HOMEWORK BURGER'S EQUATION

-FINITE ELEMENTS IN FLUIDS-

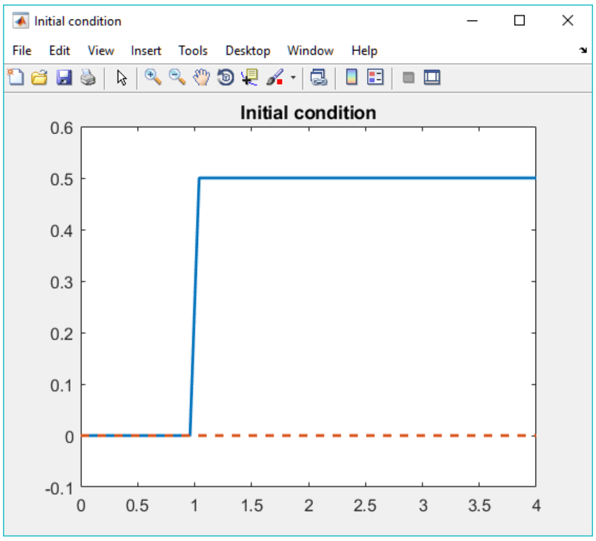
Marcos Boniquet

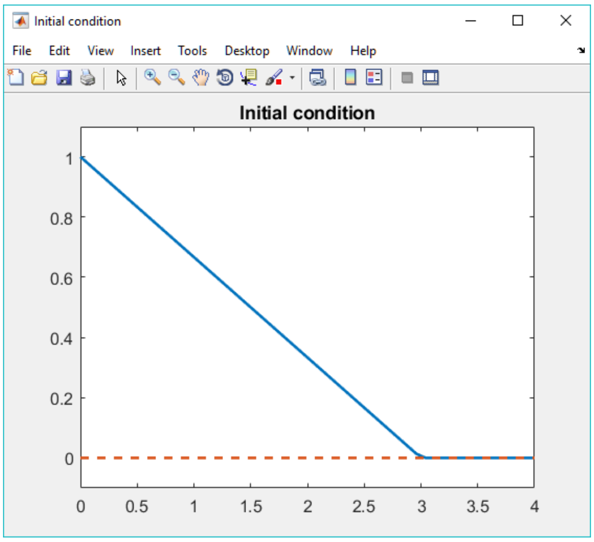
Burger's equation:

$$U_t + UU_x = 0$$

- x∈[0,4] and **m=200**
- Tf = 4;
- Δt = 0.005;

Two initial problems considered:





Integrating by parts and introducing perturbation:

$$\mathbf{M\mathring{U}+C(U)U+\epsilon KU=0}$$

where Є is perturbation (NUMERICAL DIFUSSION) and M,K;C are the mass, diffusion and convection matrices:

$M=\int NNd\Omega$
$K=\int N_x N_x d\Omega$
$C(\mathbf{U})=\int N\mathbf{U}N_x d\Omega$

4 methods to solve , depending on U and solving implicit method:

- **EXPLICIT**

  $U=U_n$; $U_t=(U_{n+1}-U_n)/\Delta T \rightarrow \ MU^{n+1}=[M-\Delta T(C(U^n)+\epsilon K]U^n$

- **IMPLICIT PICARD**

  $U=U_{n+1}$; $U_t=(U_{n+1}-U_n)/\Delta T \rightarrow [M+\Delta T(C(U^{n+1})+\epsilon K)]U^{n+1}=MU^n$
  solving by ${}^{k+1}U^{n+1}=A^{-1}({}^kU^{n+1})(MU^n)$

- **IMPLICIT NEWTON-RAPHSON**

  $U=U_{n+1}$; $U_t=(U_{n+1}-U_n)/\Delta T \rightarrow [M+\Delta T(C(U^{n+1})+\epsilon K)]U^{n+1}=MU^n$
  solving by ${}^{k+1}U^{n+1}={}^kU^{n+1}\cdot J^{-1}({}^kU^{n+1})\, f({}^kU^{n+1})$

- **CRANK-NICHOLSON** (*PICARD* by default) $\mathbf{U=(U_{n+1}+U_{n-1})/2}$;

  $U_t=(U_{n+1}-U_n)/\Delta T \rightarrow [M+\Delta T*\Theta(C(U^{n+1})+\Delta T\Theta\epsilon K)]U^{n+1}=[M-\Delta t*(1-\Theta)*C-\Delta t*(1-\Theta)*\epsilon*K]*U^n$
  with $\Theta=½$.
  solving by ${}^{k+1}U^{n+1}=A^{-1}({}^kU^{n+1})(MU^n)$

