Numerical Analysis 1 – Class 11

Friday, March 31st, 2023

Subjects covered

- Solving ODEs in 1 and N dimensions using the following methods: Forward Euler, Backward Euler, Heun's Method, Runge-Kutta.
- GTE and stability of ODE methods.
- Butcher tableaux.
- Adaptive integrators. Stiff systems.
- Examples of ODEs solved numerically: Harmonic oscillator, van der Pol oscillator, Logistic equation.
- Symplectic integrators for conservative systems.

Reading

- Kutz, chapter 7.1 – 7.3.
- "Numerically Solving Ordinary Differential Equations", S. Brorson (on Canvas).
- Chapter 7, "Numerical Computing with MATLAB", C. Moler (linked on Canvas).

Problems

Most of the following problems require you to write a program. For each program you write, please make sure you also write a test which validates your program. Please use Canvas to upload your submissions under the "Assignments" link for this problem set.

Problem 1

In 1877 the British physicist known as Lord Rayleigh wrote down the following second-order differential equation describing the oscillation of a clarinet reed:

$$m\ddot{x} = -k\,x + a\dot{x} - b\left(\dot{x}\right)^3 \tag{1}$$

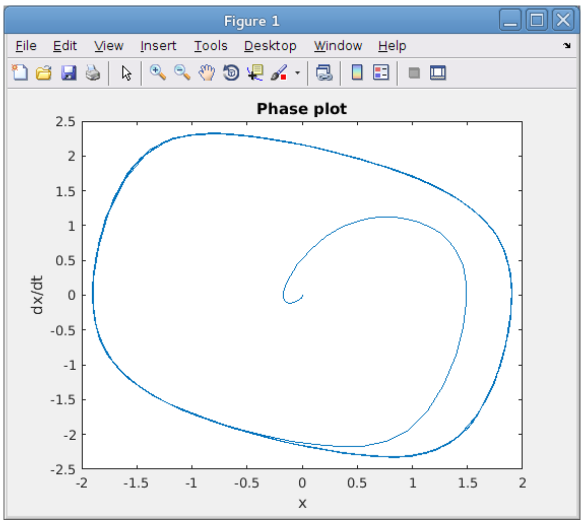
The variable *x* expresses the displacement of the reed's position from its equilibrium value. In this notation, a single dot over the variable *x* indicates a first derivative in time (i.e. $\dot{x} = dx/dt$), and two dots indicate a second time derivative. The coefficients *m, k, a*, and *b* capture information related to the physics of the problem. For example, *m* parameterizes the mass of the reed, *k* is a restoring force, and *a* and *b* are damping coefficients. You may recognize equation (1) as an expression of Newton's third law. The left hand term contains the mass and acceleration of the reed, and the right hand side expresses the three forces acting on it: a linear restoring force (Hooke's law), linear damping, and a non-linear damping term.

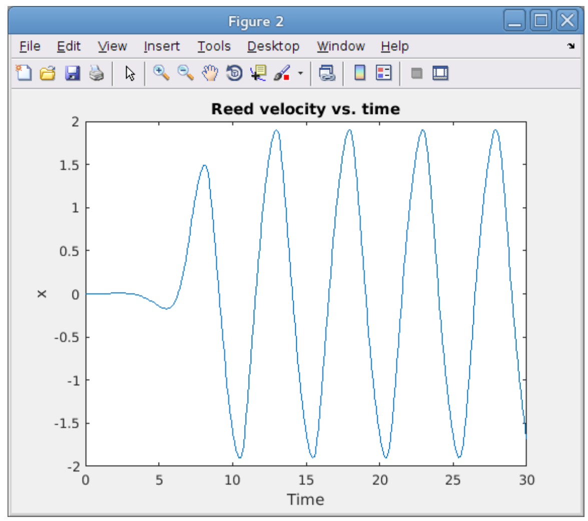
Figure 1: Clarinet

For certain values of the parameters equation (1) evinces self-oscillations. It's important that this system can support oscillations in order that the clarinet can produce a musical sound.

