Codes to solve Advection Equation for given initial conditions

1) **Using FTFS, FTCS, FTBS, LW, BW and Fromm Methods**

```
# Calling necessary libraries from Python
import numpy as np
import matplotlib.pyplot as plt

# Function to define initial conditions
def U_initials(x, condition_type):
    u = np.zeros(len(x))
    if condition_type == 1:
        u[x < 0.2] = 1
    elif condition_type == 2:
        y = (x >= 0.05) & (x < 0.35)
        u[y] = np.sin(4 * np.pi * (x[y] - 0.05) / 0.3)
    elif condition_type == 3:
        y = (x >= 0.05) & (x < 0.35)
        u[y] = np.sin(8 * np.pi * (x[y] - 0.05) / 0.3)
    elif condition_type == 4:
        y = (x >= 0.05) & (x < 0.35)
        u[y] = np.sin(12 * np.pi * (x[y] - 0.05) / 0.3)
    elif condition_type == 5:
        u = np.exp(-50 * ((x - 0.2)**2) / 0.16) #sigma =0.4
    return u

# Parameters
courant_values = [0.5, 1.0, 1.5]  # values of v
space_marching_points = 101  # total number of space grids
a = 1  # value of velocity
x = np.linspace(0, 1, space_marching_points) # list of spatial grid
points
dx = x[1] - x[0] # distance between two grid points
```

```
# Function to apply methods
def Methods(u, nu, method):
    dt = nu * dx / a
    time = np.arange(0, 0.35 + dt, dt)
    answer = np.zeros((len(x), len(time)))
    answer[:, 0] = u

    for n in range(len(time) - 1):
        for i in range(0, len(x) - 1):
            if method == "FTFS":
                answer[i, n + 1] = answer[i, n] - nu * (answer[i + 1,
n] - answer[i, n])
            elif method == "FTCS":
                answer[i, n + 1] = answer[i, n] - 0.5 * nu * (answer[i
+ 1, n] - answer[i - 1, n])
            elif method == "FTBS":
                answer[i, n + 1] = answer[i, n] - nu * (answer[i, n] -
answer[i - 1, n])
            elif method == "Lax_Wandroff":
                answer[i, n + 1] = answer[i, n] - 0.5 * nu * (answer[i
+1, n] - answer[i - 1, n]) + 0.5 * (nu**2) * (answer[i + 1, n] - 2 *
answer[i, n] + answer[i - 1, n])
            elif method == "Beam_Warming":
                answer[i, n + 1] = answer[i, n] - nu * (3 * answer[i,
n] - 4 * answer[i - 1, n] + answer[i - 2, n]) / 2 + (nu**2) *
(answer[i, n] - 2 * answer[i - 1, n] + answer[i - 2, n]) / 2
                if condition_type == 1:
                    answer[1,n+1] = 1
                    answer[len(x)-2,n+1] = 0
                else:
                    answer[1,n+1] = 0
                    answer[len(x)-2,n+1] = 0
            elif method == "Fromm_Method":
                answer[i, n + 1] = (0.5 * (answer[i, n] - 0.5 * nu *
(answer[i + 1, n] - answer[i - 1, n]) + 0.5 * (nu**2) * (answer[i + 1,
n] - 2 * answer[i, n] + answer[i - 1, n]))
```

```
                + 0.5 * (answer[i, n]- nu * (3 * answer[i, n] - 4 *
answer[i - 1, n] + answer[i - 2, n]) / 2 + (nu**2) * (answer[i, n] - 2
* answer[i - 1, n] + answer[i - 2, n]) / 2))
                if condition_type == 1:
                    answer[0,n+1] = 1
                    answer[len(x)-1,n+1] = 0
                else:
                    answer[0,n+1] = 0
                    answer[len(x)-1,n+1] = 0
    return answer, time
# Loop over nu values, initial conditions, and methods
initial_condition_types = [1, 2, 3, 4, 5]
methods = ["FTFS", "FTCS", "FTBS", "Lax_Wandroff", "Beam_Warming",
"Fromm_Method"]

for nu in courant_values:
    for condition_type in initial_condition_types:
        u = U_initials(x, condition_type)
        plt.plot(x, u, label=f'Initial Condition {condition_type}')
        plt.xlabel('x')
        plt.ylabel('u')
        plt.title(f'Initial Condition {condition_type}')
        plt.legend()
        plt.show()
        for method in methods:
            answer, time = Methods(u, nu, method)
            plt.plot(x, answer[:, -1], label=f'Final answer (nu={nu},
method={method})')
            plt.xlabel('x')
            plt.xlim(0,1)
            plt.ylim(-1.5,1.5)
            plt.ylabel('u')
            plt.title(f'Final answer with {method} method (nu={nu})')
            plt.legend()
            plt.show()
```

2) **Using BTCS Method**

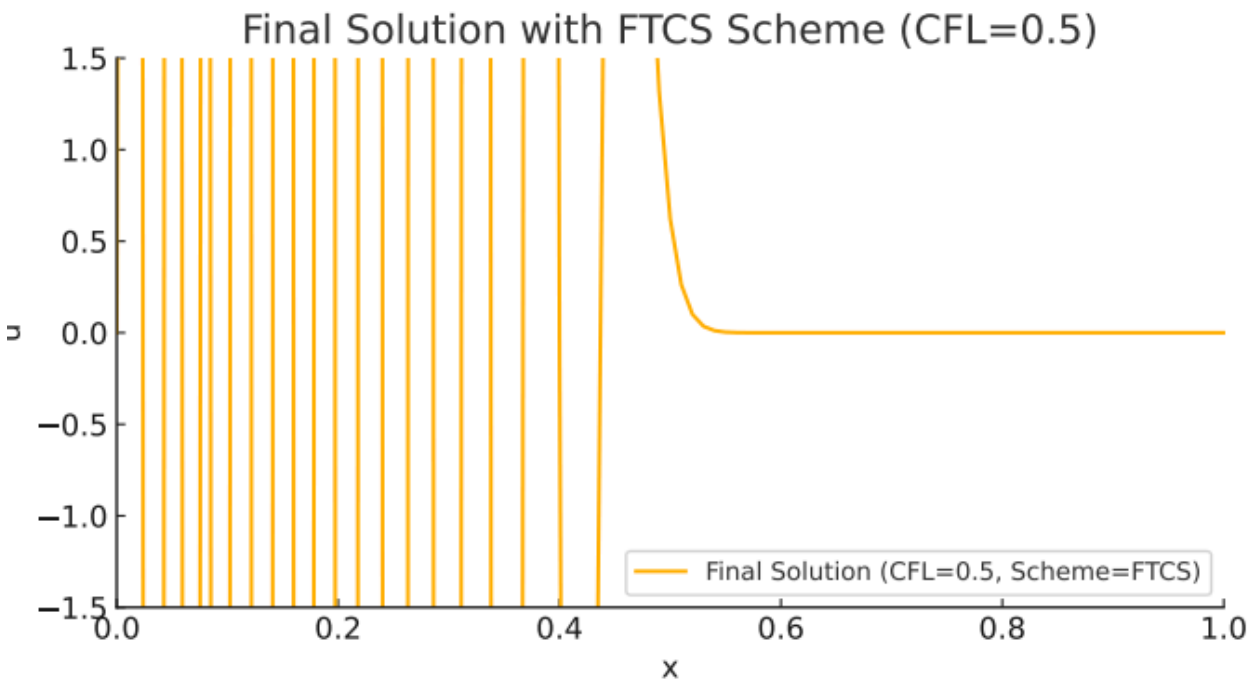
```
# Calling necessary libraries from Python
import numpy as np
import matplotlib.pyplot as plt

# Initial Conditions
def step_function(x):
    u = np.zeros_like(x)
    u[x >= 0.2] = 1.0
    return u

def discontinuous_sin4pi(x):
    u = np.zeros_like(x)
    y = (x >= 0.05) & (x < 0.35)
    u[y] = np.sin(4 * np.pi * (x[y] - 0.05) / 0.3)
    return u

def discontinuous_sin8pi(x):
    u = np.zeros_like(x)
    y = (x >= 0.05) & (x < 0.35)
    u[y] = np.sin(8 * np.pi * (x[y] - 0.05) / 0.3)
    return u

def discontinuous_sin12pi(x):
    u = np.zeros_like(x)
    y = (x >= 0.05) & (x < 0.35)
    u[y] = np.sin(12 * np.pi * (x[y] - 0.05) / 0.3)
    return u

def gaussian_distribution(x):
    return np.exp(-50 * ((x - 0.2)*2) / 0.16)

# Thomas Algorithm for Tri-diagonal Matrix Solver
def thomas_algorithm(a, b, c, d):
    n = len(d)
    c_dash = np.zeros(n - 1)
    d_dash = np.zeros(n)
```

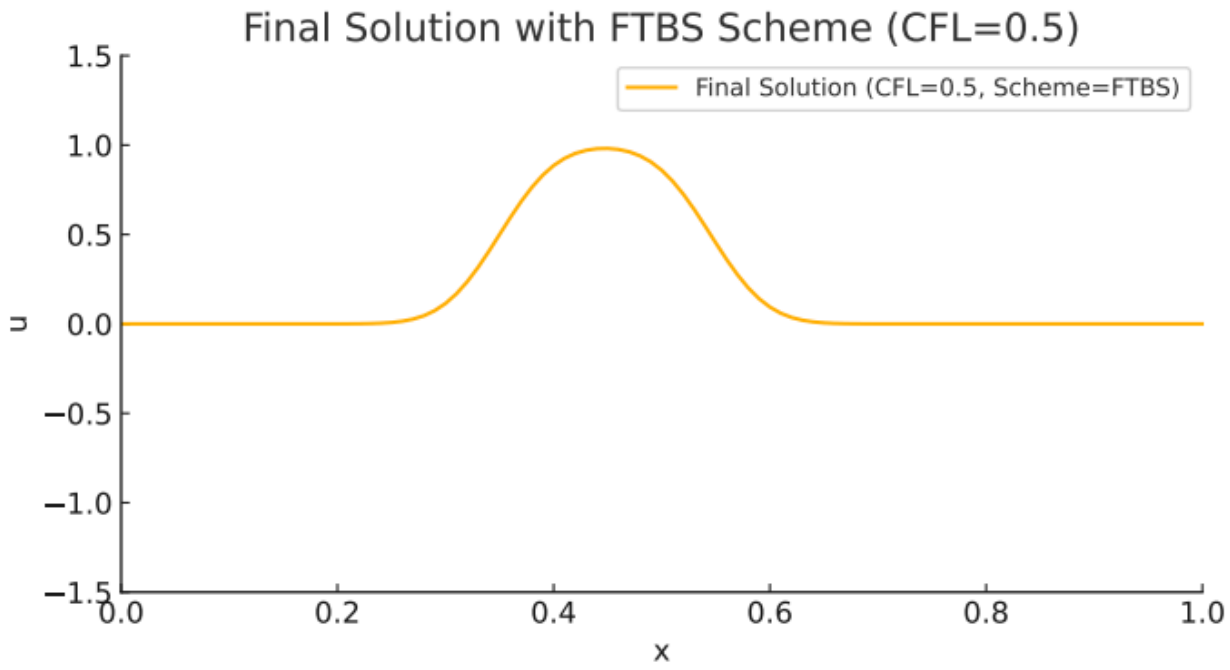
```
    # Forward sweep
    c_dash[0] = c[0] / b[0]
    d_dash[0] = d[0] / b[0]
    for i in range(1, n):
        denom = b[i] - a[i - 1] * c_dash[i - 1]
        if i < n - 1:
            c_dash[i] = c[i] / denom
        d_dash[i] = (d[i] - a[i - 1] * d_dash[i - 1]) / denom

    # Backward substitution
    x = np.zeros(n)
    x[-1] = d_dash[-1]
    for i in range(n - 2, -1, -1):
        x[i] = d_dash[i] - c_dash[i] * x[i + 1]

    return x

# Parameters
a = 1.0  # Wave speed
L = 1.0  # Domain length
nx = 101  # Number of spatial points
x = np.linspace(0, L, nx)  # Spatial grid
dx = x[1] - x[0]  # Spatial step

# BTCS Scheme
def btcs(u, nu, nt):
    n = len(u) - 2
    a = -0.5 * nu * np.ones(n - 1)  # Sub-diagonal
    b = (1 + nu) * np.ones(n)      # Main diagonal
    c = -0.5 * nu * np.ones(n - 1) # Super-diagonal

    for _ in range(nt):
        d = u[1:-1]  # Right-hand side
        u_inner = thomas_algorithm(a, b, c, d)
        u[1:-1] = u_inner

    return u
```

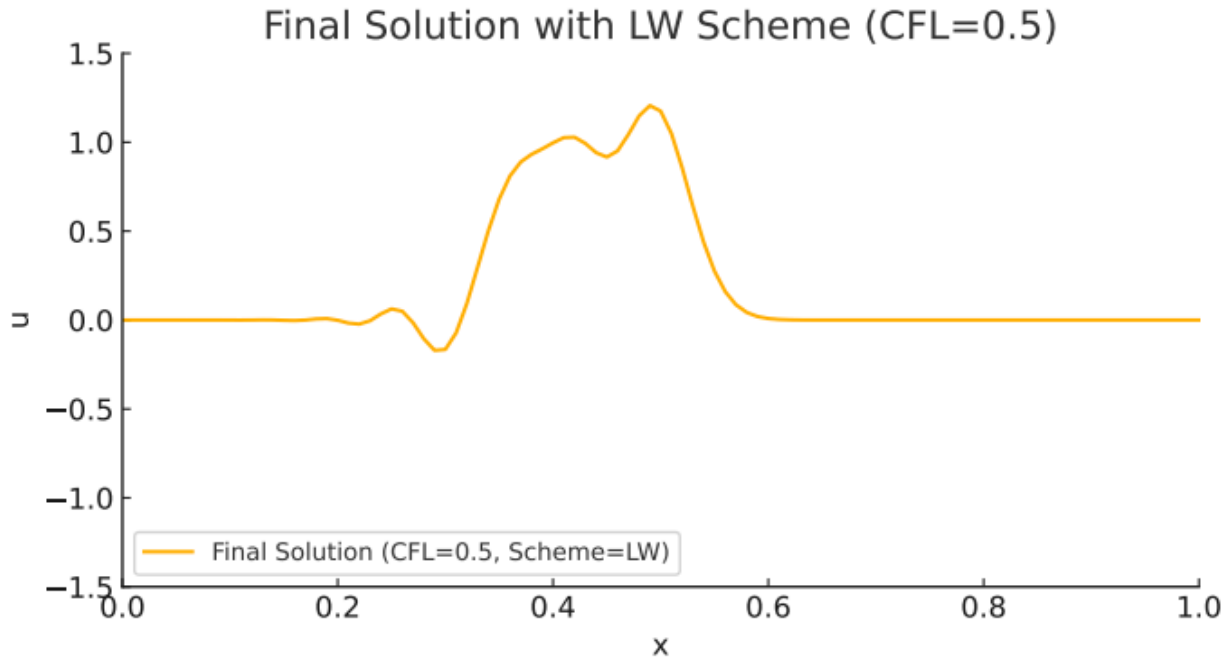
```
# Initial Conditions List
initial_conditions = [
    ("Step Function", step_function),
    ("Discontinuous(sin4pi)",discontinuous_sin4pi),
    ("Discontinuous(sin8pi)", discontinuous_sin8pi),
    ("Discontinuous(sin12pi)", discontinuous_sin12pi),
    ("Gaussian Distribution", gaussian_distribution),
]

# CFL values
v = [0.5, 1.0, 1.5]

# Solve and Plot for Each Initial Condition and CFL Value
for name, ic in initial_conditions:
    for cfl in v:
        dt = cfl * dx / a  # Compute time step for given CFL
        nt = int(0.35 / dt)  # Compute number of time steps
        nu = cfl  # CFL number is nu = a * dt / dx

        u = ic(x)

        # Apply BTCS Scheme
        u = btcs(u, nu, nt)

        # Plot Results
        plt.figure(figsize=(8, 4))
        plt.plot(x, u, label=f"CFL={nu:.2f}, t=0.35s")
        plt.title(f"BTCS Scheme: {name} Initial Condition")
        plt.xlabel("x")
        plt.ylabel("u")
        plt.ylim(-1.5, 1.5)
        plt.legend()
        plt.grid()
        plt.show()
```

