FINITE ELEMENTS IN FLUIDS
Master of Science in Computational Mechanics/Numerical Methods
Spring Semester 2019

Rafel Perelló i Ribas

Laboratory: 2nd session

Unsteady convection

In this assignment, the transport problem is solved using four different methods.

The problem is already discretised using a Crank-Nicolson method and consistent mass matrix. For the assignment, the Lax-Wendroff method has been implemented as well as the lumped matrices for both methods. The initial condition is a step. The results here are presented for the case of a Courant number of 0.6:

- Convection velocity of 1.2
- Spatial domain length of 3
- Discretized in 240 linear elements
- Total time of 1.5
- Discretized in 240 elements

Lax-Wendroff with consistent mass matrix:

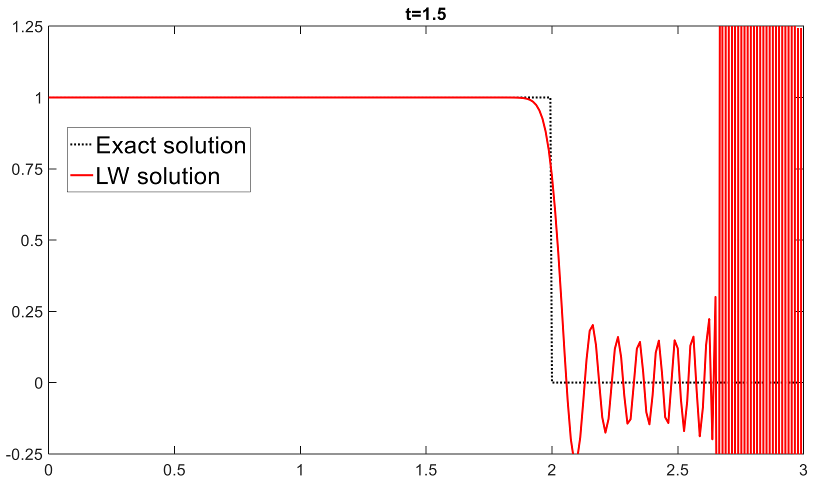

Figure 1: Lax-Wendroff method with consistent mass matrix

Lax-Wendroff with lumped mass matrix:

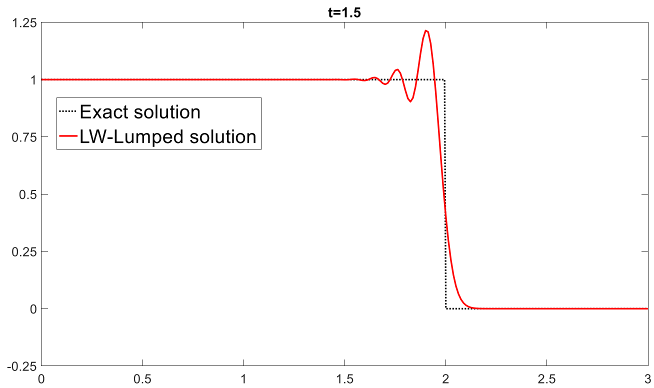

Figure 2: Lax-Wendroff method with lumped mass matrix

Crank-Nicolson with consistent mass matrix:

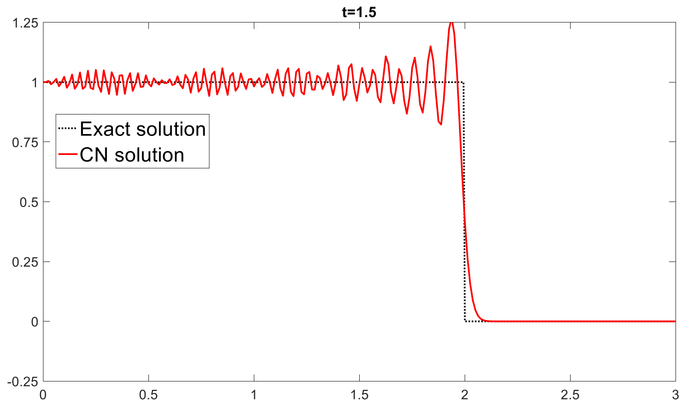

Figure 3: Crank-Nicolson method with consistent mass matrix

Crank-Nicolson with lumped mass matrix:

