Mathematical Modeling

- **Why** to model?
- **What** to model?
- **How** to model?
  - Ordinary Differential Equations
  - Partial Differential Equations
  - Etc.

Why to use models? Why not?

- To understand better the behavior of a system
- To predict the future development of a system or even to manipulate the system in order to achieve a desired result
- Models are usually cheaper and faster than conventional experiments
- Sometimes the only feasible way to find a solution is through modeling
  - Models can be too complicated or too restricted

78

What can be modeled?

- Mechanical structures: airplanes, buildings, skeletons
- Chemical reactions: batteries, pulp manufacturing
- Electromagnetic fields: mobile phones, human heart
- Fluid dynamics: airplanes, human heart
- Biological systems: simple population models, competition
- Not yet: macroscopic properties from atomic structure, long time weather forecasting ("never"...)

79

Modeling methods

- Differential equations: Ordinary & Partial DEs:
  - By far the most common method
  - Ample supply of numerical methods and computer programs
- Simulations: particle simulations, simulated annealing
  - Discrete dynamical models
  - Rules for interaction or changing from a state to another
- Game theory
- Statistical models
- Genetic algorithms

Things to consider

- Find the right mathematical description for the phenomenon to be modeled
  - Start with as simple a model as possible
  - Include only the most essential characters
- Decide how to analyze the model
- Try to estimate the reliability of your model AND your analysis
- Improve the model if needed
  - Change parameters
  - Add or remove something

Levels of models

- No model at all
- A model with no solutions
  - A frustrating situation, if it goes unrecognized for long
  - Mathematical analysis with existence theorems
- A model that can be solved numerically with computers
  - Problem: some numerical methods always give an answer - even when no solution exists!

82

Levels of models

- A model with analytical solutions
  - A relatively rare incidence with new, scientifically noteworthy models
    - The models that are easily analyzable have already been analyzed
    - Those that have defied analytical methods so far are usually very hard
  - Very useful as examples and test problems for computational methods.

83

Classical modeling

- Objectives
- Hypothesis – intuition, expertise
- Mathematical Formulation – experience
- Verification – hard work, testing over and over again
- Calibration – estimation, comparison
- Analysis and Evaluation – varies from mechanical work to very abstract proofs
- Usually, the modeling process starts from the top of this list and proceeds downwards until for some reason or another it is necessary to go back to a previous point

Objectives

- Before you start modeling, you should be able to answer these questions:
  - What is the system that you plan to model?
  - Which properties do you wish to find out about the system?
  - Are you after numbers (quantitative modeling) or just some characteristic behavior (qualitative modeling)?
  - How detailed must the model be?
  - How do you know when the model is adequate?
  - What are you going to do with the results?
    - Testing a new idea
    - Scientific publication: Journal, Conference, Newspaper article
    - Basis for a recommended action
    - Product design

Possible error sources

- New errors are introduced at every step of modeling
- Modeling error: the gap between the model and 'reality‘
- Errors due to mathematical methods used in solving the model
  - This error can be zero in simple models (where the modeling error is usually larger)
  - Several methods in different stages of solution
    - The methods should be in balance with respect to their errors
  - Theoretical error estimates for numerical methods
- Errors induced by the use of computers
  - Computer arithmetic (rounding)
  - Errors in programming
  - Errors in post-processing (visualization, data analysis)
  - Wrong interpretation of the results

Table of Contents

- Motivation & Trends in HPC
- Mathematical Modeling
- **Numerical Methods used in HPSC**
  - **Systems of Differential Equations: ODEs & PDEs**
  - Automatic Differentiation
  - Solving Optimization Problems
  - Solving Nonlinear Equations
  - Basic Linear Algebra, Eigenvalues and Eigenvectors
  - Chaotic systems
- HPSC Program Development/Enhancement: from Prototype to Production
- Visualization, Debugging, Profiling, Performance Analysis & Optimization

Finite Difference Models

- Discrete time steps
- State variables depend on their values at **previous** steps (mostly recursive systems)

$$N_{t+1}^{i} = f^{i}(N_{t}^{1}, N_{t}^{2}, \ldots, N_{t-1}^{1}, N_{t-1}^{2}, \ldots, t)$$

- Sometimes analytically solvable
- Simpler versions:
  - Only one previous time step affects the new values
  - Only one state variable
  - Linear dependence
- Analytical methods exist → **insight**
- Computers are very powerful → **numerical results**
- Example: Density Independent Population Growth

$$N_{t+1} = N_{t} + kN_{t} = N_{0}(1+k)^{t+1}$$

Useful Calculus Concepts

- A mapping $f(t):\Re\rightarrow\Re$ is a function if for each *t* there is only one value of *f(t)*
- **Difference quotient** of $f(t): \frac{f(t+h)-f(t)}{h}$
- **Derivative** $f'(t)=\lim_{h\rightarrow 0}\frac{f(t+h)-f(t)}{h}$
  - Measures the rate of change of *f(t)* at *t*
- To **differentiate** a function means to calculate its derivative
- The inverse operation is **integration**: $F'=f\Rightarrow F=\int f+C$
- However, not all functions are differentiable:
  - There a points where the derivative does not exist (
  - For example: f(t) = |t| at t = 0
  - Sometimes these functions appear in models, so proper attention must be paid to their handling

