Computational Physics; January–May 2020
Assignment 2
Due: Thursday, 20 April 2020

Instructions

- When you are asked to write a code, submit your code by posting it to Github and sending in the Github link.
- When you are asked to solve something manually, or when you are asked for a number as an answer, submit your response in writing.
- This version modified on 14 April 2020 to correct typos in problems 7 and 15 and to add a clarification to problem 5.

---

1. Use backward integration with Euler's Method in a Python code to solve the following two initial value problems:

$$\frac{\mathrm{d}y}{\mathrm{d}x} = -9y$$

with $0 \leq x \leq 1$ and $y(0) = e$, and

$$\frac{\mathrm{d}y}{\mathrm{d}x} = -20(y - x)^2 + 2x$$

with $0 \leq x \leq 1$ and $y(0) = 1/3$. Do not use any integration functions from Numpy.

2. Use Euler's Method in a Python code to approximate the solution for the initial value problem

$$y' = \frac{y}{t} - \left(\frac{y}{t}\right)^2,$$

where $1 \leq t \leq 2$ and $y(1) = 1$. Use a step size of $h = 0.1$. We know, using analytical methods, that the solution to above equation is

$$y = \frac{t}{1 + \ln t}.$$

Compute the absolute and relative error in the solution obtained using Euler's method. (Do not use any integration functions from Numpy.)

3. Use the Runge-Kutta Method in a Python code to solve the differential equation

$$y'' - 2y' + y = xe^x - x,$$

with $0 \leq x \leq 1$, $y(0) = 0$, and $y'(0) = 0$. (Do not use any integration functions from Numpy.)

4. What are stiff differential equations? Give an example of a physical process that is often described by stiff differential equations? What algorithm will you choose to solve a stiff differential equation? What Numpy function will you choose to solve a stiff differential equation?

5. Implement the shooting method in Python for the boundary value problem.

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -g, \tag{1}$$

where $g = 10$ and the boundary conditions are given as

$$x = 0 \text{ at } t = 0, \tag{2}$$
$$x = 0 \text{ at } t = t_1, \tag{3}$$

where $t_1 = 10$. Can you use `numpy.argmin` to find the solution? Make a plot that shows the exact solution, the numerical solution, and at least five candidate solutions. Even if your shooting method converges with less then five candidate solutions, just construct more candidate solutions following the general shooting idea and plot them.

6. Implement the relaxation method for the above boundary value problem in Python. Make a plot that shows the exact solution, the numerical solution, and at least five candidate solutions.

7. Use `scipy.integrate.solve_ivp` to solve the following initial value problems. In each case, make a plot of the solution and argue why it is correct.

   - $y' = te^{3t} - 2y$ with $0 \le t \le 1$ and $y(0) = 0$
   - $y' = 1 - (t - y)^2$ with $2 \le t \le 3$ and $y(2) = 1$
   - $y' = 1 + y/t$ with $1 \le t \le 2$ and $y(1) = 2$
   - $y' = \cos 2t + \sin 3t$ with $0 \le t \le 1$ and $y(0) = 1$

   Find analytical solutions to these equations using a symbolic calculator (such as Mathematica or WolframAlpha) and put them on the plot.

8. Use `scipy.integrate.solve_bvp` to solve the following boundary value problems. In each case, make a plot of the solution and argue why it is correct.

   - $y'' = -e^{-2y}$ with $1 \le x \le 2$, $y(1) = 0$, and $y(2) = \ln 2$
   - $y'' = y' \cos x - y \ln y$ with $0 \le x \le \pi/2$, $y(0) = 1$, and $y(\pi/2) = e$
   - $y'' = -(2(y')^3 + y^2 y') \sec x$ with $\pi/4 \le x \le \pi/3$, $y(\pi/4) = 2^{-1/4}$, and $y(\pi/3) = 12^{1/4}/2$
   - $y'' = 1/2 - (y')^2/2 - y \sin x/2$ with $0 \le x \le \pi$, $y(0) = 2$, and $y(\pi) = 2$

   Find analytical solutions to these equations using a symbolic calculator (such as Mathematica or WolframAlpha) and put them on the plot.

9. In a Python code, use adaptive step-size control with the fourth-order Runge-Kutta method to solve

$$y' = (y^2 + y)/t$$

with $1 \le t \le 3$ and $y(1) = -2$. The absolute accuracy of the solution should be better than 10^{-4}. Make a plot showing the solution and the mesh points.

10. Solve the intial value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{1}{x^2 + t^2}$$

over the domain $0 < t < \infty$ with $x(0) = 1$ using the fourth-order Runge-Kutta method. What is the value of the solution at $t = 3.5 \times 10^6$?

11. Use the fourth-order Runge-Kutta method to solve the initial value problem

$$\begin{aligned} u_1' &= u_1 + 2u_2 - 2u_3 + e^{-t} \\ u_2' &= u_2 + u_3 - 2e^{-t} \\ u_3' &= u_1 + 2y_2 + e^{-t} \end{aligned}$$

with $0 \le t \le 1$, $u_1(0) = 3$, $u_2(0) = -1$, and $u_3(0) = 1$. Plot the result.

12. Develop an algorithm for a sixth-order Runge-Kutta method for the initial value problem

$$\mathrm{d}y/\mathrm{d}t = f(y, t).$$

How many time does your algorithm evaluate f at each step?

13. Use Euler's method in a Python code to solve the initial value problem

$$t^2 y'' - 2ty' + 2y = t^3 \ln t$$

with $1 \le t \le 2$, $y(1) = 1$, and $y'(1) = 0$ with step size $h = 0.001$. Plot the solution together with the known exact solution $y(t) = 7t/4 + (t^3/2)\ln t - (3/4)t^3$.

14. Spend some time going through the GSL documentation on initial value problems. Write down the names of at least three GSL functions for solving initial value problems. Also write down their APIs and the name of the algorithm they use.

15. In a C code, use Euler's method with step size $h = 0.2$ to solve the initial value problem

$$y' = y - t^2 + 1$$

with $0 \le t \le 2$ and $y(0) = 0.5$. The exact solution is $y = (t+1)^2 - 0.5e^t$. In class, we derived an upper bound for the error incurred in using Euler's method. Tabulate the error for this given initial value problem and the error bound at each step.