Numerical Solution of ODEs

Exercise Class

24th October 2024

Convergence Analysis

Theorem 2.23. *Let there exist a positive constant C such that the local discretisation error is bounded by*

$$d(t+\tau, t, u(t)) \leq C\tau^{p+1},$$

for all $\tau \leq \tau_1$, $t \in [t_0, T]$. Consider an equidistant partition $\{t_j\}_{j=0}^{N}$ and approximate solution $\{u_j\}_{j=0}^{N}$, where

$$u_0 = x_0, \qquad u_{j+1} = \psi(t_{j+1}, t_j, u_j), \qquad j = 0, \dots, N-1.$$

Then,

$$\|u(t_j) - u_j\| \leq \frac{e^{\Lambda(t_j - t_0)} - 1}{\Lambda} C\tau^p, \qquad j = 1, \dots, N.$$

Here, p is the order of the method.

If we study the error at the last time step

$$\underbrace{\|u(T) - u_N\|}_{\mathcal{E}_N} \leq \underbrace{\frac{e^{\Lambda(T-t_0)} - 1}{\Lambda} C}_{K \text{ — constant}} \tau^p;$$

then,

$$\log_{10} \mathcal{E}_N \leq \log_{10} K + p \log_{10} \tau.$$

Hence, we should observe asymptotically as $\tau \to 0$ that

$$\log_{10} \mathcal{E}_N = q + p \log_{10} \tau,$$

where $q = \log_{10} K$ is a constant.

Adaptive Timestepping

Algorithm 2.1. Given *two* one step methods

$$\psi \text{ — Order } p$$
$$\overline{\psi} \text{ — Order } p+1.$$

we define adaptive timestepping at each timestep as:

$\tau \leftarrow \max(\tau, \texttt{tol})$
$\delta \leftarrow \left\|\bar{\psi}(t+\tau, t, x) - \psi(t+\tau, t, x)\right\|$
while $\delta > \texttt{tol}$ **do**
  $\tau \leftarrow \tau \left(\frac{\texttt{tol}}{\delta}\right)^{1/(p+1)}$
  $\delta \leftarrow \left\|\bar{\psi}(t+\tau, t, x) - \psi(t+\tau, t, x)\right\|$
end while

Accept τ $\qquad\triangleright$ it now holds that $\left\|\bar{\psi}(t+\tau,t,x)-\psi(t+\tau,t,x)\right\| \leq \texttt{tol}$
$t \leftarrow t+\tau$
$x \leftarrow \bar{\psi}(t+\tau,t,x)$

We provide three implementations of an algorithm with $p=1$:

`ode12_1.m` Basic algorithm (using Euler and Heun)

`ode12_2.m` Adds damping to the timestep size update to prevent large changes.

`ode12.m` Adds heuristics for the initial timestep size.

Exercises

1. Modify `one_step_order.m` to calculate the order of the Runge, Runge-Kutta, Heun, Implicit Euler, and Crank-Nicholson methods, using the logistic equation

$$\begin{aligned} x'(t) &= (a-bx(t))x(t), & t \in [0,2], \\ x(0) &= x_0, \end{aligned}$$

with $a=b=1$, $x_0=2$, and known exact solution

$$x(t) = \frac{x_0 e^t}{1-x_0(1-e^t)}.$$

Remark. Note that the implicit Euler method `ieuler` may not converge for $\tau = 1/2$. Therefore, the convergence analysis code needs to be changed to start from $\tau = 1/4$.

2. Modify `ode12` to use Euler (order $p=1$ method) and Runge (order $p+1=2$ method). Solve

$$x'(t) = \begin{pmatrix} 998 & 1998 \\ -999 & -1999 \end{pmatrix} x(t), \qquad t \in [0,0.1], \tag{1}$$

$$x(0) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \tag{2}$$

implemented by `linsystem.m`, using this method.