

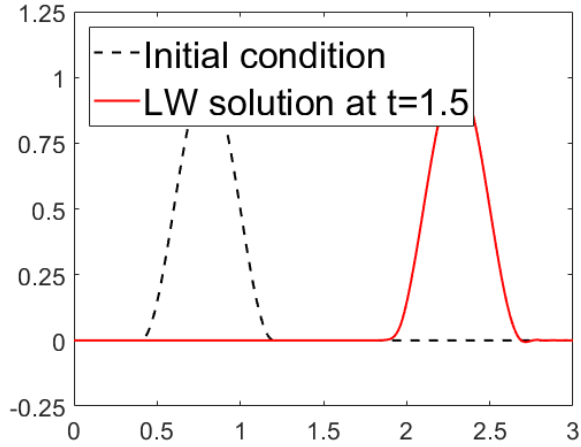
Name	Ahmed Saeed Mohamed Sherif
Course	Finite Elements in Fluids
Master	MSc Computational Mechanics
Report no.	2
Topic	1D Unsteady Transport Equation - Temporal and Spatial Discretization

Contents

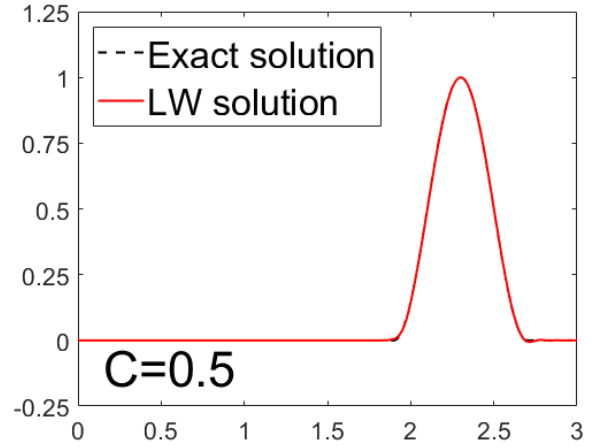
Contents	i
1 Lax-Wendroff with consistent mass matrix (LW)	1
2 Lax-Wendroff with lumped mass matrix (LW-FD)	2
3 Crank-Nicholson with consistent mass matrix (CN)	4
4 Leap-frog with consistent mass matrix (LF)	5
5 Third-order Taylor-Galerkin method (TG3)	6
6 Two-step Third-order Taylor-Galerkin method (TG3-2S)	7
7 Comparison at $C = 0.9$ or 90% of the stability limit	10
A Developed codes	11
A.1 Modifications in <code>System.m</code>	11
A.2 Modifications in <code>main.m</code>	12
A.3 New function <code>ExactG.m</code>	13
A.4 New script <code>plotPhaseDampingError.m</code>	13
References	14

1 Lax-Wendroff with consistent mass matrix (LW)

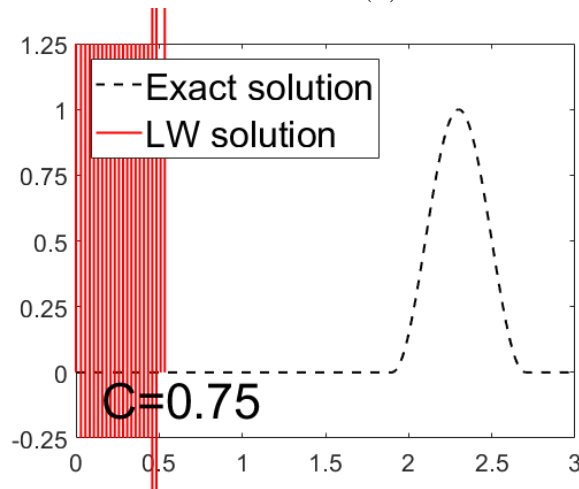
The stability condition for Lax-Wendroff time stepping scheme with consistent mass matrix is $C^2 < \frac{1}{3}$, where C is the Courant–Friedrichs–Lewy (CFL) condition. Figure 1b shows the stable solution obtained when $C = 0.5$, while Figure 1c shows the unstable solution when $C = 0.75$. In all the results presented in this report, Galerkin formulation with linear finite elements is employed for the spatial discretization.



(a) Initial and final solution - $C = 0.5$



(b) Exact and numerical solution - $C = 0.5$



(c) Exact and numerical solution - $C = 0.75$

Figure 1: Results of using Lax-Wendroff time stepping scheme with Galerkin spatial discretization at different values of C .

It is known that the accuracy of the solution depends on the dimensionless wave number ξ [1]. Figure 2 shows the relative phase error and the damping error as a function of the dimensionless wave number ξ . It is clearly seen in Figure 2a that at lower values of ξ , i.e. longer wave length and lower wave frequency, the accuracy is higher. To understand this statement more clearly, two convection problems of equal convection speed a and different wave lengths λ are solved using the same mesh size h and time step Δt , i.e. same C . The results are shown in Figure 3, where a higher phase error is observed in case of higher value of ξ .

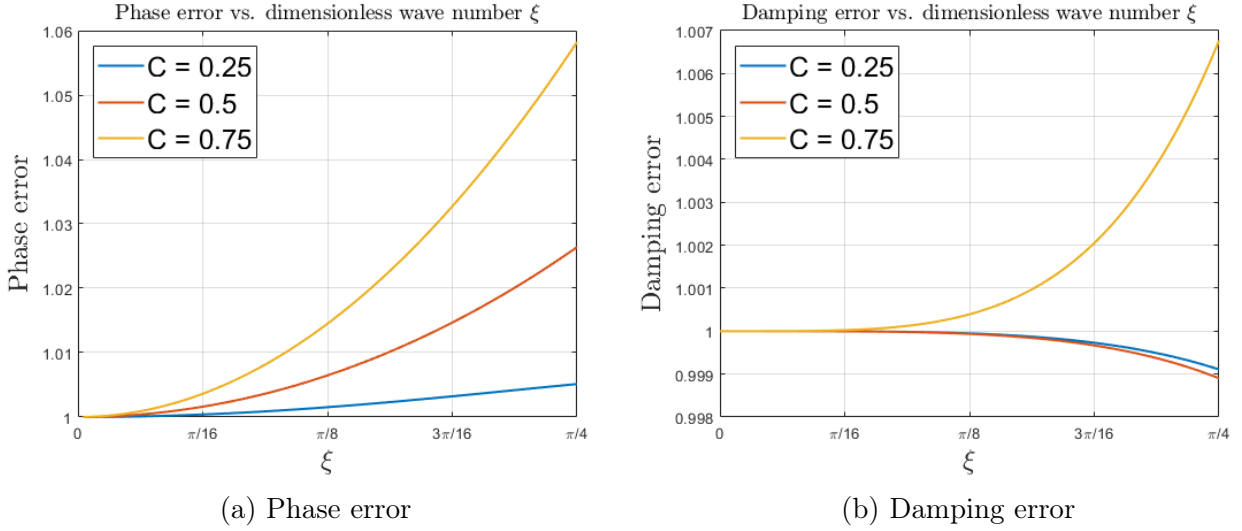


Figure 2: Relative phase error (left) and damping error (right) at different values of C for LW time stepping scheme with linear Galerkin spatial discretization.

