Homework V : Dissipation and Dispersion

Dhruv Balwada

April 27, 2015

1 Statement of problem

The problem is to solve the initial-boundary value problem :

$$IVP : [\begin{array}{c} u_t + u_x = 0 \\ u(x,0) = f(x) \\ u(0,t) = u(2\pi, t) \end{array} \tag{1}$$

The boundary conditions are periodic. Using finite difference approximations with the Courant-Issacson-Rees(CIR) and Lax-Wendroff(LW) schemes.

$$U_j^{n+1} = U_j^n - \triangle t \delta_x^- U_j^n \tag{2}$$

$$U_j^{n+1} = U_j^n - \triangle t[\delta_x^o - \frac{\triangle t}{2}\delta_x^+ \delta_x^-]U_j^n \tag{3}$$

2 Description of The Mathematics

The stability, consistency and convergence properties of the two schemes have been analyzed in the class and in previous assignments. Also the dispersion and dissipation of the schemes are solved for as part of this assignment. I only present the main results here. The dissipation and dispersion results are attached as hand written notes.

The CIR is a conditionally stable scheme. If the CFL number(λ) is positive(right travelling wave solutions to original hyperbolic problem) then it is stable for $\lambda \leq 1$. Also, it is consistent with accuracy of $O(\triangle t, \triangle x)$. The dissipation and dispersion results are as follows

Amplitude

$$|G|^2 = 1 - 4\lambda(1-\lambda)sin^2(\theta/2) \tag{4}$$

Phase Velocity

$$\alpha = \frac{1}{\lambda\theta} tan^{-1}(\frac{\lambda sin\theta}{1 - 2\lambda sin^2(\theta/2)}) \tag{5}$$

Group Velocity

$$\gamma = \frac{1 - 2(1-\lambda)sin^2(\theta/2)}{1 - 4\lambda(1-\lambda)sin^2(\theta/2)} \tag{6}$$

The LW is a also a conditionally stable scheme with the same stability criteria as CIR. Also, it is consistent with accuracy of $O(\triangle t^2, \triangle x^2)$. The dissipation and dispersion results are as follows

Amplitude

$$|G|^2 = 1 - 4\lambda^2(1-\lambda^2)sin^4(\theta/2) \tag{7}$$

Phase Velocity

$$\alpha = \frac{1}{\lambda\theta}tan^{-1}(\frac{\lambda sin\theta}{1-2\lambda^2 sin^2(\theta/2)}) \tag{8}$$

Group Velocity

$$\gamma = \frac{1-2(1-\lambda^2)sin^2(\theta/2)}{1-4\lambda^2(1-\lambda^2)sin^4(\theta/2)} \tag{9}$$

The plots of each of these expressions are shown in Figures 1, 13 and 3. I will come back to these while trying to understand the properties of the numerical solutions.

2.1 Numerical approximations to spatial derivatives

The schemes require the calculation of forward, backward and centered spatial derivatives. This is achieved using the xderiv.f90 routine.

The order of the finite difference approximation can be found using Taylor series approximation and difference calculus.

$$\delta^o = \frac{S - S^{-1}}{2\triangle x} \tag{10}$$

Where S is the forward shift operator and $\triangle x$ is the grid spacing.

We know $S = e^{\triangle x D}$ where D is the derivative operator. This gives that

Going through a similar analysis we can see that $\delta^+ = D + O(\triangle x)$ and $\delta^- = D + O(\triangle x)$

This scheme should give the exact results at all the interior points for a 2nd or lower degree polynomial and exact results for a linear polynomial on the boundary points.

2.2 Euler Solver

The time stepping scheme is the Forward Euler. This scheme is first order accurate in time. This can be seen using a similar analysis(not repeated here) as the one done before for a centered difference operator.

3 Description of the Algorithm

The code for solving this homework was written in modular form. In this section I provide the algorithm for each individual module and then the algorithm for the main program that connects the modules.

I am starting my arrays from index of 1 in my codes. So all the spatial and time indices as shown in the problem statement will be shifted by 1.