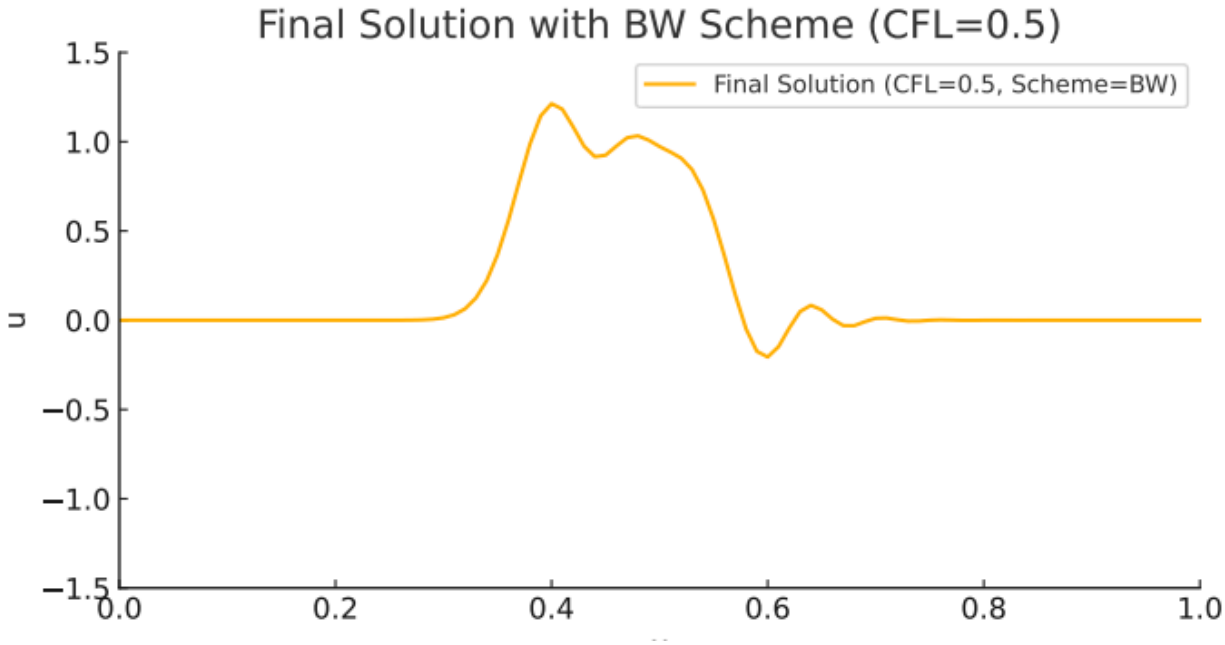
Results

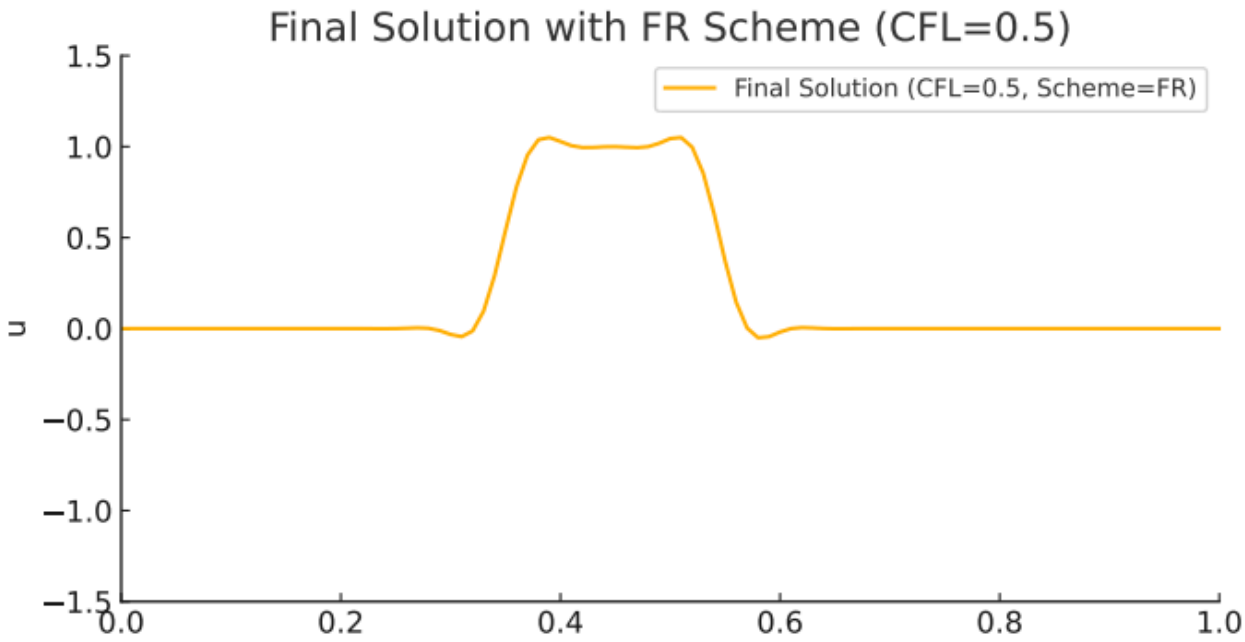
a) For 1st condition

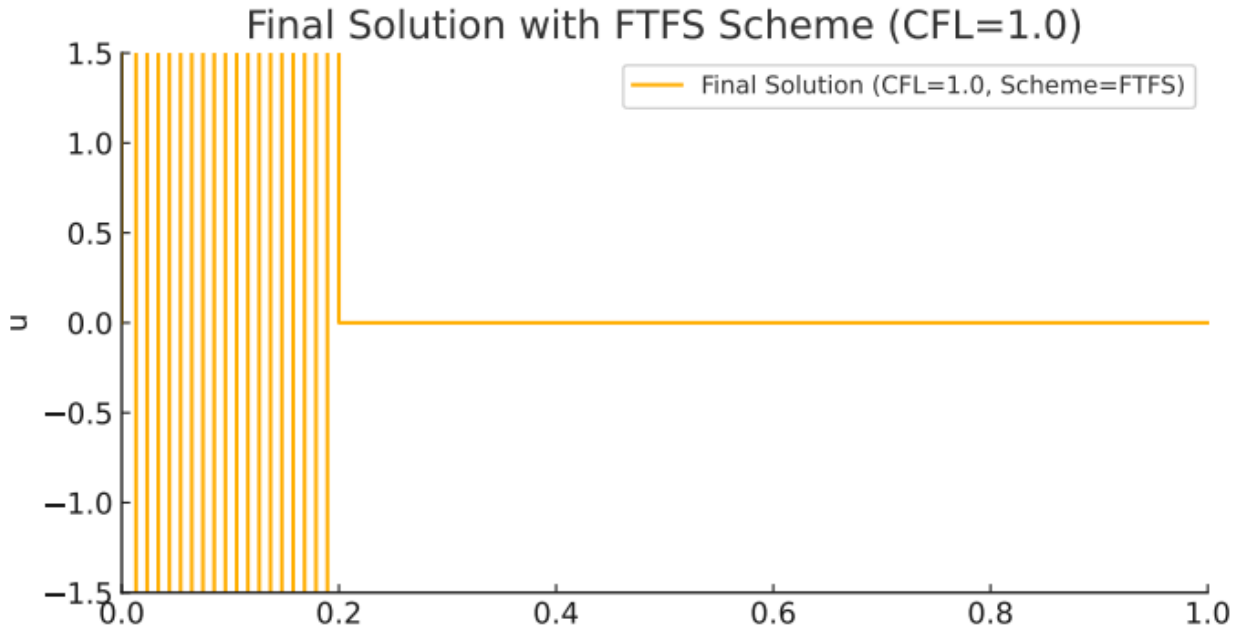


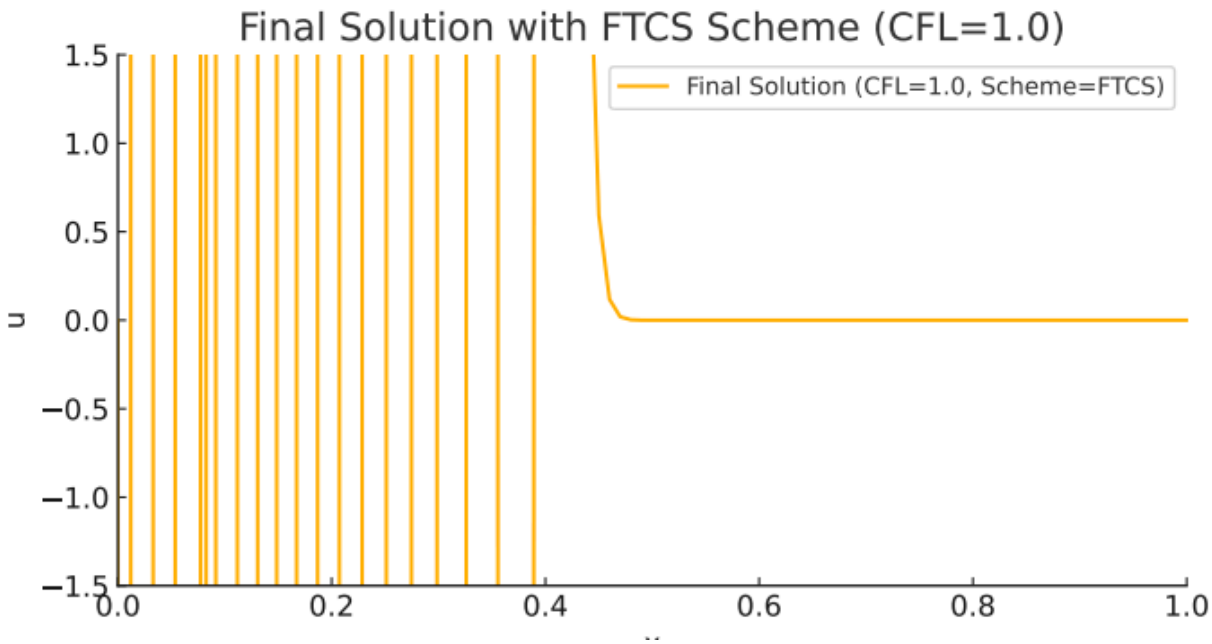


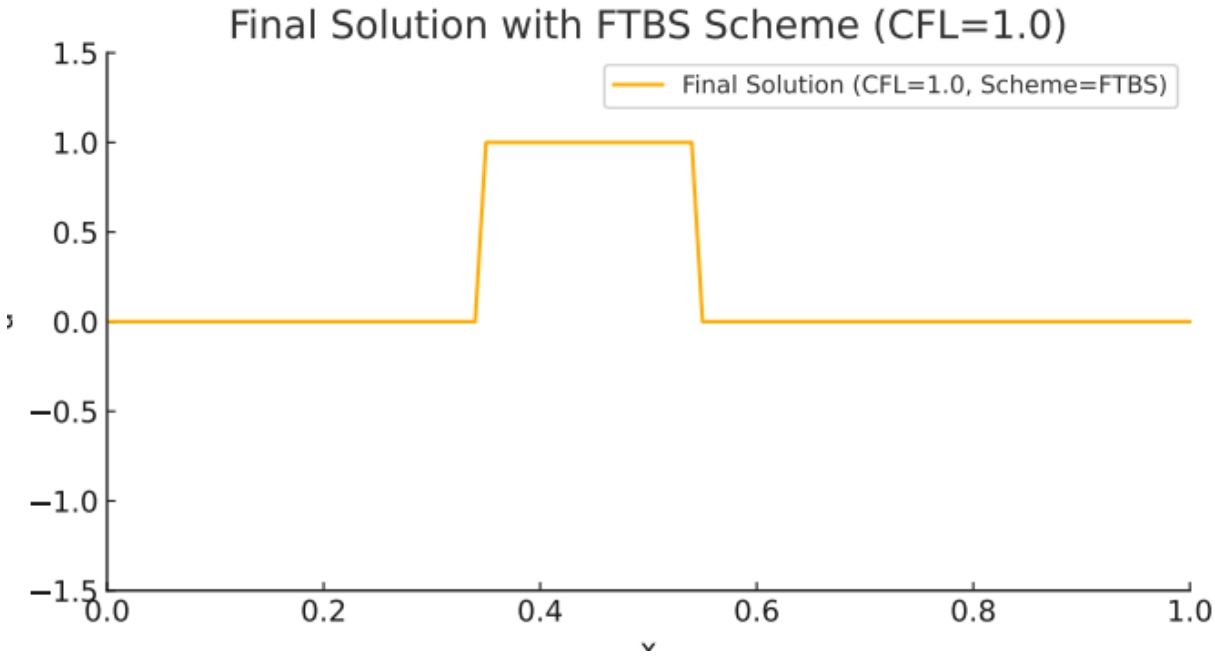

Final Solution with LW Scheme (CFL=0.5)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
Final Solution (CFL=0.5, Scheme=LW)
0.0
0.2
0.4
0.6
0.8
1.0
x

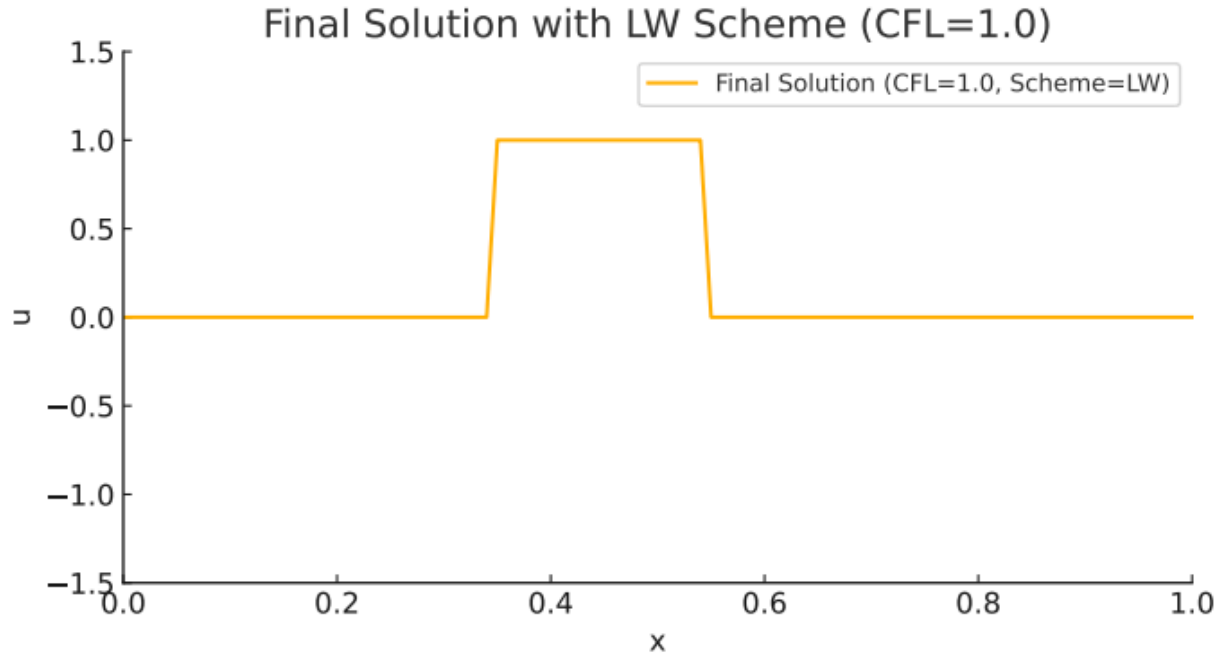

Final Solution with BW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0

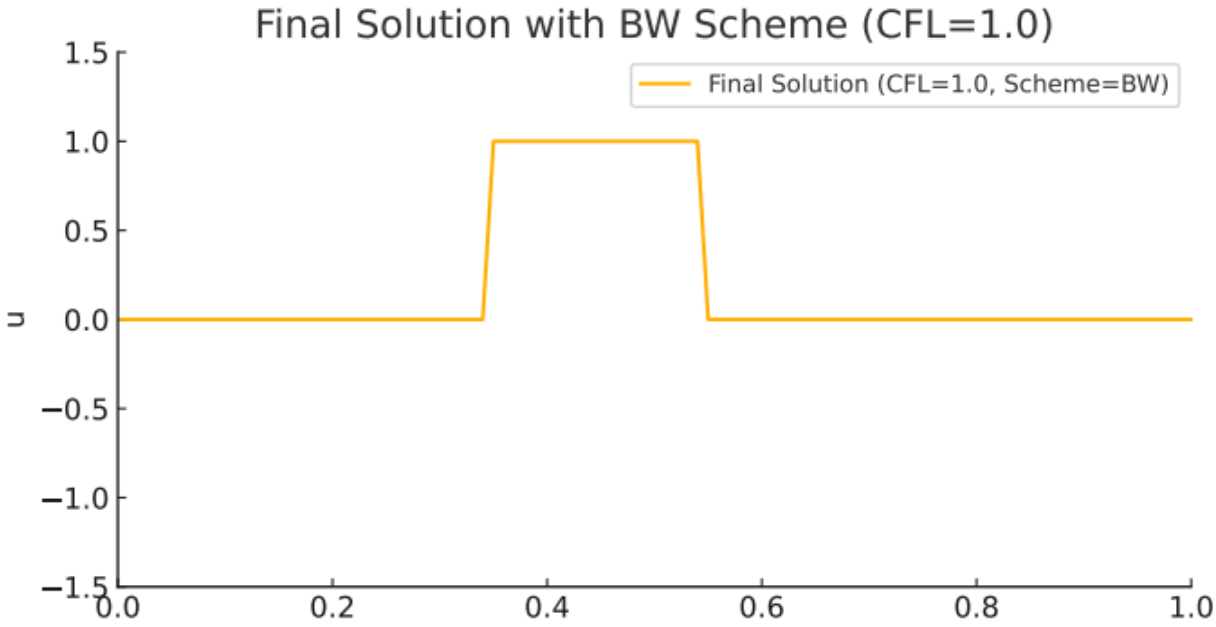

Final Solution with FR Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FR)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

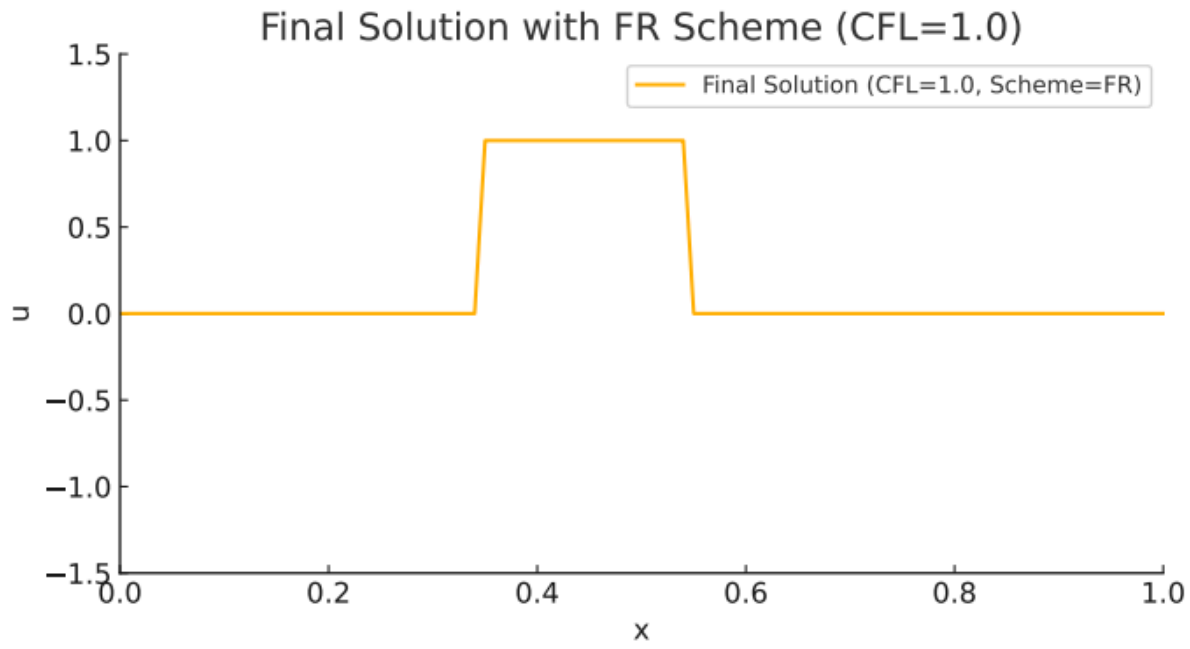

Final Solution with FTFS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTFS)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