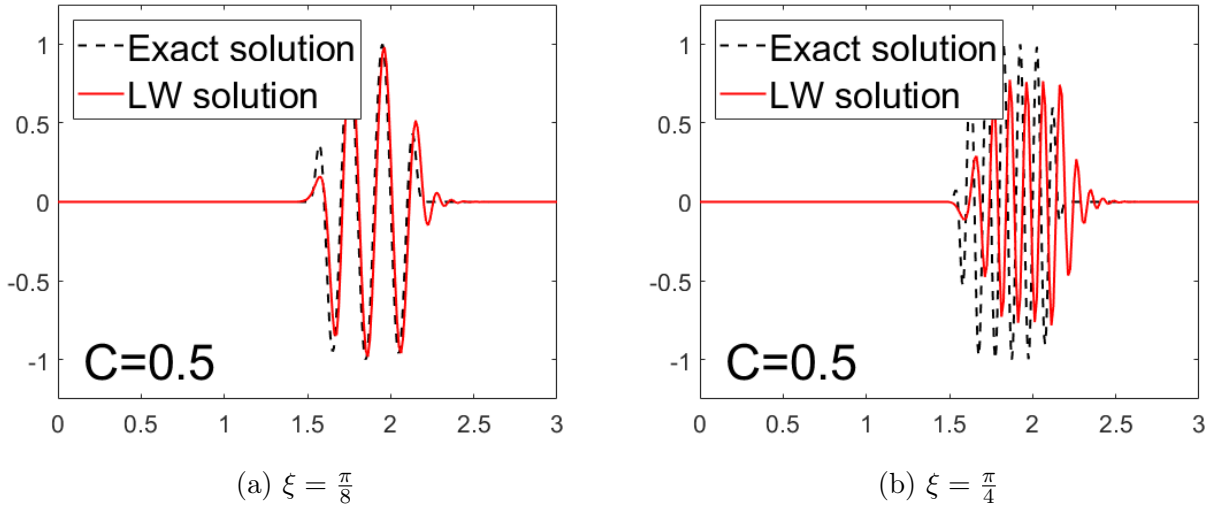
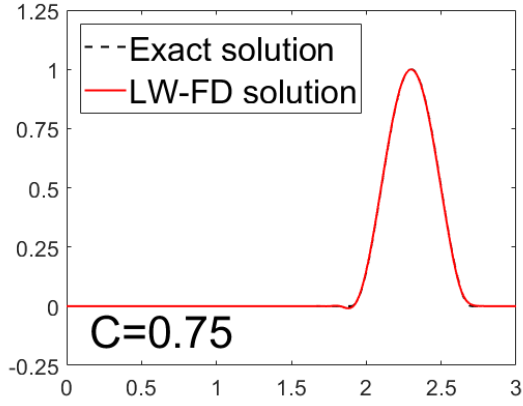


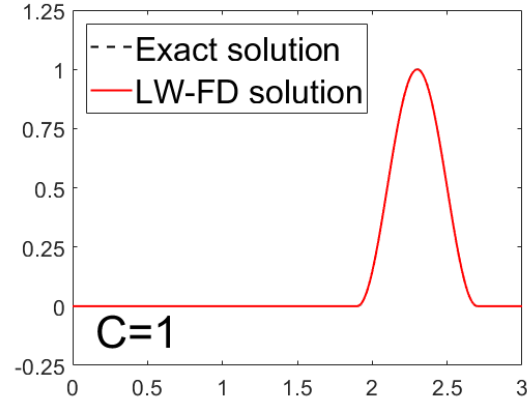
Figure 3: Solution using LW time stepping scheme with linear Galerkin spatial discretization for two convection problems with different dimensionless wave number, $\xi = \frac{\pi}{8}$ (left) and $\xi = \frac{\pi}{4}$ (right).

2 Lax-Wendroff with lumped mass matrix (LW-FD)

Using a lumped mass matrix in Lax-Wendroff time stepping scheme yields a higher stability limit, thus the stability condition becomes $C^2 < 1$. The solution of the problem shown earlier in 1 is obtained at $C = 0.75$ and $C = 1$ as shown in Figure 4. It is clearly seen that a stable solution is obtained for $C = 0.75$ unlike the case of Lax-Wendroff with consistent mass matrix. However, the cost of this is the reduced accuracy in case of using lumped mass matrix. This is clearly seen in Figure 5, where for a stable case of $C = 0.5$ and a specific dimensionless wave number ξ , Lax-Wendroff method with consistent mass matrix (LW) yields less phase and damping errors when compared to the case of lumped mass matrix (LW-FD). Furthermore, the difference in accuracy between LW and LW-FD is obviously observed in Figure 6.

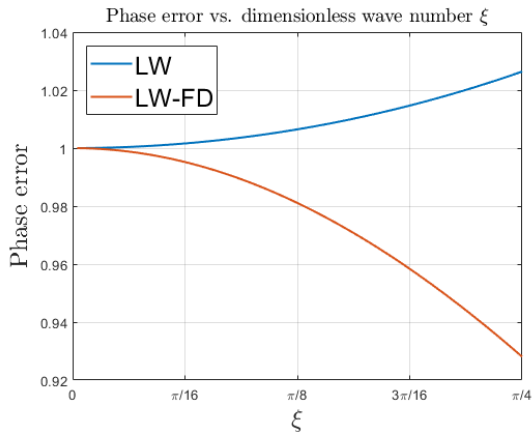


(a) $C = 0.75$

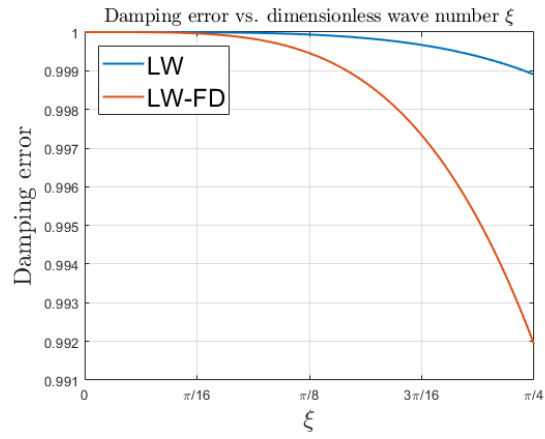


(b) $C = 1$

Figure 4: Solution using LW-FD time stepping scheme with linear Galerkin spatial discretization at different values of C .

