POLYTECHNIC UNIVERSITY OF CATALONIA
MASTER OF SCIENCE IN COMPUTATIONAL MECHANICS

Finite Element in Fluids Assignment 2

Chiluba Isaiah Nsofu

Lecture : Prof. Pablo Sáez

Date: 2nd April, 2018

Contents

1 1D convection Equation **2**

1.1 Lax-Wendroff Method . . . 2

1.2 Lax-Wendroff with lumped mass matrix + Galerkin . . . 2

1.3 Crank Nicolson . . . 3

1.4 Crank-Nicolson with lumped mass matrix + Galerkin . . . 4

1.5 Leap frog . . . 4

1.6 One step and two step Third-order Taylor-Galerkin (TG3) . . . 4

1.7 One step TG3 Solution . . . 5

1.8 Two-step third-order Taylor-Galerkin solution . . . 6

1 1D convection Equation

1.1 Lax-Wendroff Method

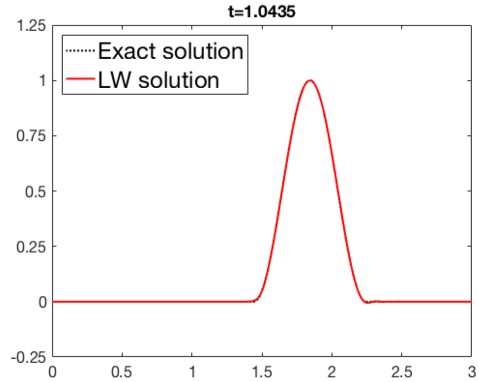

Figure 1.1: LW for problem 1, CFL=0.5

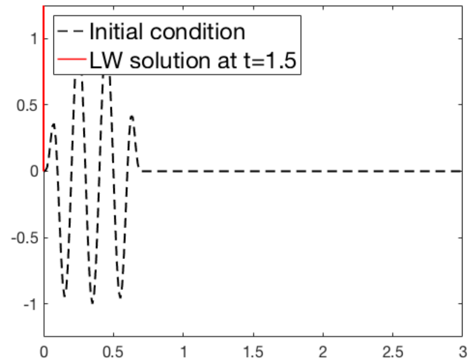

Figure 1.2: LW FD for problem 3, CFL=1

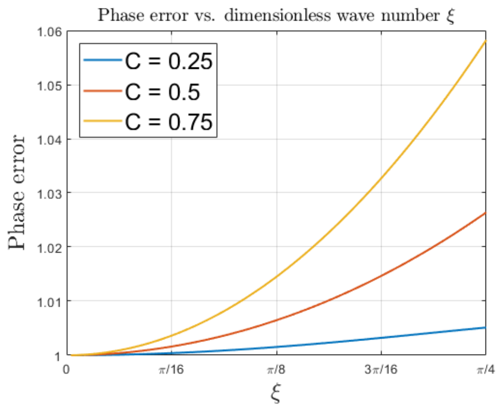

Figure 1.3: Phase error plot

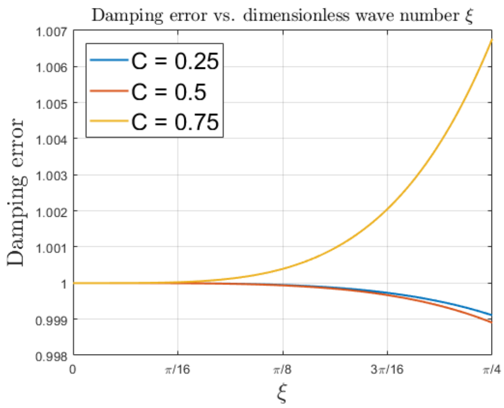

Figure 1.4: Damping error plot

1.2 Lax-Wendroff with lumped mass matrix + Galerkin

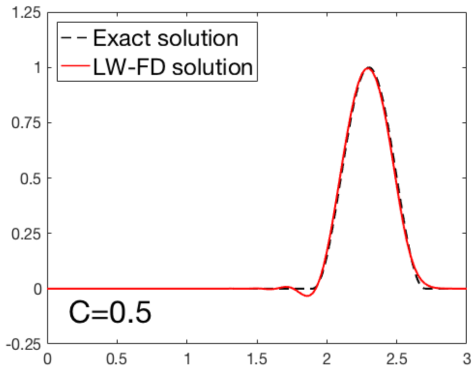

Figure 1.5: LW FD for problem 1, CFL=0.5

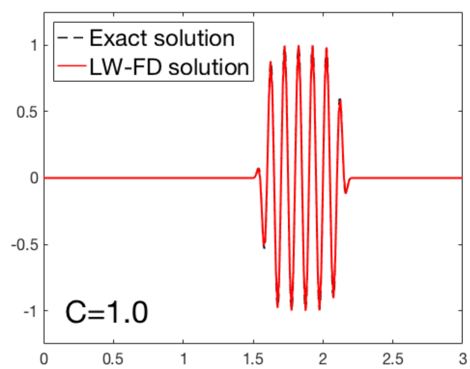

Figure 1.6: LW FD for problem 3, CFL=1

Unlike the general Lax Wendroff method this method is stable for CFL ≤ 1 as it can be seen in figure 1.5 and 1.6 . The method is also accurate this range of CFL. This method is definitely unstable for any values of CFL more that 1.