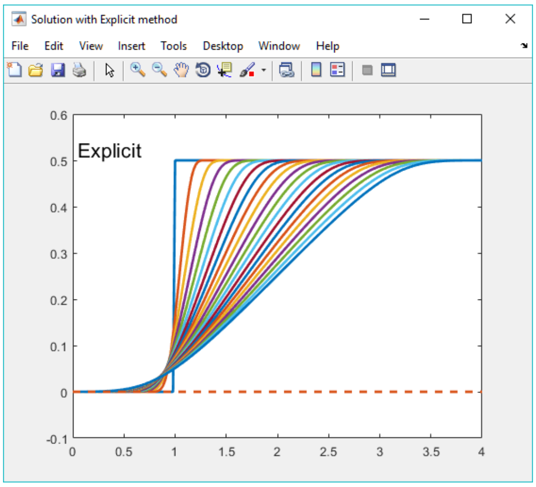
RESULTS

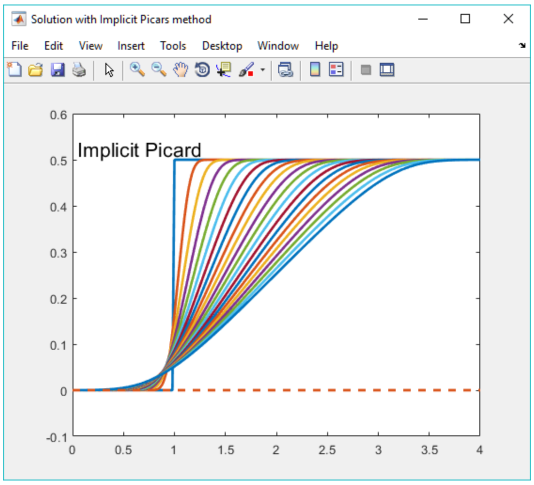
2 cases of Perturbation Є (NUMERICAL DIFFUSION):

