Finite Elements in Fluid

Homework 4b: Nonlinear hyperbolic problems numerical examples

Ye Mao

Ye Mao, mao.ye@estudiant.upc.edu
Master of Numerical methods on engineering - Universitat Politècnica de Catalunya

1. INTRODUCTION

1D Cauchy problem

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

Where $f(u)$ is a nonlinear function of the unknown u.

Example: inviscid Burgers' equation

$$\begin{cases} u_t + (\frac{u^2}{2})_x = 0 \\ u(x,0) = u_0(x) \end{cases}$$

The flux function is $f(u) = \frac{u^2}{2}$

Nonlinear transport uu_x with unknown convection u.

Nonlinear transport equation where the convection velocity is the solution itself

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

The solution is constant along each characteristic and the characteristics are straight lines.

Weak form: find $u(x,t)$ such that $u(x,0) = u_0(x)$

$\int_0^L vu_t\,dx + \int_0^L vuu_x\,dx + \varepsilon \int_0^L v_x u_x\,dx = 0 \ \ \forall v$ such that $v(0) = v(L) = 0$

Discretization:

$$u(x,t) \approx u_h(x,t) \approx \sum_j N_j(x) u_j(t)$$

$$v(x) = N_i(x)$$

$$\boldsymbol{M\dot{U}} + \boldsymbol{C(U)U} + \boldsymbol{\varepsilon KU} = \boldsymbol{0}$$

Picard method:

$$(\boldsymbol{M} + \Delta t(\boldsymbol{C}(\boldsymbol{U}^{n+1}) + \boldsymbol{\varepsilon K})\boldsymbol{U}^{n+1} = \boldsymbol{MU}^n$$

$${}^{0}\boldsymbol{U}^{n+1} = \boldsymbol{U}^n$$

$${}^{k+1}\boldsymbol{U}^{n+1} = \boldsymbol{A}^{-1}({}^{k}\boldsymbol{U}^{n+1})\ ({}^{k}\boldsymbol{MU}^n)$$

Newton-Raphson method:

$$\mathbf{f}(\boldsymbol{U}^{n+1}) = \boldsymbol{0}$$

$$\mathbf{f}(\mathbf{U}) = (\boldsymbol{M} + \Delta t(\boldsymbol{C}(\boldsymbol{U}) + \boldsymbol{\varepsilon K})\boldsymbol{U} - \boldsymbol{M}\boldsymbol{U}^n = \mathbf{0}$$

$$^{0}\boldsymbol{U}^{n+1} = \boldsymbol{U}^n$$

$$^{k+1}\boldsymbol{U}^{n+1} = {}^{k}\boldsymbol{U}^{n+1} - \boldsymbol{J}^{-1}({}^{k}\boldsymbol{U}^{n+1})\mathbf{f}({}^{k}\boldsymbol{U}^{n+1}),$$ where $\boldsymbol{J} = \frac{\boldsymbol{df}}{\boldsymbol{dU}}$ is the jacobian matrix

2. OBJECTIVE
   2.1 Complete the code to solve the problem with NR scheme.
   2.2 Explain the main changes in code.
   2.3 Compare and discuss the result.

3. METHODOLOGY AND RESULTS
   3.1 Explain the main changes in code.
   The mainly codes change could be found following,

```
    while (error_U > 0.5e-5) && k < 20
        C = computeConvectionMatrix(X,T,U0);
        F = (M + At*C+At*E*K)*U0-M*U(:,n);
        J = M+2*At*C+At*E*K;
        U1 = U0-J\F;
        error_U = norm(U1-U0)/norm(U1);
        fprintf('\t Iteration %d, error_U=%e\n',k,error_U);
        U0 = U1; k = k+1;
    end
    U(:,n+1) = U1;
end
```

We can obtain $\frac{\partial f(U)}{\partial U} = \mathbf{A}(\mathbf{U}) + \frac{\partial C(U)}{\partial U} \bullet \boldsymbol{U}$,
where $\mathbf{f}(\mathbf{U}) = \mathbf{A}(\mathbf{U}) \bullet \boldsymbol{U}, \mathbf{A}(\mathbf{U}) = \boldsymbol{M} + \Delta t(\boldsymbol{C}(\boldsymbol{U}) + \boldsymbol{\varepsilon K})$.
The difficult point is the term $\frac{\partial C(U)}{\partial U}$. Our codes should be focused on this partial differential calculation.
First of all, F is computed as $\mathrm{M} + \mathrm{At} * \mathrm{C} + \mathrm{At} * \mathrm{E} * \mathrm{K}$ time velocity vector U0. U0 takes values that U(:,n) at every nTimeStep. This vector changes while the condition is reached. Second, M multiplying a matrix U(:,n) is in n number of time steps. This matrix can be actualized while condition is reached. While the condition is obtained, the U0 vector in the second interation will take the last column of U's value. And the last solution is used to calculate the new one. Then, Newton-Raphson method's Jacobian is J, and evaluation function is F. Finally, the derivative of convection matrix with respect to the solution is approximated with means of the residual method by $2 * \mathrm{dt} * \mathrm{C}$ while implement the Jacobian.