Please do the following:

1. Re-write the second order differential equation (1) as two first-order equations in preparation for numerical solution. Hand in your derivation.
2. Please implement a program which computes the solution to this system as a function of time using 4th order Runge-Kutta. Use the following parameter values: m = 1, k = 2, a = 2, b = 0.5. Use as initial conditions $[x, \dot{x}] = [0, 0.001]$. Please make two plots of your solution: x vs. time and a phase plot. My plots are shown below. Feel free to borrow code from my implementation of the Runge-Kutta solution of the van der Pol Oscillator on Blackboard.
3. To test, verify that your solution obeys $-kx + a\dot{x} - b(\dot{x})^3 - m\ddot{x} = 0$ within some tolerance. Note that the tolerance will be some multiple of your step size h.





Problem 2

In class I discussed Heun's method for solving ODEs as an improvement on Forward Euler. The method is defined as

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right)$$

where

$$\tilde{y}_{n+1} = y_n + hf(t_n, y_n)$$

Please do this:

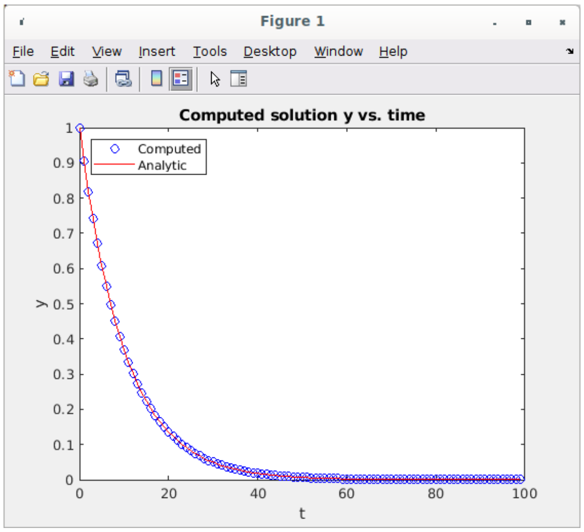
- Please derive the stability condition for Heun’s method for real $h\lambda$. Please report, over what interval on the real number line is Heun’s method stable?
- Write an implementation of Heun's method. Feel free to borrow the implementation of Forward Euler from Blackboard as a starting point.
- To test your implementation, use it to integrate the simple first-order system

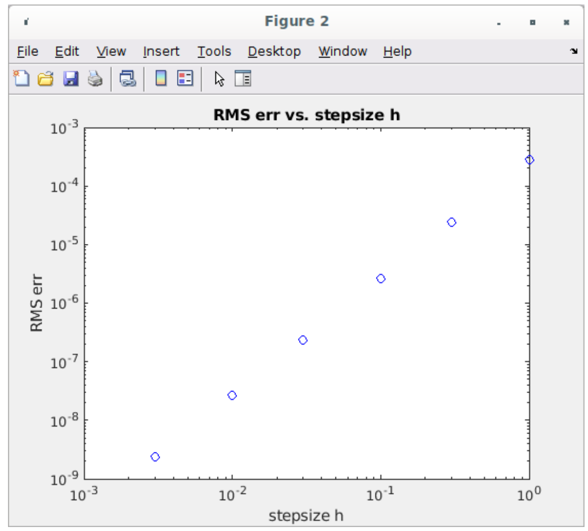
$$\frac{dy}{dt} = \alpha y$$

with initial condition $y(0) = 1$ and parameter $\alpha = -0.1$. Please compare your solution with

Heun's method to the analytic solution. My result is shown below.

- Test your implementation for varying step sizes, h = [0.003, 0.01, 0.03, 0.1, 0.3, 1]. For each step size compute the RMS error between your solution and the analytic solution. Make a plot of RMS error vs. step size h. My result is shown below. How does the error scale with h for Heun's method?





Problem 3

Another well-known simple ODE integrator is the "explicit midpoint method". Here is its Butcher tableau:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

Please do this:

- Use the Butcher tableau to write down the actual set of equations used to implement the explicit midpoint method. This is a pencil and paper exercise – please turn in your equations.
- Now write a program which implements the explicit midpoint method. Feel free to use my implementation of RK4 – or any other of my functions on Canvas as a starting point or for inspiration.
- Make sure you write a test for your implementation. I suggest you just use the simple first-order IVP you used in problem 2:

$$\frac{dy}{dt} = \alpha y \qquad y(0) = 1 \qquad \alpha = -0.1$$