Universitat Politècnica de Catalunya

Master of Science in Computational Mechanics

Finite Elements in Fluids

HW4
Unsteady and nonlinear problems

Author:
Carlos Eduardo Ribeiro Santa Cruz Mendoza

March 19, 2019

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

1 Unsteady convective transport

The proposed problem is to solve the propagation of a steep front as presented in Equation 2.1 via three different methods, namely the Crank Nicolson, the Lax-Wendroff and the Third order Taylor-Galerking (TG3) method.

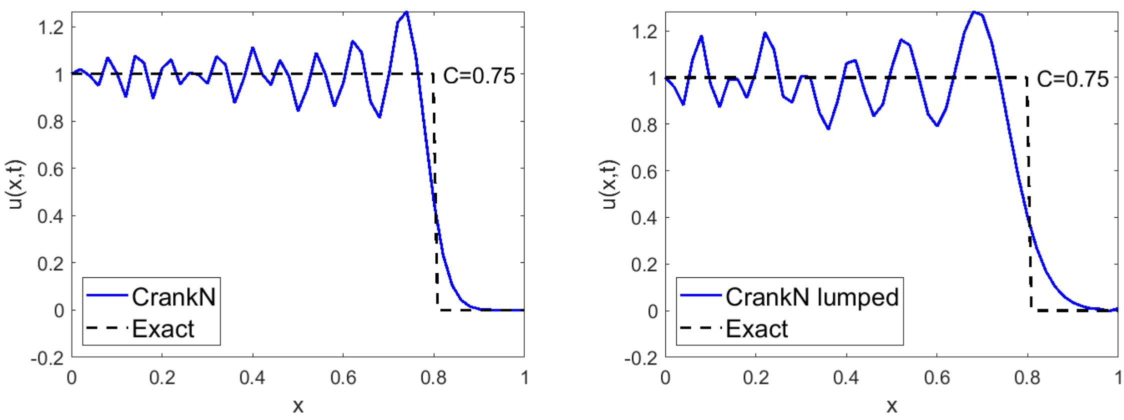
$$\begin{cases} u_t + a u_x = 0 & x \in (0,1), t \in (0, 0.6] \\ u(x,0) = u_0(x) & x \in (0,1) \\ u(0,t) = 1 & t \in (0, 0.6] \end{cases} \qquad u_0(x) = \begin{cases} 1 & \text{if} x \leq 0.2, \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

The convection velocity takes the value $a = 1$ and the space and time discretization are given by $\Delta x = 0.02$ and $\Delta t = 0.015$. Thus, the Courant number can be calculated as described in Equation 1.2.

$$C = a\frac{\Delta t}{\Delta x} = 0.75 \tag{1.2}$$

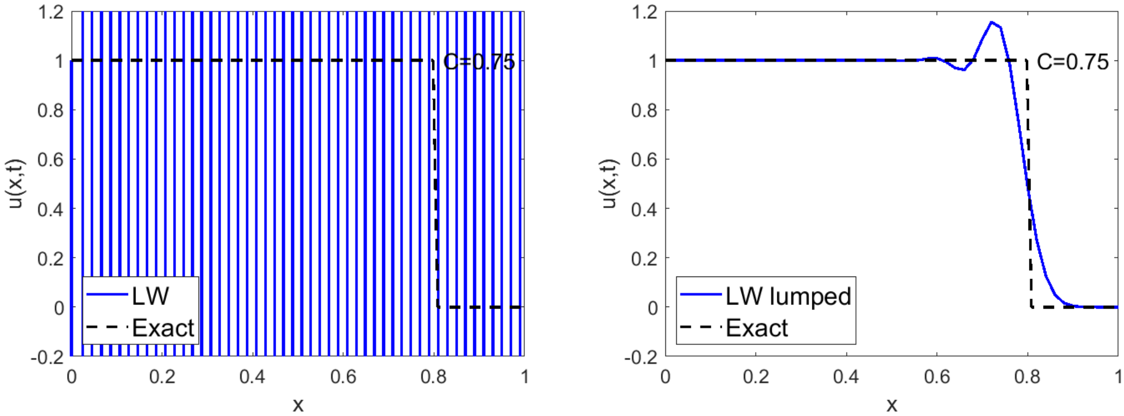
This result already points out that the Lax-Wendroff method can be subject to instabilities, since it's conditionally stable for $C^2 \leq 1/3$. On the other hand, the TG3 will be stable, for it's condition states $C^2 \leq 1$. It's also worth mentioning that the Crank Nicolson method is unconditionally stable. Given these remarks, the lumped-mass variation of the methods were also implemented due to their greater stability range [1].