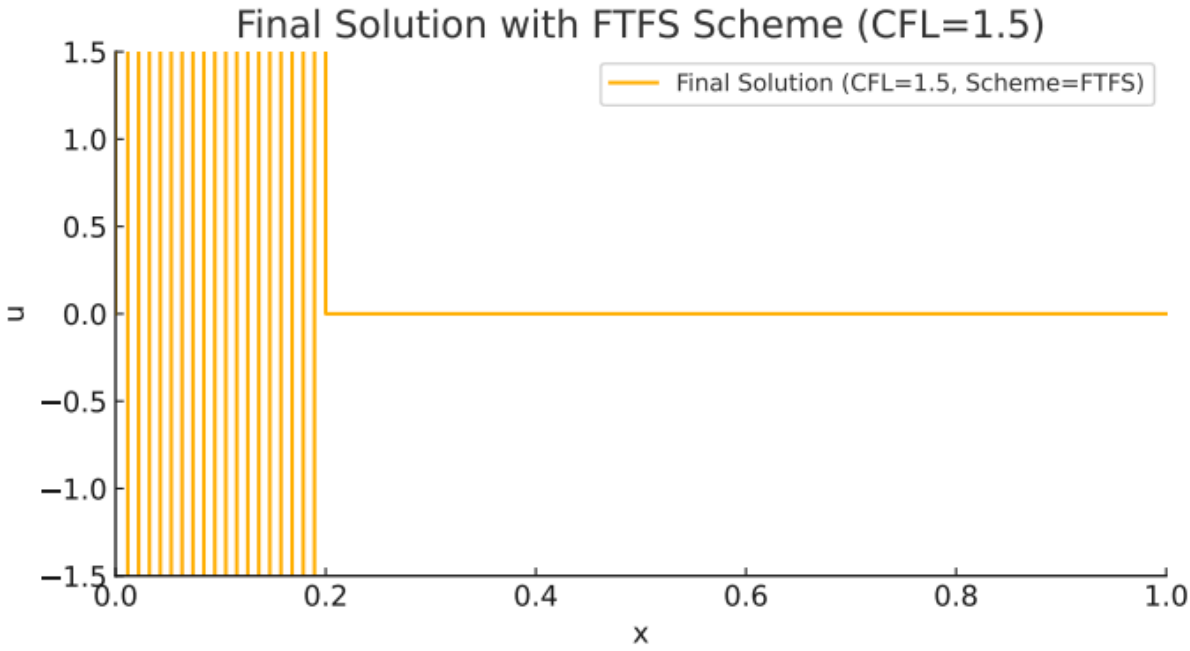

Final Solution with FTCS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

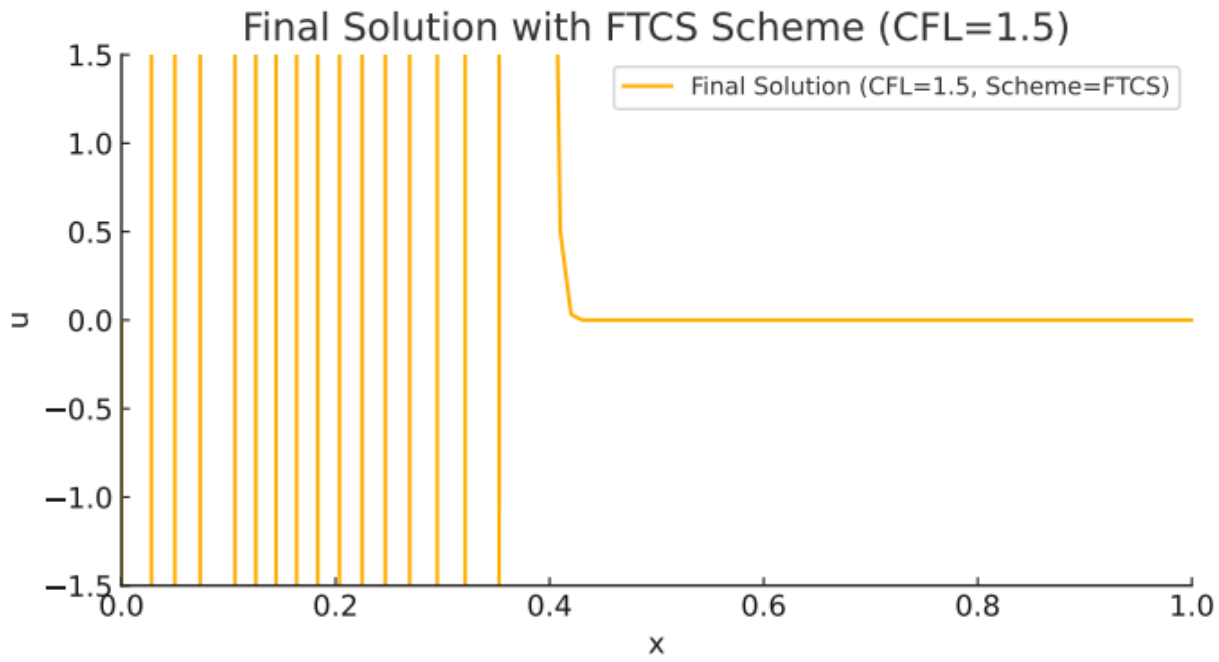

Final Solution with FTBS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

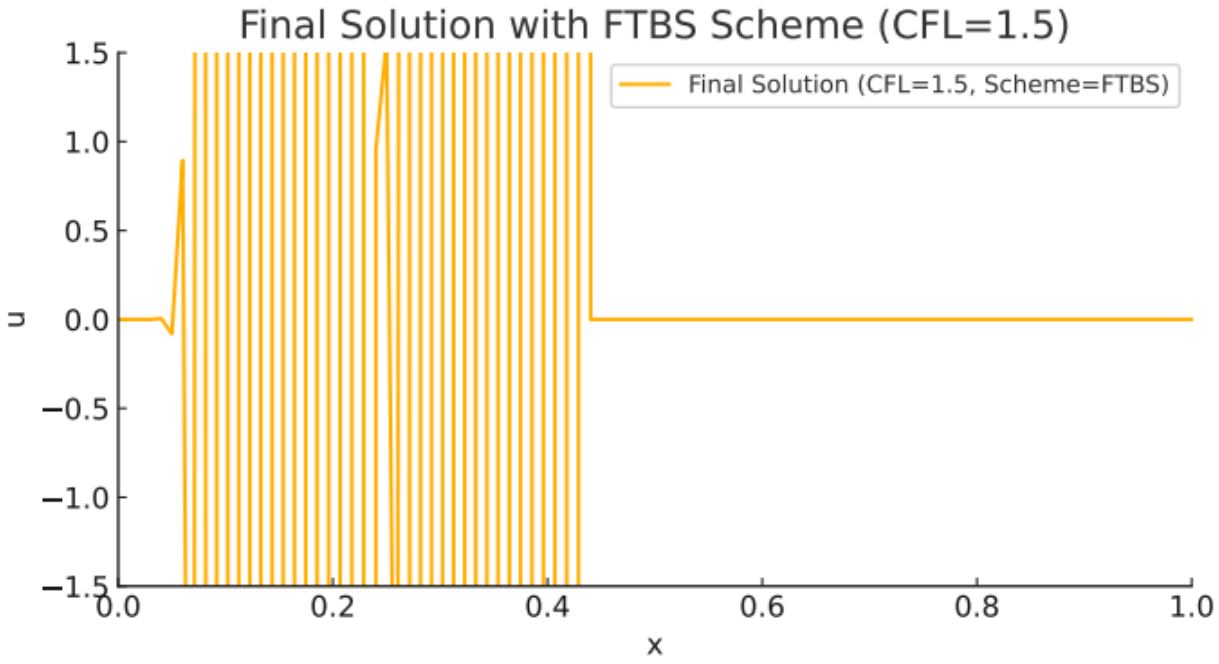

Final Solution with LW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

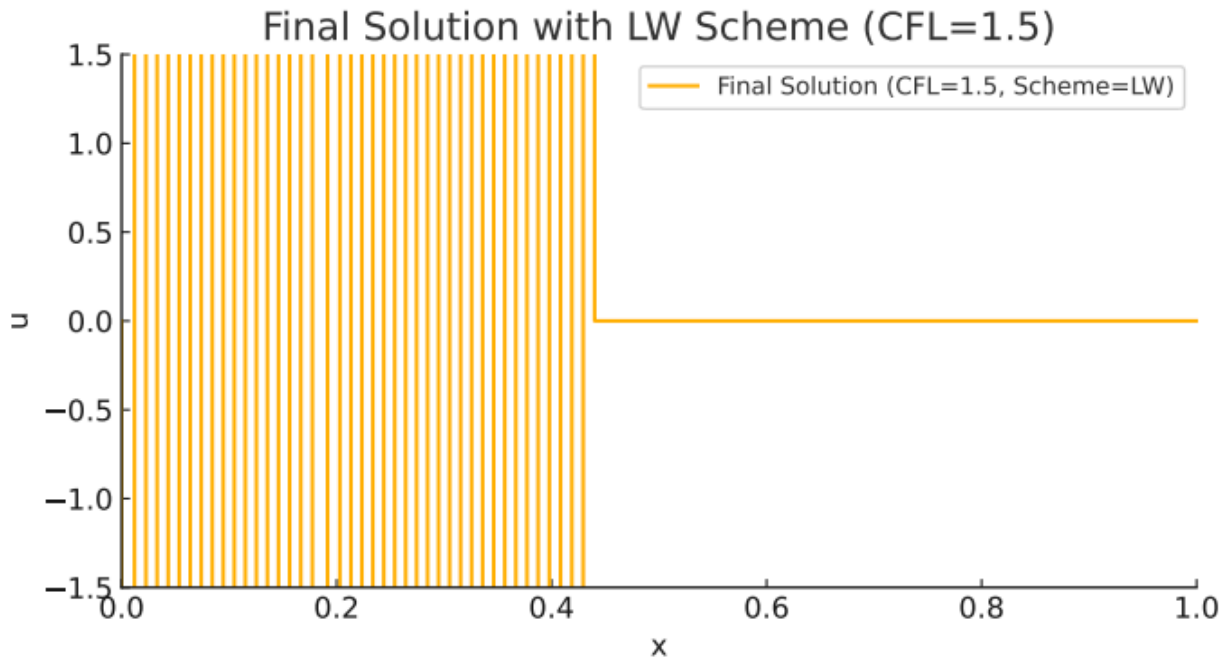

Final Solution with BW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0

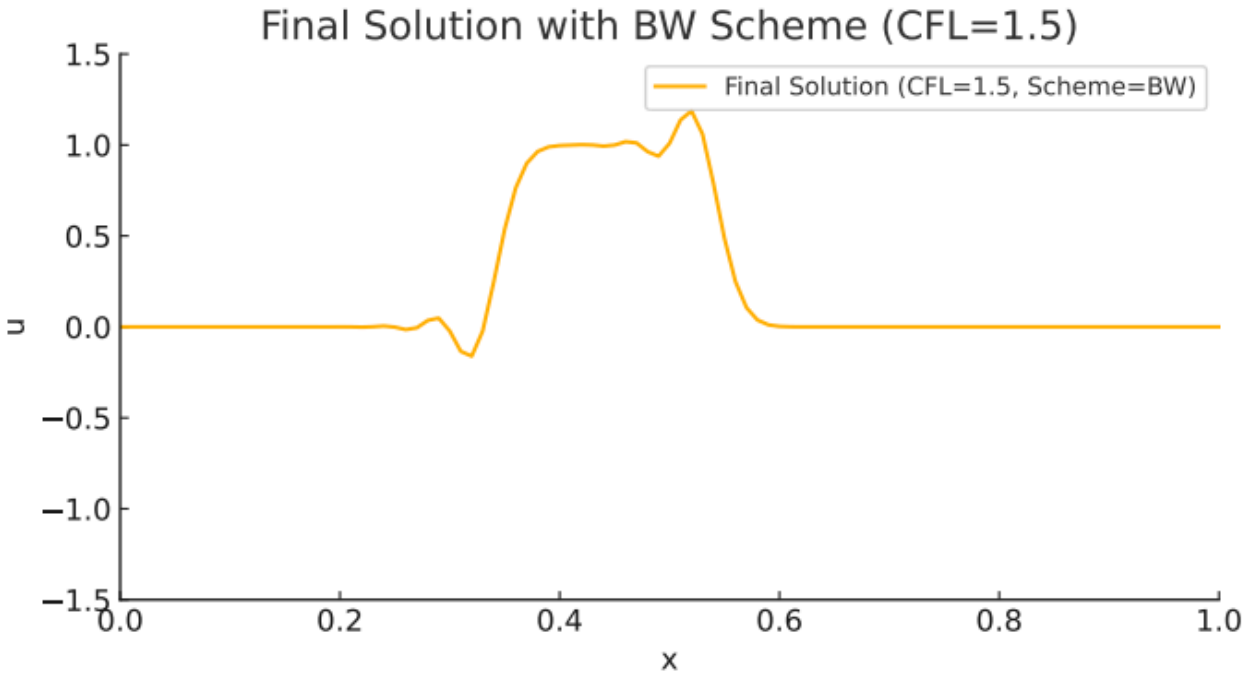

Final Solution with FR Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FR)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

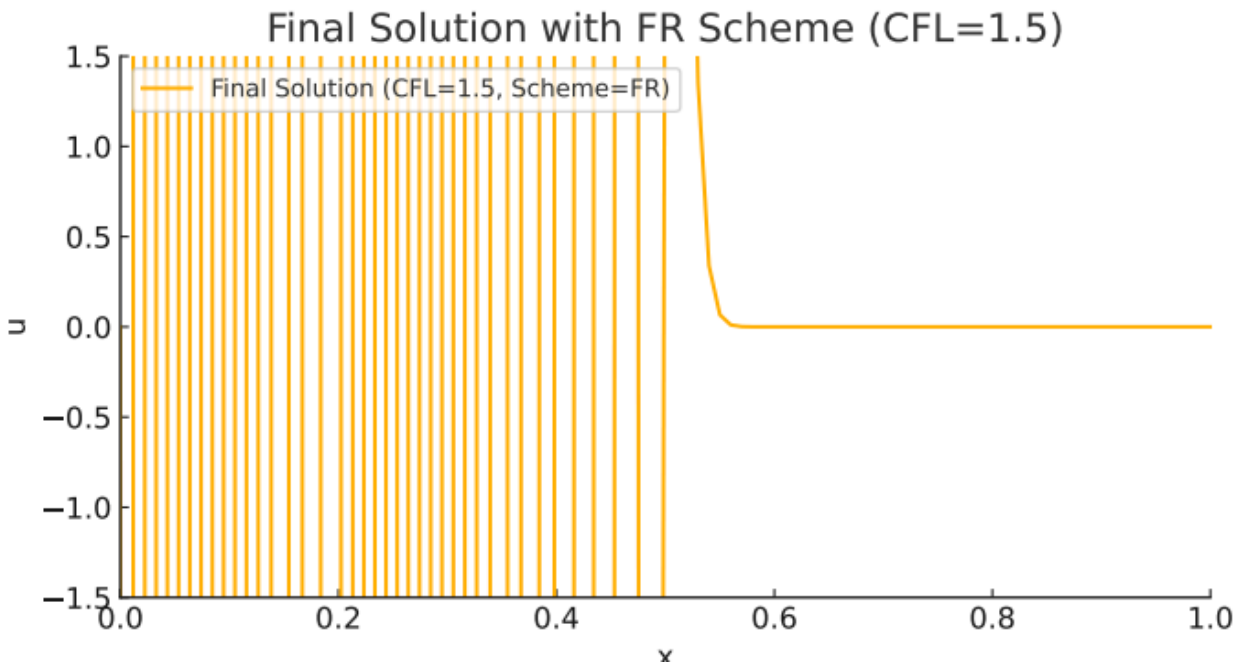

Final Solution with FTFS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTFS)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x


