Class organization

- Class webpage:
  - www.cs.ucf.edu/~dcm/Teaching/EngineeringAnalysis
- Textbook:
- Class Notes.
Grade of different activities

- Homeworks: 20%
- Projects: 30%
- Midterm: 20%
- Final: 30%

Weight of different activities
The textbook covers five categories of numerical methods:

(a) Part 2: Roots and optimization
Roots: Solve for $x$ so that $f(x) = 0$
Optimization: Solve for $x$ so that $f'(x) = 0$

(b) Part 3: Linear algebraic equations
Given the $a$'s and the $b$'s, solve for the $x$'s

\[ a_{11}x_1 + a_{12}x_2 = b_1 \]
\[ a_{21}x_1 + a_{22}x_2 = b_2 \]

(c) Part 4: Curve fitting
Regression
Interpolation

(d) Part 5: Integration and differentiation
Integration: Find the area under the curve
Differentiation: Find the slope of the curve

(e) Part 6: Differential equations
Given
\[ \frac{dy}{dt} = \frac{\Delta y}{\Delta t} = f(t, y) \]
solve for $y$ as a function of $t$
\[ y_{t+1} = y_t + f(t, y)\Delta t \]
Lecture 1

- Motivation for the use of mathematical software packages
- From Models to Analytical and to Numerical Simulation
- Example
Motivation

Science and engineering demand a **quantitative analysis** of physical phenomena. Such an analysis requires a sophisticated mathematical apparatus.

Computers are very helpful; several software packages for mathematical software exist.

- **Specialized packages** such as [Ellpack](#) for solving elliptic boundary value problems.
- **General-purpose** systems are:
  - (i) Mathematica of Wolfram Research;
  - (ii) Maple of Maplesoft;
  - (iii) Matlab of Mathworks); and
  - (iv) IDL.
Mathematica

- **All-purpose** mathematical software package.
- It integrates
  - swift and accurate symbolic and numerical calculation,
  - all-purpose graphics, and
  - a powerful programming language.
- It has a sophisticated "notebook interface" for documentating and displaying work. It can save individual graphics in several graphics format.
- Its **functional programming language** (as opposed to procedural) makes it possible to do complex programming using very short concise commands; it does, however, allow the use of basic procedural programming constructs like Do and For.
- **Drawbacks:** steeper learning curve for beginners used to procedural languages; more expensive.
Maple

- Powerful analytical and mathematical software.
- Does the same sorts of things that Mathematica does, with similar high quality.
- Maple's programming language is procedural (like C or Fortran or Basic) although it has a few functional programming constructs.
- Drawbacks: Worksheet interface/typesetting not as developed as Mathematica's, but it is less expensive.
Matlab

- Combines efficient computation, visualization and programming for linear-algebraic technical work and other mathematical areas.
- Widely used in the Engineering schools.
- Drawbacks: Does not support analytical/symbolic math.
Models

- Abstractions of physical, social, economical, systems or phenomena.
- Design to allow us to understand complex systems or phenomena.
- A model captures only aspects of the original system relevant for the type of analysis being conducted.
- Example: the study of the liftoff properties of a wing in a wind tunnel.
Theoretical studies, experiment and computer simulation are three exploratory methods in science and engineering.

In this class we are only concerned with computer models of physical systems.
Mathematical Models

- A formulation or equation that expresses the essential features of a physical system or process in mathematical terms.

- Models can be represented by a functional relationship between:
  - dependent variables,
  - independent variables,
  - parameters, and
  - forcing functions.

\[
\text{Dependent variable} = f(\text{independent variables}, \text{parameters}, \text{forcing functions})
\]
Mathematical Model (cont’d)

• *Dependent variable* ➔ a characteristic that usually reflects the behavior or state of the system

• *Independent variables* ➔ dimensions, such as time and space, along which the system’s behavior is being determined

• *Parameters* ➔ constants reflective of the system’s properties or composition

• *Forcing functions* ➔ external influences acting upon the system
Conservation laws provide the foundation for many model functions. Examples of such laws:
- Conservation of mass
- Conservation of momentum
- Conservation of charge
- Conservation of energy

Some system models will be given as implicit functions or as differential equations - these can be solved either using analytical methods or numerical methods.
Mathematical Model (cont’d)

• *Dependent variable* → a characteristic that usually reflects the behavior or state of the system

• *Independent variables* → dimensions, such as time and space, along which the system’s behavior is being determined

• *Parameters* → constants reflective of the system’s properties or composition

• *Forcing functions* → external influences acting upon the system
Analytical versus numerical methods for model solving

- Once a mathematical model is constructed one could use
  - Analytical methods
  - Numerical methods
- Analytical methods
  - Produce exact solutions
  - Not always feasible
  - May require mathematical sophistication
- Numerical methods
  - Produce an approximate solution
  - The time to solve a numerical problem is a function of the desired accuracy of the approximation.
Example: the analytical model

Consider a bungee jumper in midair. The model for its velocity is given by the differential equation:

\[
\frac{dv}{dt} = g - \frac{c_d}{m} v^2
\]

The change in velocity is affected by: the gravitational force which pulls it down and are opposed by the drag force.

- Dependent variable - velocity \( v \)
- Independent variables - time \( t \)
- Parameters - mass \( m \), drag coefficient \( c_d \)
- Forcing function - gravitational acceleration \( g \)
Example – the analytical solution

\[ v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{g c_d}{m}} t\right) \]

- The model can be used to generate a graph. Example: the velocity of a 68.1 kg jumper, assuming a drag coefficient of 0.25 kg/m
Example: numerical solution

- For the numerical solution we observe that the time rate of change of velocity can be approximated as:

\[
\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}
\]
Example: numerical results

- The efficiency and accuracy of numerical methods depend upon how the method is applied.
- Applying the previous method in 2 s intervals yields:
The solution of the analytical model

- Presented on the white board.