1D Unsteady Transport

SAMADRITA KARMAKAR

Ques. Pure Transport Equation.

$$u_t + (a.\nabla)u = s \ \in \Omega(0,T)$$

$$u(x,0) = u_0(x) \ \ on\, \Omega\, at \ \ t=0,$$

$$u = u_D \ \ on \ \ \Gamma_D^{\text{in}} \times (0,T)$$

$$-au.n = h \ \ on\, \Gamma_N^{\text{in}} \times (0,T)$$

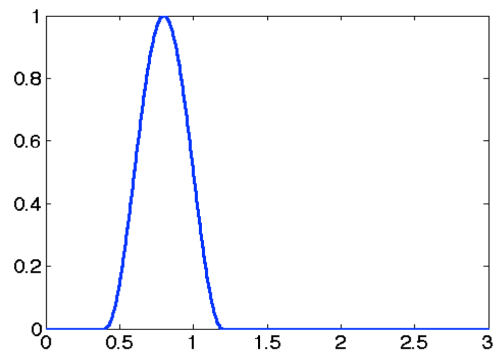
$$\Gamma^{\text{in}} = \{x \in \Omega \,|\, a.n < 0\}$$

In the example the following conditions have been considered.

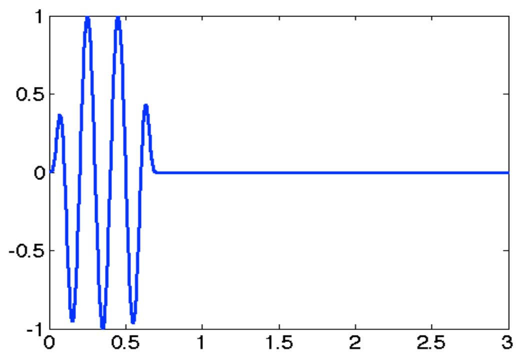
1. Zero source term,
2. Dirichlet boundary conditions on the inflow boundary.

Solution: Unsteady Convention-Diffusion Problem.

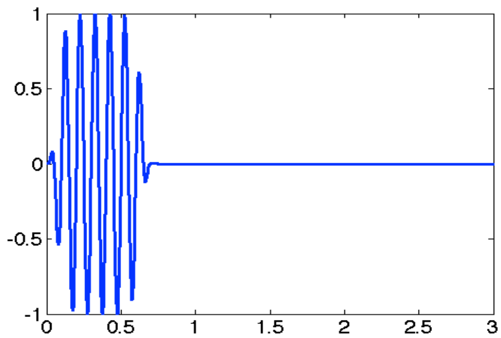
Define Problem: This graphs are the problem.



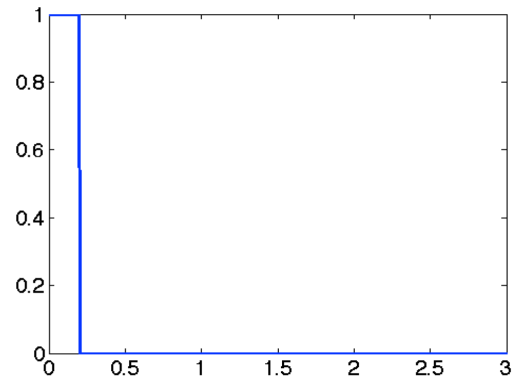
Problem 1



Problem 2



Problem 3



Problem 4

Introduction: The objective is to solve by Lax-Wendroff + Galerkin, Crank-Nicolson + Galerkin, Third order Taylor-Galerkin + Galerkin, Leap Frog + Galerkin, Two Step Third Order Taylor-Galerkin with alpha as 1/9

Code: The code has been changed in the following positions

Implementation of Taylor Galerkin

In File “System.m”

```
case 5 % Third order Taylor-Galerkin + Galerkin
    A = M + dt^3/6*a^2*K;
    B = -dt*a*C-dt^2/2*a*K;
    methodName = 'TG3';
```

Implementation of Leap Frog

In File “System.m”

```
case 6 % Leap Frog + Galerkin
    A = M;
    B = -2*a*dt*C;
    methodName = 'LF';
```

In File “main.m”

