Applied Engineering Problem Solving

Lesson #1: Getting Started with Matlab

Prof. John R. White
Chemical and Nuclear Engineering
UMass-Lowell, Lowell MA

Lesson #1 Goals

1-D Function Evaluation and Plotting

Interactive processing

Defining arrays:

1. **direct assignment**
2. **colon operator (repeat operator)**
3. **linspace command (also logspace)**

Scalar vs. vector arithmetic

Creating simple 2-D plots (f(x) vs. x)

Writing and running simple Matlab scripts

Gilat:
Chapters 1 – 2 & 4 – 5
Chapra:
Chapters 1 – 2
Lesson # 1 Lecture Notes and Illustrative Examples

Simple Interactive Demo

Evaluate and Plot: $f(t) = e^{-0.8t}\cos(5t) \quad \text{for } 0 < t < 10$

Always think discrete when implementing computer solutions:

continuous: $f(t) = e^{-0.8t}\cos(5t)$ — t is a continuous variable

discrete: $f_i = f(t_i) = e^{-0.8t_i}\cos(5t_i) \quad \text{for } i = 1, 2, \cdots$ — i is a discrete index and t_i refers to the i^{th} element of the t vector

Solution Algorithm:

1. define t vector (multiple options…)
2. evaluate f (can use scalar or vector approach)
3. plot function (use simple linear axes with proper annotations)

Simple Interactive Demo (cont.)

Evaluate and Plot: $f(t) = e^{-0.8t}\cos(5t) \quad \text{for } 0 < t < 10$

Define the t vector:

Direct assignment -- `t = [0 1 2 3 4 5 6 7 8 9 10]`

Use of the colon operator -- `t = 0:1:10`

Use of the linspace command -- `t = linspace(0,10,11)`

Evaluate the f vector:

discuss these in detail

Scalar Approach:

```
for i = 1:length(t)
  f(i) = exp(-0.8*t(i))*cos(5*t(i))
end
```

Vector Approach:

```
f = exp(-0.8*t).*cos(5*t)
```

Note the "dot" arithmetic

Simple Interactive Demo (cont.)

Evaluate and Plot: $f(t) = e^{-0.8t}\cos(5t) \quad \text{for } 0 < t < 10$

Plot the function f vs. t:

```
plot(t,f), grid
title('Sample Plot')
xlabel('Time (sec)'), ylabel('Function Value')
```

Note order: plot(independent, dependent)

Let's do it in Matlab…

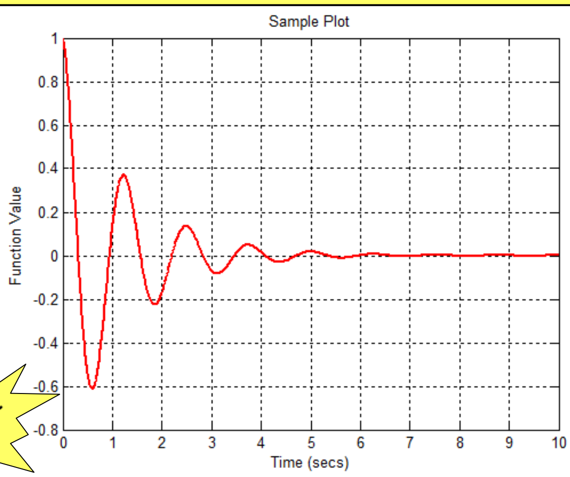
Note:

Working interactively in Matlab is great for debugging purposes, but it is not practical for solving most problems of interest.

Thus, we will do some brief illustrations using interactive mode and then quickly switch to writing and running Matlab script files for all our problem solving activities…

Simple Interactive Demo (cont.)