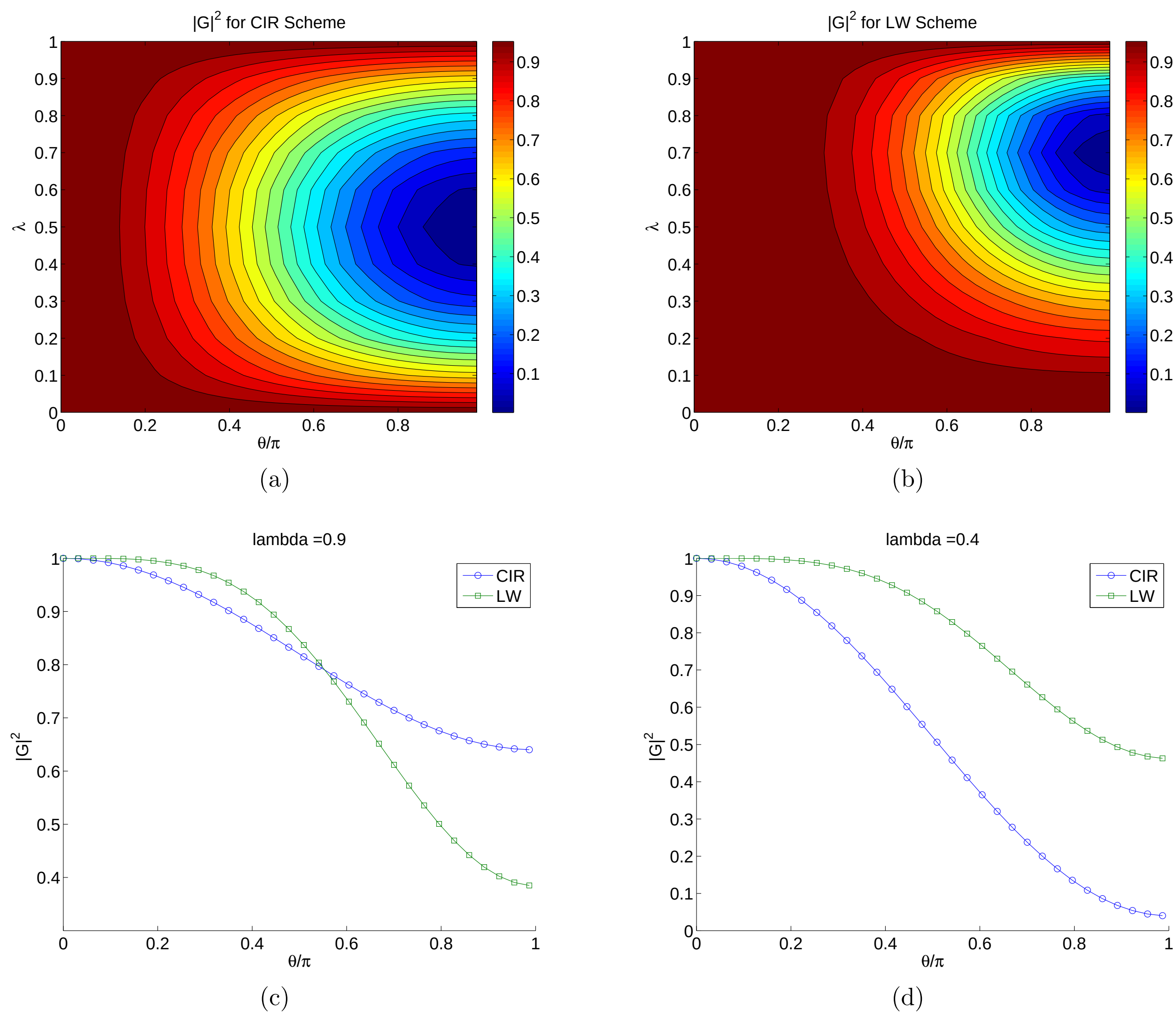


Figure 1: Theoretical growth curves. (c) and (d) show slices for the two schemes at CFL = 0.4 and 0.9

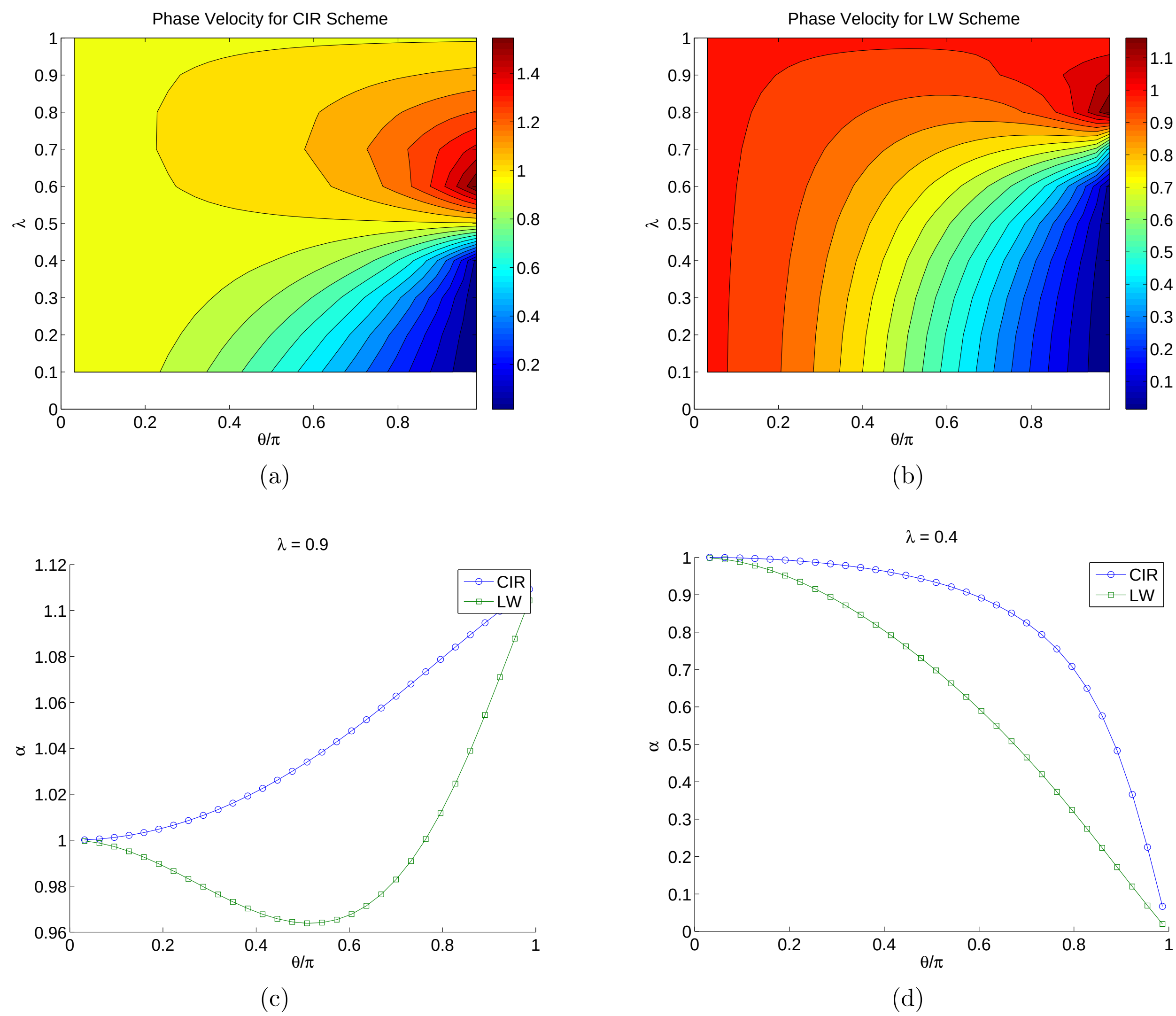
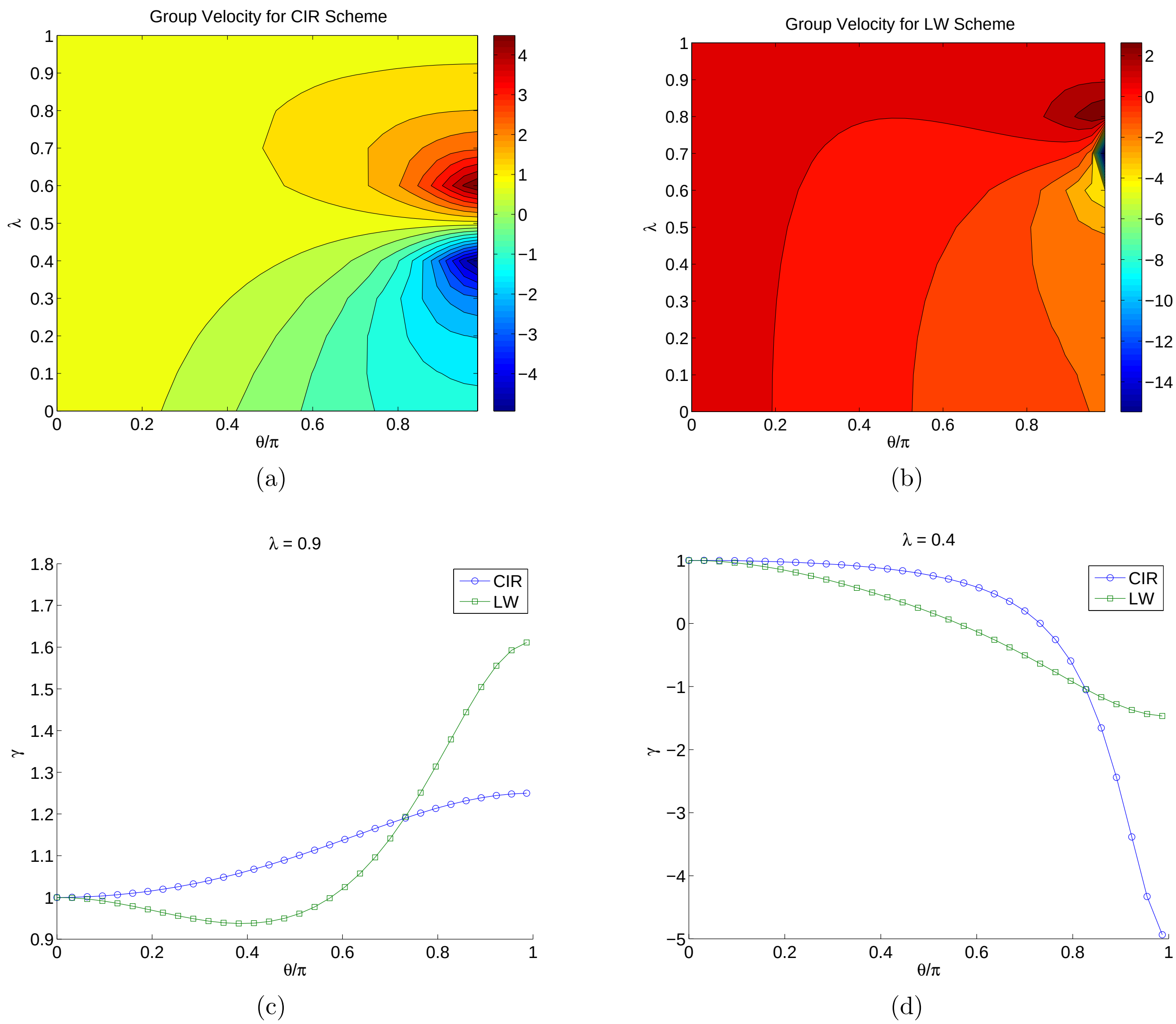