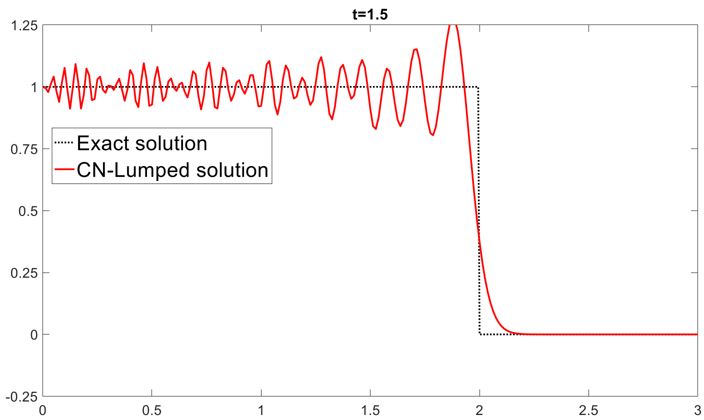

Figure 4:Crank-Nicolson method with lumped mass matrix

From the results it is seen that the solution of all methods is unstable with the exception of the Lax-Wendroff with lumped mass matrix. However, this method is dispersive as can be seen with the oscillations around the front. The behaviour of all methods is what it was expected.

Burgers equation

In this case, the Burgers equation has been solved with decreasing input data in order that a discontinuity appears.

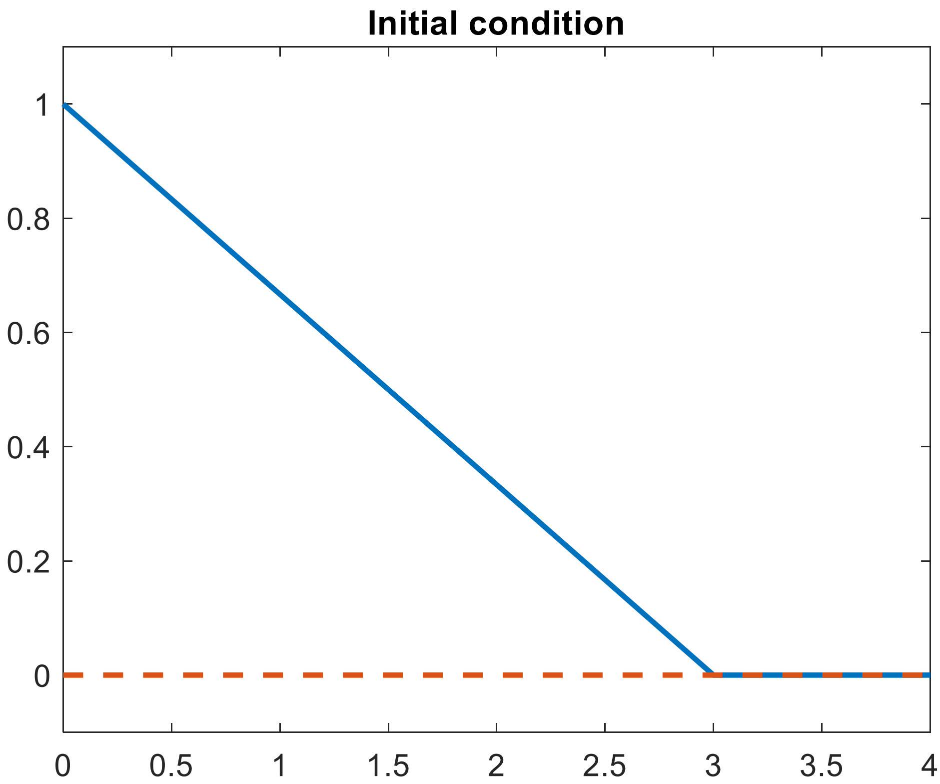

Figure 5: Initial condition for the Burgers equation

It has implemented a Newton-Rapson method. For that the equation to solve in each step is:

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

To implement the method, the Jacobian of f is needed:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{U}} = \boldsymbol{A} + \Delta t \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{U}} \boldsymbol{U}$$

The las term is:

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{U}} \boldsymbol{U}\right]_{ij} = \frac{\partial}{\partial u_k} \int N_i \cdot N_j \cdot N_j' u_j dx \, u_k = \int N_i \cdot N_j \cdot N_j' dx \, \delta_{jk} u_k = \int N_i \cdot N_j \cdot N_j' dx \, u_j = \boldsymbol{C}$$

