1 Basic Numerical Integration

Do problems 16.2 on page 303 and 16.8 (a) on page 304 in Chapra (2005) by hand.

2 Romberg Integration

Do problem 17.7 (a) and (c) on page 320 in Chapra (2005).

3 Systems of ODEs

Back near the beginning of this class we studied the numerical solution to the bungee-
 jumper problem and the trajectory of a golf ball. In both cases, we applied Euler’s method to
solve a system of non-linear ordinary differential equations (ODEs). In the case of the bungee
 jumper, we had two equations in the unknowns position and velocity; for the golf ball, we
had four equations in the unknowns $x$ and $y$ position and $v_x$ and $v_y$ velocity. The program
we wrote to solve these problems was not adaptive, and we would have to write a completely
new differential equations solver to solve a system of equations with three unknowns.

One way to make our solutions adaptive, however, is to use matrices. In the solutions
to ODEs, we define a particular new vector, called the state-space vector, which contains all
our unknowns. In the case of the bungee-jumper problem, we would have

$$\tilde{u} = \begin{bmatrix} x \\ v \end{bmatrix}$$  (1)
where \( \vec{u} \) is called the state-space vector.

In this problem, we will re-write the differential equation solver we used in the bungee jumper problem so that it can solve an arbitrary number of equations using the state-space notation. Then we will apply the new method to solve a system of three coupled equations called the Lorenz equations.

3.1 Euler’s method solution

Start by downloading a working set of m-files for the bungee jumper problem from the February 1st entry on the Calendar page of the course web-site. Rename the function `diffeq2.m` as just `diffeq.m` and make the following modifications

- Replace the original outputs \( v \) and \( x \) with the single state-space output \( u \). Replace the input initial conditions \( v_0 \) and \( x_0 \) with the single input \( u_0 \). Note that in the new function, \( u \) and \( u_0 \) are vectors.

- After the comments block, begin the new function with a command to find out how many equations are in our system of equations. You can use the command \( n = \text{length}(u_0) \).

- The output vector \( u \) will be organized so that each row contains the values of one member of the state space evaluated at all calculated times. Thus, each column of \( u \) will contain the complete state space evaluated at one time. Set the initial conditions using a `for` loop to store the initial values in \( u_0 \) in the first column of the output vector \( u \).

- Change the `feval` command so that \( v(i) \) and \( x(i) \) are replaced by a single variable \( u \). Note that you only want to send column \( i \) of \( u \) in their place.

- Replace the calculations for \( x(i+1) \) and \( v(i+1) \) with similar calculations for the state space. Since this function should work for any number of coupled equations, you should use a `for` loop to loop through all the variables in the state space. Be sure to save the new values in column \( i+1 \).

- If you like, you can test your new program by making the appropriate changes in `main.m` and `bungee.m` so that they are in state-space form.

In making all these changes to `diffeq2.m`, remember that you want this function to work for any number of coupled equations. So do not program anything that would limit your function to only solving problems with two unknowns.
3.2 Lorenz equations

American meteorologist Edward Lorenz developed the following system of equations as a simplified model of atmospheric fluid dynamics

\[
\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -ax + ay \\ bx - y - xz \\ -cz + xy \end{bmatrix}
\] (2)

In these equations \( x \) is the intensity of atmospheric fluid motion and \( y \) and \( z \) are the temperature variations in the horizontal and vertical directions (Chapra & Canale 2002).

Use your new differential equation solver to solve this system of equations with parameter values

\[
a = 10 \\
b = 28 \\
c = 8/3
\] (3)

In the numerical solution, solve for time \( t \) from 0 to 20. Use the initial condition \( x = y = z = 5 \). Create the following graphical outputs

1. Use `subplot` to create two figures side-by-side. For the left figure, plot \( y \) versus \( x \) and for the right figure, plot \( z \) versus \( x \). Create your plot by solving the equations using a time step size of 0.001. Figure 1 presents an example of what your output should look like.

2. Use `subplot` to create three plots in one column. For the top plot, plot \( x \) versus \( t \). For the middle and bottom plots, plot \( y \) versus \( t \) and \( z \) versus \( t \). Use a solid line in the figures to represent the solution using a time step of 0.001. Re-run the model with a time step of 0.002 and plot the new results on top of the old results using a dotted line type. Why are the solutions different?

3. Create a third plot similar to the second one. On this plot, present the solution using a step size of 0.001 as the solid line. Change the initial condition for \( x \) to 5.001. Re-run the model and present the new solution over the original solution using a dotted line type. If this model is to predict the weather, how important is it to know the current weather conditions? and why?

Turn in a copy of your m-files developed to solve the Lorenz equations, a copy of the three plots described above, hand notes developed to create your Matlab programs and functions, and your answers to the questions posed in parts 2 and 3 of the above list.

The behavior exhibited by these equations is called chaos. Chaos does not mean randomness, meaninglessness, or utter non-conformance to a set or rules. What chaos means
in a mathematical sense is that a system of equations is very sensitive to boundary and initial conditions and to errors. In this case, there are two stable solutions (the two orbits created in the first plot). The trigger that moves the solution from one orbit to the other is exactly predicted by the differential equations, but its occurrence is very dependent on the initial conditions or any errors that creep into the solutions. When scientists say climate is chaotic, they mean that it has multiple equilibrium states and that an incredibly high degree of information would be required to accurately predict the transition from one state to another.

References