Introduction to Numerical Modeling and Device Modeling

Lecture 1

Special Topics: Device Modeling

Outline

• Numerical modeling
  – Terminology
  – Basic operations
  – Examples of simple algorithms
  – Sources of error
• Modeling physical systems
  – Main steps and considerations

Introduction

• Most real-world applications lead to mathematical problems which cannot be solved with exact formulas, or analytically
• A common approach is to reduce a problem to special cases and simplified situations, and study those in detail
• The aim is to uncover generally applicable concepts and properties, which can guide us in more difficult problems

Numerical modeling: terminology

Sequences and series

• A sequence is a (possibly infinite) collection of numbers lined up in some order
• A series is a (possibly infinite) sum
  – Example: Taylor’s series

\[ T_n(x) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(x_0) (x-x_0)^k \]

\[ f(x) = f(x_0) + \frac{1}{2!} f'(x_0)(x-x_0) + \frac{1}{3!} f''(x_0)(x-x_0)^2 + \ldots + \frac{1}{n!} f^{(n)}(x_0)(x-x_0)^n \]

Example: sine function approximated by Taylor series expansion

The approximation of \( f(x)=\sin x \), around \( x=0 \) by the Taylor series becomes more accurate (better over a larger interval around the center) with the number of terms increasing from 1 to 13
Numerical modeling: terminology

Convergence and divergence
• Sequence \((a_j)\) with \(j=[0,\infty]\) is said to be \(\varepsilon\)-close to a number \(b\) if there exists a number \(N \geq 0\) (it can be very large), such that for all \(n \geq N\), \(|a_j - b| \leq \varepsilon\).
• A sequence \((a_j)\) with \(j=[0,\infty]\) is said to converge to \(b\) if it is \(\varepsilon\)-close to \(b\) for all \(\varepsilon > 0\) (however small).
• Notation: \(a_j \to b\), or \(\lim_{j \to \infty} a_j = b\).
• If a sequence does not converge, it diverges.

Examples: converging and diverging sequences
• Unbounded sequences, i.e., sequences that contain arbitrarily large numbers, always diverge.
• \(e^{-n} \to 0\) as \(n \to \infty\), and convergence is very fast.
• \(n/(n + 2) \to 1\) as \(n \to \infty\), and convergence is rather slow.
• \(\log(n) \to \infty\) as \(n \to \infty\), so the sequence diverges.

Numerical modeling: terminology

Convergence and divergence
• Define the \(N\)th partial sum \(S_N = a_1 + a_2 + \ldots + a_n = \sum_{j=1}^{n} a_j\).
• The series \(\sum a_j\) converges if the sequence of partial sums \(S_N\) converges to some number \(b\) as \(N \to \infty\).
• Notation: \(\sum a_j = b\).
• If a series does not converge, it diverges.

Examples: converging and diverging series
• Geometric series converges for \(|x|<1\)
  \[\sum_{j=0}^{\infty} x^j = 1 + x + x^2 + \ldots = \frac{1}{1-x}\]
• Harmonic series diverges
  \[\sum_{j=1}^{\infty} \frac{1}{j} = 1 + \frac{1}{2} + \frac{1}{3} + \ldots\]
• Series \(\sum_{j=0}^{\infty} (-1)^j = 0 + 1 + 0 + 1 + \ldots\) also diverges.

Numerical modeling: terminology
• Discretization is the process of transferring continuous functions, models, and equations into discrete counterparts.
  • Usually carried out as a first step toward making them suitable for numerical evaluation and implementation as a computer program.
• Iteration (from the Latin iterare, “to plow once again”) is a repetition of a mathematical or computational procedure applied to the result of a previous application, typically as a means of obtaining successively closer approximations to the solution of a problem.
• Iterative methods are used to produce approximate numerical solutions to mathematical problems.
• When programmed, implemented through loops.
Example: Solution of an equation $x = F(x)$ using the method of iterations

- Start with an initial approximation $x_0$, and compute the sequence $x_0 = F(x_0), x_1 = F(x_1), x_2 = F(x_2), \ldots$
- Each computation of the type $x_{n+1} = F(x_n)$ is called a fixed-point iteration: as $n$ grows we would like $x_n$ to be closer to the root
- If the sequence $\{x_n\}$ converges to a limiting value $\alpha$, then $x = \alpha$ is the root: $\alpha = F(\alpha)$

**Converging iterations**

**Diverging iterations**

**Numerical modeling:**

- Interpolation
- Derivatives
- Integration
- Root finding
  - Nonlinear equations
  - Differential equations