3.2 Compare the results.
   3.2.1 Case1,

$$\mathrm{u0} = [\mathrm{zeros}(\mathrm{size}(\mathrm{X}(\mathrm{X} < 1)));\ 0.5 * \mathrm{ones}(\mathrm{size}(\mathrm{X}(\mathrm{X} >= 1)))]$$

Condition: $\Delta \mathrm{t} = 0.005$ and $\mathrm{E} = 1e^{-3}$

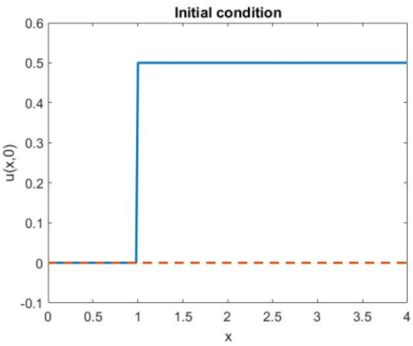

Figure 1. Initial condition

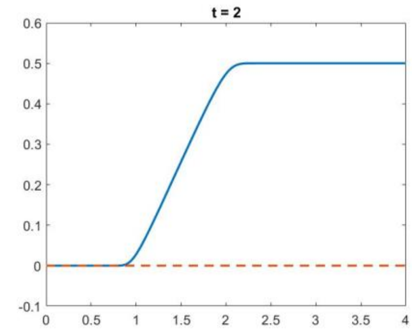

Figure 2. Final solution at t=2s

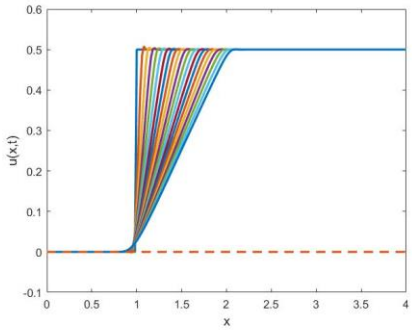

Figure 3. Explicit method at t=2s

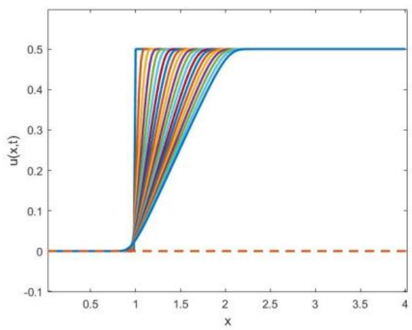

Figure 4. Picard method at t=2s

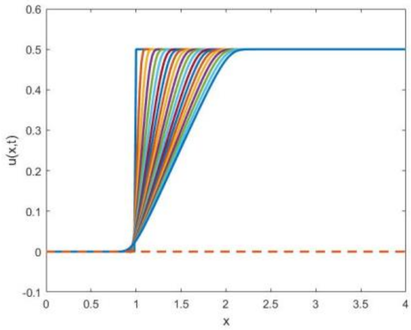

Figure 5. Newton Raphson method at t=2s

Condition: $\Delta t = 0.005$ and $E = 0$

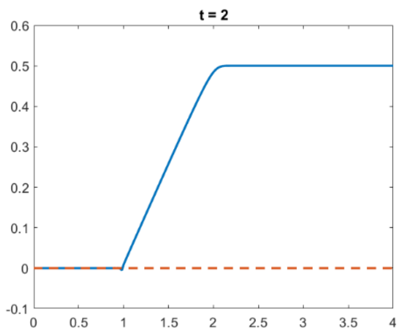

Figure 6. Final solution at t=2s

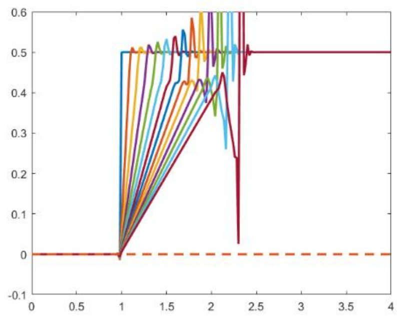

Figure 7. Explicit method at t=2s

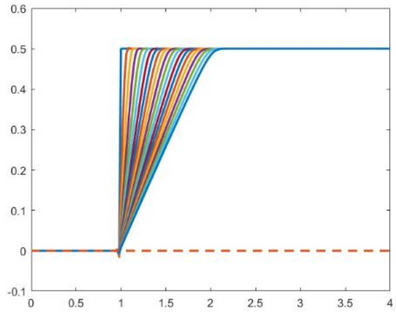

Figure 8. Picard method at t=2s

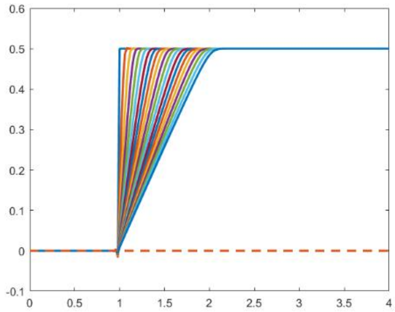

Figure 9. Newton Raphson method at t=2s

We can find that the effect of low timesteps and diffusivity coefficients close to 0. In the case $\Delta t = 0.005$ and $E = 0$, it is not accurate in solution explicit comparing with the implicit one, both Picard and Newton Raphson. The explicit method's diffusivity does not work efficiently. The solution gets the shape in entropy compliant solution of Burger's equation while the initial condition increasing.

