and Vinu Joseph (SoC): Why we are not ready for Exascale Computing?

Numerical Algorithms

Error

*different types of error (absolute, relative, discretization, convergence, roundoff)

Algorithm properties: accuracy, efficiency, robustness

Floating point system

Roundoff error accumulation

*FP system: can you tell me the range of numbers a FP system can provide?

Solving Nonlinear Equation in 1 variable: $f(x) = 0$

Bisection

Fixed point iteration

*Newton's method

*Secant

*Convergence of various methods

Function minimization

Linear algebra

Vector norm

Matrix norm

*Symmetric positive definite

Orthogonal matrices

*SVD

Linear Systems Ax = b: direct methods

Backward and forward substitution

*Gaussian elimination

*LU decomposition

Pivoting

*Cholesky decomposition

Error estimation

Condition number

Linear least squares: $\min \| b-Ax \|$

Uniqueness and normal equation: $(A^TA)x = A^Tb$

Orthogonal transformation

*QR

Householder transformation

Gram-Schmidt orthogonalization

Linear Systems $Ax = b$: Iterative methods

*Splitting (stationary methods): $A = M - N$

Jacobi method

Gauss-Seidel method

Convergence of stationary methods

Gradient descent

*Conjugate gradient: explain how this works?

Eigenvalues and singular values: Ax = \lambda x; SVD

Power method for computing dominant eigenvalue and eigen vectors

*SVD

Best lower rank approximation

*Geometric intuition behind SVD

Least squares via SVD

QR for eigenvalues

Nonlinear systems $f(x) = 0$ and optimization

Newton's method

Unconstrained optimization

Taylor's series

Gradient descent

Linear search

Quasi-Newton

Polynomial interpolation f(x) = \Sigma c phi(x)

*Piecewise linear

Piecewise constant

Monomial interpolation

*Lagrange interpolation

Divided difference (coefficients) f[x_i....x_j]

*Error

*Chebyshev interpolation

Interpolating derived values f', f''

Piecewise polynomial interpolation

Broken line

Piecewise Hermite interpolation

Cubic spline

Parametric curves

Best approximation

Continuous least squares approximation

Orthogonal basis function: Legendre polynomial

Weighted least squares

*Chebyshev polynomial: geometric intuition

Numerical differentiation

*Taylor series

2 point, 3 point, 5 point formula

Richardson extrapolation

Using Lagrange polynomial interpolation

Roundoff errors

Numerical integration

Quadrature rule: $\sum a_j f(x_j)$

*Basic rules (trapezoidal, Simpson, midpoint)

Error

*Composite rules (similar to piecewise polynomial interpolation)

Gaussian quadrature

You should be proud of what we've accomplished together!

More on the final

Notes on final exam

Open book, open notes, close internet

Please bring your calculator (recommended); TA will have a calculator that you can borrow, if needed.

20 T/F questions (2 points each)

5-10 questions that require derivation

10 T/F question for extra credits

1 extra derivation question for extra credits

Possible topics of interests (see * from topic review)

- Errors
- IEEE standard
- Fixed point methods
- GE
- Least squares
- SOR
- CG
- SD
- QR
- Condition number

- Polynomial interpolation
- SVD
- Linear systems of equations
- Taylor series
- Unconstrained optimization
- LU
- Chelosky decomposition
- Normal equations
- ...

Example questions

1. In unconstrained optimization, a necessary condition for having a global min at point x is for x being a critical point (T/F)?

2. Give polynomial interpolation to some data using different interpolation schemes.

Answer to the questions

In unconstrained optimization, a necessary condition for having a global min at point x is for x being a critical point (T/F)?

False: Page 260. A necessary condition for having a local minimum at x is that x be a critical point and that the symmetric Hessian matrix being positive semidefinite.

Example questions

3. Consider the linear system given below with $a, b > 0$:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a & b \\ b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

a) If $a \approx b$, what is the numerical difficulty in solving the linear system?
b) Suggest a numerically stable formula for computing $z = x + y$, given a and b.

Example questions

5. Consider the following:

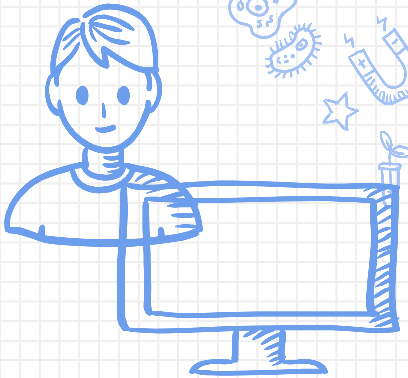
   (a) **(8 points)** Suppose you are using the trapezoidal rule to approximate an integral over an interval $[a, b]$. If you wish to obtain a more accurate approximation of the integral, which will gain more accuracy: (1) dividing the interval in half and using the trapezoidal rule on each subinterval, or (2) using Simpson's rule on the original interval? Note that either approach will use the same three function values, at the endpoints and the midpoint of the original interval. Support your answer with an error analysis. Test your conclusion experimentally with a simple integral example of your choice.