For the first iteration of Leap Frog, u_1 has been calculated by TG3 method.

```
for n = 1:nStep
if (method ==1 || method ==2 || method ==3 || method ==4 || method ==5)
    Du = A\(B*u(ind_unk,n) + f);
    u(ind_unk,n+1) = u(ind_unk,n) + Du;
    elseif (method ==6)
    if(n==1)
        Du = A\(B*u(ind_unk,n) + f);
        u(ind_unk,n+1) = u(ind_unk,n) + Du;
        clear A,B;
        [A,B,methodName] = System(method,M,K,C,a,dt);
        A = A(ind_unk,ind_unk);
        B = B(ind_unk,ind_unk);
    else
        u(ind_unk,n+1) = A\(A*u(ind_unk,n-1)+B*u(ind_unk,n) + f);
    end
```

Implementation of Two Step Third Order Taylor-Galerkin with alpha as 1/9

It has been implemented in two parts. One in File “System.h” and in File “main.h”

File “System.h”

```
case 7 % 2 step TG3-I
    A = M;
    B = -(1/3)*dt*a*C-(1/9)*dt^2*a^2*K;
    methodName = 'TG3';
case 8 % 2 step TG3-II
    A = M;
    B = -a*dt*C;
    methodName = 'TG3';
```

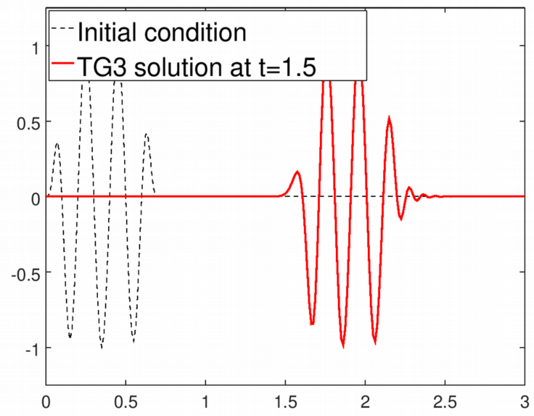
File “main.h”

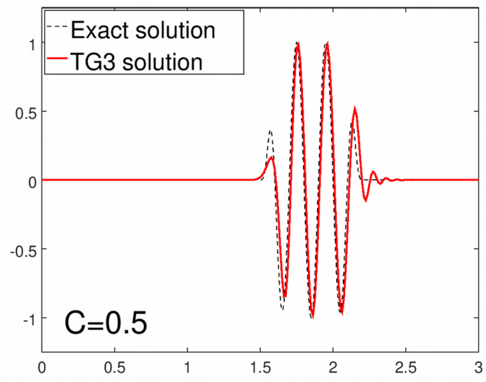
```
    elseif (method==7)

        [A,B,methodName]= System(7,M,K,C,a,dt);
        A = A(ind_unk,ind_unk);
        B = B(ind_unk,ind_unk);
        Du = A\(B*u(ind_unk,n) + f);
          u_bar= u(ind_unk,n) + Du;
        clear A,B;
        [A,B,methodName]= System(8,M,K,C,a,dt);
        A = A(ind_unk,ind_unk);
        B = B(ind_unk,ind_unk);
        K1 = K(ind_unk,ind_unk);
        Du = A\(B*u(ind_unk,n)- .5*a^2*dt^2*K1*u_bar + f);
        u(ind_unk,n+1) = u(ind_unk,n) + Du;
end
```

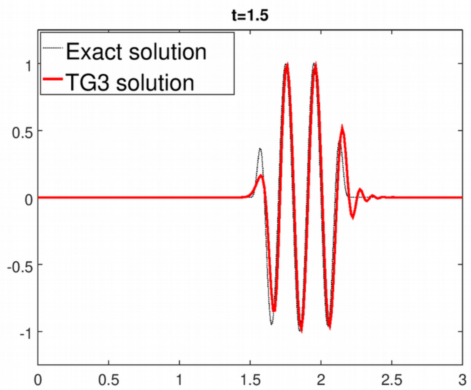
Graphs:

Graphs are generated for the above mentioned problem for all different formulation.