3.2.2 Case2,

$$u0 = [1 - X(X < 3)/3;\ X(X >= 3) * 0]$$

Condition: $\Delta t = 0.005$ and $E = 1e^{-2}$

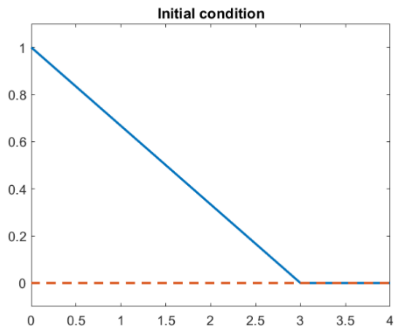

Figure 10. Initial condition

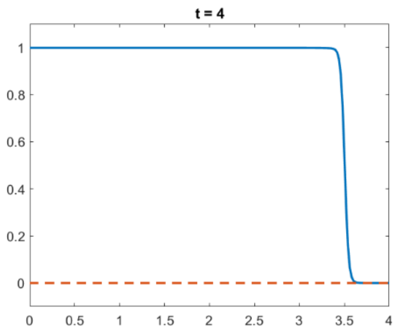

Figure 11. Final solution at t=4s

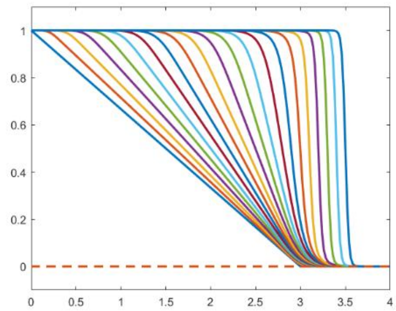

Figure 12. Explicit method at t=4s

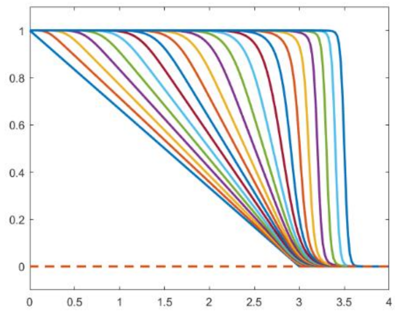

Figure 13. Picard method at t=4s

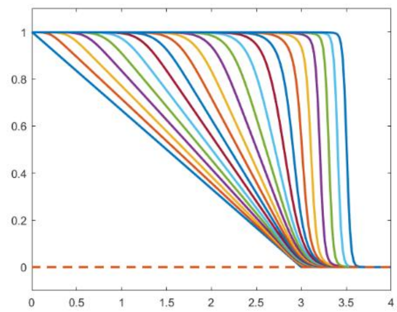

Figure 14. Newton Raphson method at t=4s

Condition: $\Delta t = 0.05$ and $E = 1e^{-2}$

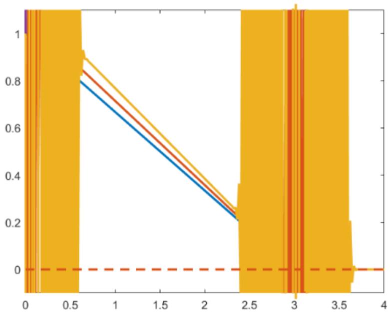

Figure 15. Explicit method at t=4s

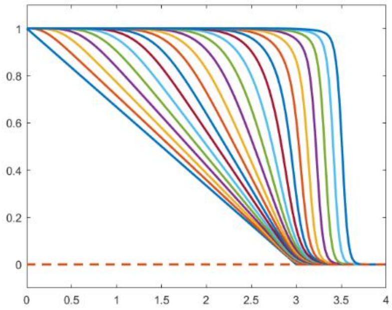

Figure 16. Picard method at t=4s

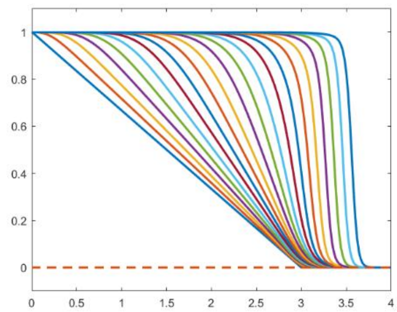

Figure 17. Newton Raphson method at t=4s

Condition: $\Delta t = 0.1$ and $E = 1e^{-2}$

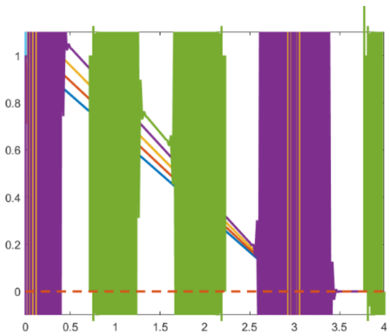

Figure 18. Explicit method at t=4s

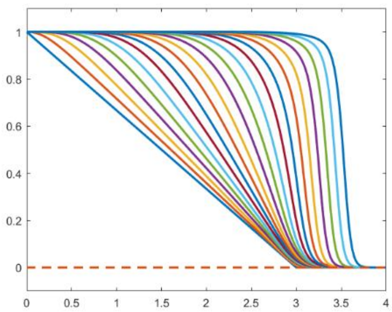

Figure 19. Picard method at t=4s