Evaluate and Plot: $f(t) = e^{-0.8t}\cos(5t) \quad \text{for } 0 < t < 10$



Final Result

Pendulum Dynamics

$$ma_t = \sum_i F_{t_i}$$

mass times acceleration is equal to the sum of the applied forces

$$ma_t = m\frac{dv_t}{dt} = mL\frac{d\omega}{dt} = mL\frac{d^2\theta}{dt^2} = mL\theta''$$

θ = angular position
ω = angular velocity = $d\theta/dt$

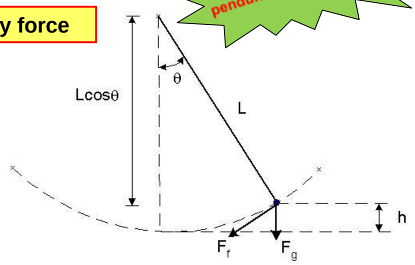
$$F_{f_t} = -cv_t = -cL\frac{d\theta}{dt} = -cL\theta'$$

friction force

$$F_{g_t} = -mg\sin\theta \approx -mg\theta$$

gravity force

See pendulum_dynamics.pdf



$$mL\theta'' + cL\theta' + mg\theta = 0$$

or

$$\theta'' + \frac{c}{m}\theta' + \frac{g}{L}\theta = 0$$

CHEN.3170 Applied Engineering Problem Solving
Lesson #1: Getting Started with Matlab
(August 2017)

Pendulum Dynamics (cont.)

This physical problem gives a simple linear constant coefficient IVP:

$$\theta'' + \frac{c}{m}\theta' + \frac{g}{L}\theta = 0 \quad \text{with} \quad \theta(0) = \theta_o \quad \text{and} \quad \omega(0) = \theta'(0) = \omega_o$$

To solve, we assume a solution of the form of e^{rt}, which gives

$$r^2 + \frac{c}{m}r + \frac{g}{L} = 0$$

characteristic eqn.

with the roots given by

$$r_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{g}{L}}$$

If $\frac{g}{L} > \left(\frac{c}{2m}\right)^2$ then, the system is underdamped which gives

$$\theta(t) = e^{\alpha t}\left(c_1\cos\beta t + c_2\sin\beta t\right)$$

ICs allow one to uniquely determine c_1 and c_2

Again see pendulum_dynamics.pdf

CHEN.3170 Applied Engineering Problem Solving
Lesson #1: Getting Started with Matlab
(August 2017)

Pendulum Dynamics (cont.)

Solution Algorithm:

1. define problem parameters: m, c, g, L, θ_o and ω_o
2. compute real and imaginary components of roots:

$$\alpha = -\frac{c}{2m} \qquad \text{and} \qquad \beta = \sqrt{\frac{g}{L} - \left(\frac{c}{2m}\right)^2}$$

3. compute equation coefficients, c_1 and c_2
4. define time vector, t
5. evaluate $\theta(t)$, $\omega(t)$, and the energy components, $E_p(t)$, $E_k(t)$, and $E_{tot}(t)$, etc.

$$E_{tot} = E_p + E_k = mgh + \frac{1}{2}mv^2 = mgL\left(1-\cos\theta\right) + \frac{1}{2}m\left(L\omega\right)^2$$

6. plot and interpret the results…

Let's do this in Matlab (see pendulum_1.m) …

The pendulum_1.m Script File

A script file simply contains a series of Matlab commands

Script file names cannot contain a blank space and you should also be careful not to use a name of an existing function

To run a script file, simply type its name at the command prompt (without the .m extension) or run from within the Editor

Be careful to note the folder where the file is stored (make this the working directory)

Most script files have at least three primary sections (input section, computational section, output section)

All your programs should have good internal documentation to describe the solution flow, to identify the key variables (including units), and to make the code easy to follow…

The pendulum_1.m Script File (cont.)