Revisit:
The Future of Scientific Computing 50 years from now

[Trefethen 2000]

1

Speech and graphics everywhere



We'll talk to computers more than type to them, and they'll respond with pictures more often than numbers.

Computer Graphics, Visualization, HCI, 3D images

2 Fully intelligent, adaptive numerics

Numerical computing will be adaptive, iterative, exploratory, intelligent – computational power will be beyond your wildest dreams.

Everything is embedded in an iterative loop, problems solved atop an encyclopedia of numerical methods

3 Loss of determinism

Determinism in numerical computing will be gone.

It is not reasonable to ask for exactness in numerical computation...we may not ask for repeatability either.

4 Floating point arithmetic: best general purpose approximation

The importance of floating point arithmetic will be undiminished.

128 bit plus word lengths, most numerical problems can not be solved symbolically still, still need approximations.

5

The quest for speed in matrix multiplication

Linear systems of equations will be solved in time $O(N^{2+e})$

Complexity of matrix multiplication = complexity of "almost all" matrix problems: inverse, determinants, solve linear systems...

How fast can we multiply two n by n matrices? Standard $O(N^3)$.

Strassen's algorithm $O(N^{2.81})$. Coppersmith and Winograd's algorithm $O(N^{2.38})$...Is $O(N^2)$ achievable?

6

Multipole methods

Multipole methods and their descendants will be ubiquitous.

Speed up the calculation of long-ranged forces in the n-body problem. Large-scale numerical computations rely more on approximate algorithms...more robust and faster than exact ones.

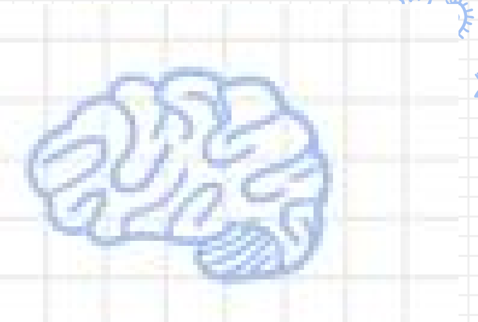
7

seamless interoperability

The dream of seamless interoperability will have been achieved.

No separation between numerical and symbolic calculations, work across different discretizations and grids, removing humans from the loop.

8 massive parallel computing

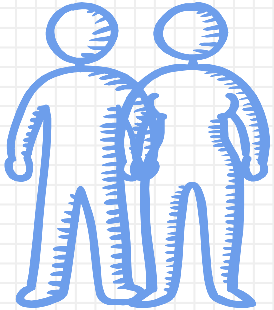


The problem of massively parallel computing will have been blown open by ideas related to human brain.

Understanding human brain and its implications for computing.

9

New programming methods

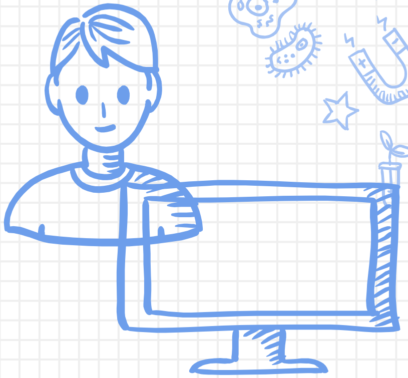


Our methods of programming will have been blown open by ideas related to genomes and natural selection.

Think digitally about the evolution of life in earth.

What's your prediction of the future of scientific computing?

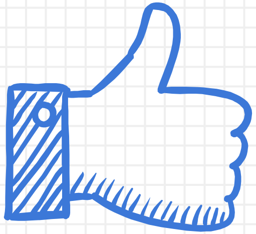
1. Supercomputing taking over!
2. Nano computing
3. Personal computing
4. Crowdsourcing
5. Human-computer interaction
6. Big data
7. Visualization needs
8. Web computing
9. Blurry boundary between SC and machine learning
10. Security
11. Your laptop and your other multimedia devices; internet of things



Take home message

1. **Think Big**: how SC could transform your research?
2. **Keep your eyes open**: identify the newest advancement in SC.
3. **Master the fundamentals**: practice makes perfect.
4. **Have some fun while learning!**

TA Friday (12/8) review hour: 1:30 p.m - 2:30 pm



THANKS!

Extra Notes

So it goes.

Credits

Special thanks to all the people who made and released these awesome resources for free:

- ✘ Presentation template by SlidesCarnival
- ✘ Photographs by Unsplash