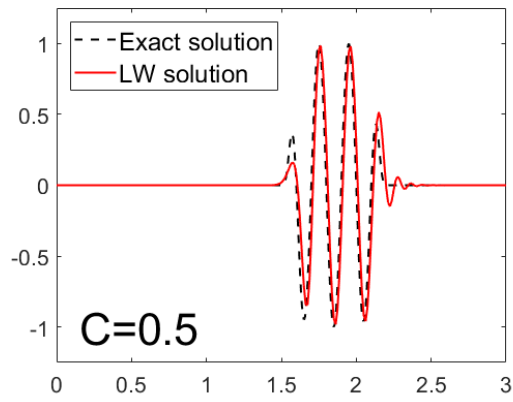


(a) Phase error

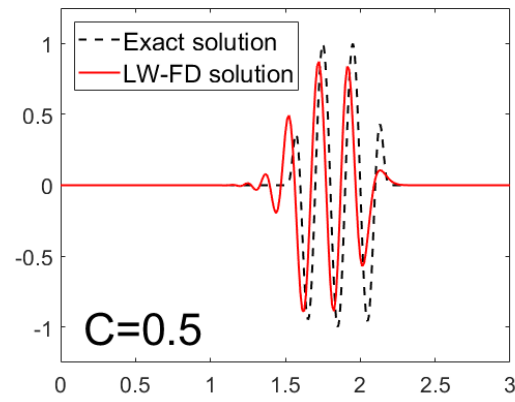


(b) Damping error

Figure 5: Relative phase error (left) and damping error (right) at $C = 0.5$ for Lax-Wendroff time stepping scheme with consistent (blue) and lumped (red) mass matrices and linear Galerkin spatial discretization.



(a) LW: Consistent mass matrix



(b) LW-FD: Lumped mass matrix

Figure 6: Solution using LW (left) and LW-FD (right) time stepping scheme with linear Galerkin spatial discretization at $C = 0.5$.

3 Crank-Nicholson with consistent mass matrix (CN)

Crank-Nicholson time integration scheme is unconditionally stable as seen in Figure 7b. However, increasing the CFL condition leads to higher phase errors as seen in Figure 7a. This is further understood by solving the same convection problem at two different values of C as seen in Figure 8.

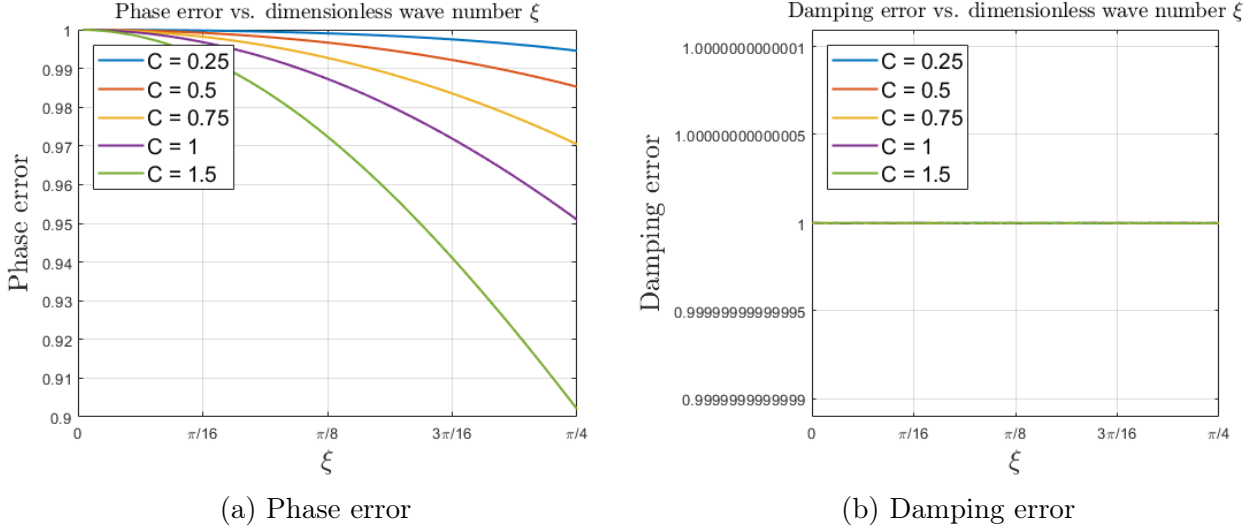


Figure 7: Relative phase error (left) and damping error (right) at different values of C for CN time stepping scheme with linear Galerkin spatial discretization.

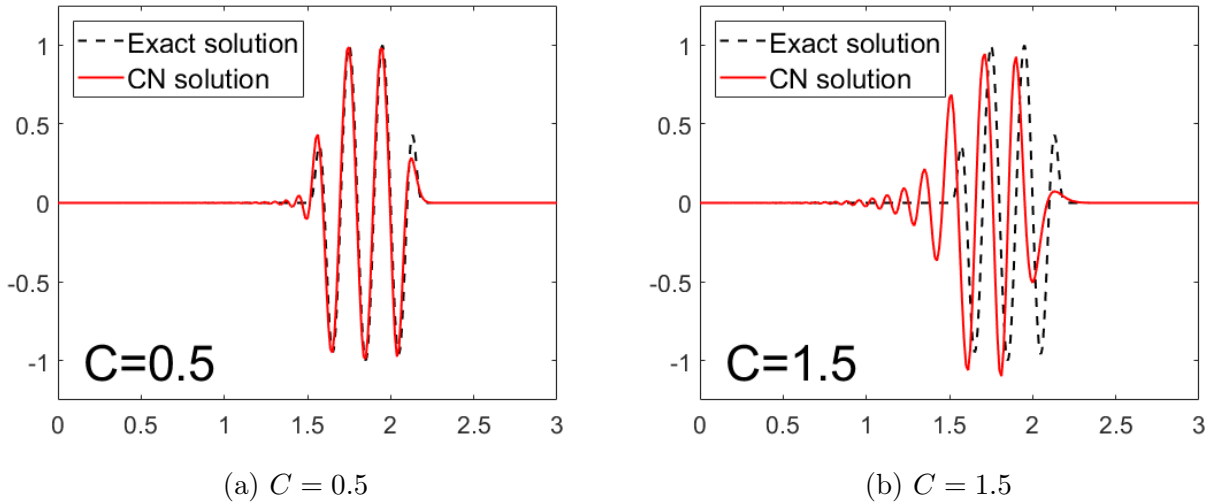


Figure 8: Solution using Crank-Nicholson time stepping scheme with linear Galerkin spatial discretization at different values of C .

