Bare Essentials

After reading Chapter 12 you should be able to

1. Write any first order differential equation in the form $\dfrac{dy}{dt} = f(y, t)$.
2. Verify that a proposed solution to an initial value problem is indeed a solution.
3. Perform manual calculations with Euler's method for any first order ODE.
4. Predict the effect of reducing stepsize on the global discretization error (GDE) of Euler's method. Specifically, evaluate the ratio $GDE(h_2)/GDE(h_1)$.

To solve basic ODEs with MATLAB you will need to

5. Write an m-file to evaluate the right hand side of a first order ODE. This m-file must accept two inputs t and y, and return dy/dt.
6. Call the `odeEuler` function (from the NMM toolbox) for the system described by the m-file in the preceding bullet.
7. Plot a comparison of the exact solution to an ODE (when it is given) and the solution to the same ODE obtained by a numerical method.

An Expanded Core of Knowledge

After mastering the bare essentials you should move on to a deeper understanding of the fundamentals. Doing so involves being able to

1. Rank the numerical methods presented in Chapter 12 in order of increasing order.
2. Convert a higher order ODE to an equivalent system of coupled first order ODEs.

To solve more advanced ODEs with MATLAB you will need to

3. Use any of the built-in ODE routines or NMM Toolbox routines to solve a first order ODE.
4. Use any of the built-in ODE routines to solve a system of first order ODEs.
5. Write an m-file to evaluate the right hand sides of a system of coupled first order ODEs. This m-file must accept two inputs, a scalar t and vector of y values (dependent variables). The m-file must return a vector of dy/dt values.
6. Write m-files that use pass-through parameters a, b, $\ldots$, to evaluates $dy/dt = f(t, a, b, \ldots)$ for use with the `ode45` command.

Developing Mastery

Working toward mastery of solving ODEs you will need to

1. Identify the circumstances when an absolute convergence tolerance is more appropriate than an relative convergence tolerance.
2. Specify an appropriate convergence tolerance for any ODE.
3. Identify the convergence rate of an unknown method for solving an ODE.
4. Reduce the convergence parameters to `ode45` so that the numerical solution is independent of the convergence parameters.