Series 11

Brusselator problem

Consider the Brusselator problem for $x \in (0,1)$ and $t \geq 0$

$$\begin{aligned} \frac{\partial u}{\partial t} &= a + u^2 v - (b+1)u + \nu \frac{\partial^2 u}{\partial x^2}, \\ \frac{\partial v}{\partial t} &= bu - u^2 v + \nu \frac{\partial^2 v}{\partial x^2}, \end{aligned} \tag{1}$$

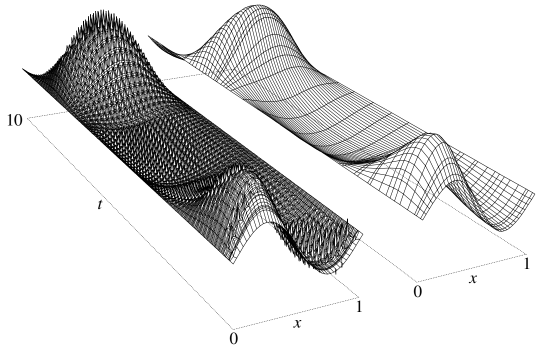
where $a = 1$, $b = 3$ and ν is the diffusion coefficient. Set the boundary conditions

$$u(0,t) = u(1,t) = 1, \qquad v(0,t) = v(1,t) = 3, \qquad t \geq 0,$$

and the initial conditions

$$u(x,0) = 1 + \sin(2\pi x), \qquad v(x,0) = 3, \qquad x \in (0,1).$$

The following plot is a numerical solution of the Brusselator problem with $\nu = 0.1$ obtained employing DOPRI5 and ROCK2.



Spatial discretization

Let $\{x_i\}_{i=0}^{N+1}$ be a discretization of the interval $[0,1]$ such that $x_i = i\Delta x$ where $\Delta x = 1/(N+1)$ and let $u_i(t)$ and $v_i(t)$ be an approximation of $u(x_i,t)$ and $v(x_i,t)$, respectively, for all $i = 0, \dots, N+1$. Then, we obtain the system of ODEs

$$\dot{y}(t) = f(y(t)) = c + By(t) + \frac{\nu}{(\Delta x)^2} Ay(t) + g(y(t)), \qquad y(0) = y_0, \tag{2}$$

where $y(t) = \big(u_1(t), \dots, u_N(t), v_1(t), \dots, v_N(t)\big)^\top \in \mathbb{R}^{2N}$, $y_0 \in \mathbb{R}^{2N}$, $c \in \mathbb{R}^{2N}$, $B \in \mathbb{R}^{2N\times 2N}$, $A \in \mathbb{R}^{2N\times 2N}$ and $g\colon \mathbb{R}^{2N} \to \mathbb{R}^{2N}$.

i) Give explicitly the system of ODEs obtained discretizing the spatial variable applying the method of lines, i.e., find y_0, c, B, A and g in (2).
Hint: use Exercise 4 of Series 10.

Linear stability analysis

The linearization of system (2) reads $\dot{z}(t) = Lz(t)$ where

$$L = \frac{\nu}{(\Delta x)^2} A + B + g'(y_0),$$

which can be seen as a perturbation of $\widetilde{L} = \frac{\nu}{(\Delta x)^2} A$ for Δx sufficiently small. Therefore, the linear stability of the solution $y(t)$ is determined up to perturbations by the eigenvalues of $\widetilde{L}$.

ii) Show that the solution $y(t)$ is stable.

iii) Perform a linear stability analysis and give the time-step restriction for the explicit Euler method. This constraint is called the Courant–Friedrichs–Lewy (CFL) condition.
Hint: use Exercise 3 of Series 10.

Chemical reaction

iv) Write a MATLAB function `[fu,fv] = reac_fun(u,v)` which computes the vectors `fu` and `fv` corresponding to the chemical reaction in (1) without the diffusion terms ($\nu = 0$), when two vectors $u = (u_1, \dots, u_N)^\top$ and $v = (v_1, \dots, v_N)^\top$ are provided.

v) Implement in MATLAB the explicit Euler method for problem (1) without diffusion ($\nu = 0$). Set $N = 20$, the final time $T = 10$ and the time step $\Delta t = T/M$ with $M = 200$. The program should output two matrices `vecu` and `vecv` of size $(N+2) \times (M+1)$ where the component (i, j) is an approximation of the solution at the position $x_{i-1} = (i-1)\Delta x$ and at the time $t_{j-1} = (j-1)\Delta t$ for $i = 1, \dots, N+2$ and $j = 1, \dots, M+1$.

vi) Plot the solution obtained in point *v*) using the following commands:

```
x = linspace(0, 1, N+2);
t = linspace(0, T, M+1);
vecx = repmat(x', 1, M+1);
vect = repmat(t, N+2, 1);
surf(vecx, vect, vecu);
surf(vecx, vect, vecv);
```

Diffusion term

vii) Write a MATLAB function `[fu,fv] = fun(u,v)` which computes the vectors `fu` and `fv` corresponding to the spatial discretization in (2), when two vectors $u = (u_1, \dots, u_N)^\top$ and $v = (v_1, \dots, v_N)^\top$ are provided.
Hint: in order to define diagonal matrices in MATLAB you can use the function `diag` (or `spdiags` to work with sparse matrices).

viii) Repeat point *v*), but now set the diffusion coefficient to be $\nu = 0.02$. Varying N and M verify that the explicit Euler method is stable under the CFL condition found in point *iii*).

ix) Implement in MATLAB the implicit Euler method for problem (1) with diffusion employing the following methods for solving the nonlinear system to compute y_{n+1} from y_n:

- fixed point iteration $y^{(k+1)} = y_n + hf(y^{(k)})$,
- Newton iteration $(I_{2N} - hf'(y^{(k)}))(y^{(k)} - y^{(k+1)}) = (y^{(k)} - y_n - hf(y^{(k)}))$,

where in both cases the value of y_{n+1} is approximated by the sequence $\{y^{(k)}\}_{k\geq 0}$ with $y^{(0)} = y_n$. Moreover, implement the following stopping criteria:

- fixed number of iteration, i.e., stop when $k > K$ with, e.g., $K = 20$,
- small increment, i.e., stop when $\|y^{(k+1)} - y^{(k)}\|_\infty < \text{tol}$ with, e.g., $\text{tol} = 10^{-10}$,
- machine precision, i.e., stop when the increment is of the order of the machine precision ϵ, that is when $\|y^{(k+1)} - y^{(k)}\|_\infty / \|y^{(k+1)}\|_\infty < \epsilon$ or $\|y^{(k+1)} - y^{(k)}\|_\infty \geq \|y^{(k)} - y^{(k-1)}\|_\infty - \epsilon$.

x) Compare the two implementations in point *ix*) for solving the nonlinear system and show that the one with the Newton iteration has the correct stability behavior, i.e., there is no restriction on the step size.