4 Leap-frog with consistent mass matrix (LF)

The time discretization using leap-frog method is given by:

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

Multiplying by a test function w and integrating over the spatial domain Ω yields the Galerkin weak form of the problem as:

$$\left(w, \frac{u^{n+1}}{2\Delta t}\right)_\Omega = \left(w, \frac{u^{n-1}}{2\Delta t}\right)_\Omega + \left(w, s^n\right)_\Omega - \left(w, \mathbf{a} \cdot \nabla u^n\right)_\Omega \quad (2)$$

In our problem of interest, the source term is zero and only Dirichlet boundary condition is imposed at the inlet boundary. Furthermore, the boundary conditions at the outflow boundary are not considered, because the wave front doesn't reach it. This yields the following discretized form:

$$\mathbf{M}\mathbf{U}^{n+1} = \mathbf{M}\mathbf{U}^{n-1} - 2a\Delta t\mathbf{C}\mathbf{U}^n \quad (3)$$

where \mathbf{M} and \mathbf{C} are the mass and convection matrices, respectively. The leap-frog method requires the data from the previous two time steps, therefore in the first time step of the process another method is used, for instance Lax-Wendroff, and this is how it was implemented in this assignment.

The code for this method is developed as a part of this assignment. The stability limit for the leap-frog method with consistent mass matrix is $C^2 < \frac{1}{3}$. Figure 9a shows the stable solution obtained when $C = 0.5$, while Figure 9b shows the unstable solution when $C = 0.75$.

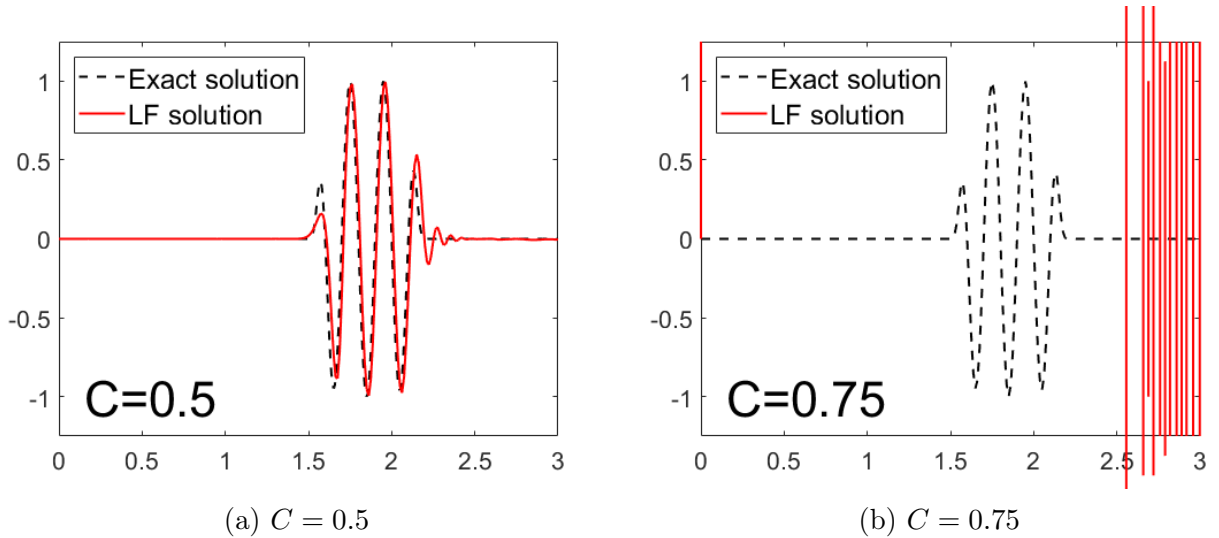


Figure 9: Solution using leap-frog time stepping scheme with linear Galerkin spatial discretization at different values of C .

5 Third-order Taylor-Galerkin method (TG3)

Neglecting the source term, the time discretization using third-order Taylor-Galerkin method is given by:

$$\left[1 - \frac{\Delta t^2}{6}(\mathbf{a} \cdot \nabla)^2\right] \frac{u^{n+1} - u^n}{\Delta t} = -(\mathbf{a} \cdot \nabla)u^n + \frac{\Delta t}{2}(\mathbf{a} \cdot \nabla)^2 u^n \quad (4)$$

and the Galerkin weak form is given by:

$$\left(w, \frac{\Delta u}{\Delta t}\right)_\Omega - \frac{\Delta t^2}{6} \left(w, (\mathbf{a} \cdot \nabla)^2 \frac{\Delta u}{\Delta t}\right)_\Omega = -\left(w, \mathbf{a} \cdot \nabla u^n\right)_\Omega + \frac{\Delta t}{2} \left(w, (\mathbf{a} \cdot \nabla)^2 u^n\right)_\Omega \quad (5)$$

Integration by parts yields:

$$\left(w, \frac{\Delta u}{\Delta t}\right)_\Omega + \frac{\Delta t^2}{6} \left(\mathbf{a} \cdot \nabla w, \mathbf{a} \cdot \nabla \frac{\Delta u}{\Delta t}\right)_\Omega = -\left(w, \mathbf{a} \cdot \nabla u^n\right)_\Omega - \frac{\Delta t}{2} \left(\mathbf{a} \cdot \nabla w, \mathbf{a} \cdot \nabla u^n\right)_\Omega \quad (6)$$

where in this process the boundary terms are neglected, because of having only Dirichlet conditions at the inlet and the outlet boundary is not considered as the wave front doesn't reach it.

Next, the discrete form is given as:

$$\left(\mathbf{M} + \frac{a^2 \Delta t^2}{6} \mathbf{K}\right) \Delta \mathbf{U} = \left(-a \Delta t \mathbf{C} - \frac{a^2 \Delta t^2}{2} \mathbf{K}\right) \mathbf{U}^n \quad (7)$$

where \mathbf{M} , \mathbf{C} and \mathbf{K} are the mass, convection and stiffness matrices, respectively.

The code for this method is developed as a part of this assignment. The stability limit for third-order Taylor-Galerkin method is $C^2 < 1$. Figure 10a shows the stable solution obtained when $C = 0.75$, while Figure 10b shows the unstable solution when $C = 1.5$. Figure 11 shows the relative phase error and the damping error as a function of the dimensionless wave number ξ . It is observed that for $C = 0.5$ and $C = 1$, the phase error is zero.