Final Solution with FTCS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

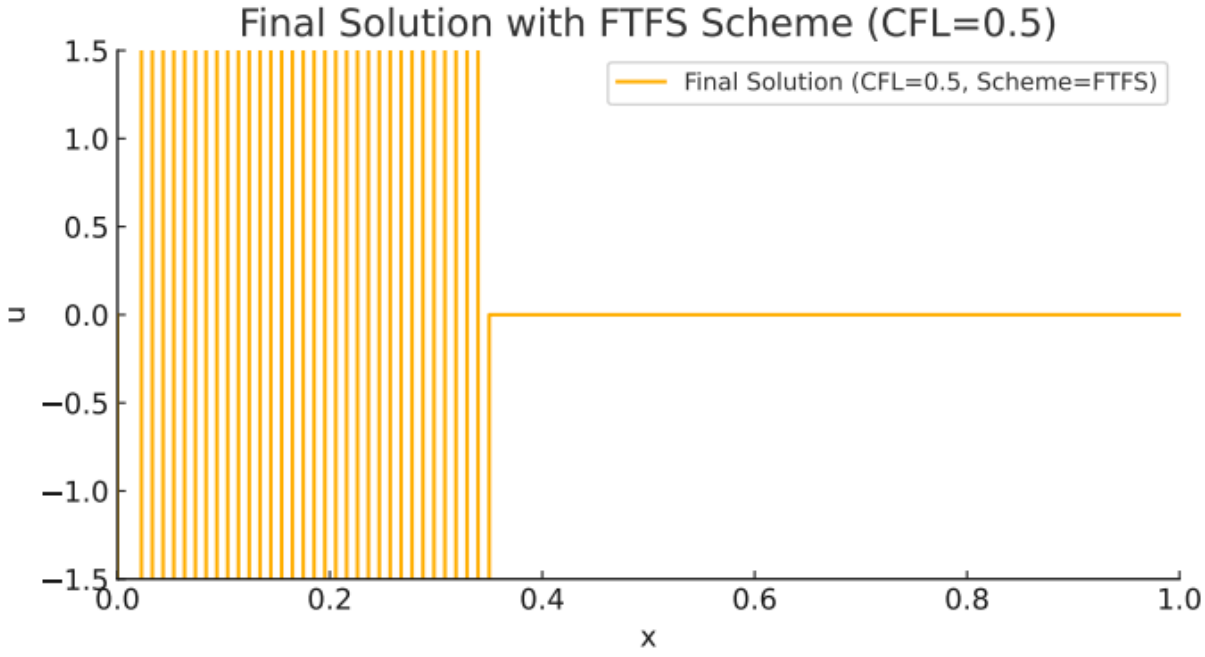

Final Solution with FTBS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

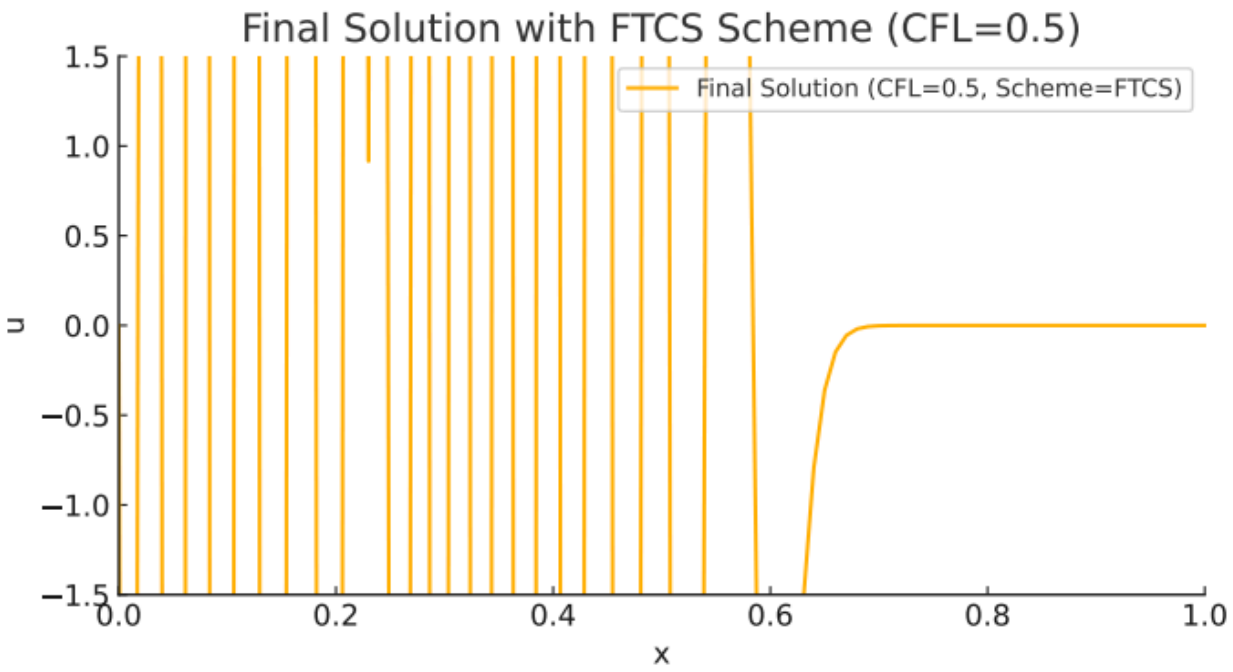

Final Solution with LW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

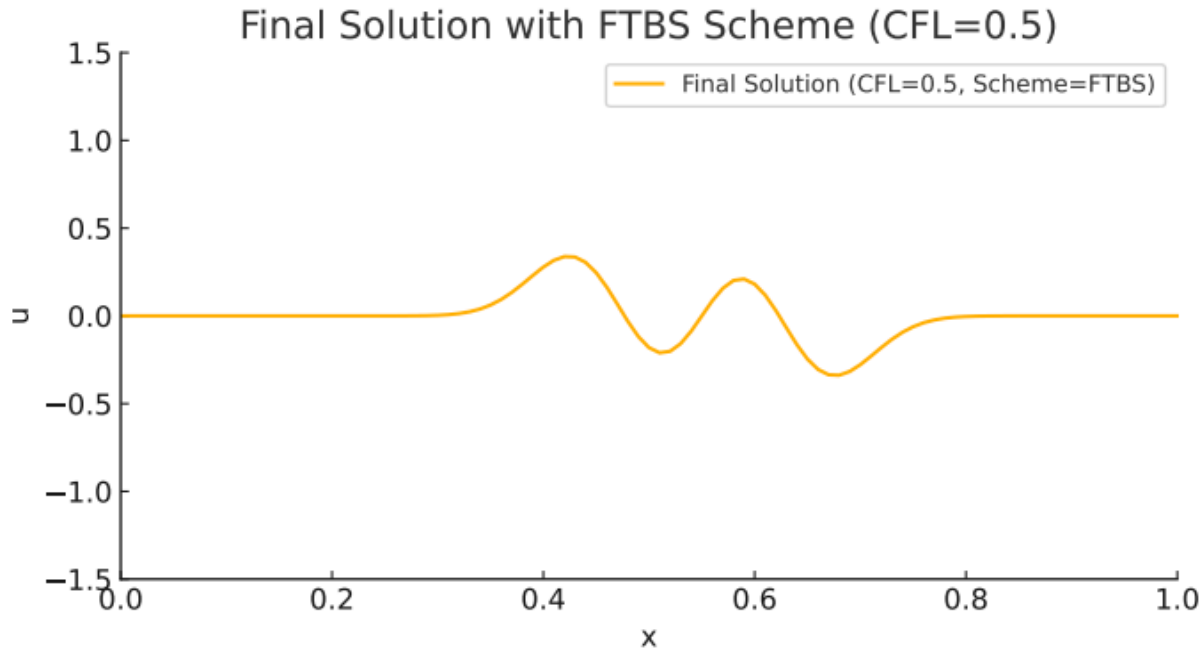

Final Solution with BW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

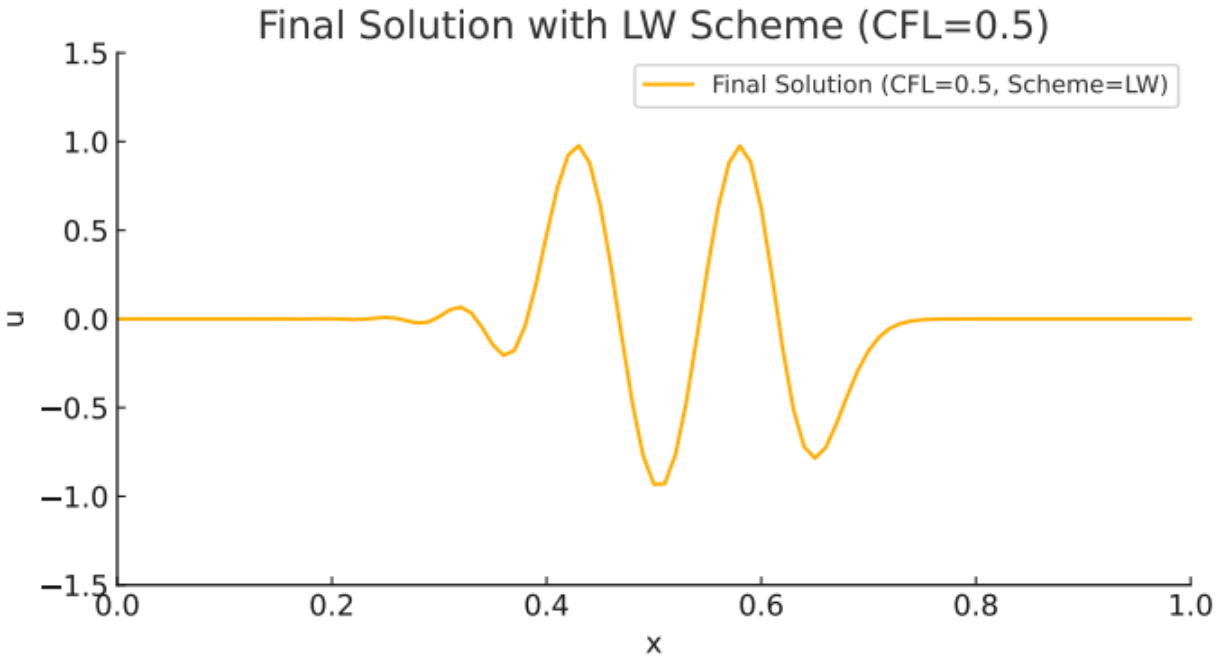


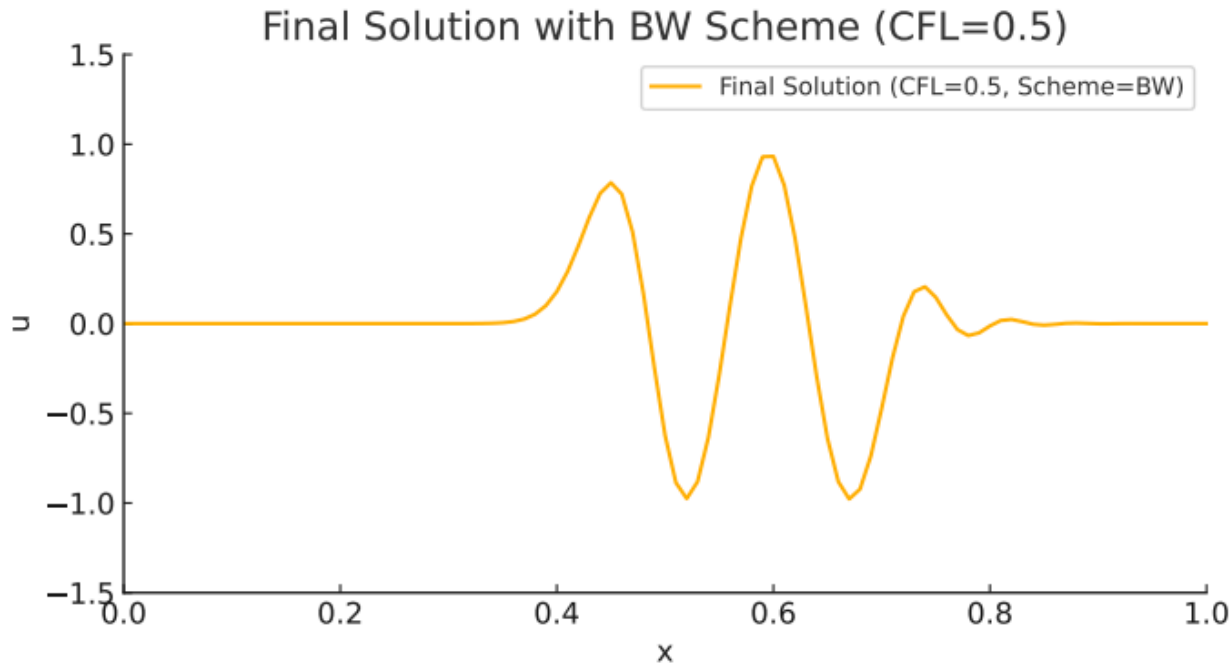
b) For 2nd Condition

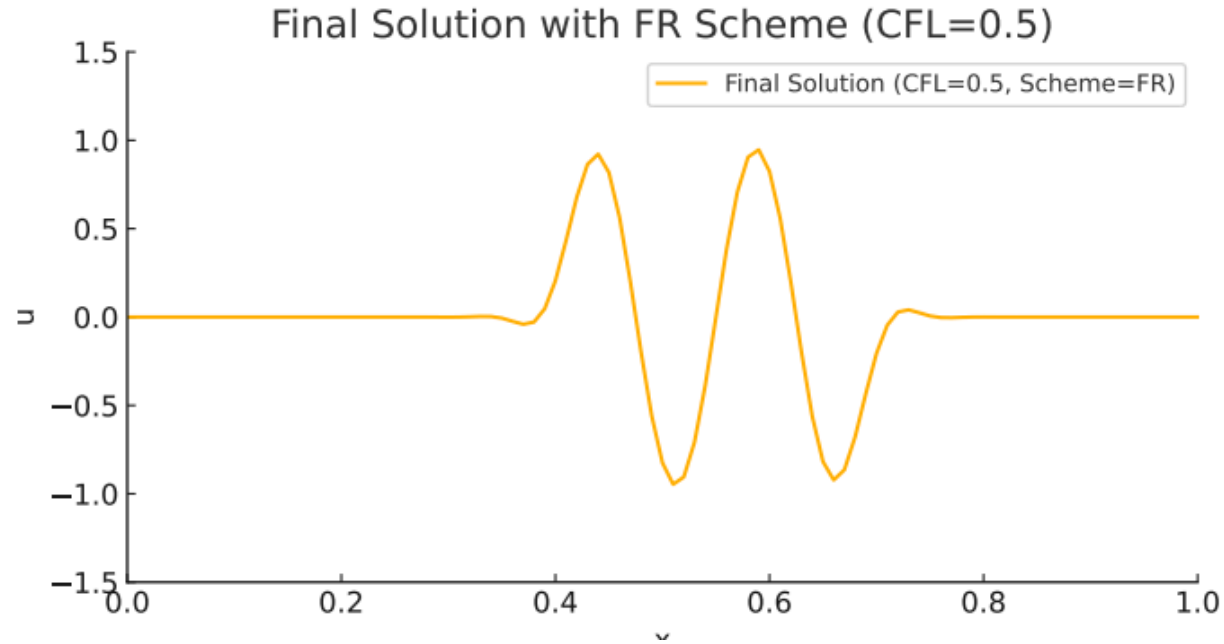


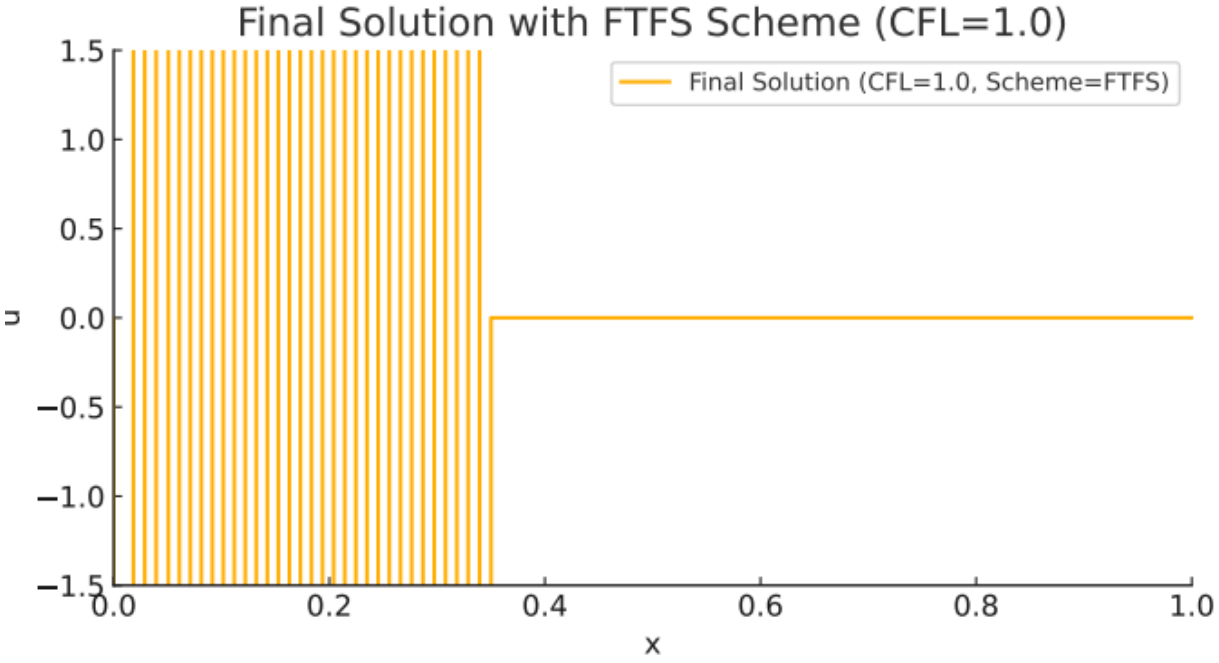

Final Solution with FTCS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

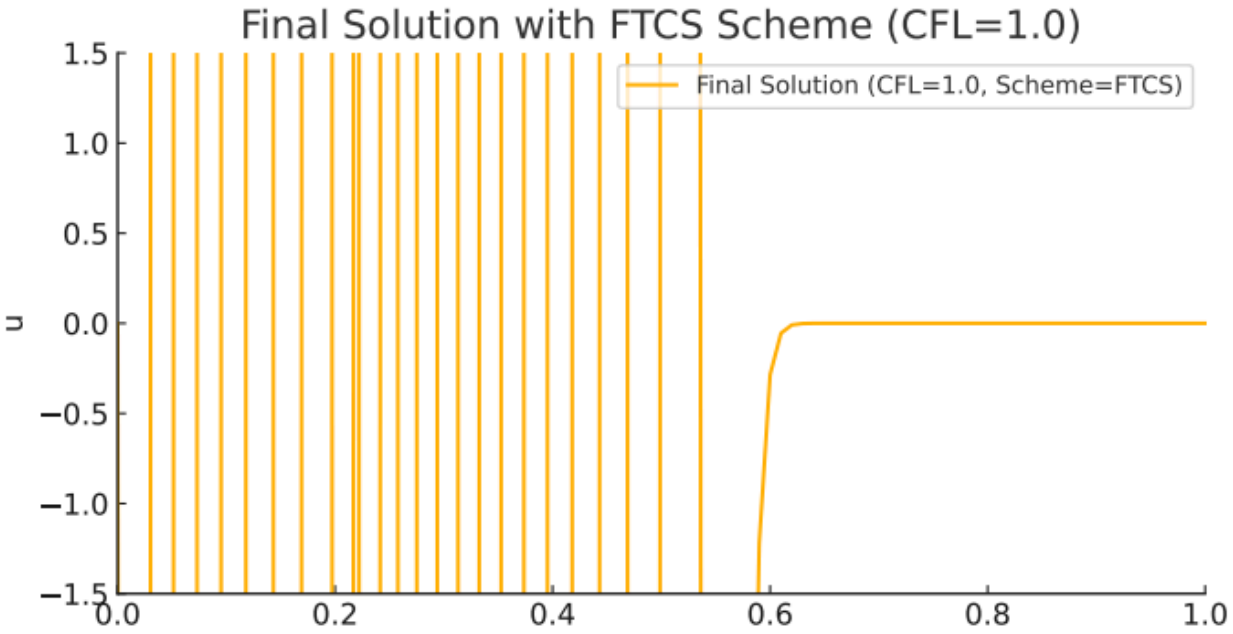

Final Solution with FTBS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

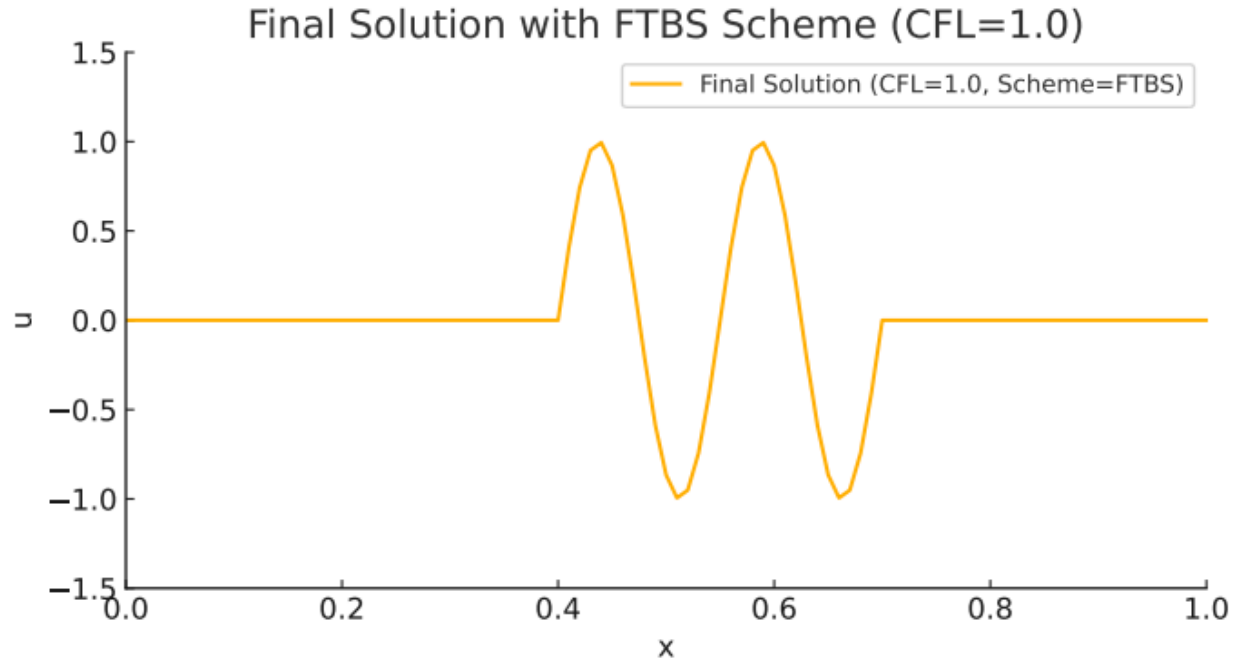

Final Solution with LW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

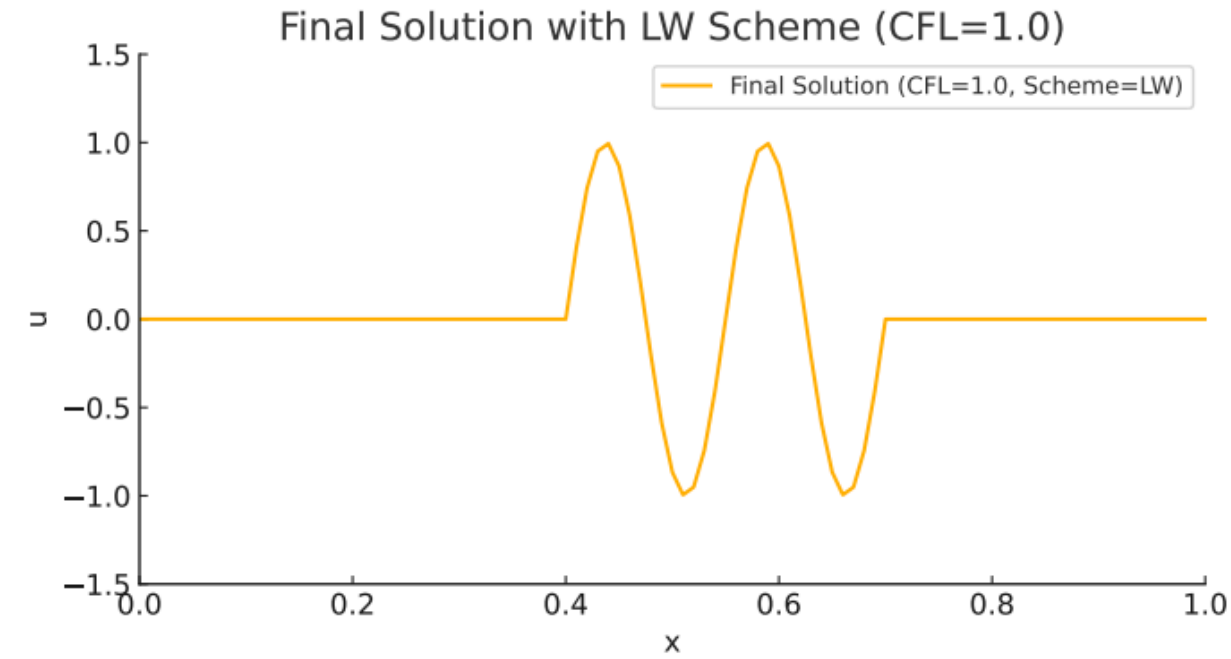

Final Solution with BW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

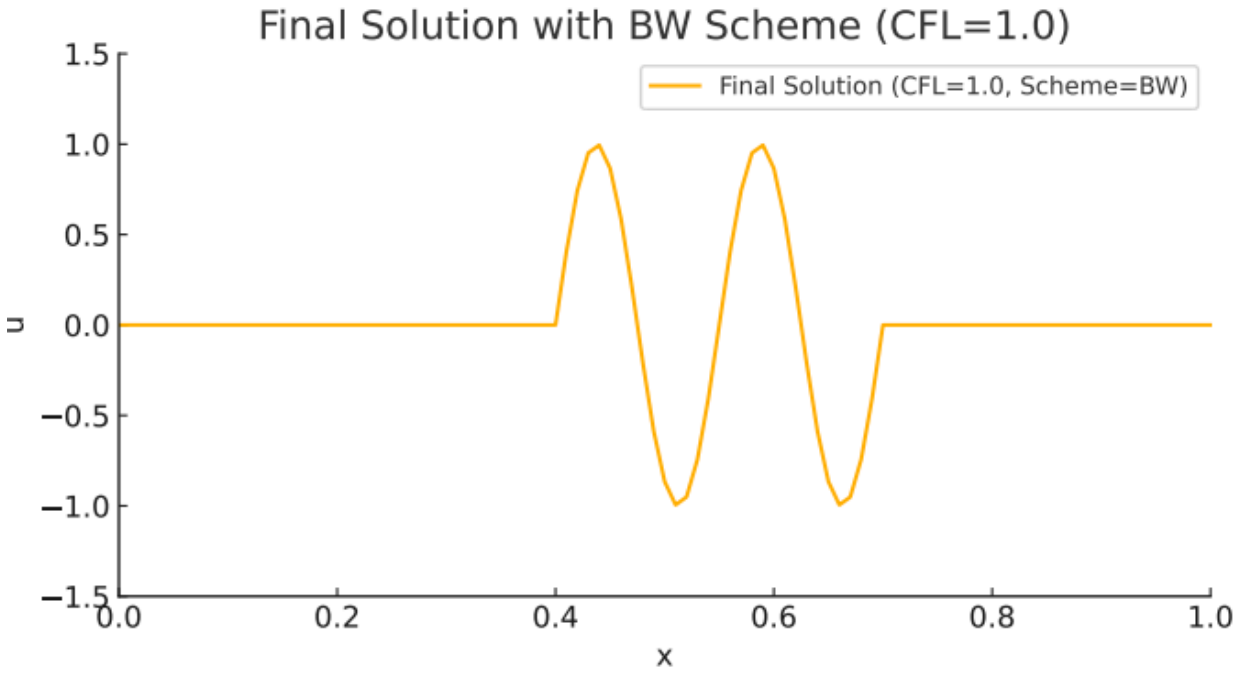

