Fundamentals of Numerical Thermo-Fluid Dynamics 322.061

Exercise 6: Burger's Equation

To be presented on June 26, 2019

In examples 6.1, 6.2, 6.3 we propose to use different techniques to solve the following non linear problem:

$$\begin{cases} \partial_t u + u\,\partial_x u - \nu \partial_{xx} u = 0 \quad , \text{on } \Omega = [0,1] \\ u(t=0,x) = e^{-\frac{(x-0.5)^2}{0.02}} \\ u(t,x=0) = 0 \quad \frac{\partial}{\partial x} u(t,x=1) = 0 \end{cases} \tag{1}$$

where the diffusivity $\nu = 10^{-2}$

6.1) Solve the problem with second-order centered finite differences in space and Euler method in time. Treat explicitly the non linear term, and implicitly the linear term (semi-implicit approach):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{\Delta x^2} = -u_i^n \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \tag{2}$$

Write a MatLab code to compute the evolution of u on a grid of $N = 1000$ points with time-step $\Delta t = 0.001$.

6.2) We propose here to solve this problem by treating the non-linear term implicitly through a fixed point algorithm.

$$\frac{V_i^{k+1} - u_i^n}{\Delta t} - \nu \frac{V_{i-1}^k - 2V_i^k + V_{i+1}^k}{\Delta x^2} = -V_i^k \frac{V_{i+1}^k - V_{i-1}^k}{2\Delta x} \tag{3}$$

- Put this problem in a fixed point function *i.e.* $V^{k+1} = f(V^k)$.
- Write a MatLab code to compute the evolution of u on a grid of $N = 1000$ points with time-step $\Delta t = 0.001$.
- See what happens for different relaxation factor θ (usually $\theta \approx 0.01$).

Algorithm 1 My fixed point

procedure GETSTEPFIXEDPOINT(u^n,tol,k_{max},θ)
  $V_1 \leftarrow u^n$; $k \leftarrow 0$
  while $tol > \delta$ and $k < k_{max}$ **do**
    $V^{k+1} \leftarrow f(V^k, u^n)$
    $\delta \leftarrow norm(V^{k+1} - V^k)/norm(V^k)$
    $V^{k+1} \leftarrow \theta V^{k+1} + (1-\theta)V^k$
    $k \leftarrow k + 1$
  end do
  return V^{k+1}

6.3) We propose in the following part to solve the Burger equation using the Newton method. The non-linear function that one consider is then:

$$F_i(V, u^n) = \frac{V_i - u_i^n}{\Delta t} - \nu \frac{V_{i-1} - 2V_i^+ V_{i+1}}{\Delta x^2} + V_i \frac{V_{i+1} - V_{i-1}}{2\Delta x}. \qquad (4)$$

- Differentiate (4) with respect to V_{i-1},V_i and V_{i+1}. Find that the jacobian J is a tridiagonal matrix. We recall that

$$J(V, u^n) = \begin{pmatrix} \partial_{V_1} F_1 & \partial_{V_2} F_1 & \cdots & \partial_{V_i} F_1 & \cdots & \partial_{V_N} F_1 \\ \vdots & & & \vdots & & \vdots \\ \partial_{V_1} F_i & \partial_{V_2} F_i & \cdots & \partial_{V_i} F_i & \cdots & \partial_{V_N} F_i \\ \vdots & & & \vdots & & \vdots \\ \partial_{V_1} F_N & \partial_{V_2} F_N & \cdots & \partial_{V_i} F_N & \cdots & \partial_{V_N} F_N \end{pmatrix}.$$

- Solve the problem numerically with Newton method (Algorithm 2) on a grid of $N = 1000$ points with time-step $\Delta t = 0.001$.

Algorithm 2 My Newton

procedure GETSTEPNEWTON(u^n,tol,k_{max})
  $V_1 \leftarrow u^n$
  $k \leftarrow 1$
  while $tol > \delta$ and $k < k_{max}$ **do**
    $F \leftarrow createVector_F(V^k, u^n)$
    $J \leftarrow createJacobian(V^k, u^n)$
    $V^{k+1} = V^k - J^{-1}F$
    $\delta \leftarrow norm(V^{k+1} - V^k)/norm(V^k)$
    $k \leftarrow k + 1$
  end do
  $u^{n+1} \leftarrow V^{k+1}$
  return u^{n+1}

6.4) Solve the two-dimensional Burger's equation

$$\begin{cases} \partial_t u + u\,\partial_x u - \nu(\partial_{xx} + \partial_{yy})u = 0 \quad ,\text{on } \Omega = [0,1]\times[01] \\ u(t=0,x,y) = 10^{-2} e^{-\frac{(x-0.5)^2}{0.04}} e^{-\frac{(y-0.5)^2}{0.04}} \\ u(t,\partial\Omega) = u(t=0,\partial\Omega) \end{cases} \tag{5}$$

with diffusivity $\nu = 1$ on a grid of 50×50 points with time-step $\Delta t = 10^{-5}$ using the semi-implicit Euler scheme in time and centered finite differences in space. You may use the enclosed template.

6.5) Find numerically the elliptic fixed point (= center of vortex) of the following steady flow:

$$\begin{aligned} u(x,y) &= y^3 \\ v(x,y) &= -x^3 \quad \text{on } \Omega = [0,1]\times[0,1] \end{aligned} \tag{6}$$

such that

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} \tag{7}$$

starting from the initial guess

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8}$$

Use the fixed-point algorithm with relaxation factor $\theta = 0.1$ and the Newton-Raphson method. Then vary θ and compare the convergence of the two methods. You may use the enclosed template.

Bonus examples

B.1) Compute the streamline from example 4.3 with 2nd order Adams-Moulton scheme

$$\vec{y}^{n+1} = \vec{y}^n + \frac{1}{2}\Delta t\left(\vec{f}(t^{n+1},\vec{y}^{n+1}) + \vec{f}(t^n,\vec{y}^n)\right) \tag{9}$$

Compare the accuracy to the second order Runge-Kutta method.

B.2) Recall the convection-diffusion equation from example 5.3

$$\begin{cases} \partial_t T = -u\partial_x T + D\ \partial_{xx} T \quad ,\text{on } \Omega = [0,1] \\ (\vec{u}\cdot\vec{n})T - D\dfrac{\partial T}{\partial \vec{n}} = 0 \quad \text{on } \partial\Omega \end{cases} \tag{10}$$

which models the transport of a passive scalar T by a flow u. Now assume that the flow velocity varies in space as

$$u(x) = x(1-x) \tag{11}$$

Solve the transport of T starting from the initial condition

$$T(t=0,x) = e^{-\frac{(x-0.5)^2}{0.02}} \tag{12}$$

using second-order central finite differences with $\Delta x = 0.01$ and Crank-Nicolson scheme with $\Delta t = 0.01$. Take a lower diffusivity $D = 0.01$ and show that the total amount of T in the domain is no longer conserved. Explain.