**Interpolation**

- Experiment is usually represented by a discrete set of datapoints $\{(x_i, f(x_i))\}$; interpolation is required to find the value of $f(x_k)$ for any arbitrary point $x_k$
- Often we want an analytical function describing the whole data set
- One of the most useful and well-known approaches to functions mapping over a range of values is with the algebraic polynomials

$P_n(x) = a_0 + a_1 x + \ldots + a_n x^n$

- Taylor's polynomials are good approximation at one point $x_0$

**Differentiation**

- The derivative of the function $f$ at $x_0$ is defined as

$f'(x) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$

- General approximation would be to compute for small $h$:

$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$

- Two problems with this approach using numerically:
  - Very small $h$ - division by zero
  - Subtract two numbers which only differ by a small amount

- Can reduce the error by using both forward- and backward differences

\[
 f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} + O(h^2)
\]

- Neglecting $O(h^2)$ leads to error $\approx h$

- Start with Taylor series for $f(x_0 + h)$ and express first derivative:

\[
 f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \left( \frac{h}{2!} f''(x_0) + \frac{h^2}{3!} f'''(x_0) + \ldots \right)
\]

**Forward difference for $h>0$, backward difference for $h<0$**
Numerical modeling: integration

• Evaluation of definite integral of a function as numerical quadrature:
  \[ \int_a^b f(x)dx = \sum_{i=1}^{n} f(x_i)\Delta x_i \]
  Divide interval into multiple slices, and find area under the curve over interval \([a,b]\)
• Simplest approximation with rectangles; slightly more sophisticated: trapezoid rule, Simpson’s rule

\[ \sum f(x_i)\Delta x_i = \int_a^b f(x)dx \]

The \([a,b]\) is divided into \(N\) intervals, equally spaced by 
\[ h = x_i - x_{i-1} \]

• Trapezoid rule:
• Simpson’s rule: approximate function with parabola over each interval

\[ \int_a^b f(x)dx = \sum_{i=0}^{N} [f(a) + f(a+h) + 2f(a + 2h) + \ldots + f(b)] \]

Use 3 points for each interval, number of intervals should be even

Numerical modeling: root finding

• Finding a root, or solution, of an equation of the form \(f(x) = 0\), for a given function \(f\)
  – Also called a zero of the function \(f\)
• Selection of a particular numerical method depends on the equation type: linear, non-linear, differential (ordinary or partial)
• System of equations

Numerical modeling: nonlinear equations

• Newton’s (or the Newton-Raphson) method exploits derivatives of \(f(x)\) to accelerate conversion to \(f(p)=0\)
• Start with Taylor’s expansion of the \(f(x)\) about \(p_0\), which is close to the root \(p\), and neglect terms of second order, since \(p - p_0\) is small
  \[ p = p_0 - \frac{f(p_0)}{f'(p_0)} \]
  \[ p_1 = p_0 - \frac{f(p_0)}{f'(p_0)} \]
• After \(n\) iterations:
  \[ p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} \]
  for \( n \geq 1 \)
• Stop when rel. error objective is reached

Wolfram website: http://www.wolframalpha.com

Numerical modeling: differential equations

• Principally, divided into two categories:
  – the ordinary differential equations (ODE) which contain functions of only one independent variable
  – the partial differential equations (PDE) having functions of several independent variables
• ODEs can themselves be grouped into subcategories according to their order \(n\) and whether they are linear
• Most general form of ODE: \(F(x,y,y',y'', \ldots ,y^{(n)})=0\)
Numerical modeling: differential equations

- In solving differential equation we make distinction between the general and a particular solution
  - Example: equation \( y' = y \) has the general solution \( y = Ce^x \), where \( C \) is an arbitrary constant
- Every \( n \)th order ODE has \( n \) integrating constants, that can be either the initial values, or boundary conditions (determined at the integration limits) leading to initial value or boundary condition problems
  - Same for first-order differential equations

\[ y'(x) = -2xy(x) \]
\[ y(0) = 5 \]
\[ [0,3] \]

Sample solution: [http://www.wolframalpha.com](http://www.wolframalpha.com)