Figure 2: Theoretical phase velocity curves. (c) and (d) show slices for the two schemes at CFL = 0.4 and 0.9



(a) (b)

(c) (d)

Figure 3: Theoretical group velocity curves. (c) and (d) show slices for the two schemes at CFL = 0.4 and 0.9

3.1 xDeriv Algorithm

GET the vector that needs to be differentiated(u), a vector to hold the derivative(dudx), number of spatial points(N) and the size of the grid(dx) as input.
$dudx_1 = (u_2 - u_1)/dx$
for j = 2 to N **do**
  $dudx_j = (u_{j+1} - u_{j-1})/(2 * dx)$
end for
$dudx_{N+1} = (u_{N+1} - u_N)/dx$

3.2 Euler time stepping Algortihm

GET Q, neqn, N ,time, dx, dt
for k =1 to kmax **do**
  tk = time + dt*b(k)
  update or assign value to dQdt
  for l=1 to neqn **do**
    for j=1 to N+1 **do**
      G = a(k)*G + dQdt(tk,Q)
      Q(j,l) = Q(j,l) + g(k)*G*dt
    end for
  end for
end for

4 Testing Results

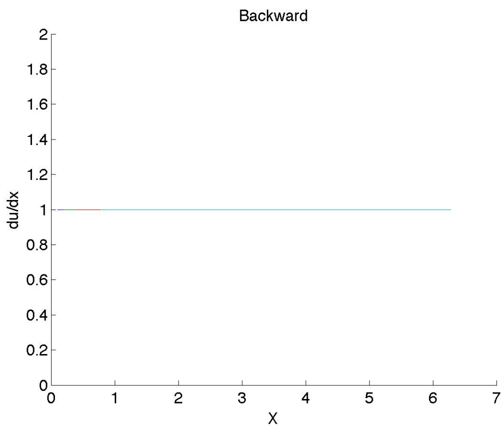
4.1 xDeriv Testing

The xDeriv routine is tested using simple functions. We expect the first order δ^+ and δ^- to work perfectly for a linear function and show order linear convergence for quadratic functions. The δ^o works perfectly for linear and quadratic functions and shows quadratic convergence for cubic functions. This is shown in Figures , and . These tests show that the derivative routines are working exactly as expected.

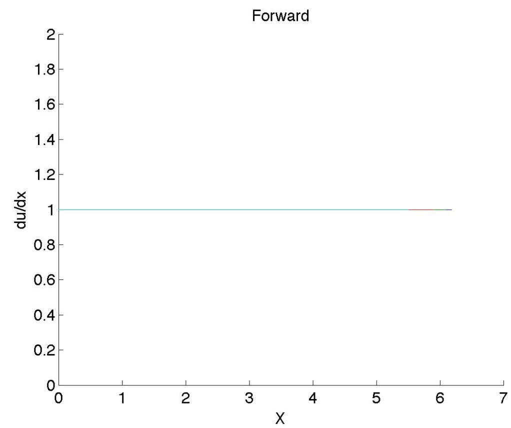
As the functions that we are using to evaluate the properties of the numerical scheme are not periodic we leave out the end points to not have to worry about the boundary conditions are we are interested in the properties of the scheme only.

4.2 Euler Solver Testing

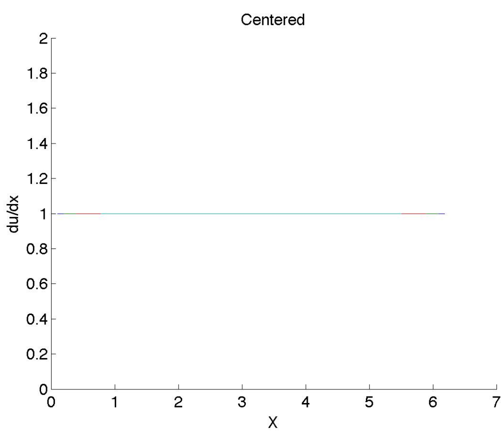
The Euler solver being a first order time stepping scheme works perfectly for a linear solution $du/dt = 1$ and shows linear convergence for a quadratic solution to equation $du/dt = t$. This is shown in Figure . For the linear function the error is on the order of the round of errors(order 10^-14. For the quadratic case the error grows linearly with dt. So we conclude that the Euler solver is working properly and as expected.



