Laboratory Exercise 4: Time stepping schemes

OBJECTIVES: To implement a variety of time stepping schemes for solving the initial value problem

$$\frac{dy}{dt} = F(y); \quad y(0) = y_0 \tag{1}$$

and to examine their accuracy and stability.

Work through the exercises on this sheet. Try to complete as much as possible. If you finish early, work through the optional material at the end.

Forward Euler, leapfrog, and Matsuno schemes

Three schemes to be investigated are

Forward (Euler) scheme: $$y^{(n+1)} = y^{(n)} + F(y^{(n)})\Delta t$$

Leapfrog scheme: $$y^{(n+1)} = y^{(n-1)} + 2F(y^{(n)})\Delta t$$

Forward/backward (Matsuno) scheme: $$\left.\begin{array}{l} y^{*} = y^{(n)} + F(y^{(n)})\Delta t \\ y^{(n+1)} = y^{(n)} + F(y^{*})\Delta t \end{array}\right\}$$

1. Write down the true solution for the two cases $F(y) = -\lambda y$ and $F(y) = i\omega y$ (λ and ω real, $\lambda > 0$). For these two cases, do stability analysis for the Matsuno scheme to obtain the amplification factor. Under what conditions is the Matsuno scheme stable?

2. Write a python procedure to solve (1) when y is a complex number. Your procedure should allow the user to choose any of the above three schemes. Use a python function to specify $F(y)$ so that only this code needs to be changed to use different functions F.

   Complex numbers in python can be written using the notation: `1j` for i. For example, $3 + i$ is written `3+1j`; $4 + 5i$ is written `4+5j`. Make sure to import functions from `cmath` rather than `math`.

3. Use your procedure to solve (1) for $F(y) = \lambda y$ and $\lambda = 1$ over the time interval $0 \leq t \leq 5$ with initial condition $y = 1 + i0$. Use $\Delta t = 0.1$. Plot the real part of the solution versus time. What kinds of errors are found for each scheme? Which schemes are stable? Are your conclusions consistent with the theory?

4. Repeat part 3 for $F(y) = i\omega y$ with $\omega = 1$. Now what kinds of errors are found with each scheme, and which schemes are stable?

Optional

Repeat parts 3 and 4 for some of the following alternative schemes:

Backward (Euler) scheme: $y^{(n+1)} = y^{(n)} + F(y^{(n+1)})\Delta t$

Trapezoidal implicit scheme: $y^{(n+1)} = y^{(n)} + \frac{1}{2}\left(F(y^{(n)}) + F(y^{(n+1)})\right)\Delta t$

2nd order Adams-Bashforth: $y^{(n+1)} = y^{(n)} + \frac{1}{2}\left(3F(y^{(n)}) - F(y^{(n-1)})\right)\Delta t$

3rd order Adams-Bashforth: $y^{(n+1)} = y^{(n)} + \frac{1}{12}\left(23F(y^{(n)}) - 16F(y^{(n-1)}) + 5F(y^{(n-2)})\right)\Delta t$

2nd order Runge-Kutta:

$$\left.\begin{array}{l} y^* = y^{(n)} + \frac{1}{2}F(y^{(n)})\Delta t \\ y^{(n+1)} = y^{(n)} + F(y^*)\Delta t \end{array}\right\}$$

4th order Runge-Kutta:

$$\left.\begin{array}{l} q_1 = F(y^{(n)})\Delta t \\ q_2 = F(y^{(n)} + q_1/2)\Delta t \\ q_3 = F(y^{(n)} + q_2/2)\Delta t \\ q_4 = F(y^{(n)} + q_3)\Delta t \\ y^{(n+1)} = y^{(n)} + \frac{1}{6}\left(q_1 + 2q_2 + 2q_3 + q_4\right) \end{array}\right\}$$

Note that some of these are implicit schemes: $y^{(n+1)}$ appears on both sides of the equation. So you will need to rearrange the equation to express $y^{(n+1)}$ in terms of $y^{(n)}$.

For the Adams-Bashforth methods, use a forward Euler step for the first step and, where required, a lower order Adams-Bashforth method for the next step(s).

Some of these schemes are very accurate, so you might need to use a larger time step to begin to see significant errors.