The results for the Crank Nicolson method are presented on Figure 1.1.



(a) Consistent mass matrix (b) Lumped mass matrix

Figure 1.1: Crank Nicolson method at $t = 0.6$



(a) Consistent mass matrix (b) Lumped mass matrix

Figure 1.2: Lax-Wendroff method at $t = 0.6$

As mentioned before, the Crank Nicolson method is unconditionally stable and, thus, the lumped mass matrix adds no benefit to the solution. In fact, it lowers accuracy due to it's non consistent formulation, as we can see comparing Figures 1.1a and 1.1b. In general, though, the method performs poorly, pointing to the need for a higher order approximation.

The results for the Lax-Wendroff method are given on Figure 1.2. As expected, the Lax-Wendroff method is severely unstable for the Courant number considered, yielding useless results. However, using the lumped mass matrix was effective in avoiding oscillations and provided a somewhat acceptable result.

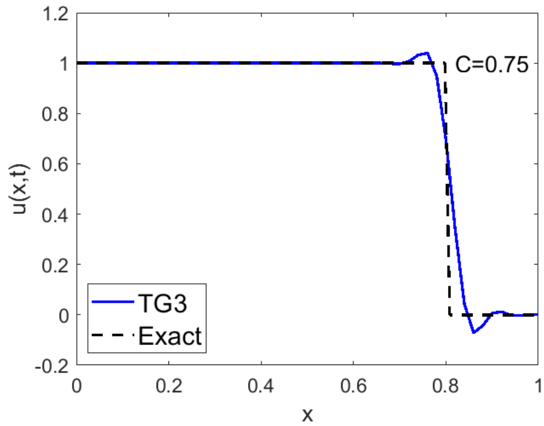


Figure 1.3: Third order Taylor-Galerkin method at $t = 0.6$

Finally, the results for the TG3 method are given on Figure 1.3. Being withing it's stability condition and having a higher order approximation in time than the other methods, the TG3 method captures well the convection of the steep front.

2 Burger's equation

The Burger's Equation (2.1) can be solved via three different schemes, namely the Explicit (Forward Euler), the Implicit Picard's method and the implicit Newton-Raphson's method. The proposed problem is defined on the $[0, 4]$ domain and it's initial condition is depicted on Figure 2.1.

$$\begin{cases} u_t + uu_x = 0 \\ u(x,0) = u_0(x) \end{cases} \tag{2.1}$$

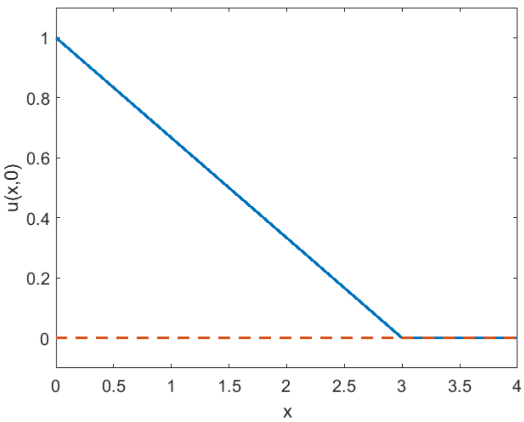


Figure 2.1: Initial condition ($t = 0$)

Such initial conditions generate discontinuous solutions, requiring the usage of the vanishing viscosity approach. A viscosity is added to Equation 2.1 as seen on Equation 2.2.

$$u_t + tt_x = \epsilon u_{xx} \tag{2.2}$$

The result is, then, acquired as the viscosity tends to zero.