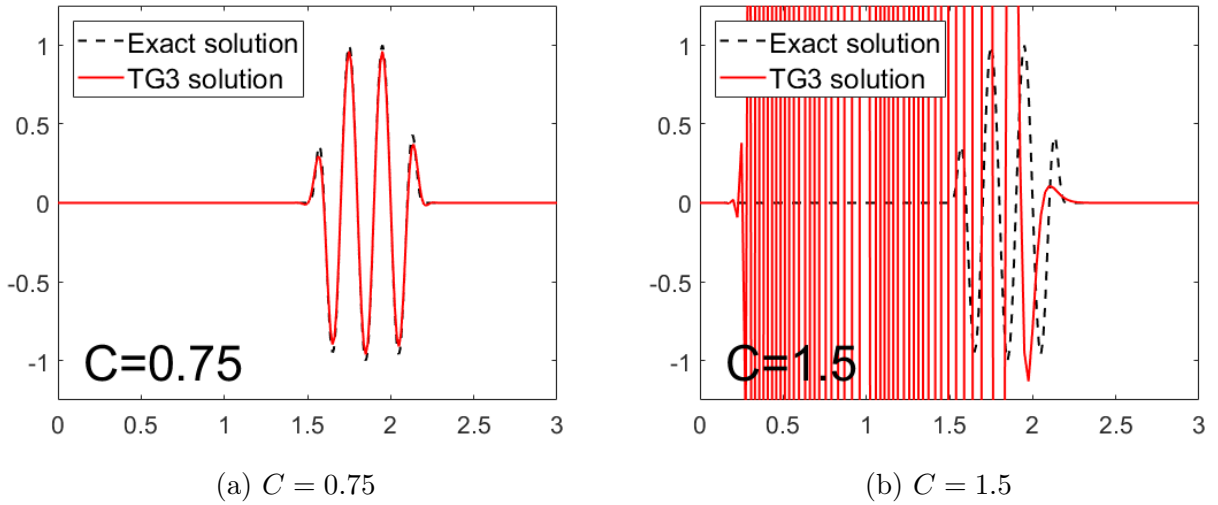


Figure 10: Solution using TG3 time stepping scheme with linear Galerkin spatial discretization at different values of C .

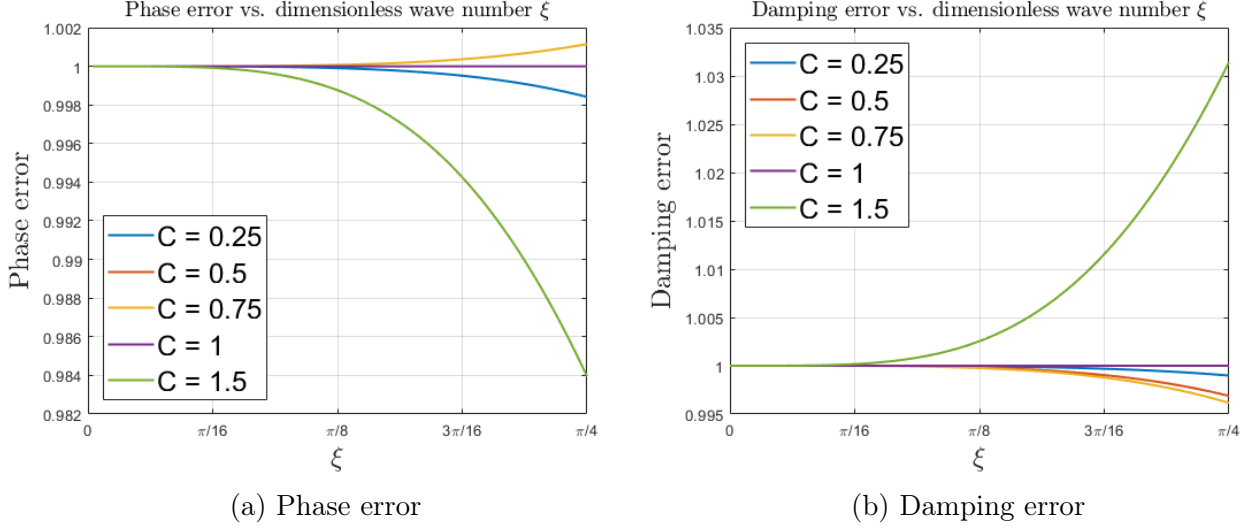


Figure 11: Relative phase error (left) and damping error (right) at different values of C for TG3 time stepping scheme with linear Galerkin spatial discretization.

6 Two-step Third-order Taylor-Galerkin method (TG3-2S)

The time discretization of two-step third-order Taylor-Galerkin method is given by:

$$\begin{aligned}\tilde{u}^n &= u^n + \frac{1}{3}\Delta t u_t^n + \alpha \Delta t^2 u_{tt}^n, \\ u^{n+1} &= u^n + \Delta t u_t^n + \frac{1}{2}\Delta t^2 \tilde{u}_{tt}^n,\end{aligned}\tag{8}$$

where the fully discrete version of the 1D case of linear convection is given in [2] by:

$$\begin{aligned}\left(1 + \frac{1}{6}\delta^2\right)(\tilde{u}^n - u^n) &= -\frac{1}{6}C\delta u^n + \alpha C^2\delta^2 u^n, \\ \left(1 + \frac{1}{6}\delta^2\right)(u^{n+1} - u^n) &= -\frac{1}{2}C\delta u^n + \frac{1}{2}C^2\delta^2 \tilde{u}^n,\end{aligned}\tag{9}$$

The discrete operators shown in (9) are defined in [3] as:

Discrete operator	Matrix form
$1 + \frac{1}{6}\delta^2$	$\frac{1}{h^2}\mathbf{M}$
$\frac{1}{2}C\delta$	$\frac{a\Delta t}{h^2}\mathbf{C}$
$C^2\delta^2$	$-\frac{a^2\Delta t^2}{h^2}\mathbf{K}$

Therefore, equation (9) is written in matrix form as:

$$\begin{aligned}\mathbf{M}(\tilde{\mathbf{U}}^n - \mathbf{U}^n) &= -\frac{1}{3}a\Delta t\mathbf{C}\mathbf{U}^n - \alpha a^2\Delta t^2\mathbf{K}\mathbf{U}^n, \\ \mathbf{M}(\mathbf{U}^{n+1} - \mathbf{U}^n) &= -a\Delta t\mathbf{C}\mathbf{U}^n - \frac{a^2\Delta t^2}{2}\mathbf{K}\tilde{\mathbf{U}}^n,\end{aligned}\tag{10}$$

The solution procedure is to solve for \tilde{U}^n in the first step, then substitute the obtained solution into the second step to obtain U^{n+1} .

The code for this method is developed as a part of this assignment. The stability limit for two-step third-order Taylor-Galerkin method is $C < \sqrt{3}/2$. Figure 12a shows the stable solution obtained when $C = 0.75$, while Figure 12b shows the unstable solution when $C = 1$. Figure 13 shows the relative phase error and the damping error as a function of the dimensionless wave number ξ . It is observed that for $C = 0.5$ the phase error is zero.

