Ideas and Algorithms

- Not perfect
  - Mathematics courses teach analytical techniques
  - Engineering courses work on real problems
- Not efficient
  - Mathematics courses analyze numerical algorithms
  - Engineering and computer science implement the software

- Formula-based → solution-based

Introduction

- Modeling: applied mathematics
  - Identify the key quantities in the problem
  - Connect them by differential equations or matrix equations
  - Study of special functions
  - Constructing the equations of equilibrium and motion (balance equations)
- Solving: scientific computing
  - Those equations
  - Numerical analysis of the algorithm to test its accuracy and stability
  - Solving steady state and time-dependent matrix and differential equations

Four Simplifications (1)

- Nonlinear → Linear
  - Hooke's law in mechanics: displacement $\propto$ force
  - Ohm's law in networks: current $\propto$ voltage difference
  - Scaling law in economics: output $\propto$ input
  - Linear regression in statistics: a straight line or a hyperplane can fit the data
  - Physical nonlinearity vs. geometric nonlinearity
  - Curvature formula in the bending of a beam: $\dfrac{u''}{\sqrt{\left(1+\left(u'\right)^2\right)^3}} \xrightarrow{\text{small } u'} u''$
- Continuous → Discrete
  - Laplace's equation: gradient, divergence, curl
  - To discretize the continuous equation into Ku = f (Ch.3)
    - Finite element method
    - Finite difference method

Four Simplifications (2)

- To solve for u (Ch.7)
  - Direct elimination
  - Iterations with preconditioning
- For very large K, good solution algorithm (multigrid) better than a supercomputer
- Stability for the discretization in the initial-value problems (Ch.6)

- Multidimensional → One-dimensional
  - Separation of variables: $u(x, t) = A(x)B(t)$
  - Spectral method (Ch.5)
- Variable coefficients → Constants
  - Fourier transform (Ch.4)
    - Sines, cosines, exponentials
    - physical domain (x, t) → frequency domain (k, ω)

Computational Science and Engineering

- Large scale computation?
- Fundamentals of scientific computing with short codes to implement the key concepts
  - Symmetric positive definite matrices A^TCA: stiffness, conductance, multigrid matrices
  - A: geometry, C: physical properties
  - Spring-mass systems, networks, least squares, differential equations like div(c grad u)=f

Applications (1)

- Computational Engineering
  - Finite element method (computational mechanics)
  - Reliable and efficient for the mechanics of structures
  - 1D introduction in Section 3.1, 2D code in Section 3.6
  - Differential equations in weak form integrated against test functions
  - Stresses and displacements approximated by polynomials
  - Fluid-structure interactions, high-speed impact?
- Computational Electromagnetics
  - SPICE codes: solve network problems with many circuit elements
    - Variations of Newton's method $J\Delta u=-g$ (Section 2.6) for $g(u)=0$
  - Finite differences on a staggered grid (Yee's method) for Maxwell's equations
  - Helmholtz equation separates variables: indefinite equation with negative eigenvalues for high frequencies

Applications (2)

- Computational Physics and Chemistry
  - Large computation over many time steps, resolution for high accuracy → multigrid (Section 7.3)
  - Stability and accuracy over many periods: $mu''=f(u)$ in Section 2.2 → atomic oscillation, orbits in space
  - Computational fluid dynamics (Sections 6.6-6.7): convections with diffusion, divergence-free projection to solve the Navier-Stokes equations
- Computational Biology
  - Differential equations (part of biology)
  - Another large part: networks of nodes and edges (Section 2.4), combinatorial problems, probability, very large amount of data
  - Data mining: probability, statistics, signal processing, optimization

Applications (3)

- Computational Simulation and Design
  - Quicker and easier than physical experiments
  - Series of tests with varying parameters to approach an optimal design
  - Numerical simulations reliable? Problem of quantifying uncertainty
  - Validation (solving the right equations): modeling
  - Verification (solving the equations right): estimate of numerical error
    - Discretization: $O((\Delta x)^2)$, $O((\Delta t)^4)$
    - derivatives→difference, functions→polynomials, integrals→finite sums
  - Sensitivity analysis: d(output)/d(input) Section 8.7
  - Ill-posed inverse problem vs. well-posed forward problem

Applications (4)

- Computational Finance
  - Black-Scholes equation: Ito’s lemma, Brownian motion (Section 6.5)
  - Constant volatility: reduce to the heat equation $u_t=u_{xx}$
  - Variable volatility: finite differences
  - Monte Carlo methods
- Computational Optimization
  - Compute a minimum or maximum subject to constraints
  - Lagrange multipliers: sensitivity of the solution to changes in the data
  - duality for least squares (Section 8.1)
  - Calculus of variations: function of x or t
    - Lagrangians and Hamiltonians in mechanics

Basics of Scientific Computing (1)

- Matrix equations (problems of linear algebra)
  - lu(A): $Ax = b$ by elimination and $A=LU$ (triangular factors)
  - eig(A): $Ax=\lambda x$ leading to diagonalization $A=S\Lambda S^{-1}$ (eigenvalues in Λ)
  - qr(A): $Au \approx b$ by solving $A^TA\hat{u}=A^Tb$ with orthogonalization $A=QR$
  - svd(A): Best bases from the Singular Value Decomposition $A=U\Sigma V^T$

- Differential equations in space and time: boundary and initial values
  - Explicit solution by Fourier and Laplace transforms
  - Finite difference solutions with tests for accuracy and stability
  - Finite element solutions using polynomials on unstructured meshes
  - Spectral methods of exponential accuracy by Fast Fourier Transform

Basics of Scientific Computing (2)

- Large sparse systems of linear and nonlinear equations
  - Direction solution by reordering the unknowns before elimination
  - Multigrid solution by fast approximation at multiple scales
  - Iterative solution by conjugate gradients and MINRES
  - Newton’s method: linearization with approximate Jacobians

Applied Mathematics for Computational Design and Analysis
전산설계 및 해석을 위한 응용수학

Ch	Contents	
1	Applied Linear Algebra	○
2	A Framework for Applied Mathematics	○
3	Boundary Value Problems	○
4	Fourier Series and Integrals	
5	Analytic Functions	
6	Initial Value Problems	
7	Solving Large Systems	
8	Optimization and Minimum Principles	○

Advanced Numerical Methods in Engineering
수치해석특론

Ch	Contents	
1	Applied Linear Algebra	
2	A Framework for Applied Mathematics	
3	Boundary Value Problems	
4	Fourier Series and Integrals	○
5	Analytic Functions	
6	Initial Value Problems	○
7	Solving Large Systems	○
8	Optimization and Minimum Principles	
	Variational Method	○