Final Solution with FR Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FR)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

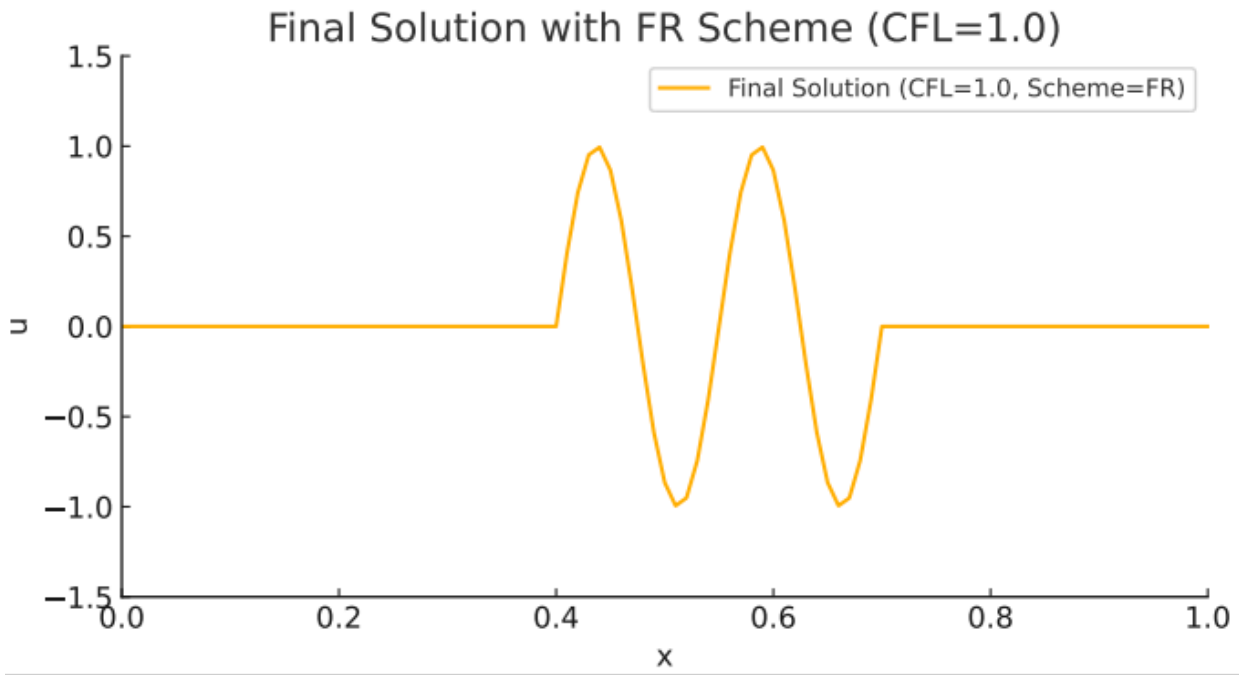

Final Solution with FTFS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTFS)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0
x

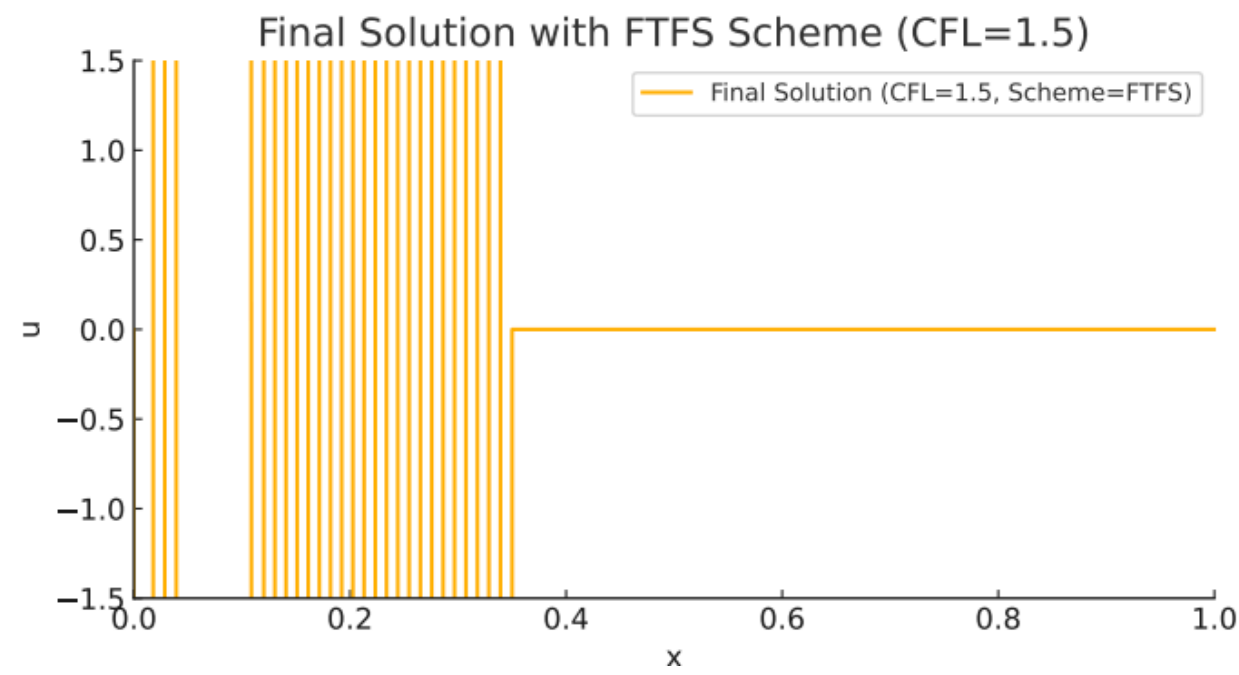

Final Solution with FTCS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTCS)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

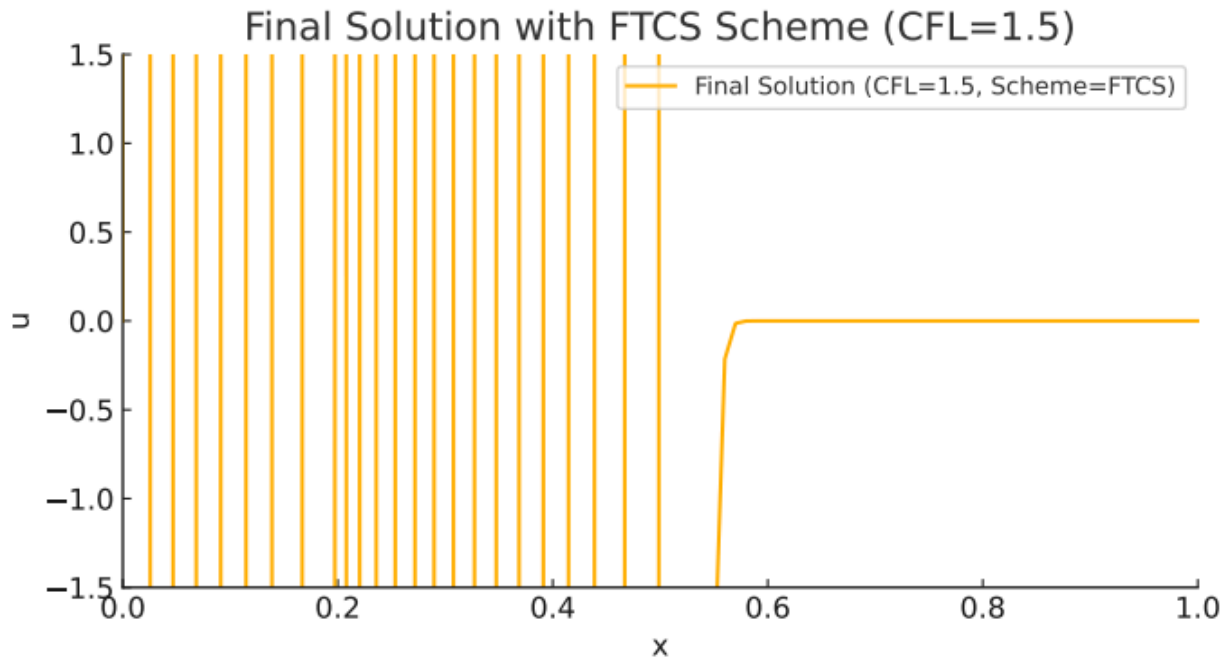

Final Solution with FTBS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTBS)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0
x

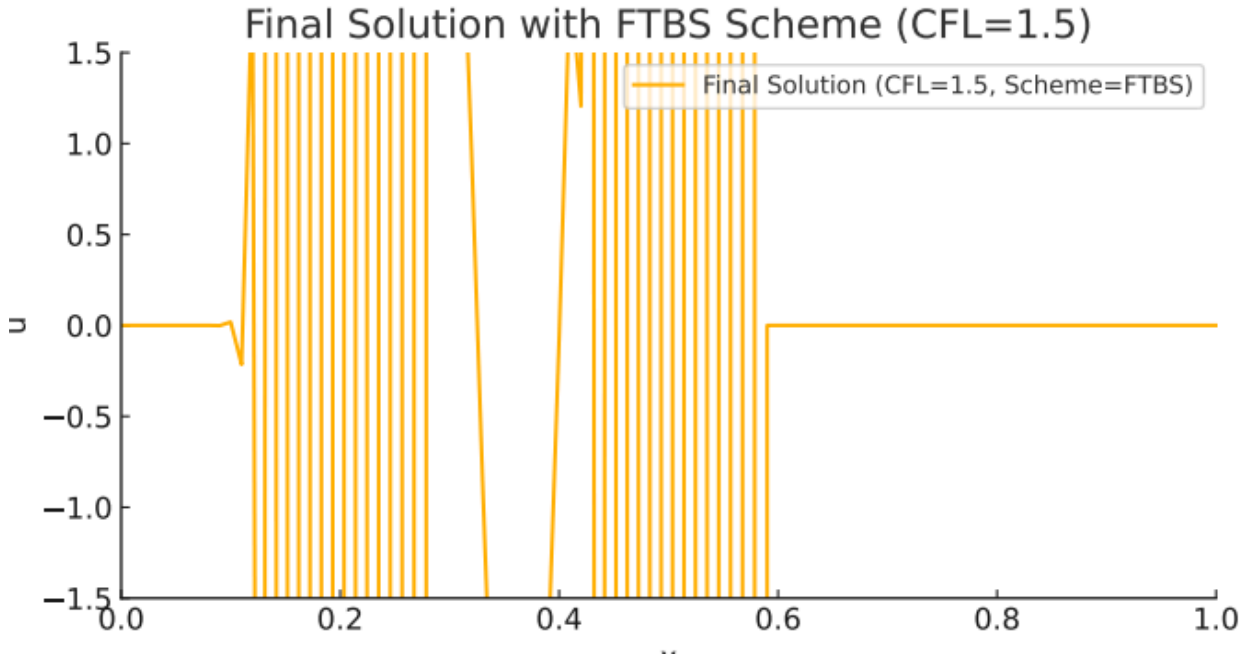

Final Solution with LW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

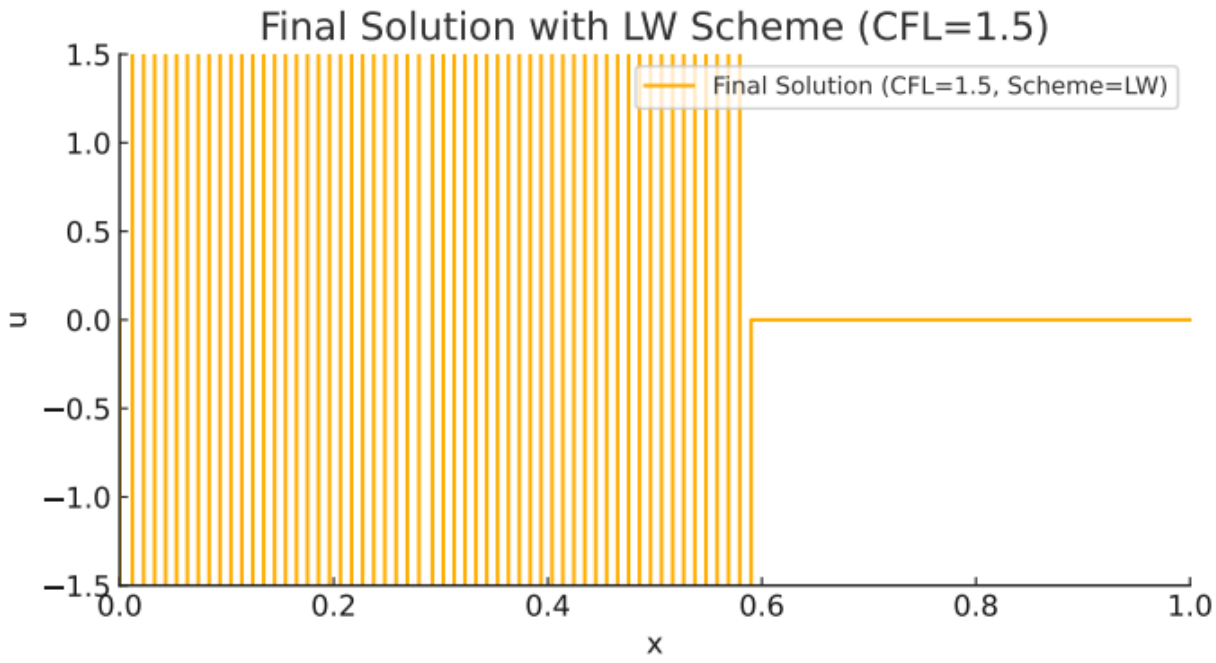

Final Solution with BW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

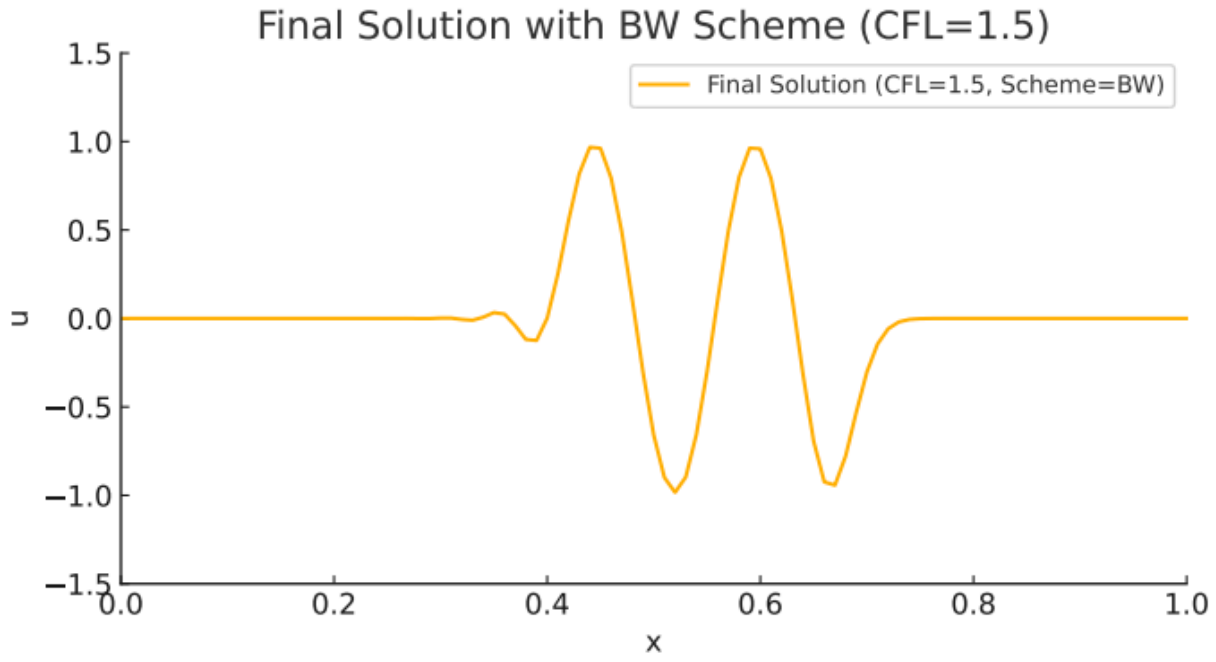

Final Solution with FR Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FR)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

