School of Mathematics and Statistics
MAST30028 Numerical Methods & Scientific Computing
Week 10

Drag and drop the folder `Week10` from `L: \MAST30028` to `C:\...\MATLAB` and include it in the path. Now MATLAB knows how to find the files in `Week10`.

Exercise Set 1

This relates to material in Lecture 18.

Levenberg-Marquardt method

a. To see how L-M performs on the banana function, run `bandem.m` in MATLAB R2015b or earlier, which comes from the Optimization Toolbox. In the GUI which pops up, click on the last button L-M to run the Levenberg-Marquardt method for this problem. It finds the minimum after 34 function evaluations.

b. An implementation of Levenberg-Marquardt by Kelley is `levmarCorr.m`. A driver for Heath's problem is `kelleyLMdriver`. Run it to see how L-M performs for both good and poor guesses, and see how the Levenberg-Marquardt parameter λ is changed at each iteration.

How does L-M perform compared to undamped and damped G-N?

The Optimization toolbox

The Curve fitting toolbox is just a GUI interface for the commands `lsqcurvefit` and `lsqnonlin` from the Optimization Toolbox. `lsqcurvefit` is itself a wrapper for data fitting purposes of the nonlinear least squares solver `lsqnonlin`. Options to these functions are set using the command `optimset`. We will see a similar method for setting options when we meet ODE solvers.

a. Since the 'banana' function has the form of a sum of squares, we can fool MATLAB into minimizing it by calling `lsqnonlin`. It requires as input a function handle that returns the residual vector $\mathbf{R}(\mathbf{x})$. `lsqnonlin` forms the sum of squares itself i.e. the cost function. Other functions in the Optimization Toolbox require the cost function itself as input.

   To see how `lsqnonlin` performs on the banana function, see my `bananaScript.m` which uses both the default MATLAB method, then sets an option to use Levenberg-Marquardt. Notice how the L-M parameter λ changes by factors of 10 in MATLAB, different to Kelley's code. What is the stopping criterion used?

b. To see how these codes can be used on the Heath fitting problem, see my `heathScript.m`. It first uses `lsqnonlin`, then uses `lsqcurvefit` with identical results. `lsqcurvefit` requires a function handle to describe the fitting model, as well as the data $\{T_i, Y_i\}$. Finally, we set an option to use Levenberg-Marquardt on the same problem.

Exercise Set 2: Euler's method for a scalar 1st order ODE

This relates to material in Lecture 19.

Error propagation in Euler's method

In solving initial value problems, it is important to understand that one is in effect jumping between neighbouring solutions from a family of solutions. If the differential equation is asymptotically stable with respect to initial conditions, these solutions get closer to each other; if it's unstable, they diverge from each other.

a. To see the error propagation of the Euler method, run the script `ShowTrunc`. At each time step, the previous error is propagated by following the appropriate solution curve —- whether this grows or shrinks depends on $J = \frac{\partial f}{\partial y}$. In this case, $J = \frac{\partial f}{\partial y} = -5$. In addition, at each time step a new local truncation eror is made, so the numerical solution hops onto a new solution curve.

b. Now edit `ShowTrunc` to make $J = \frac{\partial f}{\partial y} = 5$ and run to see what difference it makes to the error propagation.

This explains why errors tend to grow with time when $J > 0$ and decay when $J < 0$.

Accuracy and stability

a. Look at the code `FixedEuler` to see how fixed step Euler's method could be coded.

b. Run the code `testEuler` that I showed in lectures. It uses `FixedEuler` on 3 problems with known solutions, so I can compute the errors. It pauses between the 3 problems, so you have to hit a key to continue.

   I used them to illustrate the accuracy (order) of the method and the (numerical) stability of the method. By comparing the ten times error using $n = 1000$ (`10*error3`) with the error using $n = 100$ (`error2`) (and similarly for the error using $n = 10$), I am testing to what extent the global error

$$GE_k \equiv y(t_k) - y_k^{EM}$$

   is proportional to h. Except for the case when the errors blow up catastrophically, it seems to hold fairly well.

c. In the next lecture, I will show that Euler's method is numerically unstable unless

$$|1 + hJ| < 1$$

   Using this criterion, explain the case when the errors blow up catastrophically.

Try yourself

a. Use `FixedEuler` to solve the IVP

$$y' = y^2$$

   where $y(0) = 1$, over the interval $[0, 2]$. First try with $n = 10$, then with $n = 100$. What do you think is going on?

   Hint: find the exact solution using first year maths. Also use the Variable Browser (from the Workspace) to see the values of the numerical solution.

Exercise Set 3: Euler's method for systems

Converting to systems

Most IVP software expects higher order ODEs to be converted to systems of first order DEs by the user. Try your hand at the following.

a.

$$\ddot{\theta} + \sin\theta = 0$$

the simple pendulum

b.

$$y''' + 3x^2 y y'' - \sin(y) y'^2 = \cos(x)$$

c.

$$\ddot{x} = -x/r^3; \quad \ddot{y} = -y/r^3; \quad r = \sqrt{x^2 + y^2}$$

the equations for a planet orbiting the Sun (the Kepler problem). The equations describing this are Newton's equations for the x and y components of acceleration.

Vectorized Euler's Method

a. To apply Euler's method to systems of ODEs, we need to *vectorize* `FixedEuler` i.e. write it so that it will work with a function that accepts a scalar t and a vector $\mathbf{y}$ and returns a vector of derivatives $\mathbf{y}'$. One possible solution is given in my code `FixedEulerVec`.

Run `FixedEulerVec` over the interval [0,20] with $n = 2,000$ for the following systems:

(a)

$$\ddot{\theta} + \sin\theta = 0$$

using ICs: $\theta(0) = 1, \dot{\theta}(0) = 0$

(b)

$$\begin{aligned} \dot{y_1} &= -(\frac{8}{3})y_1 + y_2 y_3 \\ \dot{y_2} &= -10y_2 + 10y_3 \\ \dot{y_3} &= -y_1 y_2 + 28y_2 - y_3 \end{aligned}$$

using ICs: $y_1(0) = 1; y_2(0) = 1; y_3(0) = 1$

Repeat for $n = 20,000$.

Create suitable plots to answer the questions:

- Is system a above periodic? (Use a phase plot).
- Is system b above periodic? (Use a 3D phase plot).