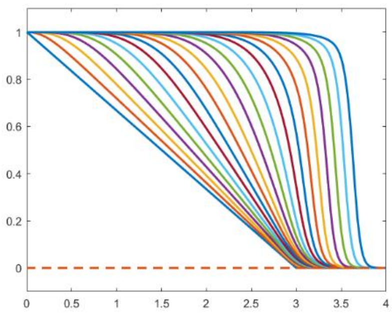

Figure 20. Newton Raphson method at t=4s

Condition: $\Delta t = 0.005 \text{ and } E = 1e^{-4}$

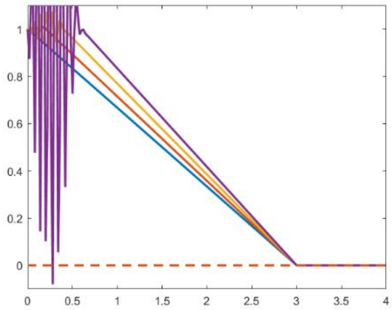

Figure 21. Explicit method at t=4s

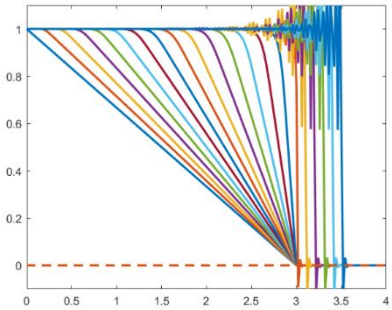

Figure 22. Picard method at t=4s

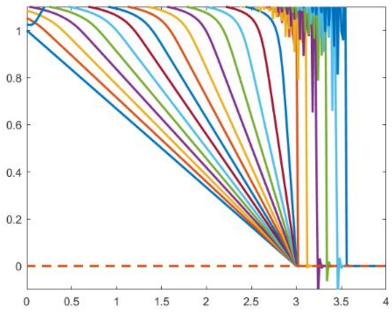

Figure 23. Newton Raphson method at t=4s

In this case, the initial date decreasing. The explicit method does not work while time step increasing and diffusion lower, for example what happen between

$\Delta t = 0.005 \text{ and } E = 1e^{-4}$ and $\Delta t = 0.005 \text{ and } E = 1e^{-2}$.

The diffusivity is useful in implicit methods. When the initial date decreasing, it is continuities in solution. And when the diffusivity is large enough, the time step increment can keep the implicit methods' solution stable.

4. Reference

[1] Lecture slides in Finite elements in fluid.

[2] Finite Element Methods for Flow Problems, Jean Donea and Antonio Huerta.