Differential equations

- A differential equation relates the behavior of a function with its derivative(s).

$$G(t, f(t)), f'(t), f''(t),...,t^{(n)})=0$$

- In addition to the actual equation, other conditions are required in order to guarantee a unique solution.
- These can be either initial or boundary conditions.
  - Initial conditions are given at one point $t=t_0$
  - Boundary conditions are given at separate points

$$t=t_1, t=t_2,...,t=t_n$$

90

Numerical treatment of differential equations

- Ordinary differential equations (ODE) & partial differential equations (PDE) are used in physics to
  - Describe phenomena such as the flow of air around an aircraft
  - Bending of a bridge under various stresses
- Obtaining useful information however
  - "How much does this bridge sag if there are a hundred cars on it" is not that easy
- Techniques are required to turn ODEs & PDEs into computable problems

91

Ordinary Differential Equations

- Ordinary differential equations describe
  - How a quantity (scalar/vector) depends on a single variable
  - Typically, this variable denotes **time**
  - The value of the quantity at some starting time is given
  - This type of equation is called an Initial Value Problem (**IVP**)

Numerical methods for ODEs

- The most elementary / commonly used methods are **finite difference methods**
- Numerical solution results in two sets of data:
  - The argument **points**
  - The **function values** corresponding to these arguments
- Difference methods
  - Based on **approximating** the derivative(s) of a function in some interval with linear combinations of the function values

Numerical methods for ODEs (2)

- The first derivative of a function $f(t)$ is approximated by $$f'(t_k) \approx \sum_{j=k-j_-}^{k+j_+} a_j f(t_j)$$
- If the summation index $\boldsymbol{j}$ takes
  - Only values smaller than $\boldsymbol{k}$, the method is **explicit**
  - Otherwise it is **implicit** and requires one or more algebraic equations
- The difference between the arguments is called step size
  - Usually denoted by $\boldsymbol{h}$
  - $\boldsymbol{h}$ does not have to be constant

Partial Differential Equations

- Partial differential equations describe functions of several variables:
  - Denoting space and time
  - Similar initial values in ODEs, PDEs need values in space to give a uniquely determined solution
  - These are called boundary values → the problem is called a Boundary Value Problem (BVP)
  - Boundary value problems typically describe static mechanical structures

Sample Problem

- A heat equation has aspects of both IVPs and BVPs as:
  - It describes heat spreading through a physical object such as a rod
  - The initial value describes the initial temperature
  - The boundary values give prescribed temperatures at the ends of the rod
- Simplifications:
  - All functions involved have sufficiently many higher derivatives
  - Each derivative is sufficiently smooth

Initial Value Problems – IVP

- Many physical phenomena change over time, and typically the laws of physics give a description of the change, rather than of the quantity of interest itself.
  - Newton's second law $F = ma$
  - Is a statement about the change in position of a point mass expressed as

$$a = \frac{d^2}{dt^2} x = F / m$$

  - It states that acceleration depends linearly on the force exerted on the mass

Initial Value Problems – IVP (2)

- A closed form description $x(t) = \ldots$ can sometimes be derived analytically
- But usually some form of approximation or numerical computation is needed
- Newton's equation is a second order ODE, since it involves a second derivative
- This can be reduced to first order if we allow vector quantities: $u(t) = (x(t), x'(t))$

$$u' = Au + B, \qquad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ F / a \end{pmatrix}$$

Initial Value Problems – IVP (3)

- Here we only consider scalar equations → the equation becomes:

$$u'(t) = f(t, u(t)), \qquad u(0) = u_0, \qquad t > 0$$

- Several numerical methods can solve it
- The initial value in some starting point t, we are interested in the behavior of u

$$t = 0, \qquad u(0) = u_0, \qquad \mathrm{t} \to \infty$$

- As an example: $f(x) = x \quad \rightarrow \quad u'(t) = u(t)$
- The equation states that the rate of growth is equal to the size of the population

Initial Value Problems – IVP (4)

- We consider the numerical solution and the accuracy of this process
- In a numerical method:
  - We consider **discrete size time steps** to approximate the solution of the **continuous time-dependent** process
  - This introduces a certain amount of **error**:
    - Analyze the error introduced in each time step
    - How this error adds up to a global error
- The need to limit the global error will impose restrictions on the used numerical scheme

100

Error and Stability

- Numerical computation involves inaccuracies
  - Machine arithmetic
  - Incremental errors: small perturbation in the initial value leads to large perturbations in the solution
- A differential equation is **stable** if solutions corresponding to **different initial values** u_0 converge to one another as $t \to \infty$
- Let us limit ourselves to the 'autonomous' ODE $u'(t) = f(u(t))$ in which the right hand side does not explicitly depend on $\boldsymbol{t}$

101

Criterium for Stability

- A sufficient criterium for stability is:

$$\frac{\partial}{\partial u} f(u) = \begin{cases} > 0 & \text{unstable} \\ = 0 & \text{neutrally stable} \\ < 0 & \text{stable} \end{cases}$$

- A simple example
  $f(u) = -\lambda u$ with solution $u(t) = u_0 e^{-\lambda t}$
- This problem is stable if $\lambda > 0$