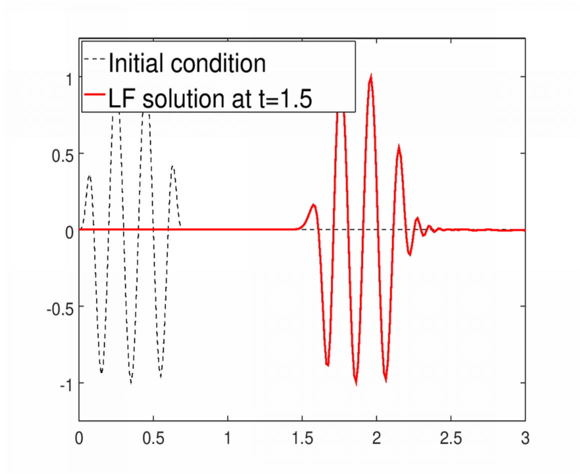
Taylor Galerkin

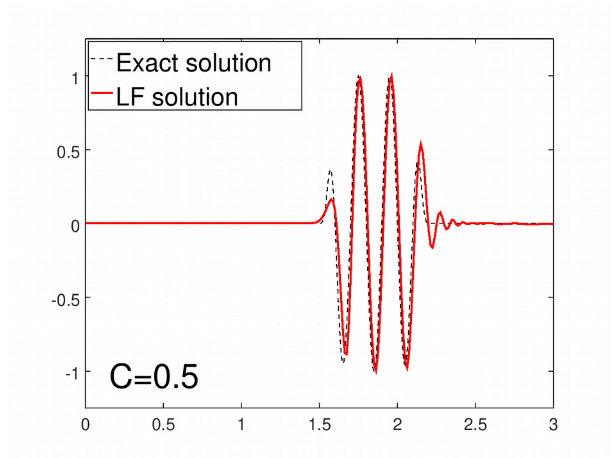


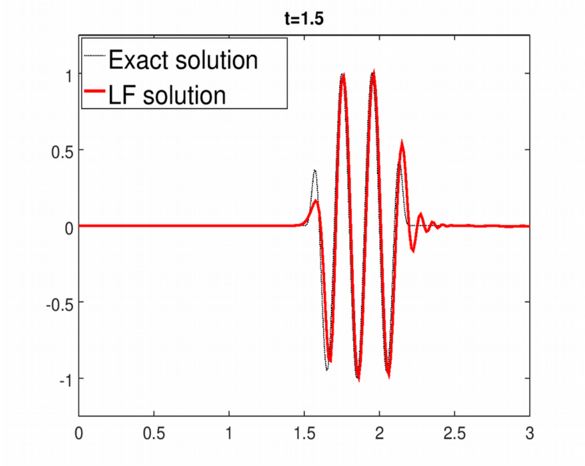




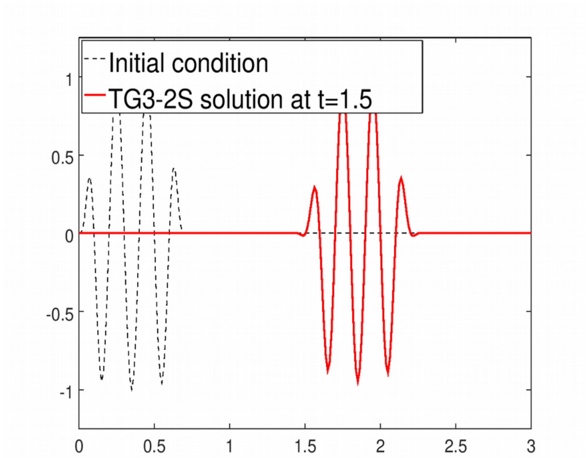
Leap Frog Method

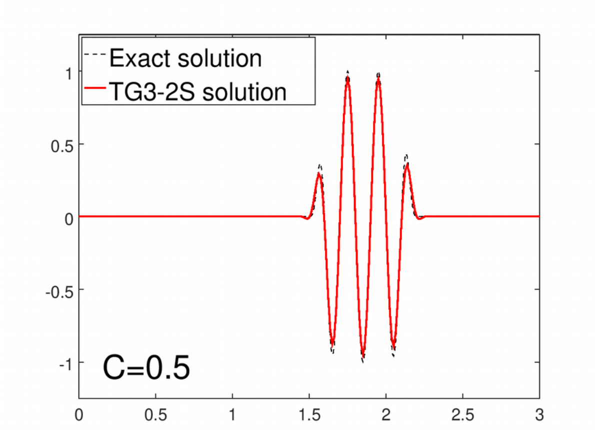