Some specific things to look for in pendulum_1.m:

1. use of **comments** (i.e. good internal documentation)
2. use of **linspace** command
3. proper use of **"dot" arithmetic**
4. the **figure** command
5. the **plot** and **subplot** commands
6. the **phase plane plot**
7. **proper plot annotation** (**grid**, **axis**, **title**, **xlabel**, **ylabel**, and **legend** commands, etc…)
8. use of Greek symbols (i.e. **\theta** and **\omega**)
9. use of **_** to get an underscore instead of a subscript
10. etc…

Maxwell Boltzmann Distribution

In a **dilute gas**, the kinetic energies of the molecules are distributed according to the Maxwellian distribution function:

f(E,T) dE = probability of finding a particle in energy interval dE for a particular gas temperature T

Since f(E,T) is a probability density function (i.e. probability per unit energy), then

$$\int_0^\infty f(E,T)\,dE = 1$$

See maxwell_1.pdf

At a particular gas temperature,

$$f(E,T) = \frac{2\pi}{(\pi kT)^{3/2}} E^{1/2} e^{-E/kT}$$

Our job is to evaluate, plot, and interpret this function

where E is in eV, T is the absolute temperature in K, and the Boltzmann constant, k, has a value of k = 8.6170x10^{-5} eV/K

Maxwell Boltzmann Distribution (cont.)

Solution Algorithm:

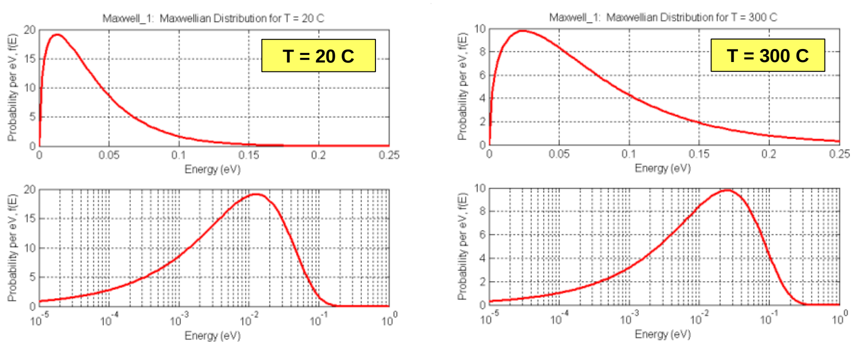
1. **define k and the absolute gas temperature, T**
2. **define a discretized energy grid, E**
3. **evaluate f(E,T) using element-by-element vector arithmetic**
4. **plot and label f(E,T) as appropriate and interpret…**

Let's do this in Matlab (see maxwell_1.m) …

Some specific things to look for when using maxwell_1.m:

1. **basic distribution of f(E) for a given T (should match expectations)**
2. **use of logspace and semilogx commands**
3. **proper use of "dot" arithmetic**
4. **use of the num2str, input, and axis commands**

Maxwell Boltzmann Distribution (cont.)



Final Result

More Illustrative Examples

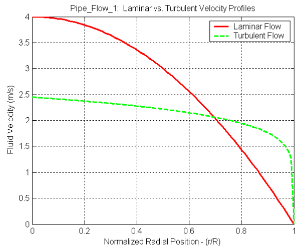
Laminar vs. Turbulent Flow in a Pipe

Laminar:

$$u_L(r) = u_{L_{max}}\left[1-(r/R)^2\right]$$

Turbulent:

$$u_T(r) = u_{T_{max}}\left[1-r/R\right]^m \qquad (m = 1/7)$$



Pipe Insulation Considerations

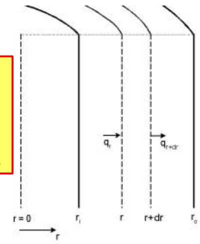
Fourier's Law:

$$q_{cond} = -kA(r)\frac{dT}{dr}$$

Newton's Law of Cooling:

$$q_{conv} = hA(r_o)\left(T(r_o) - T_\infty\right)$$

Sensitivity Study:
thermal conductivity, k
heat transfer coeff., h
insulation thickness, r_o



Lesson #1 Summary

In this Lesson we have discussed the following topics:

1-D Function Evaluation and Plotting in Matlab

Interactive processing

Defining arrays:

1. **direct assignment**
2. **colon operator (repeat operator)**
3. **linspace command (also logspace)**

Scalar vs. vector arithmetic

Creating simple 2-D plots (f(x) vs. x)

Writing and running simple Matlab scripts

You should now be much more comfortable with these topics…