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-word 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

Introduction

- The simplified analytical model (e.g., set of equations) for problem is converted into a *numerical* model
  - Can be solved in a finite number of basic arithmetic operations
  - Approximation of an analytical model (error)
- A numerical model is solved with *numerical methods*, introducing additional errors
- Machine representation: computers have a limit on how small or large a number can be

Numerical modeling: terminology

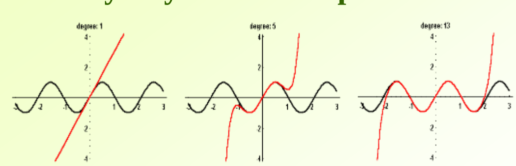
<u>Sequences and series</u>

- 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

$$T_n(x) = \sum_{k=1}^{n} \frac{1}{k!} f^{(k)}(x_o)(x - x_o)^k =$$

$$f(x_o) + f'(x_o)(x - x_o) + \frac{1}{2} f''(x_o)(x - x_o)^2 + \ldots + \frac{1}{n!} f^{(n)}(x_o)(x - x_o)^n$$

Example: sine function approximated by Taylor series expansion



Images from: http://www.sosmath.com/calculus/taypol/taypol05.html

The approximation of $f(x)=\sin x$, around $x_0=0$ 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,∞] is said to be ε-close to a number *b* if there exists a number N ≥ 0 (it can be very large), such that for all n ≥ N, $|a_j - b| \leq \varepsilon$
- A sequence (a_j) with j=[0,∞] is said to *converge* to *b* if it is ε-close to *b* for all ε> 0 (however small)
- Notation: $a_j \rightarrow b$, or $\lim_{j\rightarrow\infty} a_j = b$
- 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
- $e^{-n} \rightarrow 0$ as $n \rightarrow \infty$, and convergence is very fast
- $n/(n + 2) \rightarrow 1$ as $n \rightarrow \infty$, and convergence is rather slow
- $\log(n) \rightarrow \infty$ as $n \rightarrow \infty$, so the sequence diverges

Numerical modeling: terminology

Convergence and divergence

- Define the *N*th partial sum $S_N = a_0 + a_1 + ... + a_n = \sum_{j=0}^{N} a_j$
- The series $\sum_j a_j$ converges if the sequence of partial sums S_N converges to some number *b* as $N \rightarrow \infty$
- Notation: $\sum_{j=0}^{\infty} a_j = b$
- If a series does not converge, it *diverges*

Examples: converging and diverging series

- Geometric series converges for |x|<1

$$\sum_{j=0}^{\infty} x^j = 1 + x + x^2 + ... = \frac{1}{1-x}$$

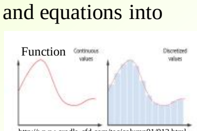
- Harmonic series diverges

$$\sum_{j=1}^{\infty} \frac{1}{j} = 1 + \frac{1}{2} + \frac{1}{3} + ...$$

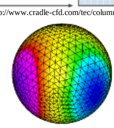
- Series $\sum_{j=0}^{\infty} (-1)^j = 0 + 1 + 0 + 1 + ...$ also diverges

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



http://www.cradle-cfd.com/tec/column01/012.html



Surface mesh

http://mfem.org/examples/

Numerical modeling: terminology

- *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

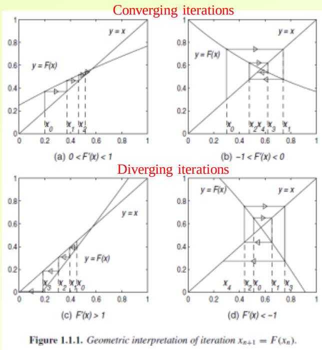


Figure 1.1.1. Geometric interpretation of iteration $x_{n+1} = F(x_n)$.

Example: Solution of an equation $x=F(x)$ using the method of iterations

- Start with an initial approximation x_0, and compute the sequence

$x_1=F(x_0)$ $x_2=F(x_1)$ $x_3=F(x_2)$ …

- 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 α, then x= α is the root: α=F(α)

Numerical modeling: basic operations

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

Numerical modeling: interpolation

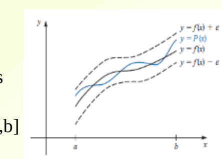
- Experiment is usually represented by a discrete set of datapoints $[x_i, f(x_i)]$; *interpolation* is required to find the value of $f(x_k)$ for any arbitrary point x_k
  - Often we want an analytical function describing the whole data set
- One of the most useful and well-known approaches to functions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

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \dots a_1 x + a_0$$

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

Numerical modeling: interpolation

- For a function defined and continuous over an interval [a,b], for each $\varepsilon>0$ there exists polynomial P(x) such that $|f(x) - P(x)| < \varepsilon$ for all x in [a,b]



- Another important reason for considering the class of polynomials in the approximation of functions is that the derivative and indefinite integral of a polynomial are easy to determine and are also polynomials

Numerical modeling: 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:

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

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

Numerical modeling: differentiation

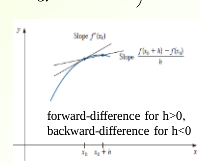
- Start with Taylor series for $f(x_0+h)$ and express first derivative:

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \left(\frac{h}{2!} f''(x_0) + \frac{h^2}{3!} f'''(x_0) + \dots\right)$$

- Neglecting ( ) leads to error ~h
- Can reduce the error by using both forward- and backward differences

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

remainder ~h^2



Numerical modeling: integration

- Evaluation of definite integral of a function as numerical quadrature:

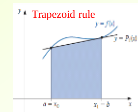
$$\int_a^b f(x)dx = \sum_{i=0}^{n} f(x_i)\Delta x_i$$

