Fundamentals of Numerical Thermo-Fluid Dynamics 322.061

Exercise 7: Burger's Equation

June 20, 2018

In this exercise we propose to use different techniques to solve the following non linear problem:

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

7.1) Solve numerically the problem by treating explicitly the non linear term, and implicitly the linear terms. *(Functions from previous exercises can be used ...)*

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

7.2) We propose here to solve this problem by solving a non-linear problem 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.* $V^{k+1} = f(V^k)$.
- Find the solution, implementing Algorithm 1 to solve one time step.
- 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}

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

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

- 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 numerically the problem, by implement a Newton method solving U for a time step (*cf.* Algorithm 2).

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}

7.4) What can you conclude concerning the convergence of each method, and how fast each method converges to a solution ? Can you think of a case in which the Newton method cannot be applied ?