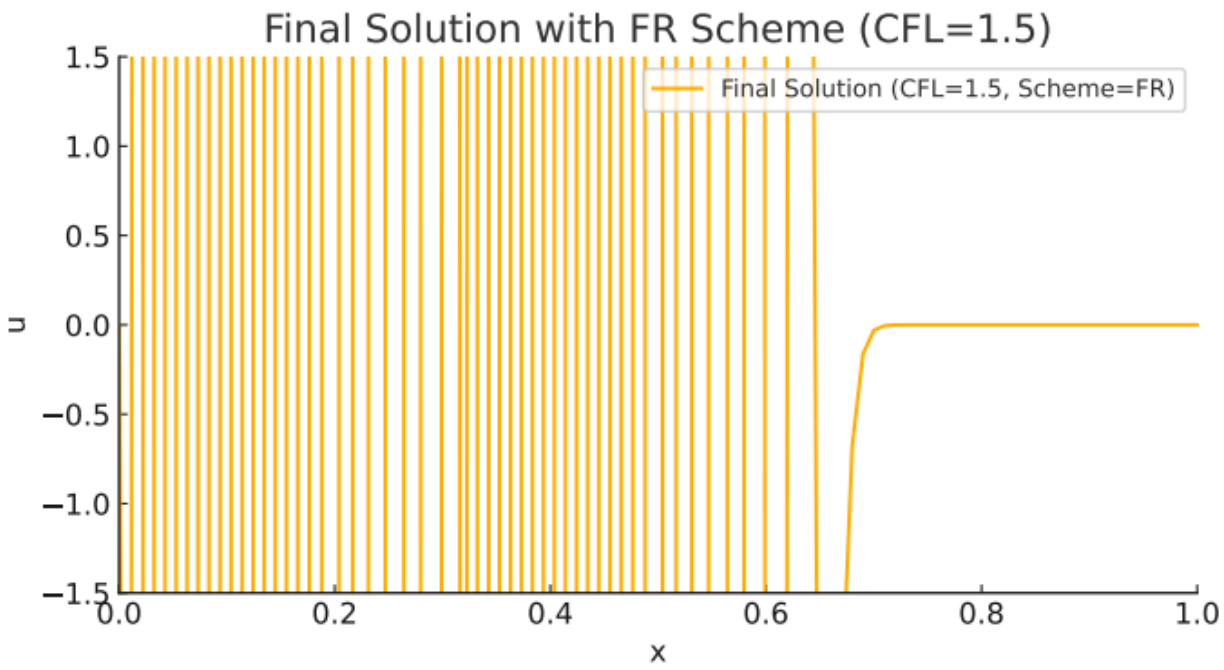

Final Solution with FTFS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTFS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0


Final Solution with FTCS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

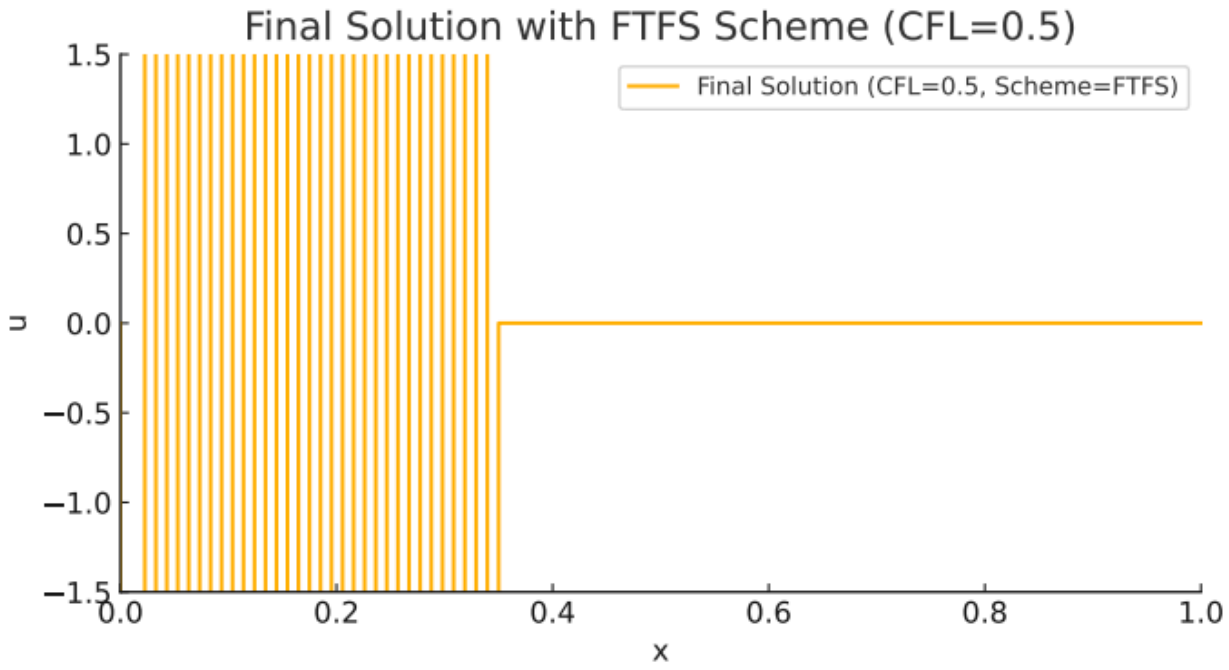

Final Solution with FTBS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTBS)
u
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