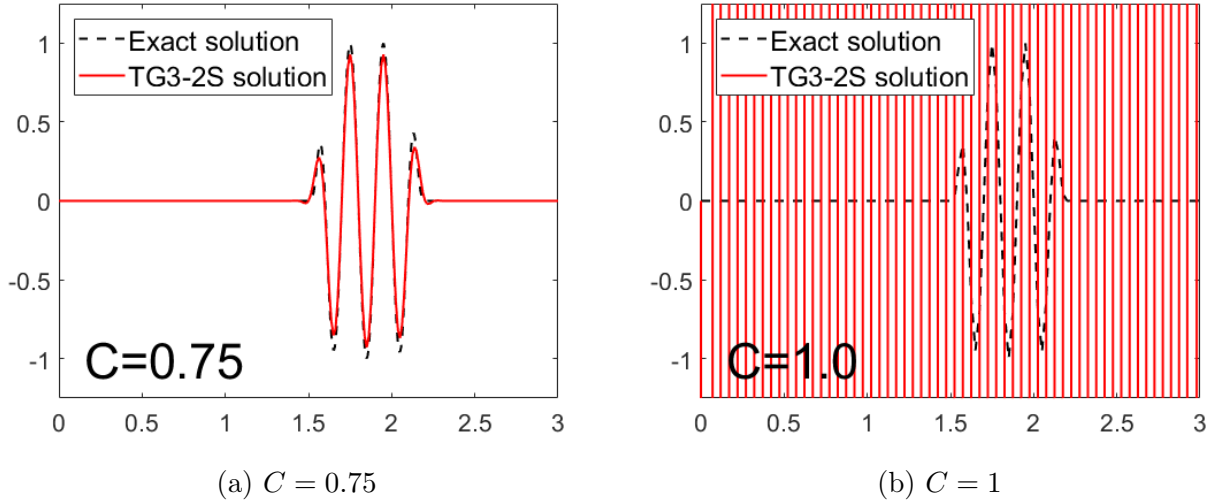


Figure 12: Solution using TG3-2S time stepping scheme with linear Galerkin spatial discretization at different values of C .

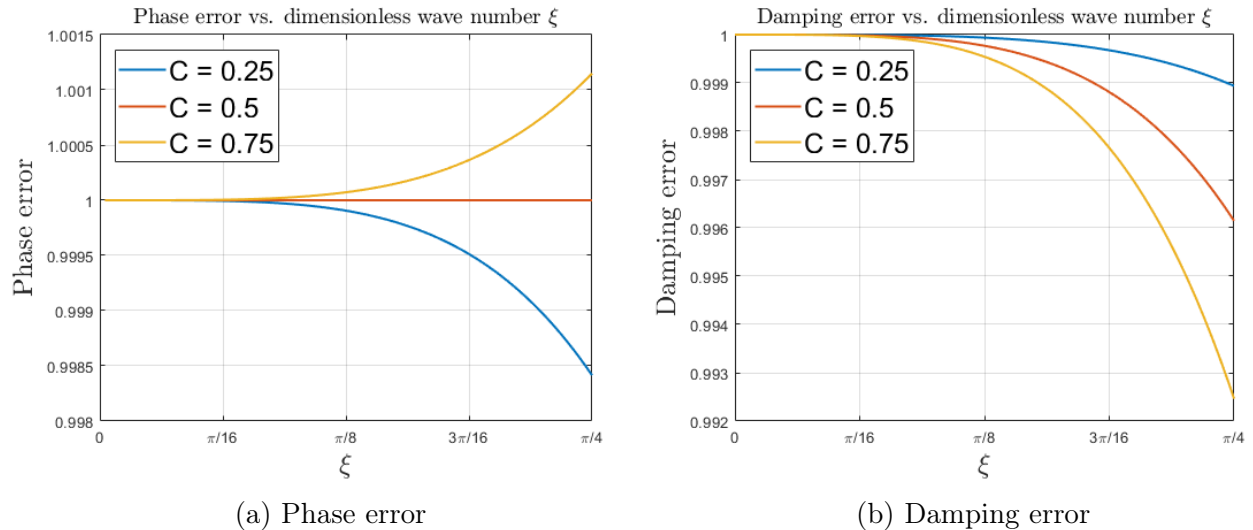


Figure 13: Relative phase error (left) and damping error (right) at different values of C for TG3-2S time stepping scheme with linear Galerkin spatial discretization.

A comparison between the one-step and two-step third-order Taylor-Galerkin method is shown in Figure 14 where the relative phase error and amplification factor are shown at $C = 0.75$. It is observed that the phase-speed characteristics is exactly the same for TG3 and TG3-2S, this is due to using the parameter $\alpha = 1/9$. Furthermore, it is noticed that the two-step method (TG3-2S) is more dissipative when compared to the one-step method (TG3).

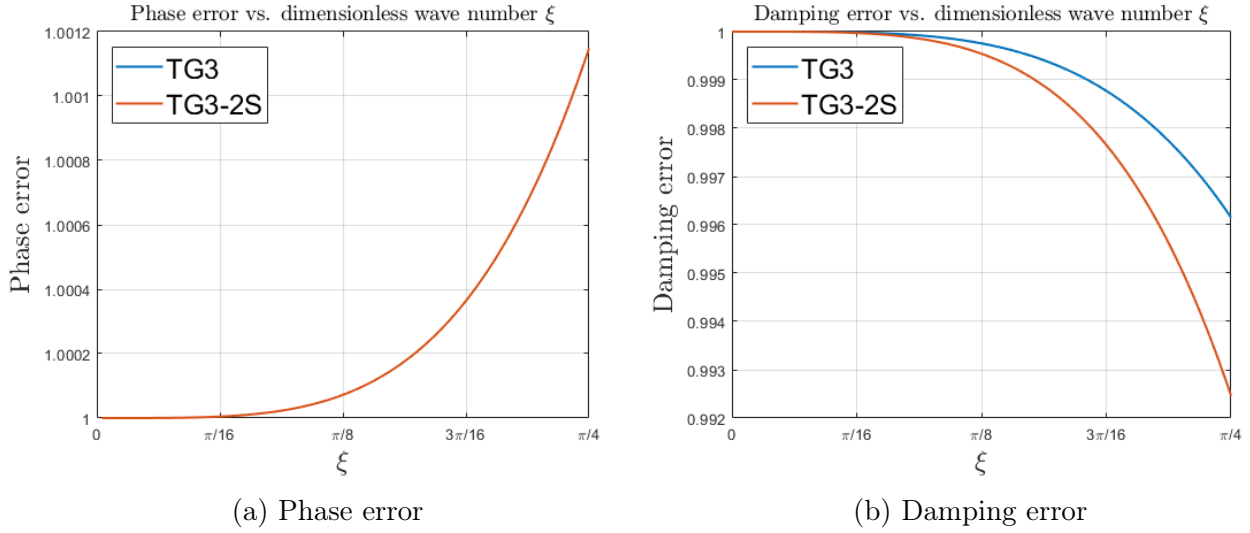
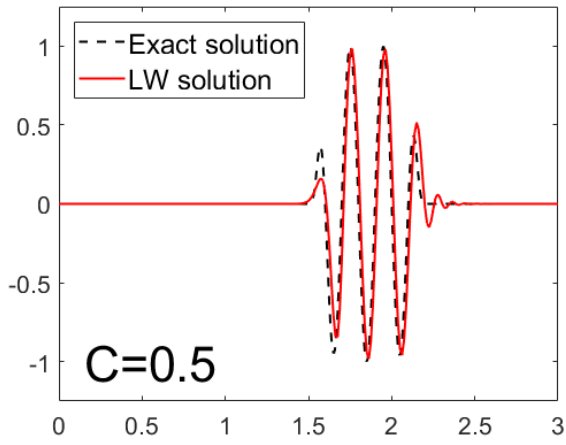
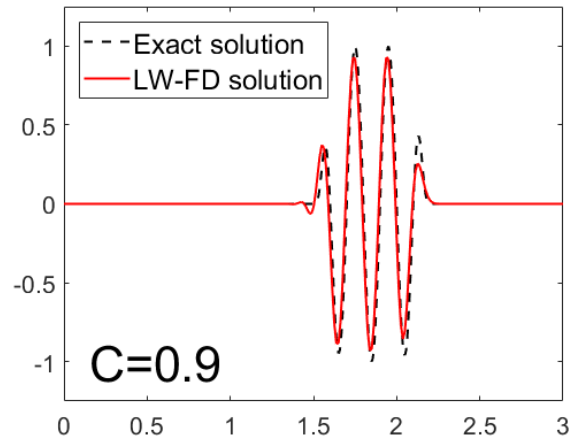