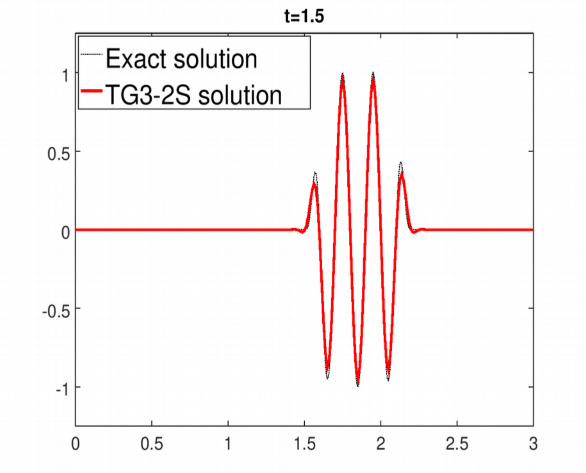




Two Step Third Order Taylor-Galerkin with alpha as 1/9

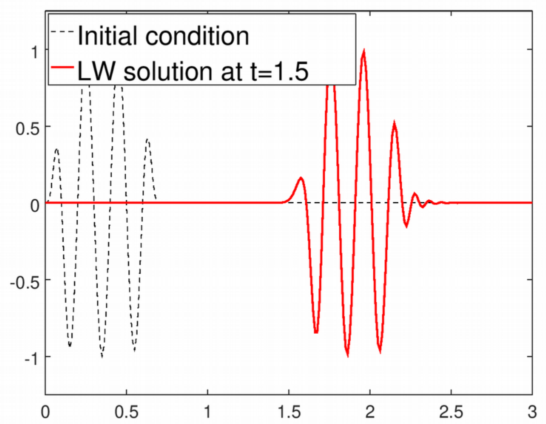


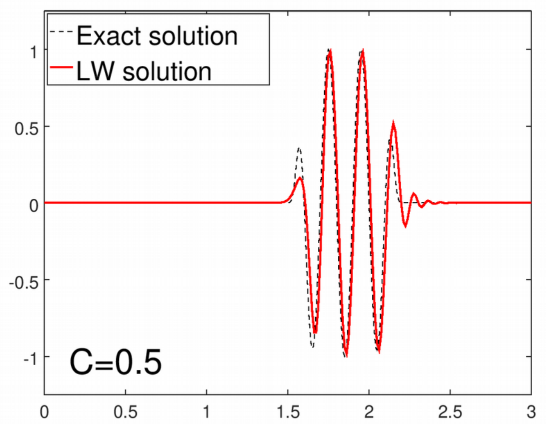


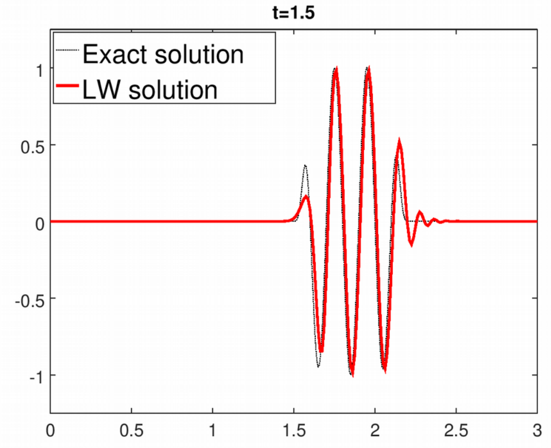


Pre-Solved Implementations

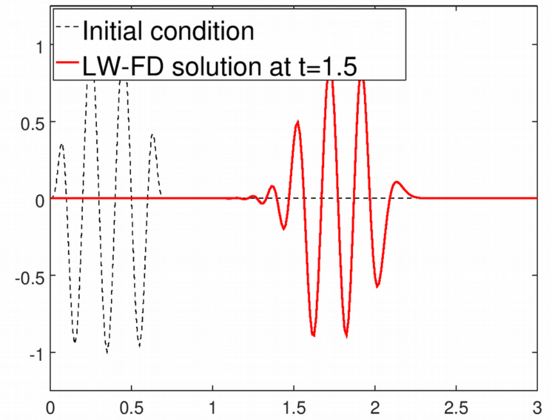