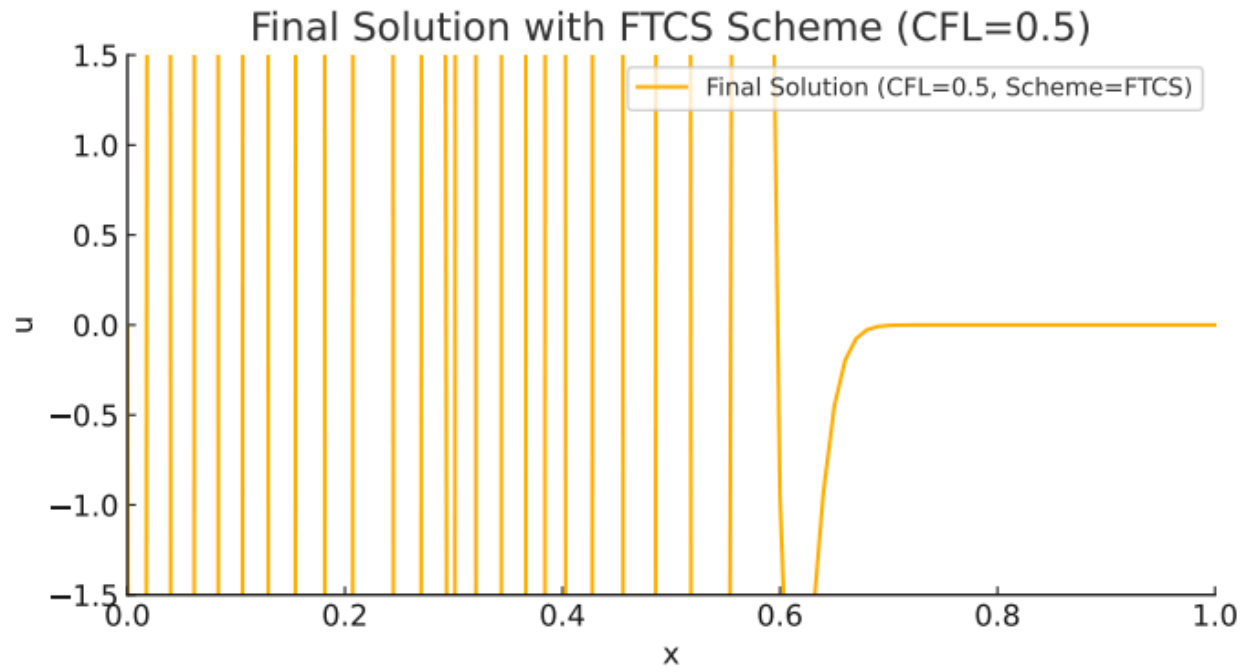

Final Solution with LW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

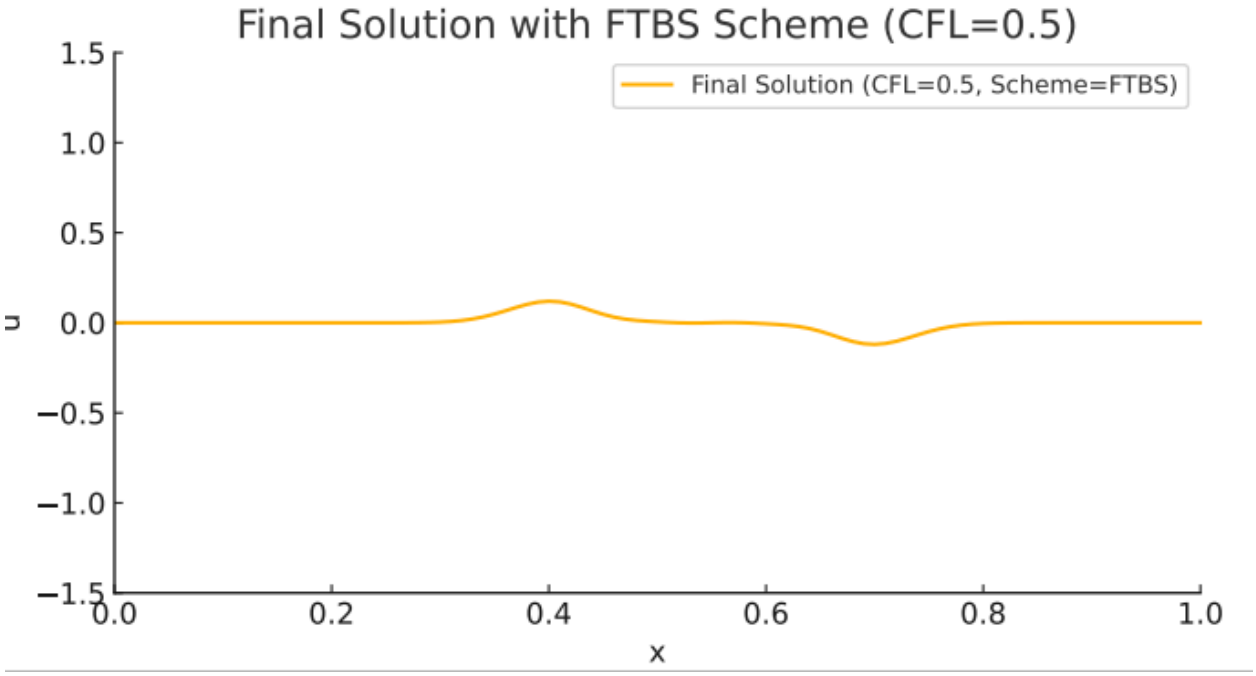

Final Solution with BW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

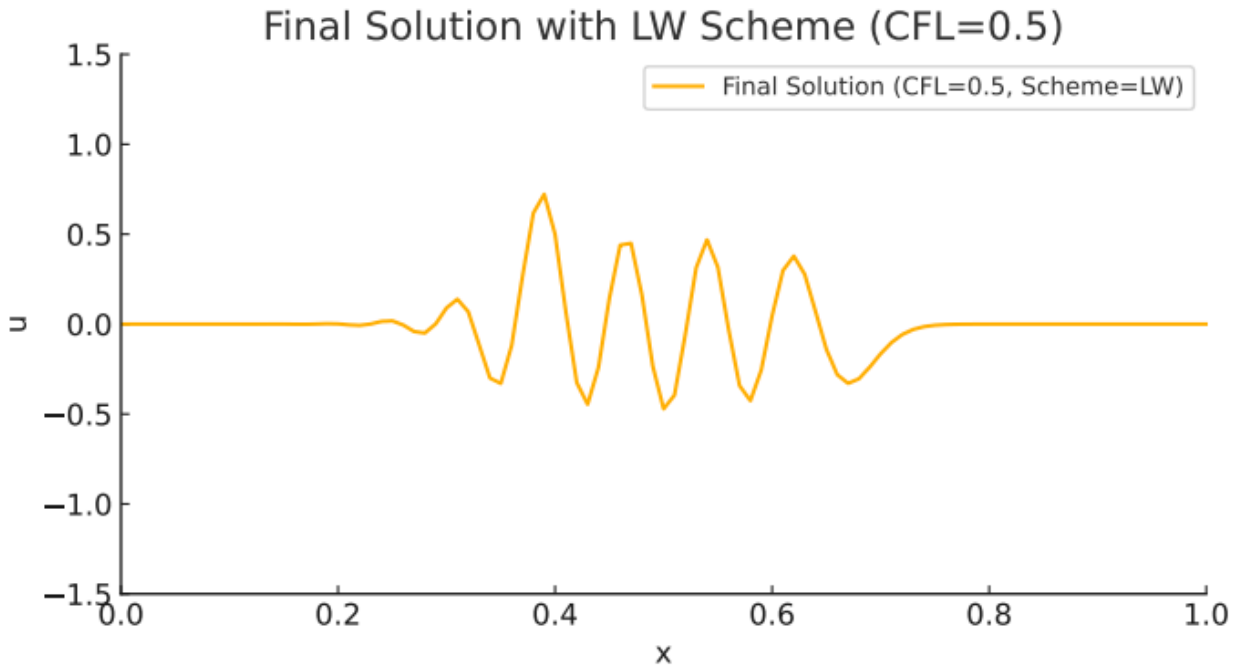


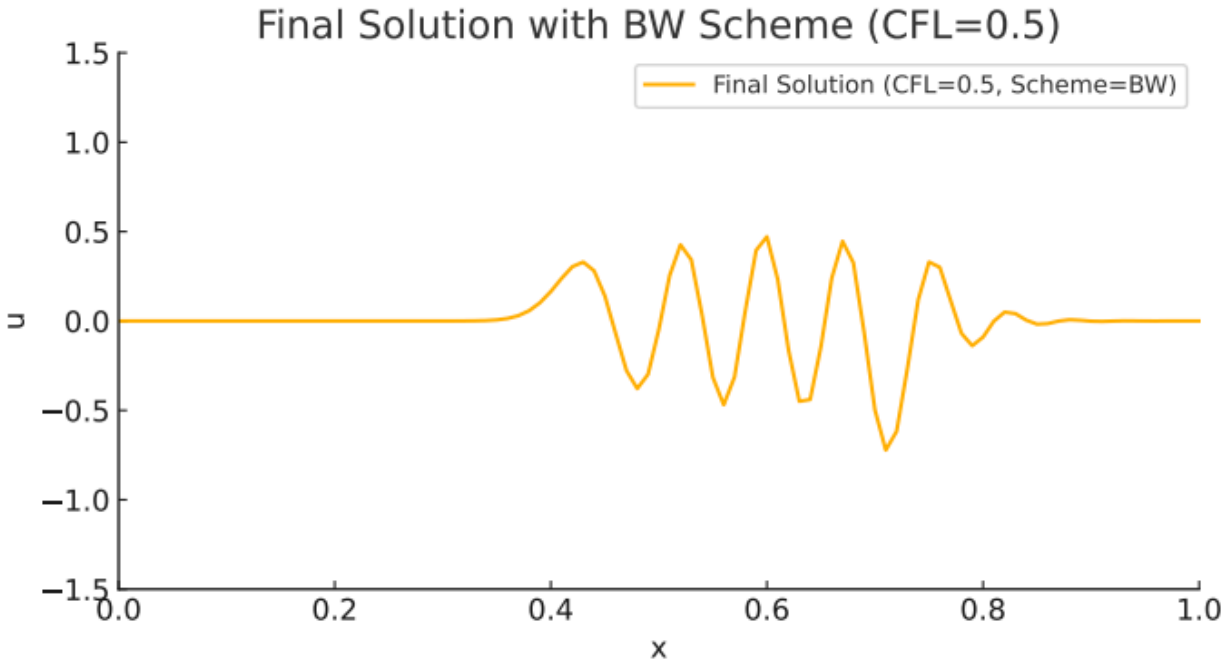
c) For 3rd Condition

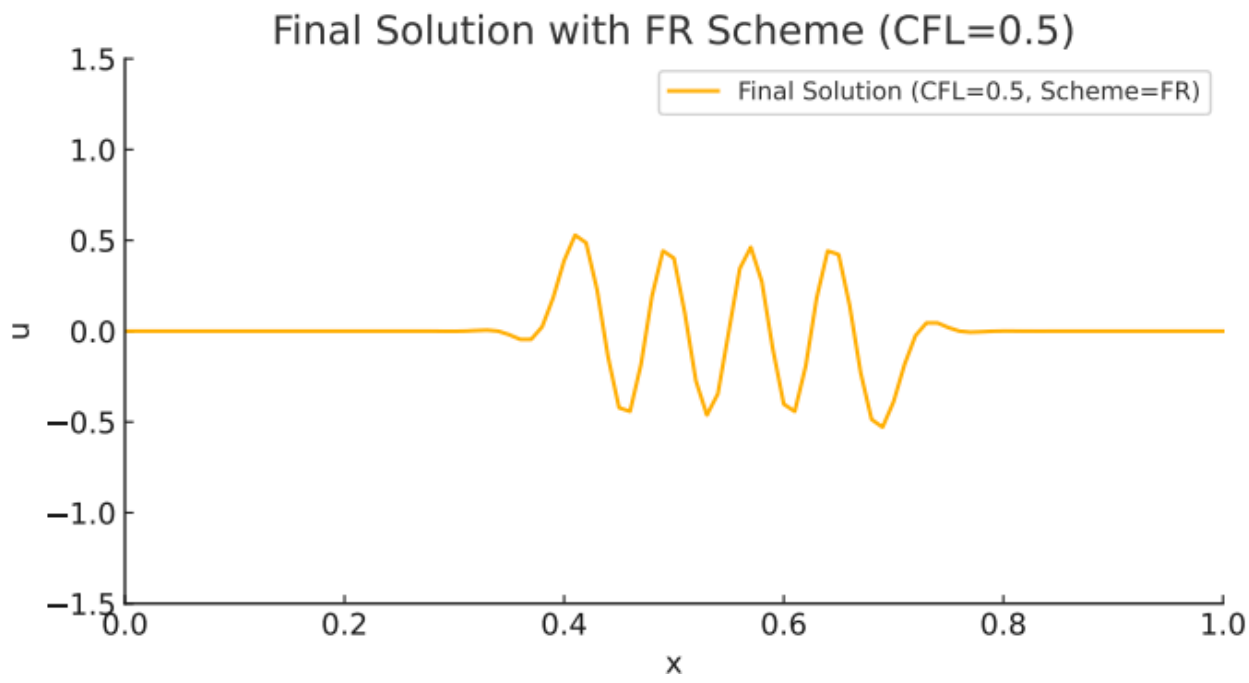


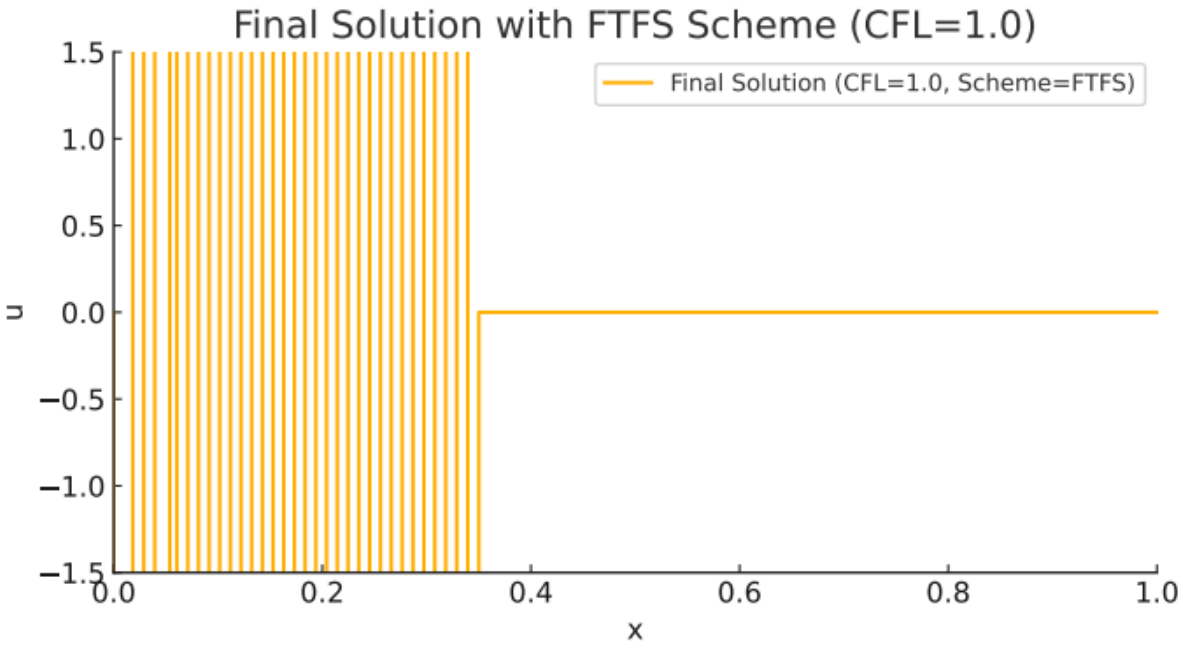

Final Solution with FTCS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

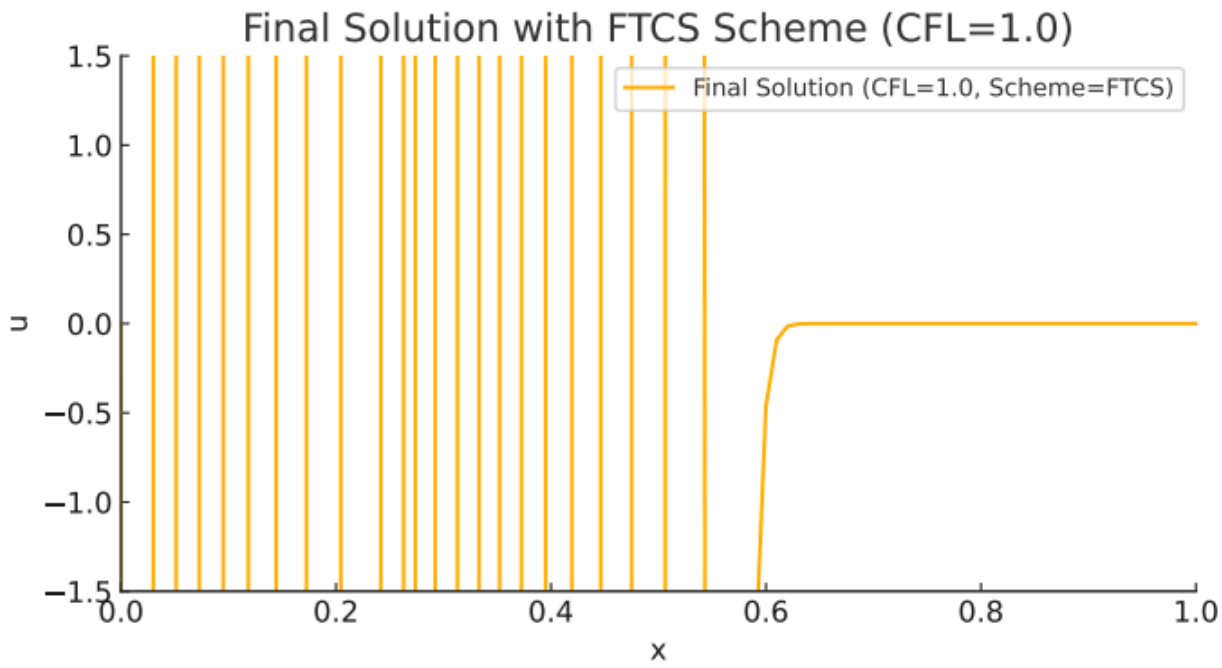

Final Solution with FTBS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

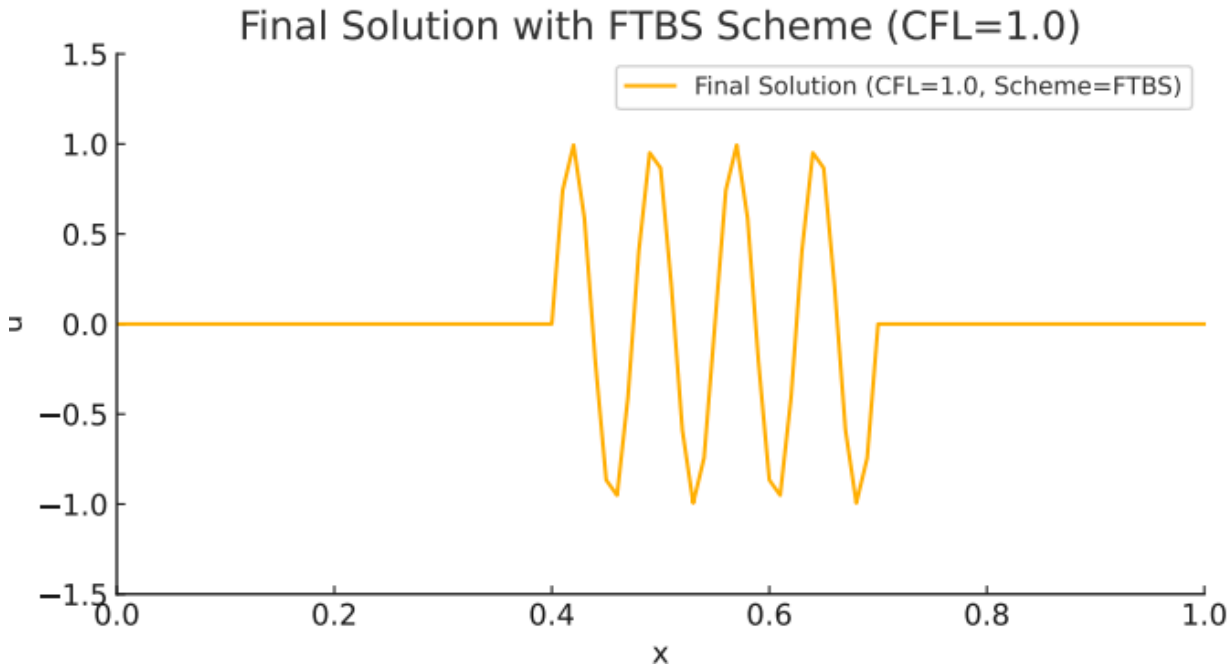

Final Solution with LW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

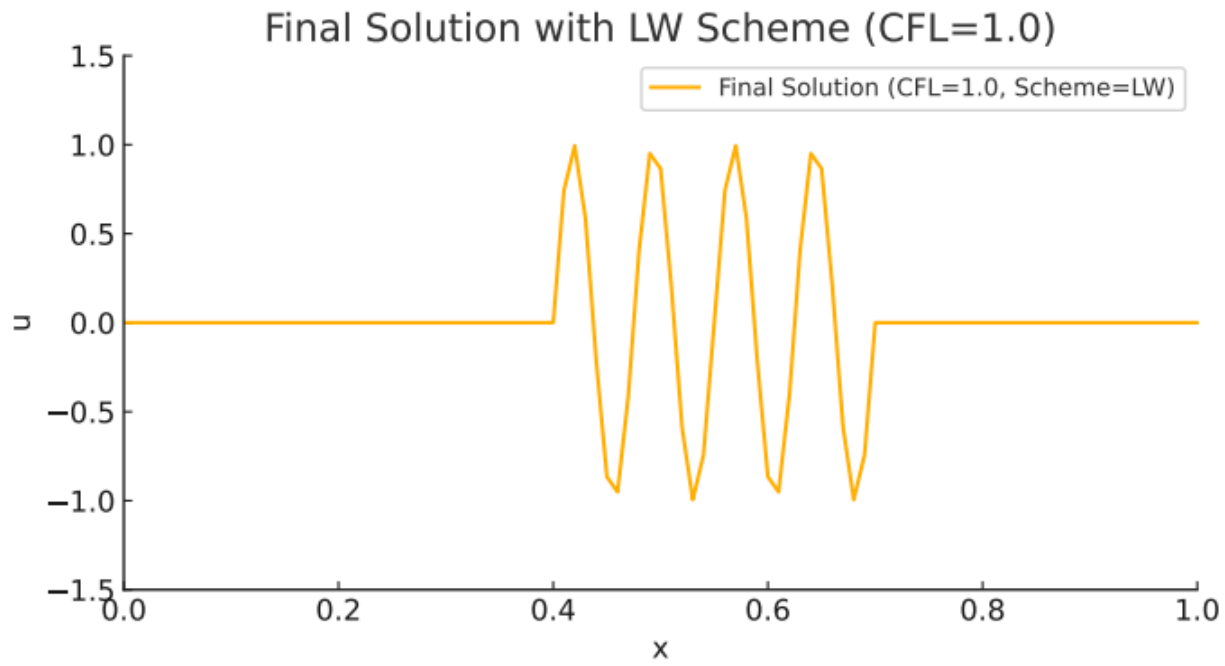

Final Solution with BW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

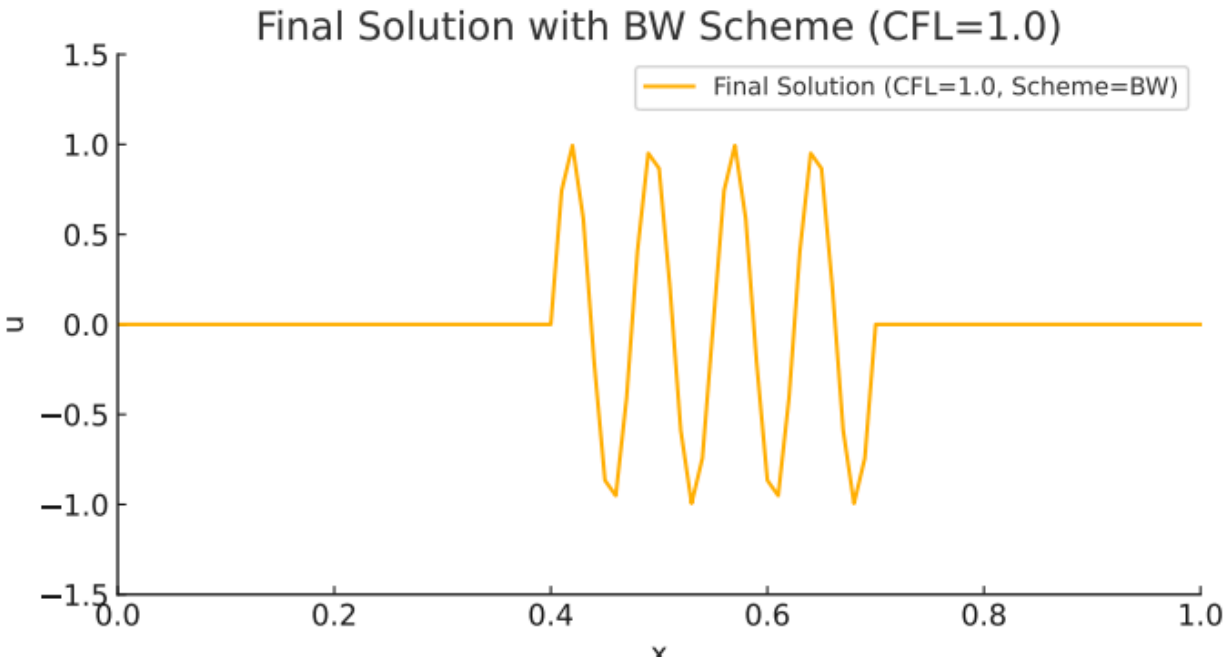

Final Solution with FR Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FR)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

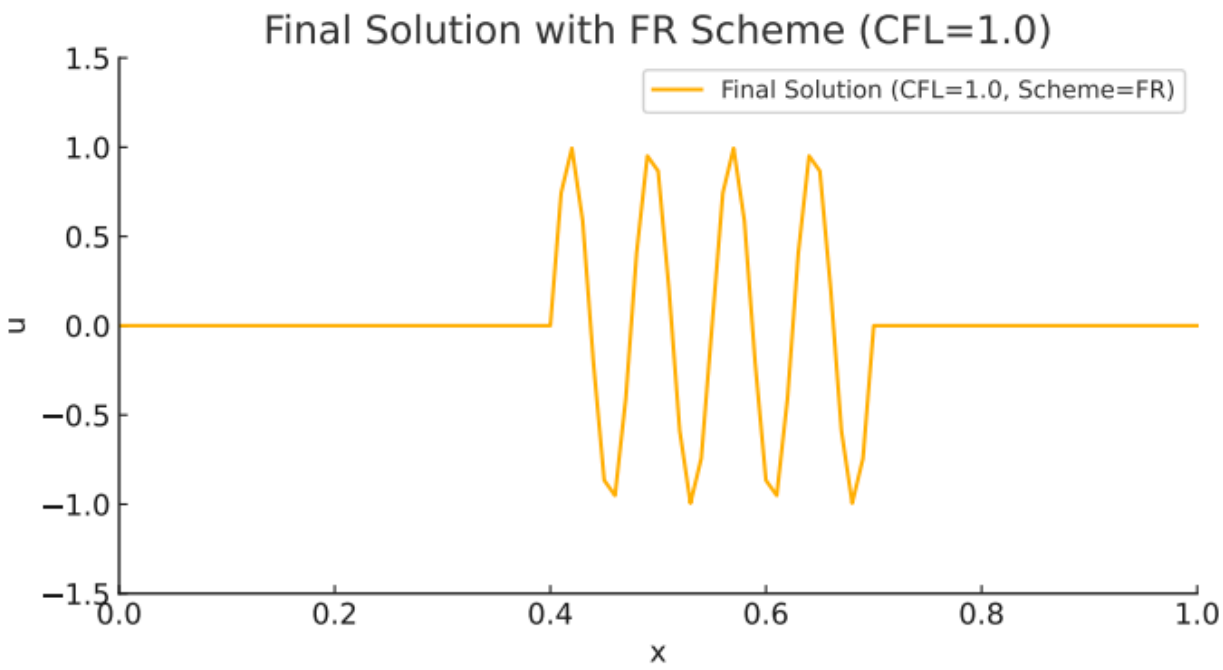

Final Solution with FTFS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTFS)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