Figure 1.7: Phase error plot

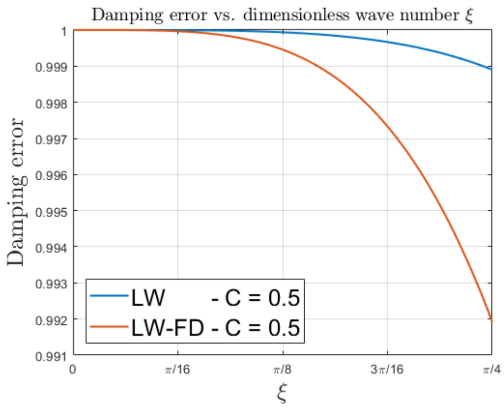

Figure 1.8: Damping error plot

Figures (1.7) and (1.8) shows the phase and dampming error of both the LW and LW-FD methods. These conparisong clearly justifies the differences in the two method as seen in the results which means that the the consistent formulation yields better results

1.3 Crank Nicolson

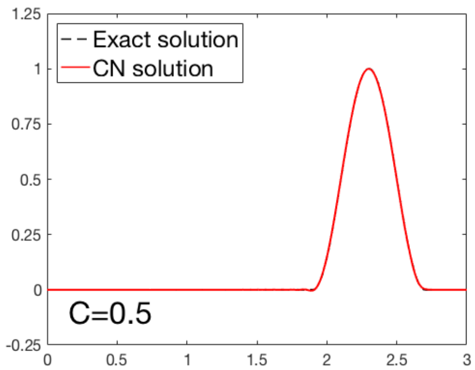

Figure 1.9: Crank Nicolson for problem 1 with CFL=0.5

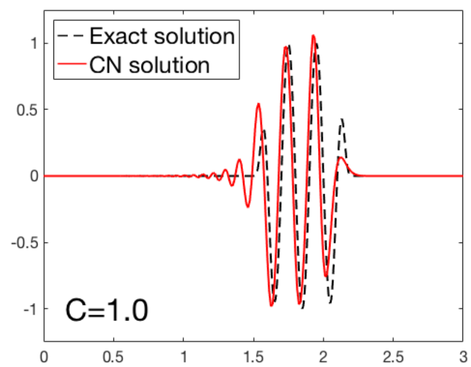

Figure 1.10: C-N for problem 1 with CFL=1

As it can be see in both figure 1.9 and 1.10 Crank Nicolson method is both stable and accurate for the convection equation for both CFL =0.5 and CFL = 1. This shows the method is unconditionally stable for this equation. The plots to support the unconditionally stability option for this method are shown in the figures (1.11) and (1.12) on the next page.