That means that the Jacobian has the following form:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{U}} = \boldsymbol{A} + \Delta t \boldsymbol{C}$$

This has been implemented in the MATLAB code with low diffusion in order to see the differences between the methods. The parameters are the following:

- $\text{Total time} = 4$
- $\Delta t = 5 \cdot 10^{-3}$
- $\epsilon = 1 \cdot 10^{-4}$

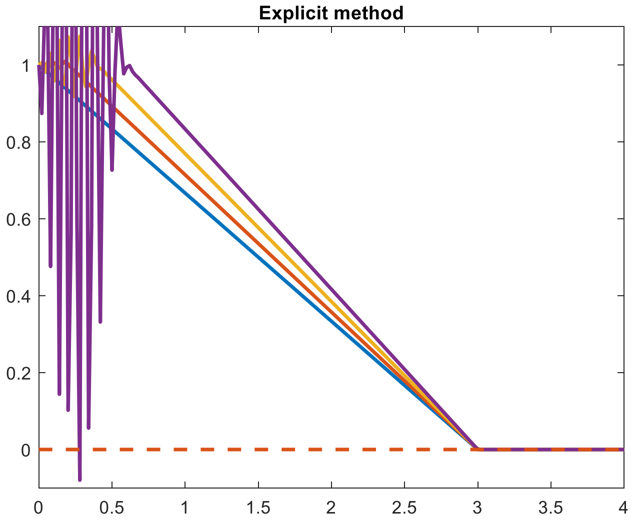

Figure 6: Solution with explicit method

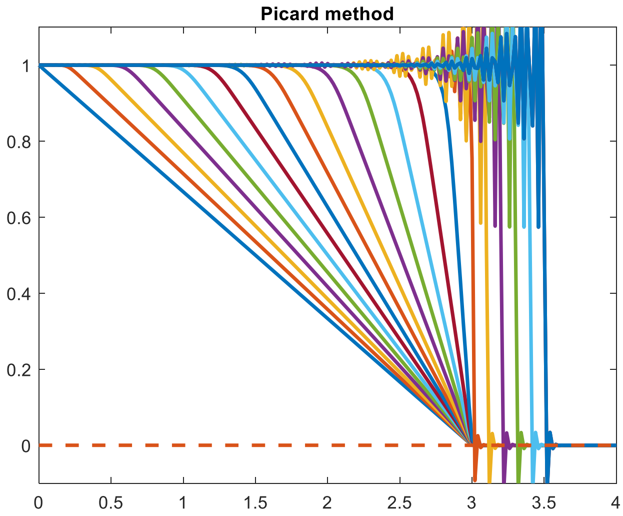

Figure 7: Solution with Picard method

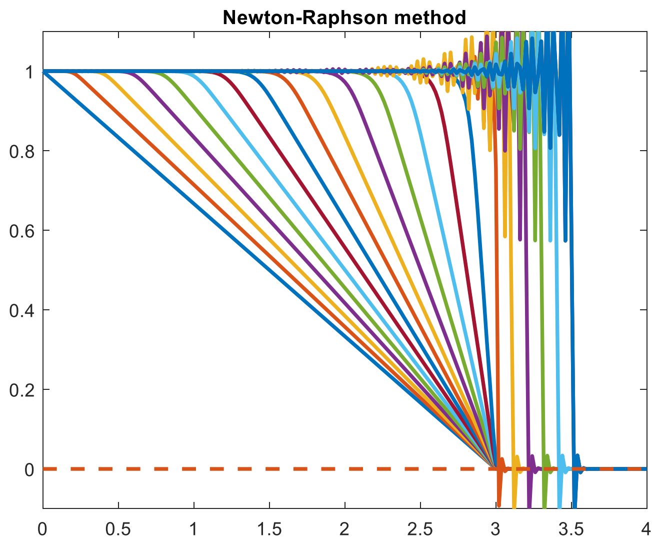

Figure 8: Solution with Newton-Raphson

Here it is seen that the solution is unstable because the diffusion should be increased. In the case of the explicit method the solution explodes after a few time steps. However, in the implicit, the solution become unstable when the shocked is formed. Solution for the Picard and Newton-Raphson method are the same as expected.