Numerical Programming 2 (CSE) 2015

Worksheet 7

Notation

All exercises on this sheet are about autonomous ODEs

$$\dot{y}(t) = f\left(y(t)\right), \; y(0) = y_0.$$

The approximate solution computed by some numerical method with step size h is $y_n \approx y(nh)$, $n = 0, 1, \ldots$.

The flow map $\Psi^\tau : y_0 \mapsto y(\tau)$ maps an initial value $y_0 = y(0)$ to the exact solution $y(\tau)$.

The discrete flow map $\widetilde{\Psi}^h : y_0 \mapsto y_1$ maps an initial value y_0 to the approximate solution y_1 after one time step of length h.

Exercise 1 (θ-method)

The family of θ-methods given by

$$y_{n+1} = y_n + h\left(\theta f(y_n) + (1-\theta) f(y_{n+1})\right), \; \theta \in [0,1]$$

includes both Euler methods and the trapezoidal rule.

- Interpret the θ-method as a 2-stage implicit Runge-Kutta scheme and write down its Butcher tableau.
- Determine the stability function $R(z)$.
- For which values of θ is the method A-stable?

  Hint: Express $|R(a+ib)|$ in terms of a and b. It may be helpful to distinguish between $\theta < \frac{1}{2}$, $\theta = \frac{1}{2}$, $\theta > \frac{1}{2}$.
- For which values of θ is the method L-stable?

Exercise 2 (B-Stability)

An ODE $\dot{y} = f(y)$ is called *dissipative* if

$$\operatorname{Re}\left\langle f(x) - f(y), x - y\right\rangle \leq 0 \;\forall x, y,$$

where $\operatorname{Re}(a+ib) = a$ is the real part of a complex number.

A numerical integration method is called *B-stable* if for every dissipative ODE the discrete flow map $\widetilde{\Psi}^h$ is *non-expansive*, i.e. it fulfills

$$\left\|\widetilde{\Psi}^h(x) - \widetilde{\Psi}^h(y)\right\| \leq \|x - y\| \;\forall x, y \;\forall h \geq 0.$$

- Show that the exact flow map Ψ^τ of a dissipative ODE is non-expansive, i.e. that it fulfills

$$\|\Psi^\tau(x) - \Psi^\tau(y)\| \leq \|x - y\| \ \ \forall x, y \ \forall \tau \geq 0.$$

- Prove that the implicit Euler method is B-stable.
- Prove that B-stable methods are A-stable.

Exercise 3 (Implementing an implicit method)

Write a MATLAB function that performs one step of the trapezoidal method and solves the implicit equation using Newton's method. f and Df should be passed to your function as function handles.

Use your implementation of the trapezoidal method to solve the two-dimensional ODE

$$\dot{y} = \begin{pmatrix} -100y_1^3 + y_2 \\ -y_1 \end{pmatrix}, \ y(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ t \in [0, 20]$$

and plot the error at $t = 20$ for different step sizes to verify that your implementation achieves the right order.