Numerical modeling: differential equations

- Euler method: first order ODE with initial value
  - Consider the problem of calculating the shape of a curve \( f(x) \): it starts at a given point \( x_0 \) (initial value) and satisfies a given differential equation, so we can calculate the derivative at \( x_0 \) (find tangent)
  - Take a small step \( h \) along the tangent and repeat the procedure

\[
\begin{align*}
  y_{n+1} & = y_n + hf(x_n, y_n) \\
  y(0) & = 5
\end{align*}
\]

\[ y'(x) = f(x, y(x)), \quad y(x_0) = y_0 \]

Numerical modeling: differential equations

- Finite differences method - finite differences approximate the derivatives
- Finite elements method subdivides a large problem into smaller, simpler, parts - finite elements (FE’s)
- The simple equations that model these FE’s are then assembled into a larger system of equations that models the entire problem
- FEM then uses variational methods to approximate a solution by minimizing an associated error function

Numerical modeling: matrices

- Two principal computational problems are associated with matrices
  - Systems of linear equations: \( Ax = b \)
  - Eigenvalue problems: \( Ax = \lambda x \)
- Calculation of determinants (\( \det A, \det (A - \lambda I) \))
- Typical problems: large round-off errors (iterative approaches such as Gaussian elimination), ill-conditioned matrices (\( \det A \) close to 0, solution is unstable)
Numerical modeling: Monte Carlo simulations

- Multiple real-life processes are stochastic in nature: characterized by probability distribution and expectation value.
- Monte Carlo simulations are designed to repeat the same process very large number of times to obtain the solution close to the expectation value.
- Rely heavily on random number generators.
  - Eventually the sequence of numbers from the generator will repeat itself.

Numerical modeling: errors

- The accuracy is always lower than in analytical solution.
- Sources of error:
  - Errors in given input data, operator error, etc.
  - Simplification error
  - Rounding (or chopping) error during computation
  - Truncation error (e.g., an infinite series is broken off after a finite number of terms)
  - Discretization error

Modeling physical system

- Model is defined through:
  - Objectives: problem formulation
  - A set of governing equations
  - Geometry
  - Boundary conditions (or initial values)
  - Material properties and other parameters
- Most models do not have simple analytical solution.
  - Example: calculation of the motion of a cannon ball in two dimensions if include drag force.

Modeling physical system: Problem formulation

- The general goal of modeling is to improve understanding of physical processes.
- It is critical to define the specific goals of a problem formulation.
- Example: drug delivery through skin, using a patch.
- The general problem is quite complex.

Modeling physical system: Problem formulation (image)

Goal: to study the drug transport primarily in the skin region.

Goal: to study the drug transport inside the patch as well as in the skin.

Modeling physical system: Geometry

- The computational domain is the chosen region of the physical domain (actual geometry) where computations will be performed.
- The larger the computational domain, the more computation is required.
- Choice of 1D vs 2D vs 3D
- How can symmetry be used to reduce the domain?
- What regions need to be included.
Modeling physical system: Geometry

Example: arterial tissue surrounding blood

Modeling physical system: Governing equations

- We need as many equations to describe the model as there are distinct “physics”
- Example: to model transport processes in a biomedical system have to consider three most distinct “physics” - fluid flow, heat transfer, and mass transfer. The following equations should be included:
  - Conservation equation for total mass (continuity equation)
  - Momentum conservation equations (fluid flow equations)
  - Energy conservation equation (heat transfer equation)
  - Mass species conservation equation (mass transfer equation)

Modeling physical system: Governing equations

- Next simplification step: what terms should remain in the governing equations
- Transient or steady-state (time dependence)
- Gradient terms and resultant transport vs uniform distributions
- Generation (depletion) or source (sink) terms
- If approximating functions – how many terms to retain?

Modeling physical system: Boundary conditions

- Boundary conditions are statements describing how the process relates to its surroundings
  - alternatively, initial conditions are specified
- How many boundary conditions are needed
  - all of the external boundaries of the computational domain need boundary conditions for each of the primary variables
- What they are specifically – opportunity for simplification

Modeling physical system: Boundary conditions

- Less realistic: the patch will lose drug while tissue will gain

Modeling physical system: Material properties

- Are we likely to find the property data needed for the exact material that we need?
- Can we estimate the property using empirical predictive equations
- Possible substitutions, simplifications, and their effect on the accuracy of the solution
- Uniform vs. non-uniform properties
- Best decided at the problem formulation stage
Summary

- Most real-world applications lead to mathematical problems which cannot be solved with exact formulas, or analytically
- A common approach is to reduce a problem to special cases and simplified situations, and study those in detail
- Errors accumulate through every simplification step at the model level, and at every approximation related to model implementation

References