- 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

Numerical modeling: integration

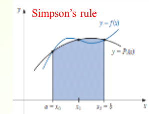
- The [a,b] is divided into N intervals, equally spaced by $h = x_i - x_{i-1}$
- Trapezoid rule:

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(x_0 = a) + f(x_0 + h) + \ldots + f(x_N = b)\right]$$



- Simpson's rule: approximate function with parabola over each interval

$$\int_a^b f(x)dx = \frac{h}{3}[f(a) + 4f(a+h) + 2f(a+2h) + $$
$$+ 4f(a+3h) + \ldots + f(b)]$$



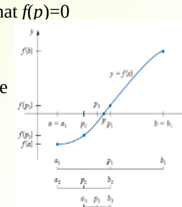
Use 3 points for each interval, number of intervals should be even

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

- *Bisection* method – the simplest and most robust; works for one or more roots
- Let $f(x)$ be a continuous function on [a,b] with $f(a)$ and $f(b)$ of opposite sign; we search for p such that $f(p)=0$
- The method calls for a repeated halving (or bisecting) of subintervals of $[a, b]$ and, at each step, locating the half containing p
- After n steps interval is reduced to $(b-a)/2^n$, continue until relative error objective is reached



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

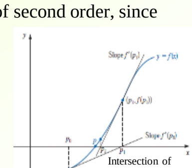
$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} \equiv p_1$$

- After n iterations:

$$p_n \approx p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for} \quad n \geq 1$$

- Stop when rel. error objective is reached

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



Intersection of tangent with x gives next point p1

Numerical modeling: differential equations

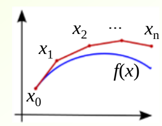
- Principally, divided into two categories:
  - the ordinary differential equations (ODE) which contain functions of only one independent variable
  - the partial differential equations (PDE) having functions of several independent variables
- ODEs can themselves be grouped into subcategories according 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 nth 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

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



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

$$y_{n+1} = y_n + hf(x_n, y_n), \quad y_n \approx y(x_n)$$ - approximate solution

Numerical modeling: differential equations

- *Runge–Kutta* method: 4th order method for ODE
- Offers much better accuracy than the Euler method (2nd degree method); can further adapt using variable step size

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)$$

$$k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

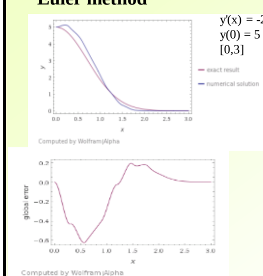
$$y_{n+1} = y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 + O(h^5)$$

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

Sample solution: http://www.wolframalpha.com

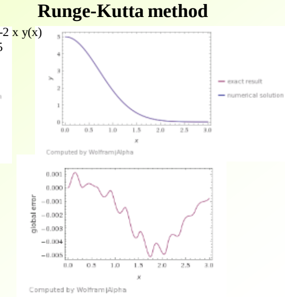
Numerical modeling: differential equations

Euler method



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

Runge-Kutta method



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: $\mathbf{Ax} = \mathbf{b}$
  - Eigenvalue problems: $\mathbf{Ax} = \lambda\mathbf{x}$
- Calculation of determinants(det$\mathbf{A}$, det$|\mathbf{A} - \lambda\mathbf{I}|$)
- Typical problems: large round-off errors (iterative approaches such as Gaussian elimination), ill-conditioned matrices (det$\mathbf{A}$ 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

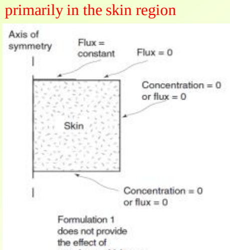
- Model is defined through
  - Objectives: problem formulation
  - A set of governing equations
  - Geometry
  - Boundary conditions (or initial values)
  - Material properties and other parameters
- Most models do not have simple analytical solution
  - Example: calculation of the motion of a cannon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

Modeling physical system: Problem formulation

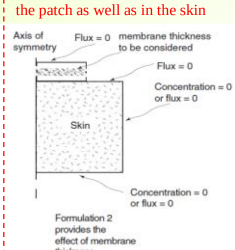
- The general goal of modeling is to improve understanding of physical processes
- It is critical to define the specific goals of a problem formulation
- Example: drug delivery through skin, using a patch
- The general problem is quite complex

Modeling physical system: Problem formulation

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



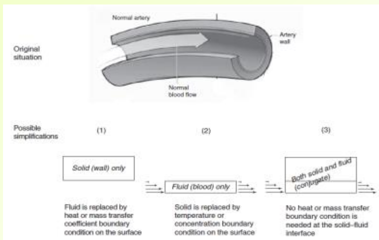
Goal: to study the drug transport inside the patch as well as in the skin



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)

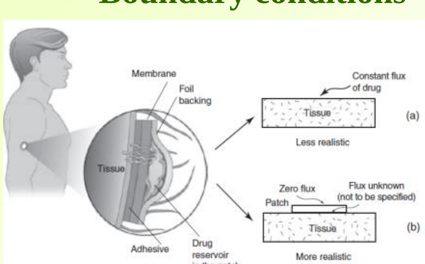
Modeling physical system: Governing equations

- 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 simlification

Modeling physical system: Boundary conditions



Less realistic

More 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-word 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

- A. Klein and A. Godunov, Introductory Computational Physics, Cambridge University Press, 2006
- R. L. Burden, J. D. Faires, Numerical Analysis, 9th edition, Brooks/Cole, Cengage Learning, 2011
- G. Dahlquist, Å. Björck, Numerical methods in scientific computing, Vol.1, SIAM, Philadelphia, 2008
- A. Datta and V. Rakesh, An Introduction to Modeling of Transport Processes, Cambridge University Press, 2010