Numerical Simulation for Graphics and Animation

Week 2: Primer

Orestis Vantzos
ovantzos@cs.technion.ac.il
Scientific Computing

- Dawn of Time: Mainframes + Assembly

System/370 Reference
Scientific Computing

- Heroic Age: (Clusters of) PCs + Fortran/C/C++
Scientific Computing

• Now: Multicore CPUs/GPUs + Scientific IDEs
The 80-20 Rule

- Most Applications ← Few Algorithms

“80%” of time spent on:
- Linear Systems
- Singular/Eigenvalue Problems
- Finding Roots
- Linear Optimisation
- Convex Optimisation
The 80-20 Rule

- Most Applications ← Few Algorithms ← Linear Algebra

Highly Optimised Libraries:
- BLAS Level 1 (Vector Ops)
- BLAS Level 2 (Matrix-Vector)
- BLAS Level 3 (Matrix-Matrix)
- LAPACK (Matrix Algorithms)
Why Linear Algebra?

- Scientists/Engineers ❤️ Linearity

Balance Laws

\[ F[\phi(\vec{x}, t)] = 0 \]

Variational Principles

\[ \min G[\phi(\vec{x}, t)] \]

Approximate

Linear Problems

\[ f + F'[\phi] = 0 \]

Convex Programs

\[ \min \{ g + G''[\phi] + \frac{1}{2} G'''[\phi, \phi] \} \]

Discretise

Linear Systems

\[ A_h \cdot \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_n \end{pmatrix} = b_h \]
Why Linear Algebra?

- Computers ❤ Arrays
Why Linear Algebra?

- Computers ❤ Arrays
Core Algorithms

• Linear Systems: Matrix Decomposition

Factorise $A = A_1 A_2 \ldots$

• LU Decomposition
• Cholesky Decomposition
• QR Decomposition
• Schur Decomposition
• SVD Decomposition
Core Algorithms

- Bonus Round: Singular Value Decomposition

Find orthogonal $U$ & $V$, diagonal $D$, such that $A = UDV^T$.

Everything you want to know about $A$
- Spectral radius and Condition number
- Rank, Range and Null-space
- Principal Component Analysis (PCA)
- Pseudoinverse

Numerical Recipes
by Press et al.
Core Algorithms

• Linear Systems: Iterative Methods

Split $A = A_1 + A_2$

• Jacobi Iteration
• Gauss-Seidel Iteration
• SOR Iteration

Krylov Subspace $\{y, Ay, A^2y, \ldots\}$

• Conjugate Gradients
• Biconjugate CG (BICG)
• GMRES

Preconditioners $LA(Rz) = Ly$

Solve $Ax = y$ for $x$, with sparse $A$.

*Templates for the Solution of Linear Systems by Barrett et al.*
Core Algorithms

• Linear Systems: Fast Solvers

For Structured Matrices
• Band Matrices, esp. Tridiagonal
• Multigrid
• Fast Multipole Method
• Fast Fourier Transform
• Fast Poisson Solvers

Solve $Ax = y$ for $x$, with special $A$.

Multigrid by Trottenberg et al.
Core Algorithms

- Convex Optimisation (with Constraints)

Linear Programming
- Simplex Method

Unconstrained Problems
- Gradient Descent
- Newton Method
- Quasi-Newton Methods

Constrained Problems
- Active-Set
- Augmented Lagrangian

Minimise convex $F(x)$ with $Ax \geq b$. 

Convex Optimization by Boyd & Vandenberghe
Numerical Optimization by Nocedal & Wright