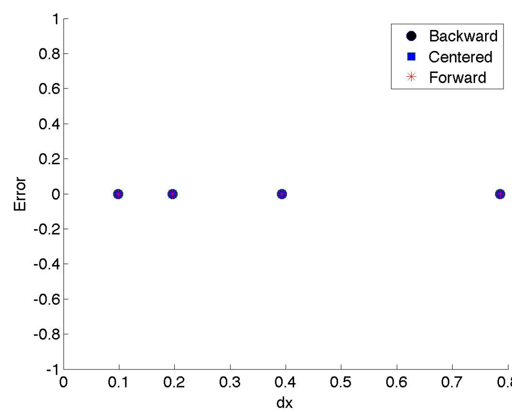
(a)



(b)

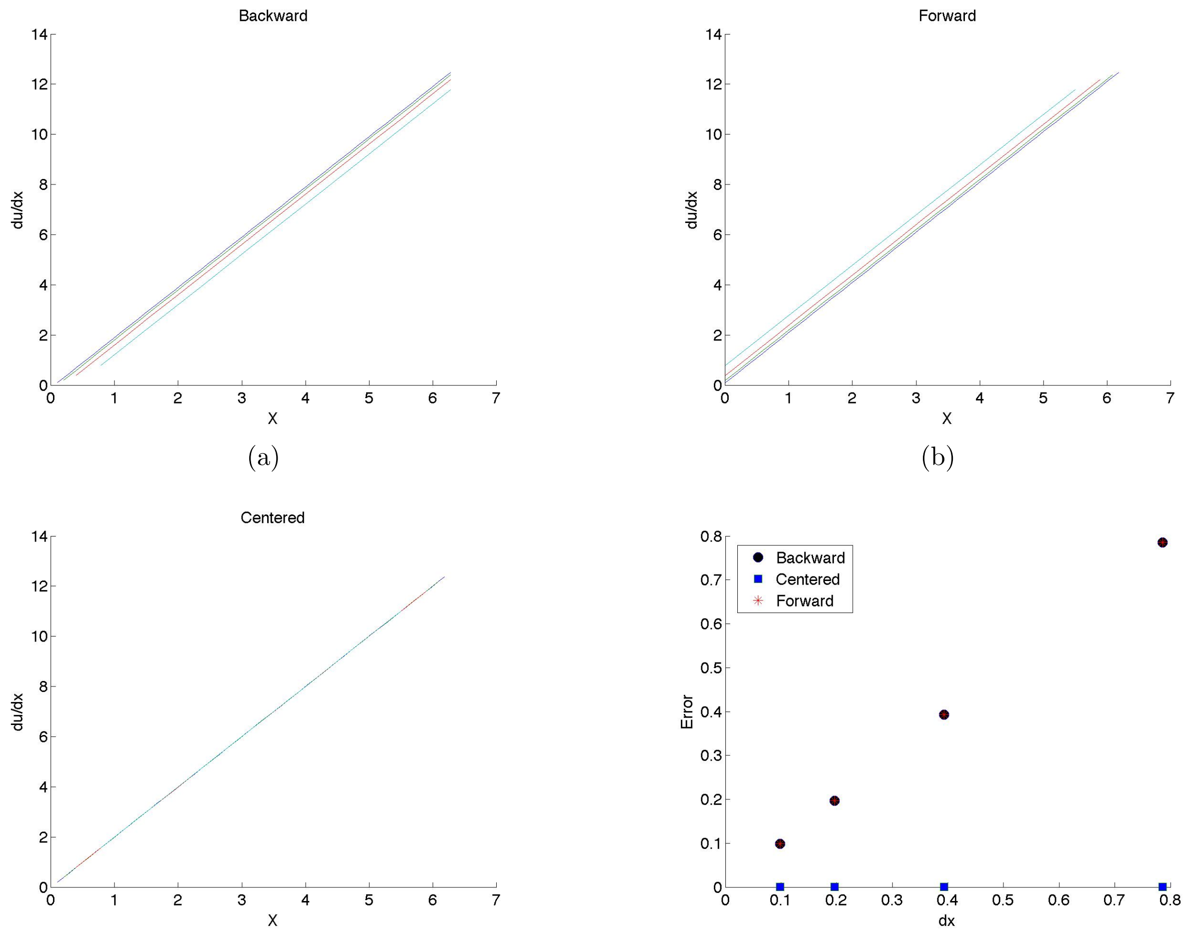


(c)



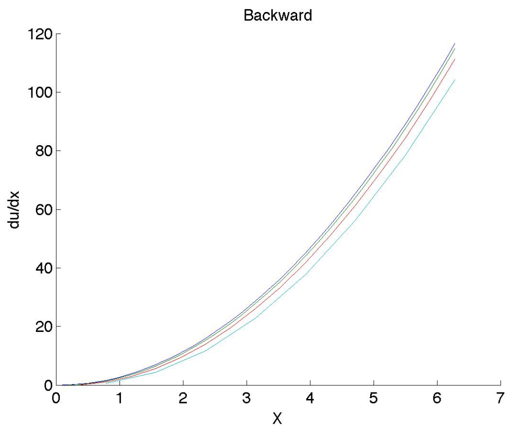
(d) Error vs dx

Figure 4: Tests of derivatives on a linear function. All schemes give accurate results

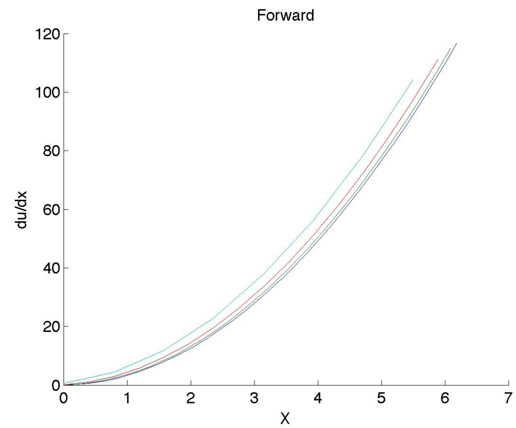


(a) (b) (c) (d) Error vs dx

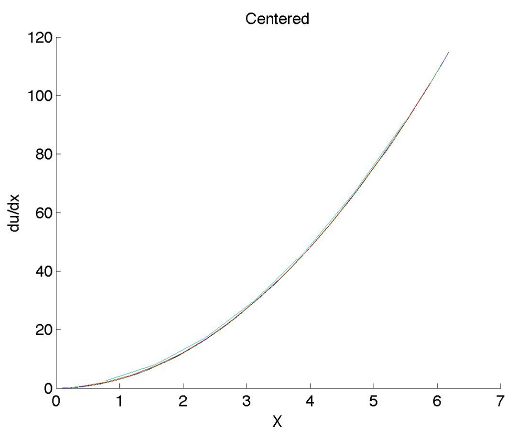
Figure 5: Tests of derivatives on a quadratic function. Centered difference has no errors while forward and backward differencing have errors that increase linearly with dx