After the Galerkin discretization the Burger's equation can be written as:

$$\boldsymbol{M}\frac{\Delta \boldsymbol{U}}{\Delta t} + \boldsymbol{C}(\boldsymbol{U})\boldsymbol{U} + \epsilon \boldsymbol{K}\boldsymbol{U} = 0 \tag{2.3}$$

The Newton-Raphson method consists of solving the equation $f(\boldsymbol{U}^{n+1}) = 0$ every time step, where $f(\boldsymbol{U})$ is giving by:

$$\boldsymbol{f}(\boldsymbol{U}) = (\boldsymbol{M} + \Delta t \boldsymbol{C}(\boldsymbol{U}) + \epsilon \Delta t \boldsymbol{K})\boldsymbol{U} - \boldsymbol{M}\boldsymbol{U}^n \tag{2.4}$$

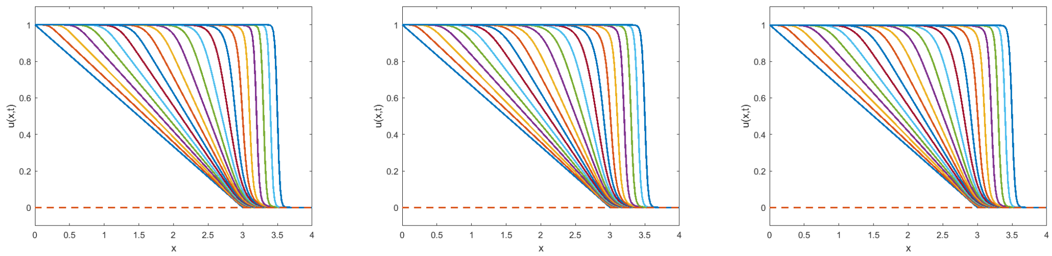
Then, an iteration is made starting from the previous time-step $\boldsymbol{U}^{n+1} = \boldsymbol{U}^n$ until it converges according to a specified tolerance $(0,5 \cdot 10^{-5})$ using the following expression:

$$\boldsymbol{U}_{k+1}^{n+1} = \boldsymbol{U}_k^{n+1} - \boldsymbol{J}^{-1}(\boldsymbol{U}_k^{n+1})\boldsymbol{f}(\boldsymbol{U}_k^{n+1}) \tag{2.5}$$

where $\boldsymbol{J} = \frac{d\boldsymbol{f}}{d\boldsymbol{U}}$ is the jacobian.

when the iteration U_{k+1}^{n+1} is within the tolerance in comparison to the previous value, the program has found the solution for the given time step and starts the calculation of the following one.

The results for all methods are presented on Figure 2.2, followed by a comparison of all methods on the last time-step on Figure 2.3.



(a) Explicit (b) Implicit Picard's (c) Implicit NR's

Figure 2.2: Solution for each scheme

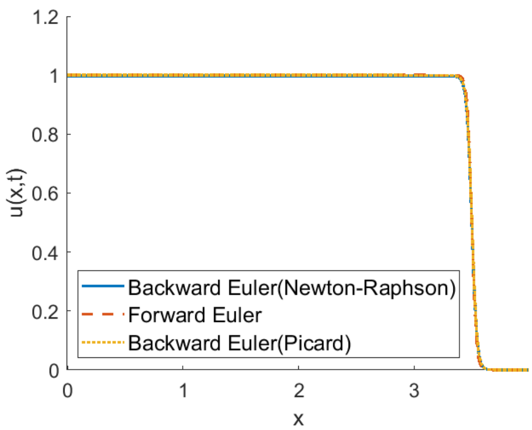


Figure 2.3: Comparison at final time-step ($t = 4$)

As we can see all methods perform similarly for the given conditions of discretization, time-step and tolerance. However, the Newton-Raphson's method has a quadratic convergence as opposed to a linear convergence of the Picard's method, which might play a significant role on computational cost when requiring the same accuracy.

Bibliography

[1] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley & Sons, 2003.