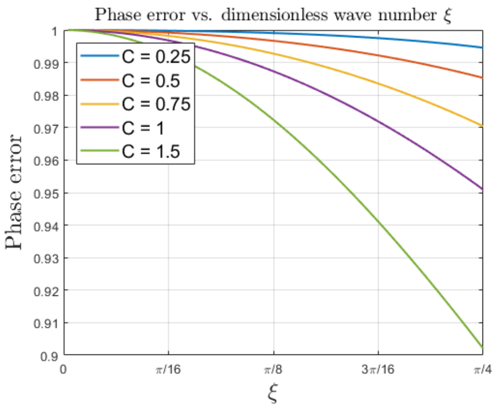

Figure 1.11: Phase error plot

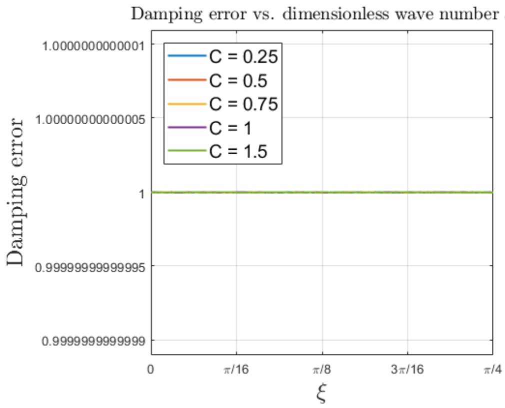

Figure 1.12: Damping error plot

1.4 Crank-Nicolson with lumped mass matrix + Galerkin

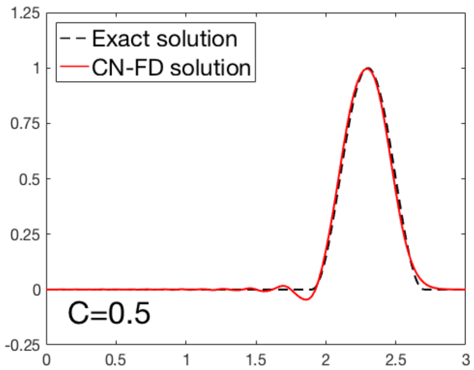

Figure 1.13: CN-FD for problem 1, CFL=0.5

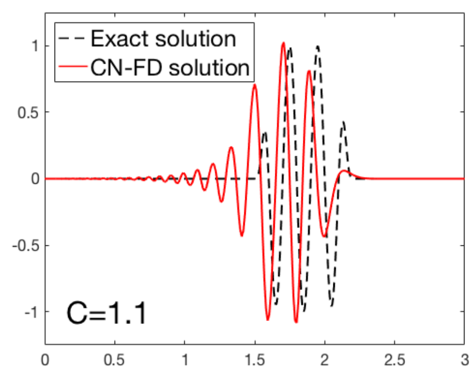

Figure 1.14: CN-FD for problem 2, CFL=1.1

This method is unconditionally stable for the convection equation. Although as it can be seen in figure 1.14 the accuracy of the method reduces with as CFL begins to increase above 1.

1.5 Leap frog

The formulation is as follows

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} = u_t^n = s^n - \mathbf{a} \cdot \nabla u^n \tag{1}$$

Since the source is zero $s^n = 0$ therefore introducing the test function w we get;

$$(w, u^{n+1}) = (w, u^{n-1}) - 2\Delta t(w, \mathbf{a} \cdot \nabla u^n) \tag{2}$$

This is implemented in the code as

$$A = M$$
$$B = -2 * dt * a * C;$$

Since for the first time step we need to calculate the previous values we have employed Lax wendroff method as it can be shown in the figure (1.15). The leap frog method is conditionally stable when it come to the convection diffusion equation. This conclusion is based on the results shown in figures (1.16) and (1.17).

1.6 One step and two step Third-order Taylor-Galerkin (TG3)

Implementation of both TG3 one step and two step method.

```
if method ==5
for n = 1
    [Alax,Blax,methodNamelax] = System(1,M,K,C,a,dt);
    % Reduced system to impose du(0) = 0:
    ind_unk = 2:nPt;
    Alax = Alax(ind_unk,ind_unk);
    if problem == 4
        flax = [Blax(2,1); zeros(nPt-2,1)];
    else
        flax = zeros(nPt-1,1);
    end
    Blax = Blax(ind_unk,ind_unk);

    Du = Alax\(Blax*u(ind_unk,n) + flax);
    u(ind_unk,n+1) = u(ind_unk,n) + Du;
end
for n = 2:nStep
    u(ind_unk,n+1) = u(ind_unk,n-1) + A\B*u(ind_unk,n);
end
```

Figure 1.15: Implementation of the Leap-Frog method

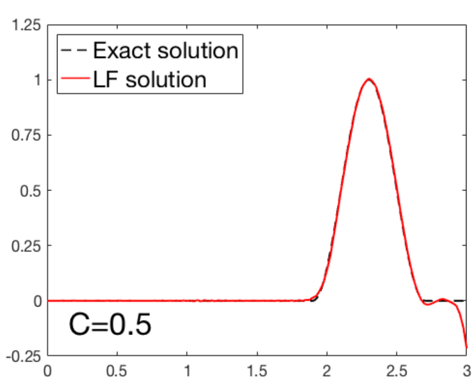

Figure 1.16: LF solution AT CFL =0.5

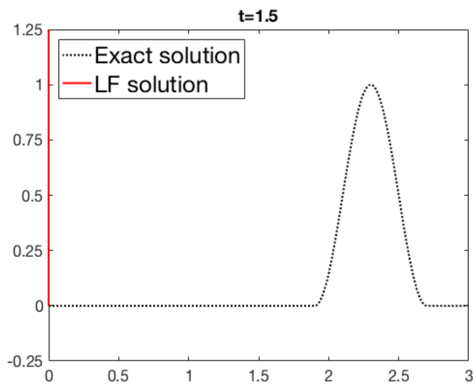

Figure 1.17: LF solution CFL =1

```
case 7 % Third order Taylor-Galerkin + Galerkin
    A = M + 1/6*dt^2*a^2*K;
    B = -a*dt*C - 0.5*a^2*dt^2*K;
    methodName = 'TG3';
case 8 % Two-steps third order Taylor-Galerkin + Galerkin
    A = M\(M - a*dt*C);
    B = -1/2*dt^2*a^2*(M\K);
    methodName = 'TG3-2S';
```