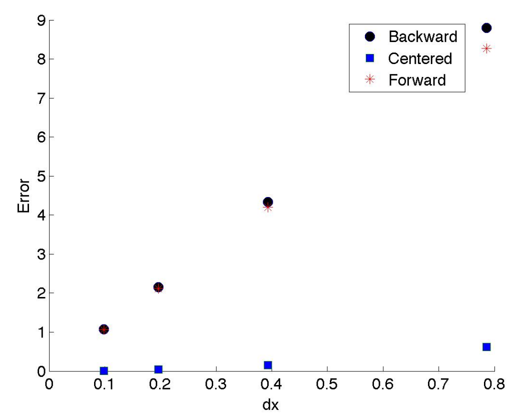
(a)



(b)

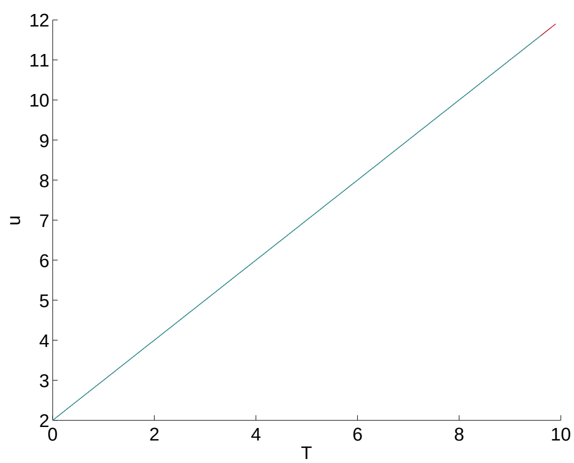


(c)

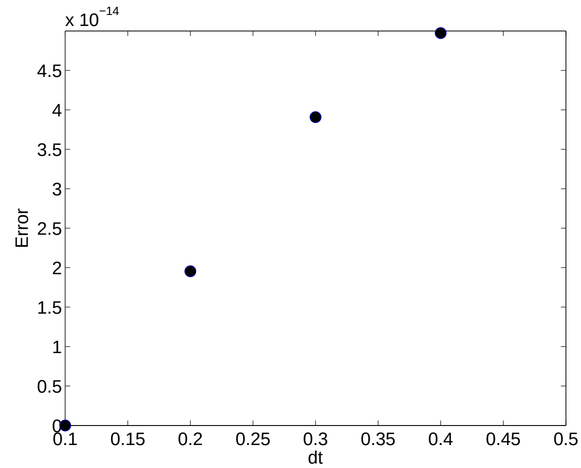


(d) Error vs dx

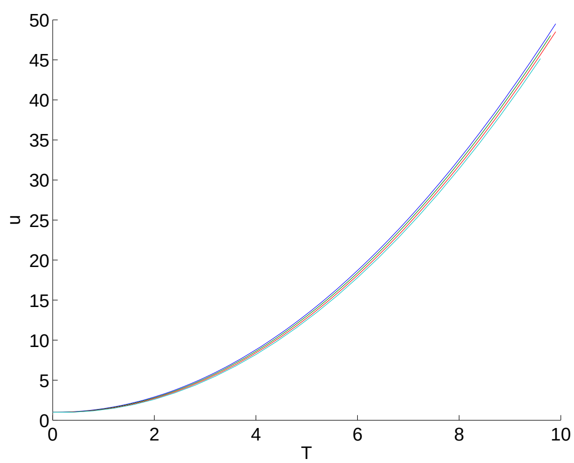
Figure 6: Tests of derivatives on a cubic function. Centered difference have errors that grow quadratically while forward and backward differencing have errors that increase linearly with dx