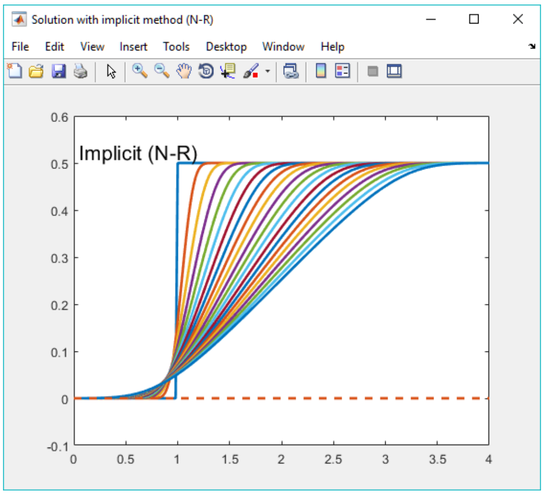
- Є= $1*10^{-2}$;
- Є = $1*10^{-4}$

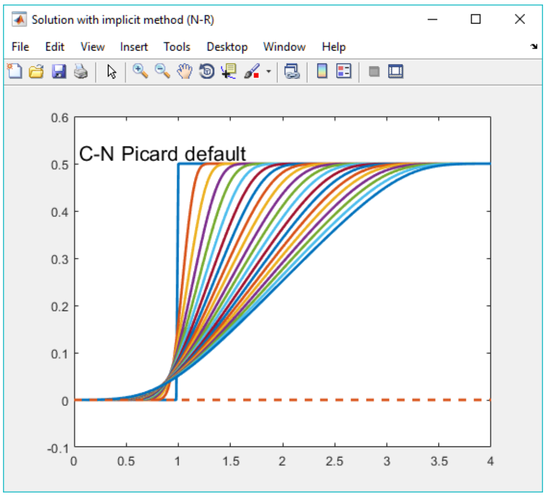
Є= $1*10^{-2}$

PROBLEM 1 **Є=** $\mathbf{1*10^{-2}}$

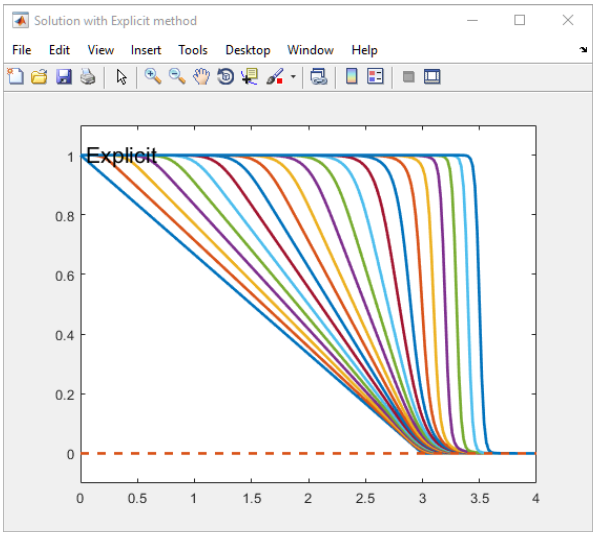


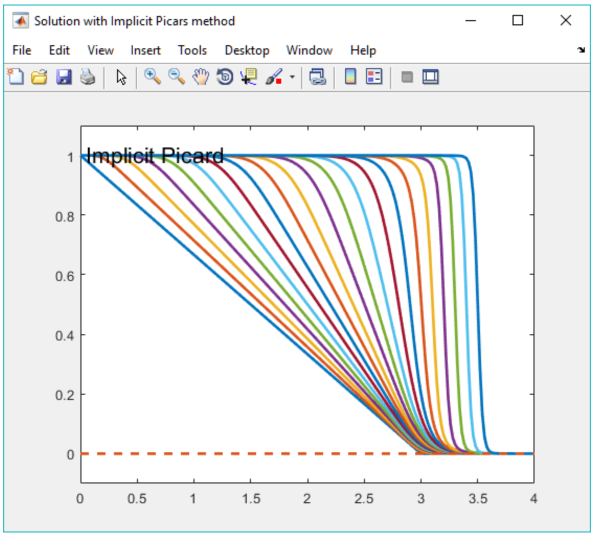


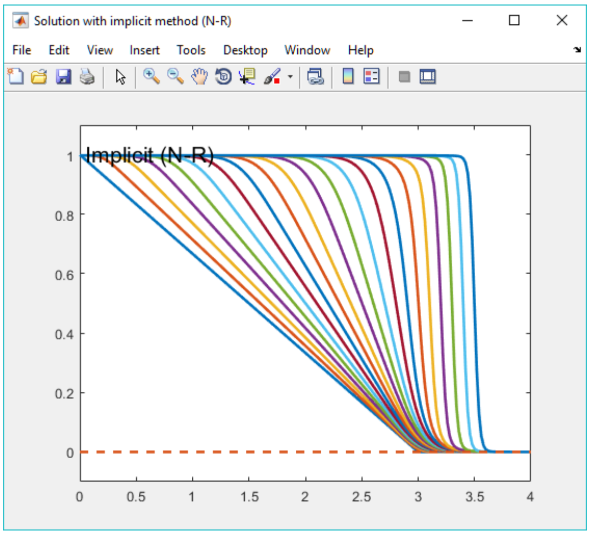


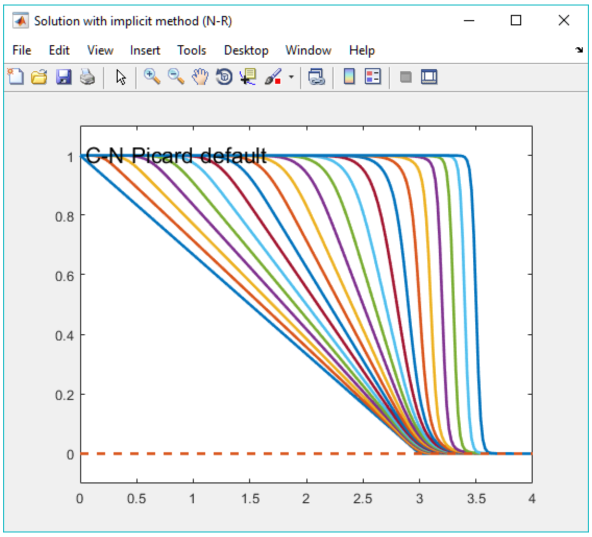


PROBLEM 2 **Є= 1*10^{-2}**



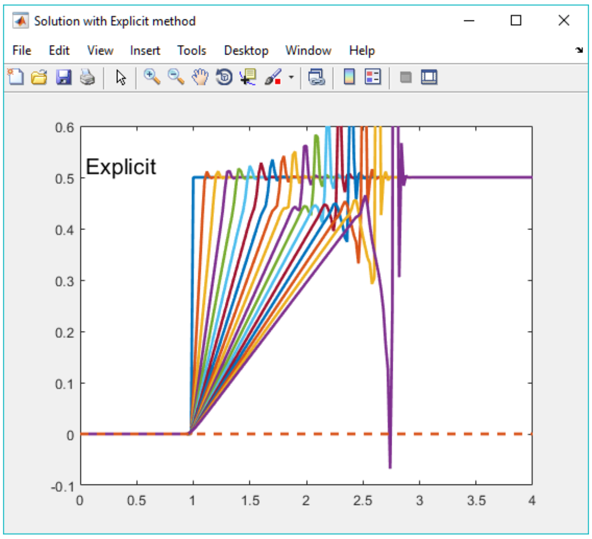


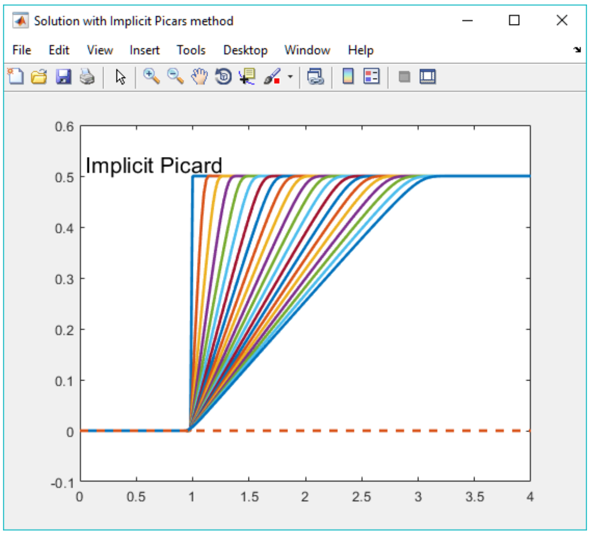