Figure 14: Relative phase error (left) and damping error (right) at $C = 0.75$ for one-step (blue) and two-step (red) third-order Taylor-Galerkin combined with linear Galerkin spatial discretization.

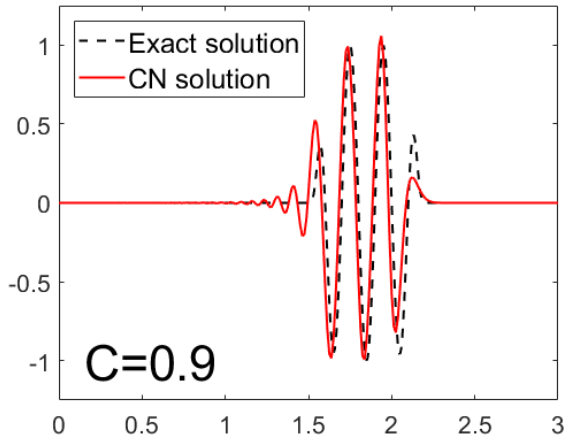
7 Comparison at $C = 0.9$ or 90% of the stability limit



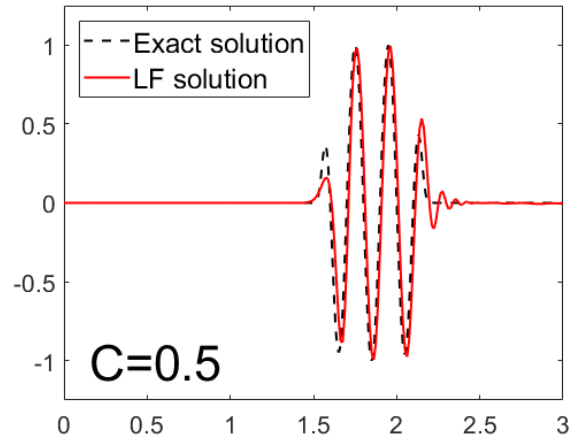
(a) LW



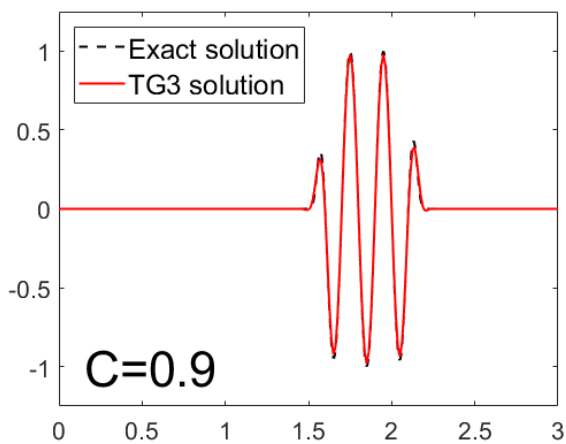
(b) LW-FD



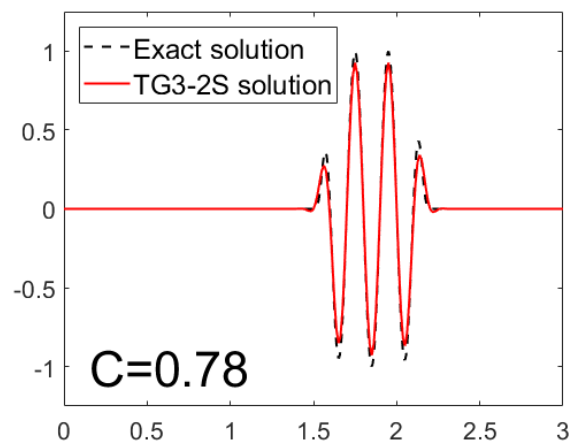
(c) CN



(d) LF



(e) TG3



(f) TG3-2S

Figure 15: Solution obtained using different time stepping schemes combined with linear Galerkin spatial discretization.

A Developed codes

A.1 Modifications in System.m

Four time-stepping schemes are added to this function which are denoted by case 5 to case 8.

```
1 function [A,B,methodName] = System(method,M,K,C,a,dt)
2
3 switch method
4     case 1 % Lax-Wendroff + Galerkin
5         A = M;
6         B = -a*dt*C - 0.5*a^2*dt^2*K;
7         methodName = 'LW';
8     case 2 % Lax-Wendroff with lumped mass matrix + Galerkin
9         A = diag(sum(M));
10        B = -a*dt*C - 0.5*a^2*dt^2*K;
11        methodName = 'LW-FD';
12    case 3 % Crank-Nicolson + Galerkin
13        A = M + 1/2*a*dt*C;
14        B = -a*dt*C;
15        methodName = 'CN';
16    case 4 % Crank-Nicolson with lumped mass matrix + Galerkin
17        A = diag(sum(M)) + 1/2*a*dt*C;
18        B = -a*dt*C;
19        methodName = 'CN-FD';
20    case 5 % Leap-frog method + Galerkin
21        A = M;
22        B = -2*a*dt*C;
23        methodName = 'LF';
24    case 6 % Leap-frog method with lumped mass matrix + Galerkin
25        A = diag(sum(M));
26        B = -a*2*dt*C;
27        methodName = 'LF-FD';
28    case 7 % Third order Taylor-Galerkin + Galerkin
29        A = M + 1/6*dt^2*a^2*K;
30        B = -a*dt*C - 0.5*a^2*dt^2*K;
31        methodName = 'TG3';
32    case 8 % Two-steps third order Taylor-Galerkin + Galerkin
33        A = M \ (M - a*dt*C);
34        B = -1/2*dt^2*a^2*(M\K);
35        methodName = 'TG3-2S';
36    otherwise
37        error('not available method')
38 end
```