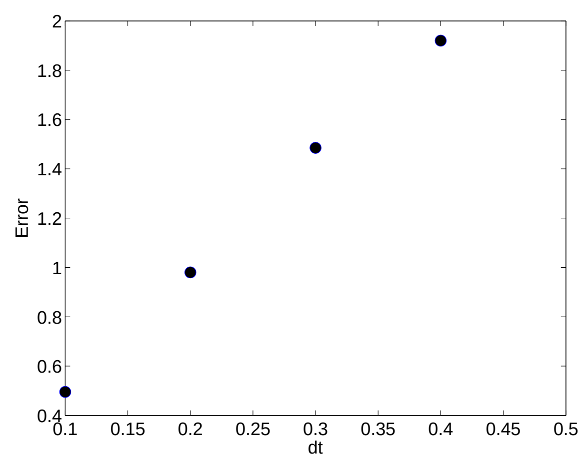
(a) u



(b) Error vs dt

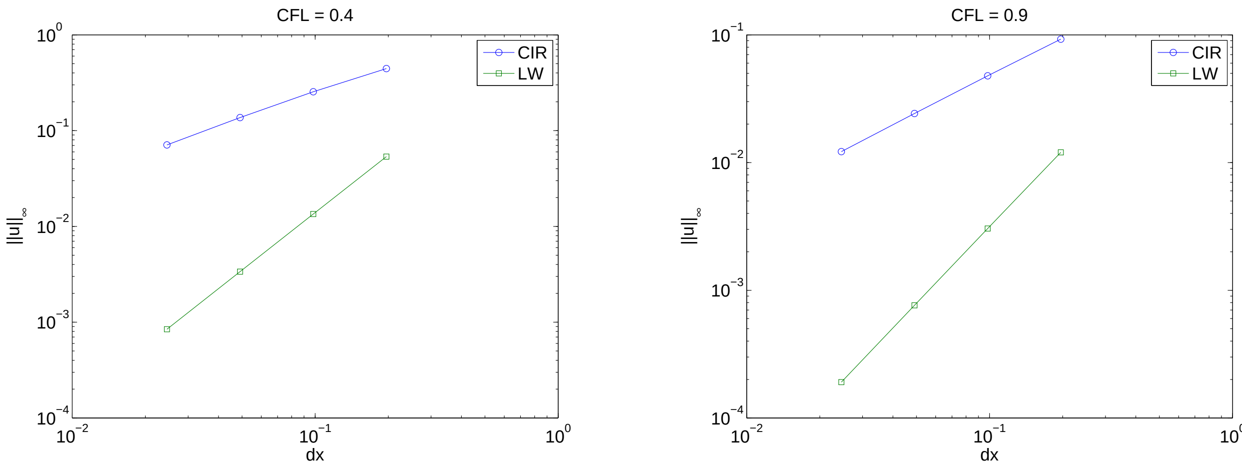


(c) u



(d) Error vs dt

Figure 7: Test cases for the Euler time stepping Scheme



(a) u

(b) Error vs dt

Figure 8: Errors for different CFL using CIR and LW schemes

4.3 CIR solver/LW solver

I first test the solvers on a simple sine wave initial condition to see if the system has the general expected properties. As the dissipation and dispersion results are tested later, here I only look at the order of accuracy of the schemes. Here I fix the lambda and then see how the error changes. As fixing the CFL number makes the dt and dx change together(linearly) we only look at the results for varying dx. Figure shows that the errors actually decrease with dx as expected with slopes of 1 and 2 for CIR and LW respectively.

5 Main Results

Now that we have tested the individual components of the code and also the working of them together we can proceed ahead and quantify the dispersion and dissipation errors as asked in the assignment.

5.1 Dispersion and Dissipation Results

The schemes are tested on 3 kinds of initial conditions. The first one is a sine wave with a long wavelength. This gives us an idea about the property of the numerical schemes at low wavenumbers. The second is a short wavelength wave with $2\triangle x$ wavelength and gives us information about the other extreme of the wavenumbers. The third case uses a gaussian enveloped wave packet. This will give us information about all wavenumbers and how the schemes do in regards to replicating the group velocity.

5.1.1 Sine Wave

The initial condition is $f(x) = sin(4x)$ and for $\triangle x = 2\pi/32$. The two schemes are run at CFL of 0.9.

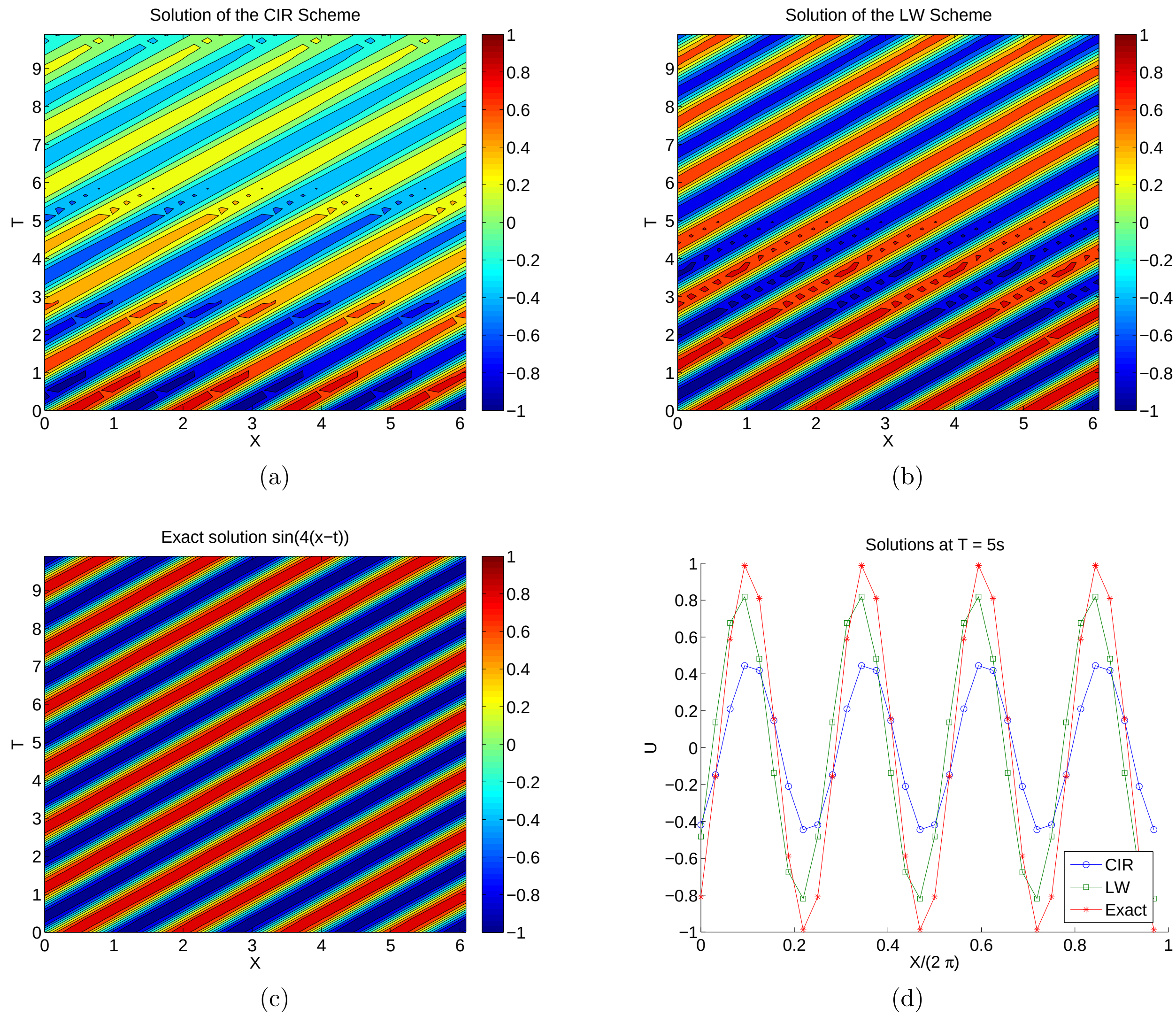


Figure 9

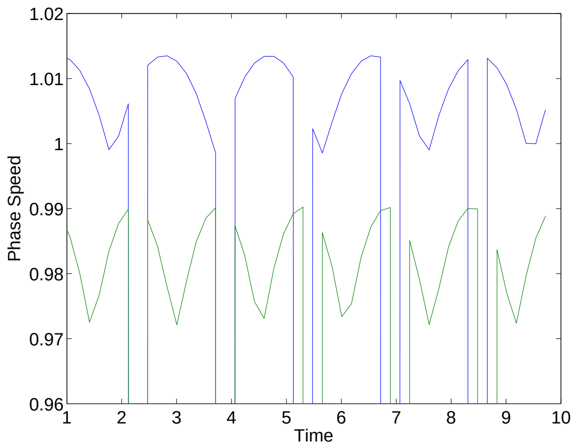
The results are in figures :

5.1.2 T

he initial condition is $f(x_j) = (-1)^j$. The two schemes are run at CFL of 0.9.