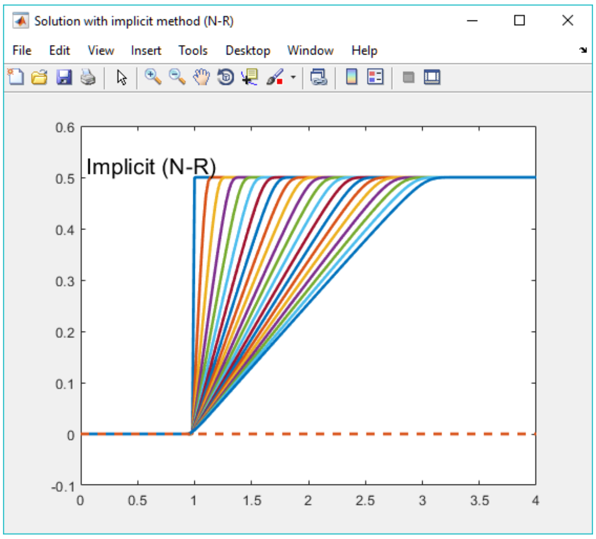


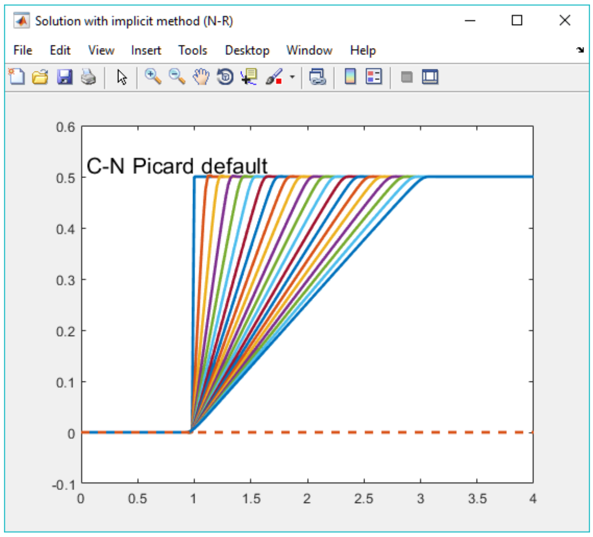
Є= 1*10^{-4}

<u>PROBLEM 1</u> **Є= 1*10^{-4}**

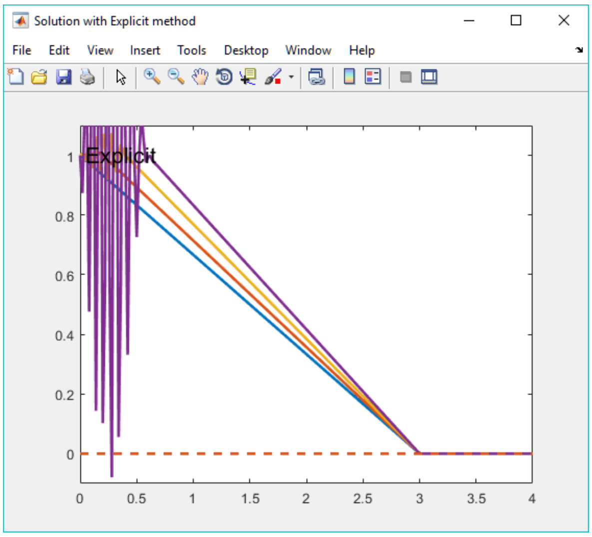


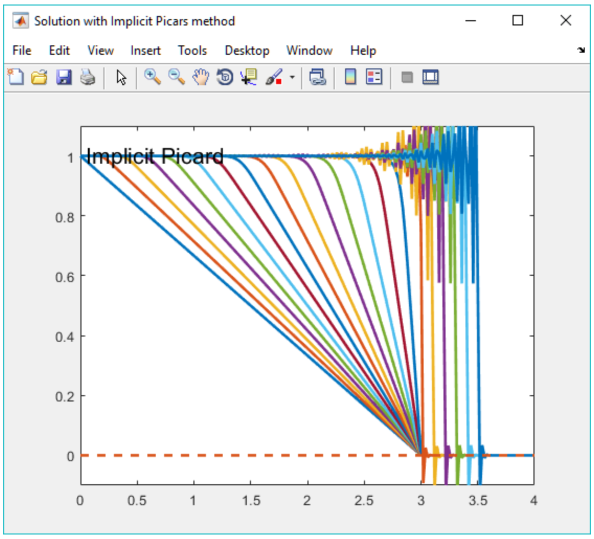


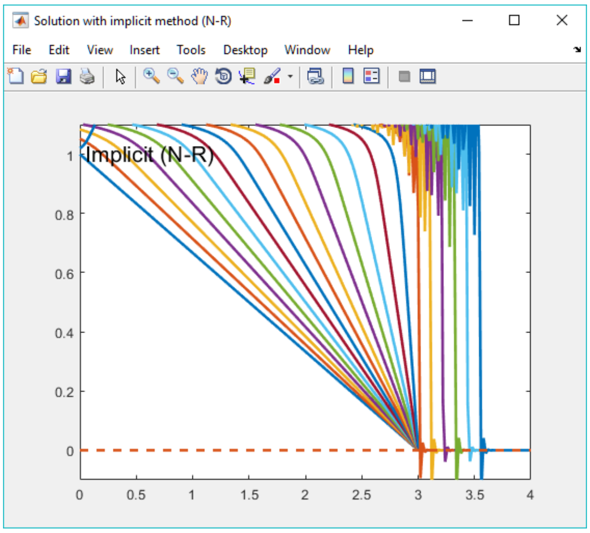


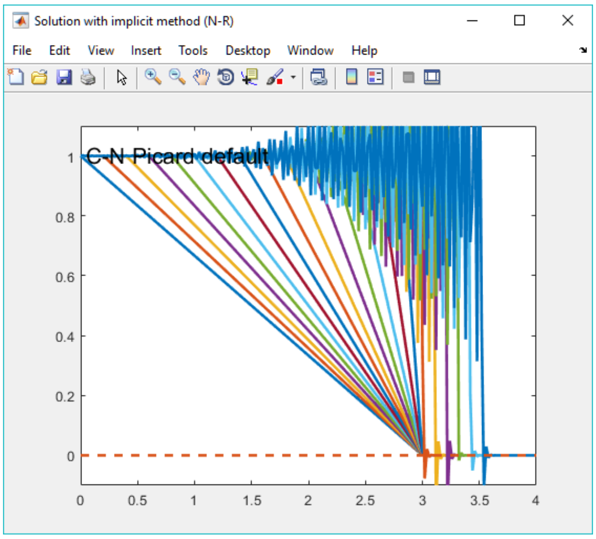


<u>PROBLEM 2</u> **Є= 1*10^{-4}**









Conclusions

For numerical diffusion 10^{-2},all methods are stable and consistent between them for both problems.

Yet, for numerical diffusion 10^{-4}, *for the first problem*, instabilities appear and increase with time for the **explicit method**, while for the second problem all of the methods are unstable, being the explicit is not a reliable method and the explicit N-R the "smoothest" one.

The first problem has INCREASING initial data, while the second has DECREASING initial data. The latter causes signals to pile up, typical of a compression profile and analog to supersonic compression ramp.