A.2 Modifications in main.m

The main modification here is to use Lax-Wendroff scheme for the first time step in case of employing leap-frog method, as well as adding a specific for-loop for TG3-2S scheme.

```
1  if strcmp(methodName, 'LF')
2
3      [A_aux, B_aux, methodName_aux] = System(1, M, K, C, a, dt);
4
5      % Reduced system to impose du(0) = 0:
6      ind_unk = 2:nPt;
7      A_aux = A_aux(ind_unk, ind_unk);
8      if problem == 4
9          f_aux = [B_aux(2,1); zeros(nPt-2,1)];
10     else
11         f_aux = zeros(nPt-1,1);
12     end
13     B_aux = B_aux(ind_unk, ind_unk);
14
15     for n = 1:nStep
16         if n == 1
17             Du = A_aux \ (B_aux*u(ind_unk, n) + f_aux);
18             u(ind_unk, n+1) = u(ind_unk, n) + Du;
19         else
20             u(ind_unk, n+1) = A \ (A*u(ind_unk, n-1) + B*u(ind_unk, n) + f);
21         end
22     end
23
24 elseif strcmp(methodName, 'TG3-2S')
25
26     alpha = 1/9;
27     uTelda = u;
28     for n = 1:nStep
29         % First step
30         uTelda(:, n) = (M \ (M - 1/3*dt*a*C - alpha*dt^2*a^2*K)) * u(:, n);
31         % Second Step
32         u(ind_unk, n+1) = A*u(ind_unk, n) + B*uTelda(ind_unk, n);
33     end
34
35 else
36     for n = 1:nStep
37         Du = A \ (B*u(ind_unk, n) + f);
38         u(ind_unk, n+1) = u(ind_unk, n) + Du;
39     end
40 end
```

A.3 New function ExactG.m

This function gives the exact amplification factor.

```
1 function G = ExactG(C,xi)
2
3 I = sqrt(-1);
4
5 G = exp(-I*C*xi);
6
7 end
```

A.4 New script plotPhaseDampingError.m

This script was made to plot the figures of phase error and amplification factor.

```
1 xi = linspace(0,pi/4,100);
2
3 C = [0.25,0.5,0.75,1,1.5];
4
5 for iC = 1:length(C)
6     G = NumericalG(C(iC),xi);
7     Gex = ExactG(C(iC),xi);
8     legendNameLW = (['C = ',num2str(C(iC))]);
9     legendNameLWFD = (['C = ',num2str(C(iC))]);
10    legendNameCN = (['C = ',num2str(C(iC))]);
11    legendNameLF = (['C = ',num2str(C(iC))]);
12    legendNameTG3 = (['C = ',num2str(C(iC))]);
13    legendNameTG32S = (['C = ',num2str(C(iC))]);
14    relativePhaseErrorLW = angle(G.TG2)./angle(Gex);
15    relativePhaseErrorLWFD = angle(G.TG2_FD)./angle(Gex);
16    relativePhaseErrorCN = angle(G.CN)./angle(Gex);
17    relativePhaseErrorLF = angle(G.LF)./angle(Gex);
18    relativePhaseErrorTG3 = angle(G.TG3)./angle(Gex);
19    relativePhaseErrorTG32S = angle(G.TG32s)./angle(Gex);
20
21    figure(1)
22    plot(xi,abs(G.TG2),'LineWidth',1.5,'DisplayName',legendNameLW)
23    % plot(xi,abs(G.CN),'LineWidth',1.5,'DisplayName',legendNameCN)
24    % plot(xi,abs(G.LF),'LineWidth',1.5,'DisplayName',legendNameLF)
25    % plot(xi,abs(G.TG3),'LineWidth',1.5,'DisplayName',legendNameTG3)
26    % plot(xi,abs(G.TG32s),'LineWidth',1.5,'DisplayName',legendNameTG32S)
27    hold on
28    % plot(xi,abs(G.TG2_FD),'LineWidth',1.5,'DisplayName',legendNameLWFD)
29    % plot(xi,abs(G.TG32s),'LineWidth',1.5,'DisplayName',legendNameTG32S)
30
31    figure(2)
32    plot(xi,relativePhaseErrorLW,'LineWidth',1.5,'DisplayName',legendNameLW)
33    % plot(xi,relativePhaseErrorCN,'LineWidth',1.5,'DisplayName',legendNameCN)
34    % plot(xi,relativePhaseErrorLF,'LineWidth',1.5,'DisplayName',legendNameLF)
35    % plot(xi,relativePhaseErrorTG3,'LineWidth',1.5,'DisplayName',legendNameTG3)
36    % ...
37    plot(xi,relativePhaseErrorTG32S,'LineWidth',1.5,'DisplayName',legendNameTG32S)
38    hold on
```

```

38 % ...
    plot(xi,relativePhaseErrorLWFD,'LineWidth',1.5,'DisplayName',legendNameLWFD)
39 % ...
    plot(xi,relativePhaseErrorTG32S,'LineWidth',1.5,'DisplayName',legendNameTG32S)
40
41 end
42
43 figure(1)
44 title('Damping error vs. dimensionless wave number ...
    $\xi$', 'Interpreter', 'latex', 'FontSize', 14)
45 xlabel('$\xi$', 'Interpreter', 'latex', 'FontSize', 18, 'FontName', 'cmr12')
46 ylabel('Damping ...
    error', 'Interpreter', 'latex', 'FontSize', 18, 'FontName', 'cmr12')
47 xlim([0,pi/4])
48 xticks([0 pi/16 pi/8 3*pi/16 pi/4])
49 xticklabels({'0', '\pi/16', '\pi/8', '3\pi/16', '\pi/4'})
50 grid on
51 legend('Location', 'NorthWest')
52
53 figure(2)
54 title('Phase error vs. dimensionless wave number ...
    $\xi$', 'Interpreter', 'latex', 'FontSize', 14)
55 xlabel('$\xi$', 'Interpreter', 'latex', 'FontSize', 18, 'FontName', 'cmr12')
56 ylabel('Phase error', 'Interpreter', 'latex', 'FontSize', 18, 'FontName', 'cmr12')
57 xlim([0,pi/4])
58 xticks([0 pi/16 pi/8 3*pi/16 pi/4])
59 xticklabels({'0', '\pi/16', '\pi/8', '3\pi/16', '\pi/4'})
60 grid on
61 legend('Location', 'NorthWest')

```

References

- [1] Donea, J., Huerta, A. (2004). *Finite Element Methods for Flow Problems*. Chichester: Wiley, pp.99-105.
- [2] Donea, J., Huerta, A. (2004). *Finite Element Methods for Flow Problems*. Chichester: Wiley, pp.115.
- [3] Donea, J., Huerta, A. (2004). *Finite Element Methods for Flow Problems*. Chichester: Wiley, pp.100.