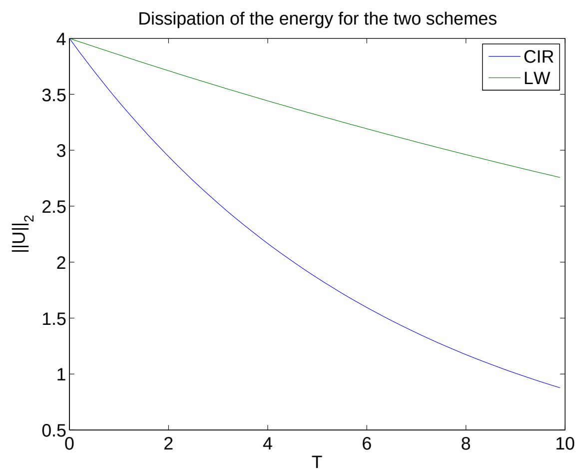
5.1.3 Gaussian Wave Packet

6 Conclusion



1.02
1.01
1
0.99
0.98
0.97
0.96
1 2 3 4 5 6 7 8 9 10
Time
Phase Speed

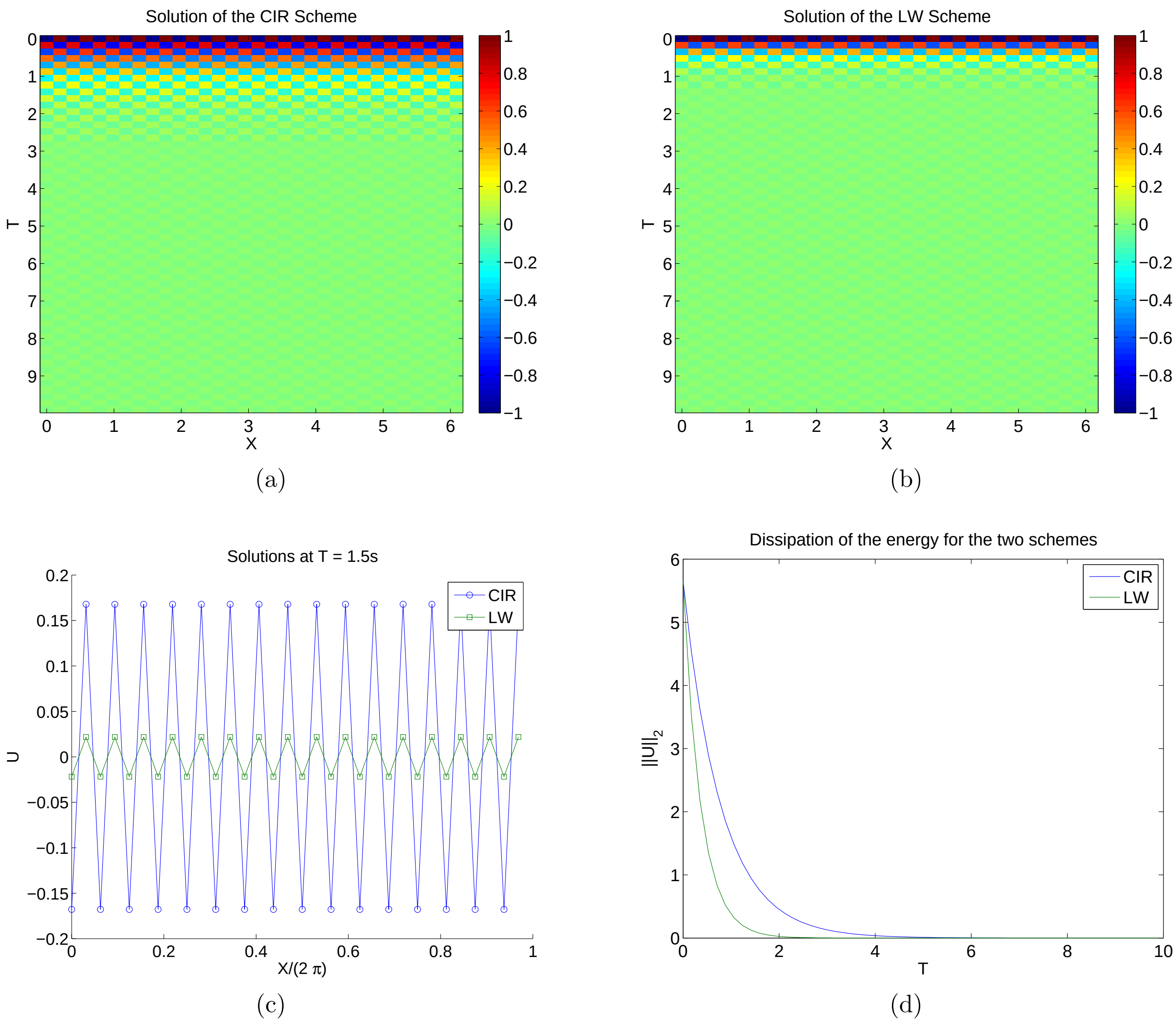
(a)



Dissipation of the energy for the two schemes
CIR
LW
4
3.5
3
2.5
2
1.5
1
0.5
0 2 4 6 8 10
T
$||U||_2$

(b)

Figure 10



Solution of the CIR Scheme
T
X
Solution of the LW Scheme
T
X
(a)
(b)
Solutions at T = 1.5s
CIR
LW
U
X/(2 π)
Dissipation of the energy for the two schemes
CIR
LW
||U||2
T
(c)
(d)