Figure 1.18: Implementation of the TG3 and the 2-step TG3 method in the system function

```
alpha = 1/9;
uTelda = u;
for n = 1:nStep
    % First step
    uTelda(:,n) = (M\(M - 1/3*dt*a*C - alpha*dt^2*a^2*K))*u(:,n);
    % Second Step
    u(ind_unk,n+1) = A*u(ind_unk,n) + B*uTelda(ind_unk,n);
end
```

Figure 1.19: Implementation of the 2 step TG3 in the main function

1.7 One step TG3 Solution

This method is conditionally stable for the convection equation as it seen in both figure 1.20 and figure 1.21. Its counterpart the Two step method is accurate and stable for CFL not more than or equal to 1 as shown in figure (1.23). Nevertheless this method is good for smaller values of CFL as shown in Figure (1.22) and (1.24)

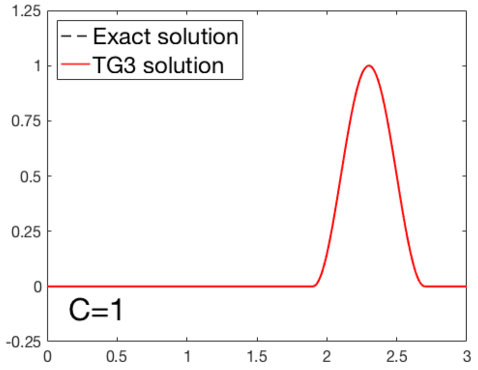

Figure 1.20: TG3 for problem 1 ,CFL = 1

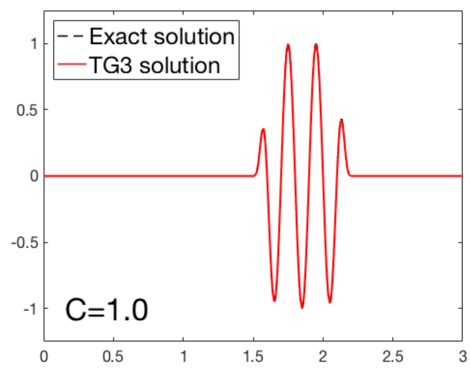

Figure 1.21: TG3 for problem 3 ,CFL = 1

1.8 Two-step third-order Taylor-Galerkin solution

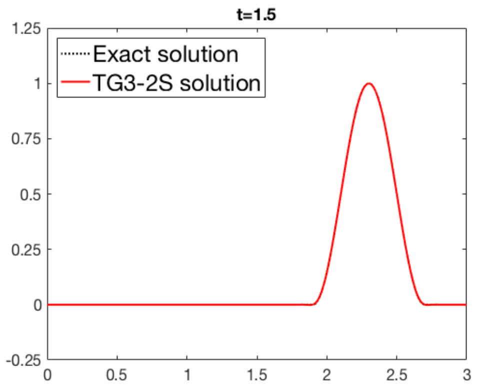

Figure 1.22: TG3 2S for problem 1 ,CFL = 1

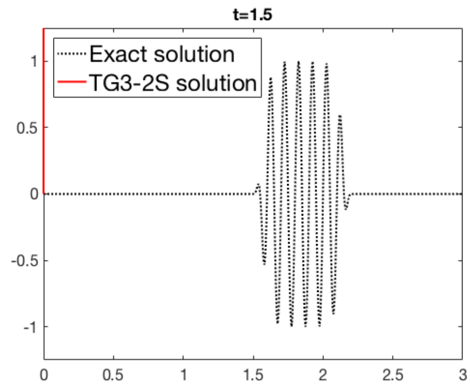

Figure 1.23: TG3 for problem 3 ,CFL = 1

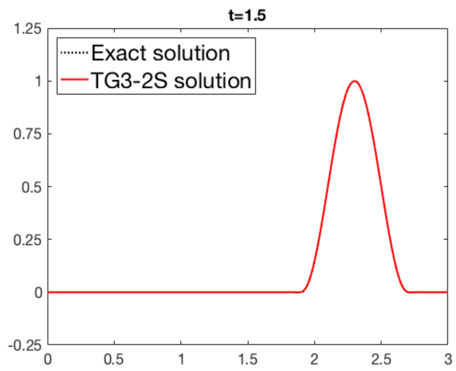

Figure 1.24: ITG3 for problem 3 ,CFL = 0.75