Lax-Wendroff

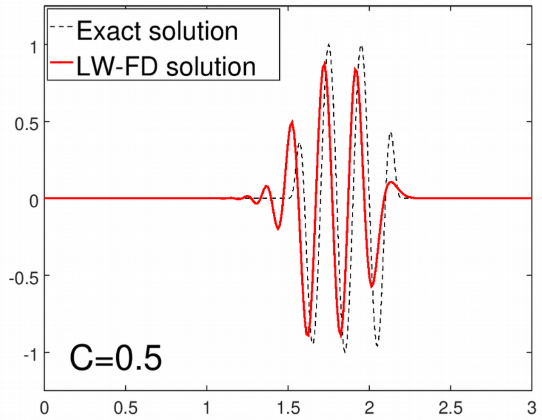


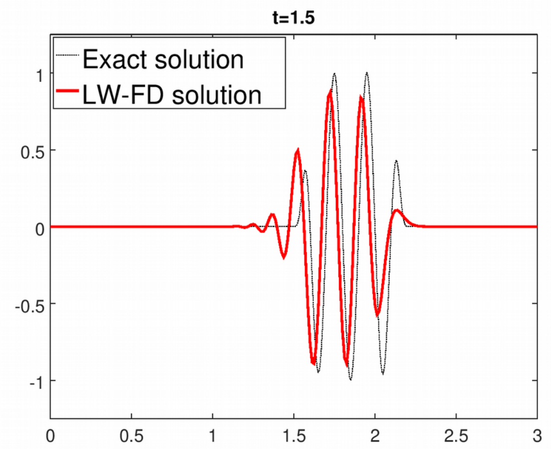




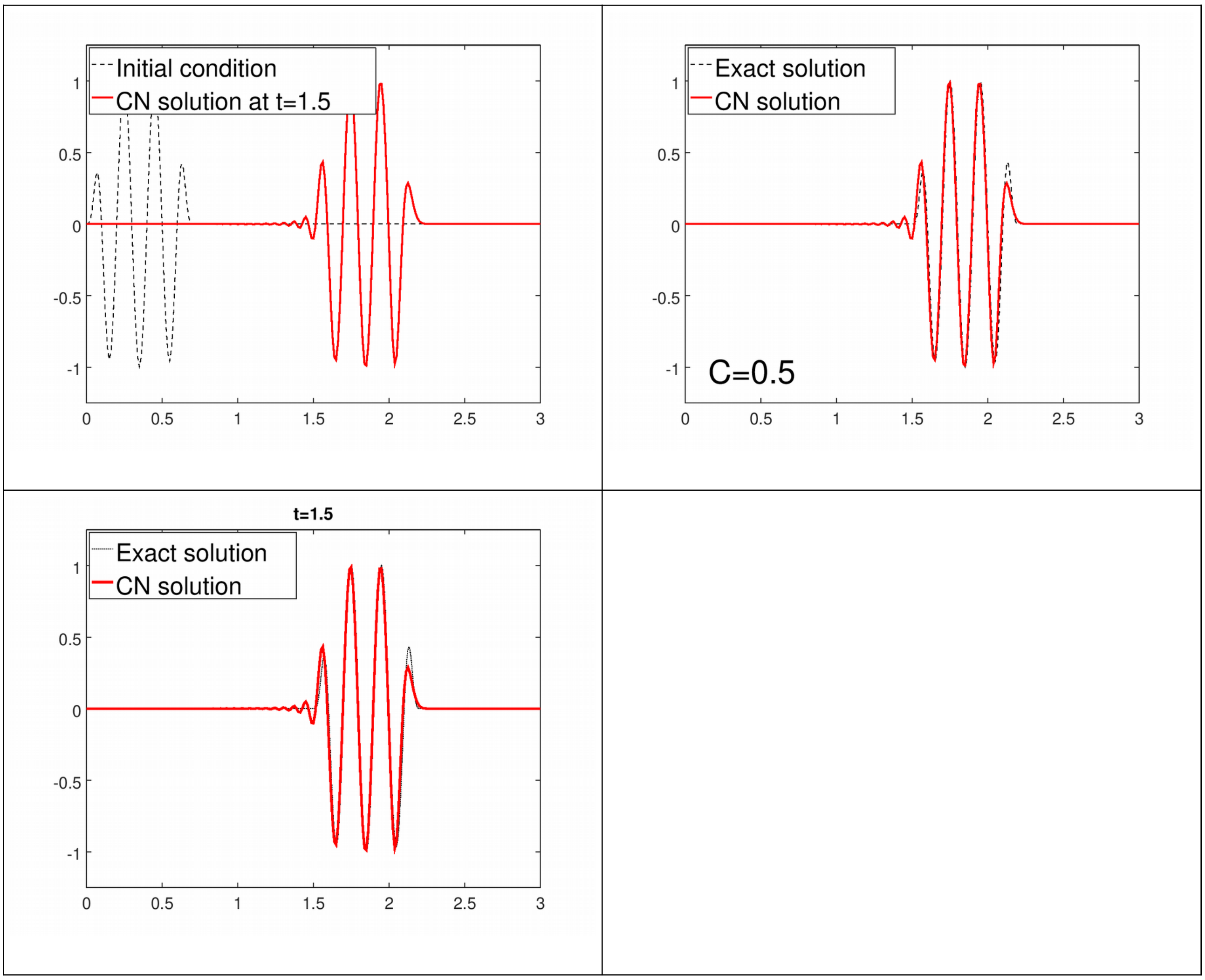
Lax-Wendroff with lumped mass matrix



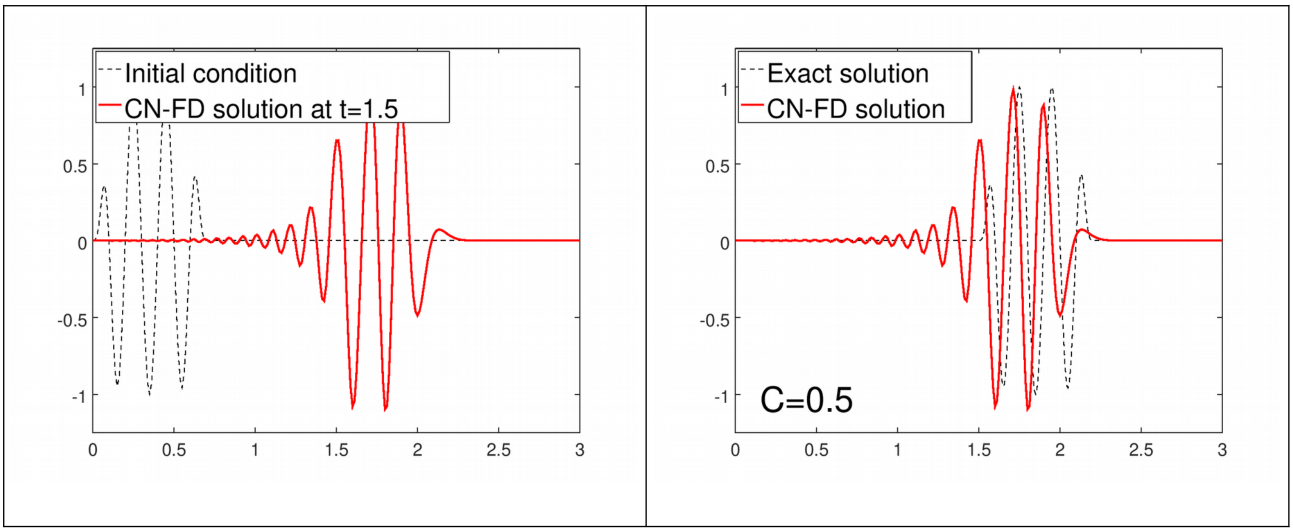


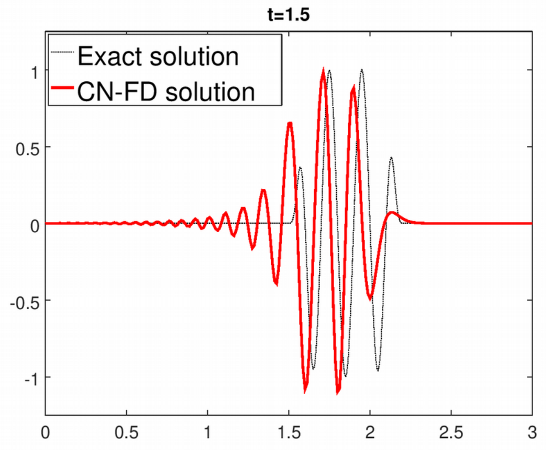


Crank-Nicolson



Crank-Nicolson with lumped mass matrix





Conclusion:

As expected, Leap Frog had the most deviation, Third order Taylor Galerkin had some deviation, and Third order Taylor Galerkin two step method had the least deviation.