Figure 11

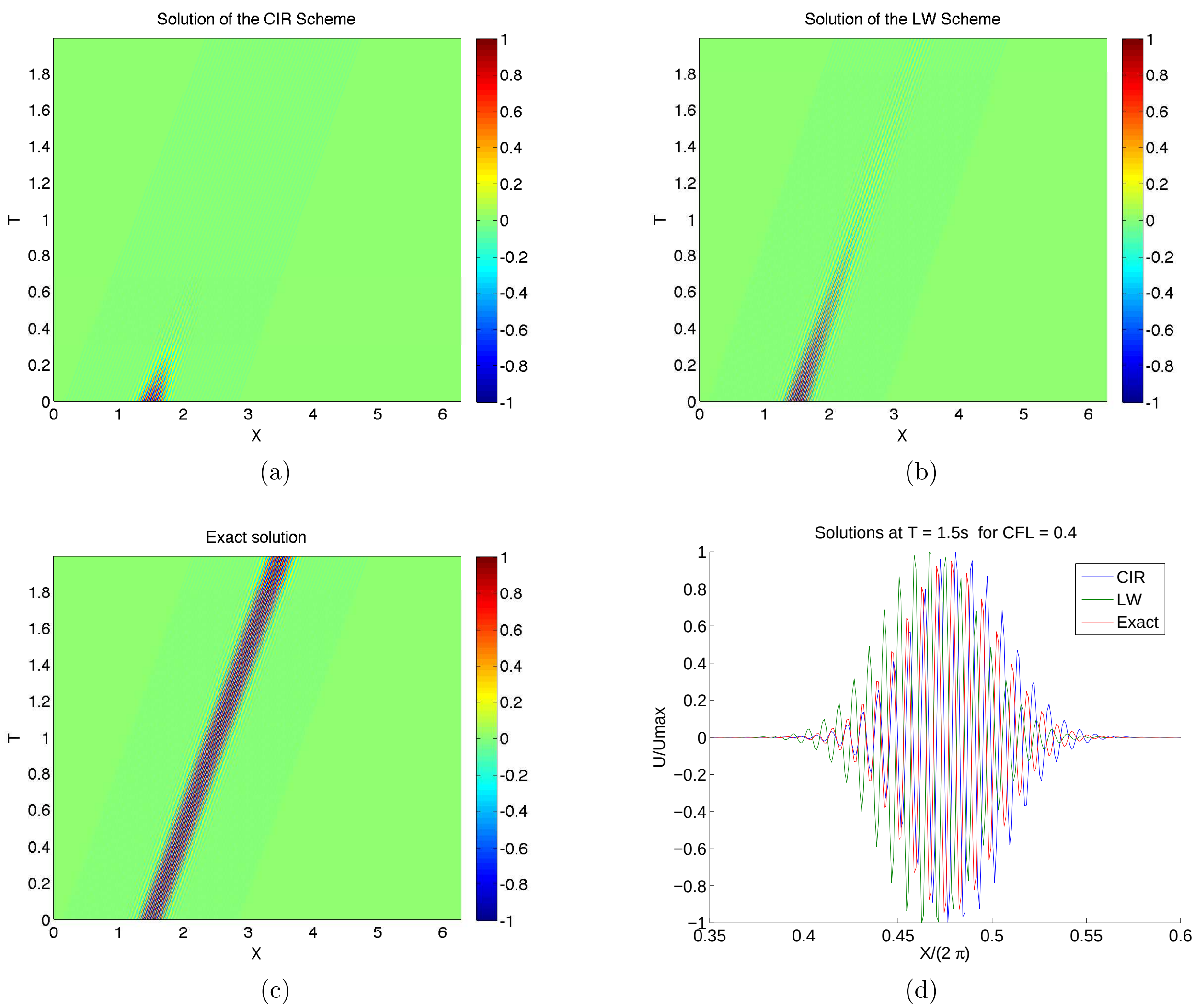
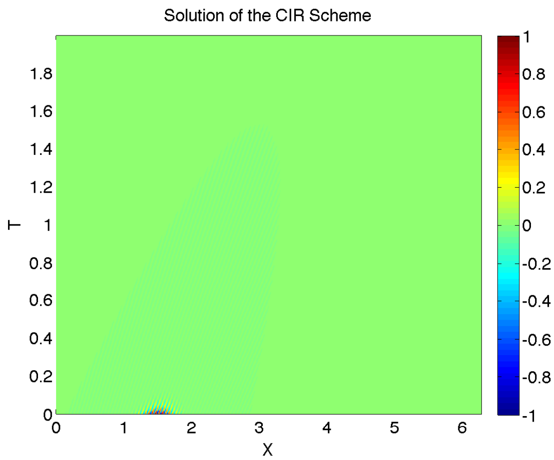
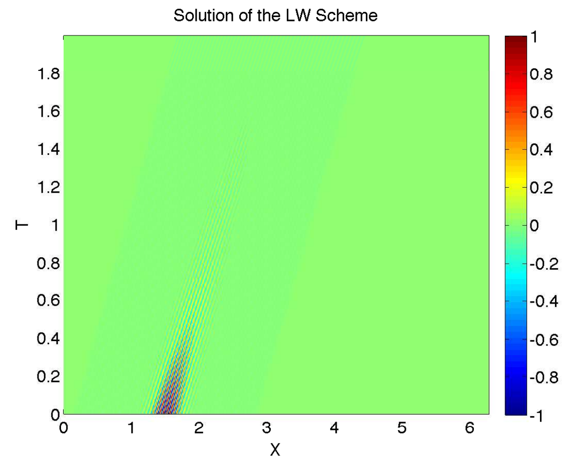

Solution of the CIR Scheme
Solution of the LW Scheme
Exact solution
Solutions at T = 1.5s for CFL = 0.4
CIR
LW
Exact
T
X
U/Umax
X/(2 π)
(a)
(b)
(c)
(d)

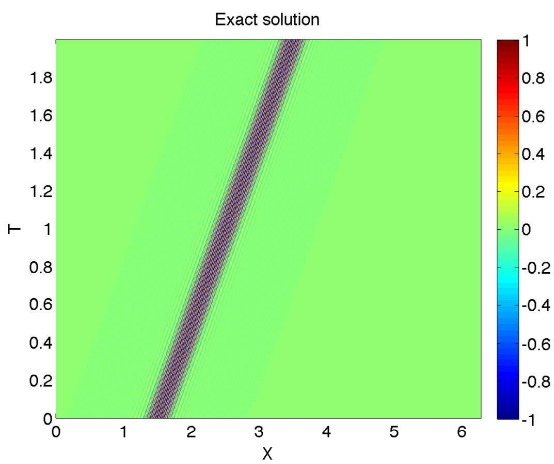
Figure 12



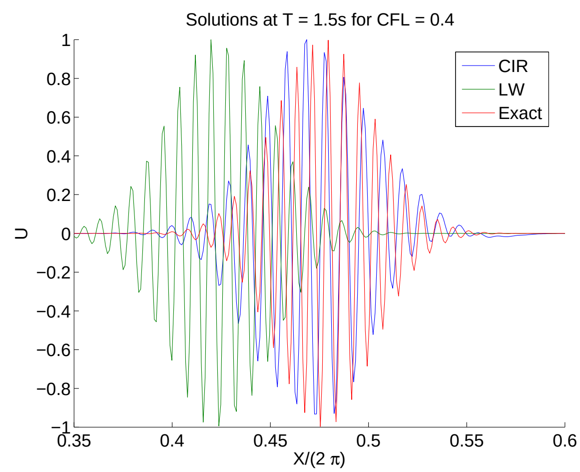
(a)



(b)



(c)



(d)

Figure 13