MAS212 Scientific Computing and Simulation

#6: Numerical methods for ODEs: (b) implicit methods and multistep methods

`http://sam-dolan.staff.shef.ac.uk/mas212`

Key resources:

- Lec 6: `http://sam-dolan.staff.shef.ac.uk/mas212/docs/l6.pdf`
- `http://en.wikipedia.org/wiki/Numerical_methods_for_ordinary_differential_equations`

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

1. Implicit methods. Consider the 2D autonomous system

$$\begin{aligned} \frac{dx}{dt} &= 398x + 798y \\ \frac{dy}{dt} &= -399x - 799y, \qquad x(0) = 1, \;\; y(0) = 0. \end{aligned} \tag{1}$$

(a) By direct substitution, or otherwise, verify that the exact solution is $x(t) = 2e^{-t} - e^{-400t}$, $y(t) = -e^{-t} + e^{-400t}$. Plot the exact solution on the domain $t \in [0, 1]$.

(b) Show that the **forward Euler method** leads to numerical instability for step sizes $h \gtrsim 1/200$.

(c) The **backward Euler method** is

$$\begin{aligned} x_{k+1} &= x_k + h\left[+398x_{k+1} + 798y_{k+1}\right], \\ y_{k+1} &= y_k + h\left[-399x_{k+1} - 799y_{k+1}\right]. \end{aligned}$$

Note that the method is **implicit**, because x_{k+1} and y_{k+1} appear on both sides of these equations.

In this case, the equations are linear and so they may be inverted to obtain

$$\begin{aligned} x_{k+1} &= \frac{1}{\Delta}\left\{(1 + 799h)x_k + 798\,h\,y_k\right\}, \\ y_{k+1} &= \frac{1}{\Delta}\left\{-399\,h\,x_k + (1 - 398h)y_k\right\}, \qquad \Delta = 1 + 401h + 400h^2. \end{aligned}$$

Implement the backward Euler method, and demonstrate that it is **stable** even for step sizes $h \lesssim 1/200$.

---

2. Multistep methods. Consider the 1D differential equation

$$\frac{dx}{dt} = f(x,t), \qquad f(x,t) = -x + \{1 + \tanh(t-20)\}\sin t, \qquad x(t<0) = 0.$$

(a) Two-step method. Write code to find a numerical solution of the ODE system with the linear two-step method

$$x_{k+1} = x_k + h\left(\frac{3}{2}f(x_k,t_k) - \frac{1}{2}f(x_{k-1},t_{k-1})\right).$$

Plot a numerical solution of the ODE with step size $h = 0.02$ over the domain $t \in [0, 60]$.
(N.B. As $x = 0$ for $t \leq 0$, and as $f(0,0) = 0$, one may initially set $x_1 = x_0 = 0$ and start with $k = 1$ in the above equation).

(b) Ratio test. Apply the ratio test to show that your implementation of the two-step method is 2nd-order accurate (i.e. $r \sim 2^2 = 4$) over the domain $t \in [30, 60]$. (You will need to restrict the range of the y-axis using `plt.ylim(2,6)`).

(c) Adams-Bashforth methods. Experiment with higher-order multistep methods,

$$\begin{aligned}
x_{k+1} &= x_k + h\left(\frac{23}{12}f_k - \frac{4}{3}f_{k-1} + \frac{5}{12}f_{k-2}\right), \\
x_{k+1} &= x_k + h\left(\frac{55}{24}f_k - \frac{59}{24}f_{k-1} + \frac{37}{24}f_{k-2} - \frac{3}{8}f_{k-3}\right), \\
x_{k+1} &= x_k + h\left(\frac{1901}{720}f_k - \frac{1387}{360}f_{k-1} + \frac{109}{30}f_{k-2} - \frac{637}{360}f_{k-3} + \frac{251}{720}f_{k-4}\right).
\end{aligned}$$

Demonstrate their order of accuracy using the ratio test.
