Numerical Solution of ODEs

Exercise Class

17th October 2024

Explicit One-Step Methods

Euler Implemented by `eul.m`:

$$\begin{aligned}\kappa_1 &= f(t,x),\\ \psi(t+\tau,t,x) &= x+\tau\kappa_1.\end{aligned}$$

Runge Implemented by `runge.m`:

$$\begin{aligned}\kappa_1 &= f(t,x),\\ \kappa_2 &= f\left(t+\frac{\tau}{2}, x+\frac{\tau}{2}\kappa_1\right),\\ \psi(t+\tau,t,x) &= x+\tau\kappa_2.\end{aligned}$$

Runge-Kutta Implemented by `rk_classical.m`:

$$\begin{aligned}\kappa_1 &= f(t,x)\\ \kappa_2 &= f\left(t+\frac{\tau}{2}, x+\frac{\tau}{2}\kappa_1\right),\\ \kappa_3 &= f\left(t+\frac{\tau}{2}, x+\frac{\tau}{2}\kappa_2\right),\\ \kappa_4 &= f\left(t+\tau, x+\tau\kappa_3\right),\\ \psi(t+\tau,t,x) &= x+\tau\left(\frac{1}{6}\kappa_1+\frac{1}{3}\kappa_2+\frac{1}{3}\kappa_3+\frac{1}{6}\kappa_4\right).\end{aligned}$$

Heun

$$\begin{aligned}\kappa_1 &= f(t,x),\\ \kappa_2 &= f\left(t+\tau, x+\tau\kappa_1\right),\\ \psi(t+\tau,t,x) &= x+\frac{\tau}{2}\left(\kappa_1+\kappa_2\right).\end{aligned}$$

Implicit One-Step Methods

Implicit Euler Implemented by `ieuler.m`:

$$\begin{aligned}\kappa_1 &= f(t,x+\tau\kappa_1),\\ \psi(t+\tau,t,x) &= x+\tau\kappa_1.\end{aligned}$$

Crank-Nicholson

$$\begin{aligned}\kappa_1 &= f(t,x),\\ \kappa_2 &= f\left(t+\tau, x+\frac{\tau}{2}\kappa_1+\frac{\tau}{2}\kappa_2\right),\\ \psi(t+\tau,t,x) &= x+\frac{\tau}{2}\left(\kappa_1+\kappa_2\right).\end{aligned}$$

Fixed Point

Computing κ_1 for the *Implicit Euler* method requires solving a potentially nonlinear equation. One method is via the use of a fixed point iteration: Compute the sequence $\{\kappa_1^{(n)}\}_{n\geq 0}$ with the iteration

$$\begin{aligned} \kappa_1^{(n+1)} &= f(t+\tau, x+\tau\kappa_1^{(n)}), \qquad n \geq 1, \\ \kappa_1^{(0)} &= f(t,x). \end{aligned}$$

Continue the iteration until

$$\left\| \kappa_1^{(n+1)} - \kappa_1^{(n)} \right\| \leq \mathtt{TOL},$$

where `TOL` is a desired tolerance.

Newton's Method

As an alternative, we can also use Newton's method for solving the implicit equation (see `ieuler_newton.m`). Defining

$$\boldsymbol{F}(\kappa_1) = \kappa_1 - f(t+\tau, x+\tau\kappa_1),$$

we try to find a root of $\boldsymbol{F}(\kappa_1) = 0$, by defining the sequence $\{\kappa_1^{(n)}\}_{n\geq 0}$ as

$$\kappa_1^{(n+1)} = \kappa_1^{(n)} - \left(\frac{\partial \boldsymbol{F}}{\partial \kappa_1}(\kappa_1^{(n)}) \right)^{-1} \boldsymbol{F}(\kappa_1^{(n)})$$

where, for $\kappa_1 \in \mathbb{R}^n$ and $\boldsymbol{F}(\kappa_1) = (F_1(\kappa_1), \ldots, F_n(\kappa_1))$, we define the *Jacobian* as

$$\frac{\partial \boldsymbol{F}}{\partial \kappa_1}(\kappa_1) = \begin{pmatrix} \frac{\partial F_1}{\partial \kappa_{1,1}}(\kappa_1) & \cdots & \frac{\partial F_1}{\partial \kappa_{1,n}}(\kappa_1) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial \kappa_{1,1}}(\kappa_1) & \cdots & \frac{\partial F_n}{\partial \kappa_{1,n}}(\kappa_1) \end{pmatrix}.$$

Note, that

$$\frac{\partial \boldsymbol{F}}{\partial \kappa_1}(\kappa_1) = I - \tau f_x(t+\tau, x+\tau\kappa_1),$$

where f_x is the first derivative of f with respect to the second argument.

Exercises

1. Compare the solution obtained by the Euler, Runge, and Runge-Kutta methods with $\tau = 1/2, 1/4, 1/8$ to the solution obtained with `ode23` for the following problems:

   (a) Logistic equation

   $$\begin{aligned} x(t)' &= (a - bx(t))x(t), \qquad t \in [0,3], \\ x(0) &= x_0, \end{aligned}$$

   with $a = b = 1$ and $x_0 = 2$.

   (b) The pendulum problem:

   $$\begin{aligned} x''(t) &= -k\sin(x(t)), \\ x(t_0) &= x_0 \end{aligned}$$

   with $k = 1$, $t = (0, 6\pi)$, and various initial conditions

   $$x_0 = \begin{pmatrix} -1.5 \\ 0 \end{pmatrix}, \begin{pmatrix} -3 \\ 0 \end{pmatrix}, \begin{pmatrix} -\pi \\ 1 \end{pmatrix}.$$

(c) The harmonic oscillator

$$x''(t) + bx = c\cos(\omega t),$$
$$x(t_0) = x_0$$

with

- $a = 0$, $b = 9$, $c = 10$
- $t = [0, 50]$
- $x_0 = (1, 0)^\top$
- $\omega = 2.5, 2.9, 3.1, 3, \sqrt{3}$

2. Implement the Heun method as a MATLAB function, and test with the logistic equation with $\tau = 1/2, 1/4, 1/8$.

3. Modify `compare.m` to compare Euler (`eul.m`), Implicit Euler using a fixed point iteration (`ieuler.m`), and Implicit Euler using a Newton iteration (`ieuler_newton.m`), for solving the ODE

$$x'(t) = \begin{pmatrix} 998 & 1998 \\ -999 & -1999 \end{pmatrix} x(t), \qquad t \in [0, 0.1], \tag{1}$$
$$x(0) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \tag{2}$$

(`linsystem.m` and `linsystem_newton.m`) with $\tau = 0.002, 0.0021, 0.0019$. Also try with smaller values of τ, such as $\tau = 0.0002$ to try to reduce oscillations in the numerical solution.

Remark. Make sure to print and check the values obtained from the solver, some of these methods will return `NaN` (Not a Number) values.

4. Implement the Crank-Nicholson method as a MATLAB function using a fixed point iteration, and test for the linear system (1)–(2).