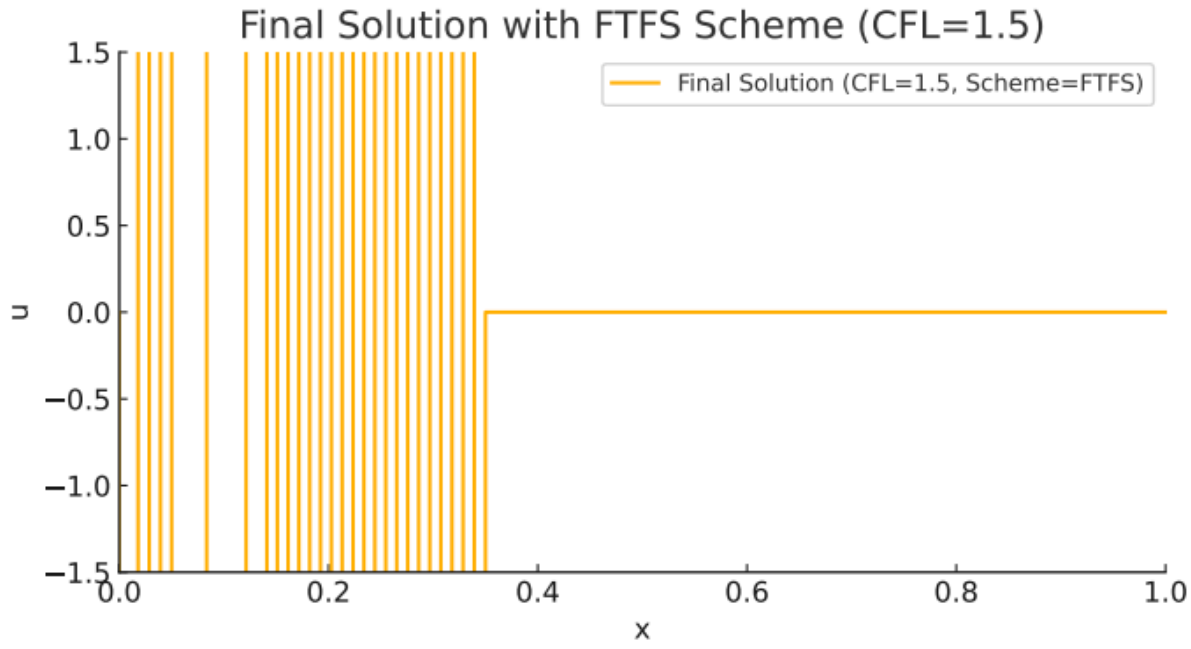

Final Solution with FTCS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

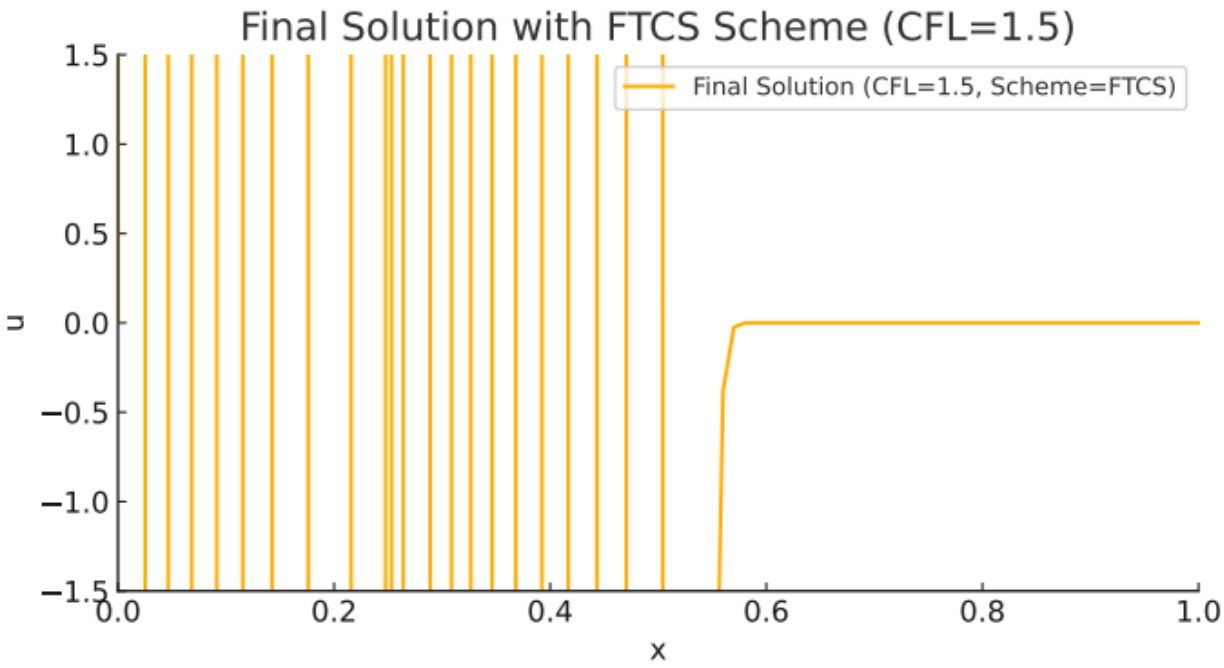

Final Solution with FTBS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

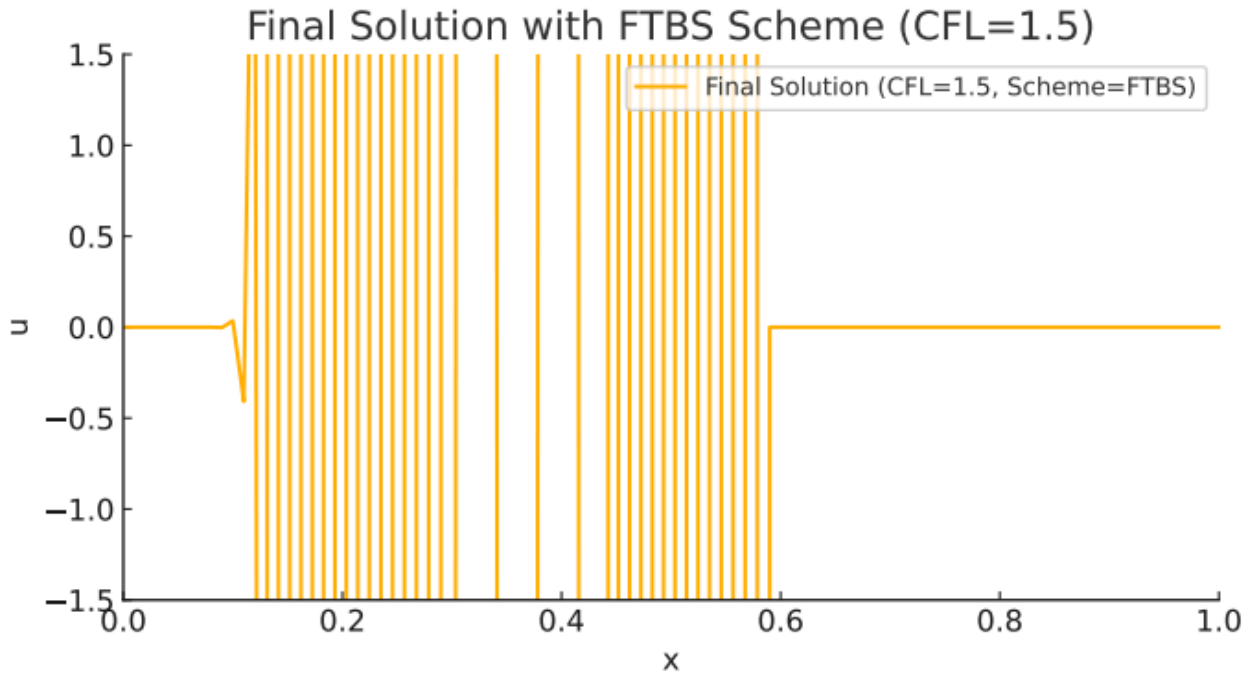

Final Solution with LW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

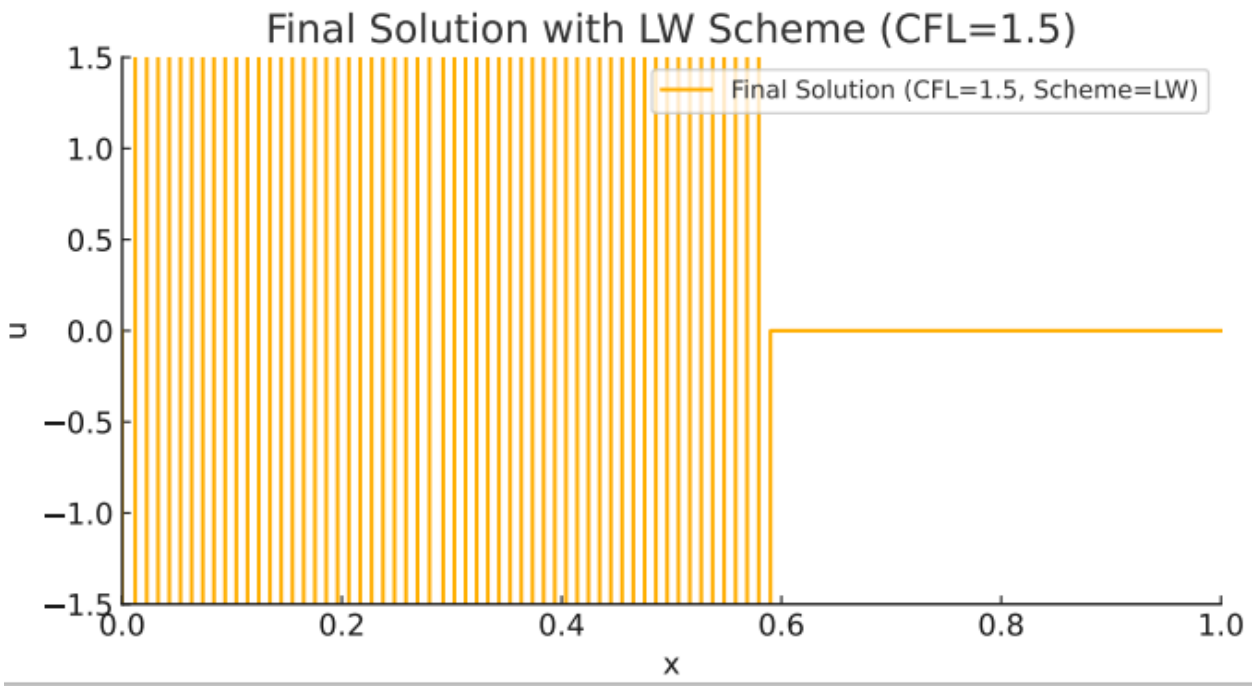

Final Solution with BW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

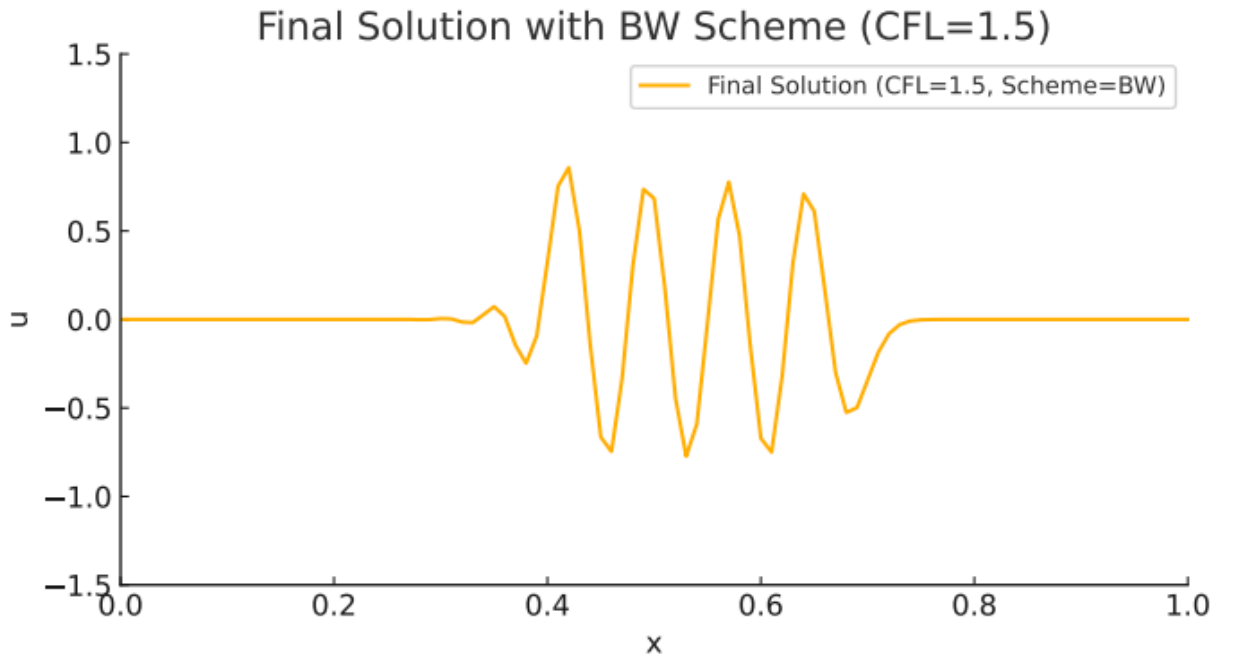

Final Solution with FR Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FR)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

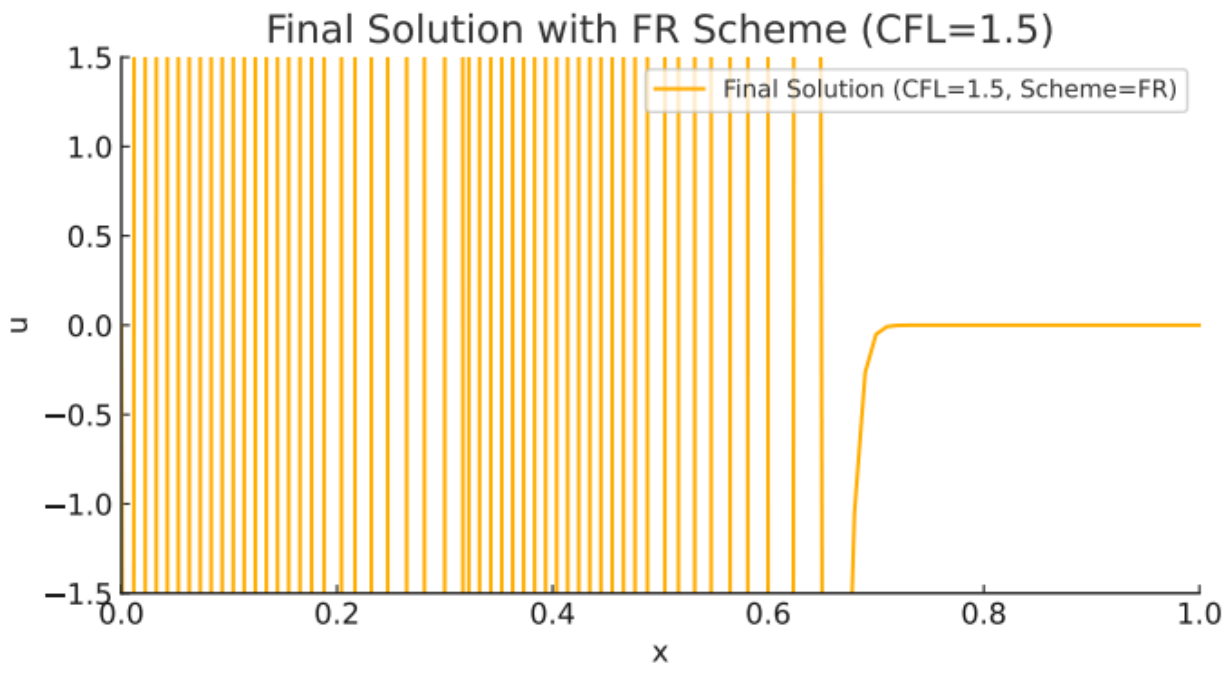

Final Solution with FTFS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTFS)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x


Final Solution with FTCS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

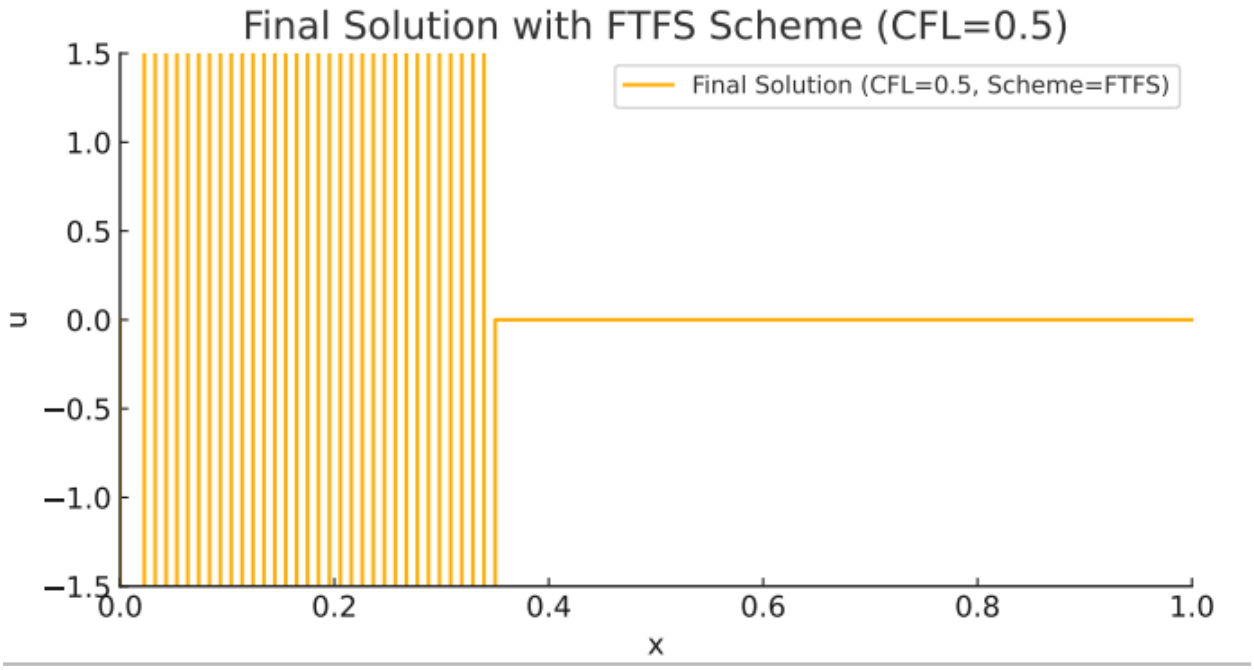

Final Solution with FTBS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

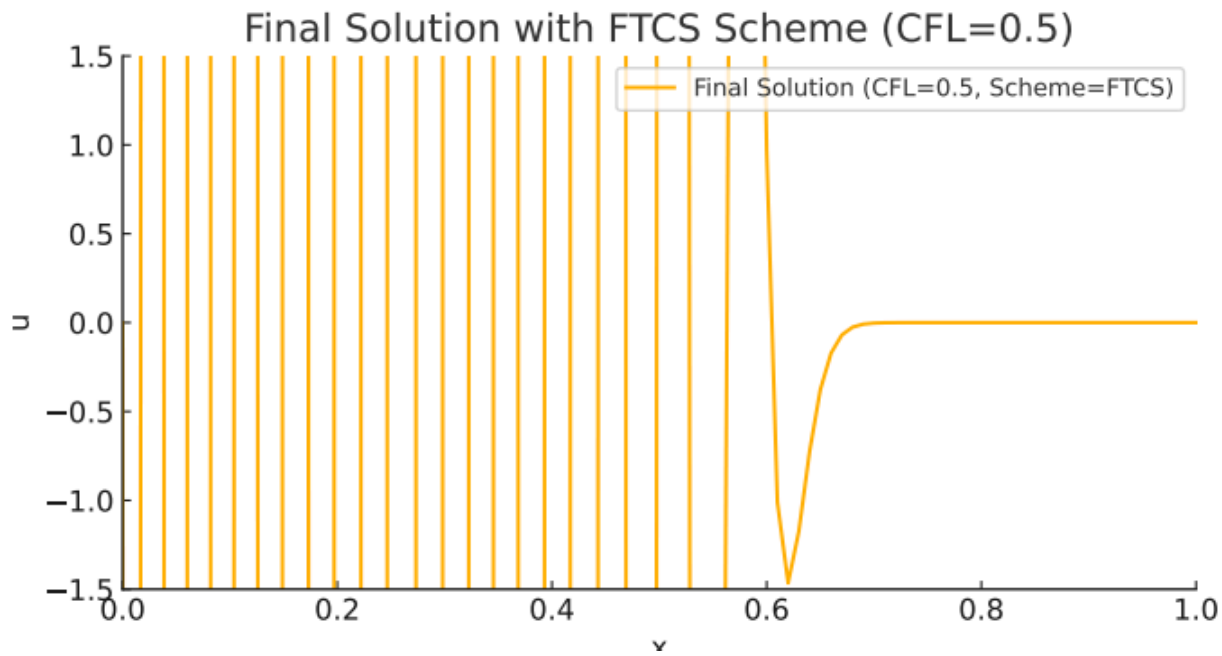

Final Solution with LW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

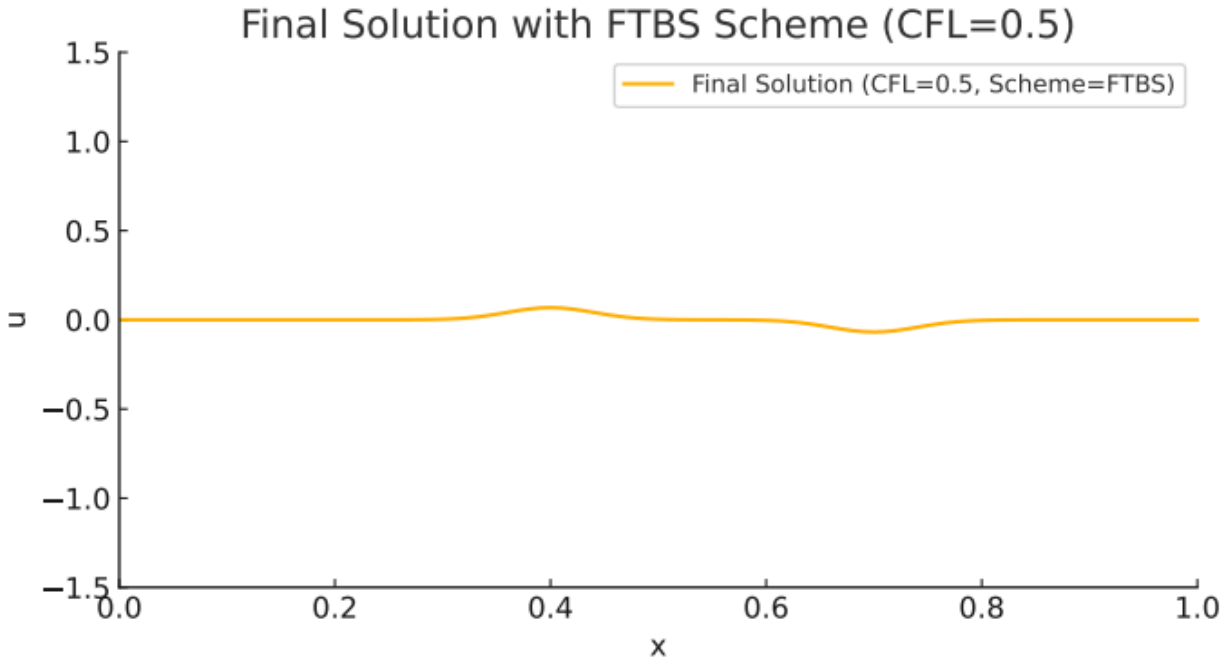

Final Solution with BW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

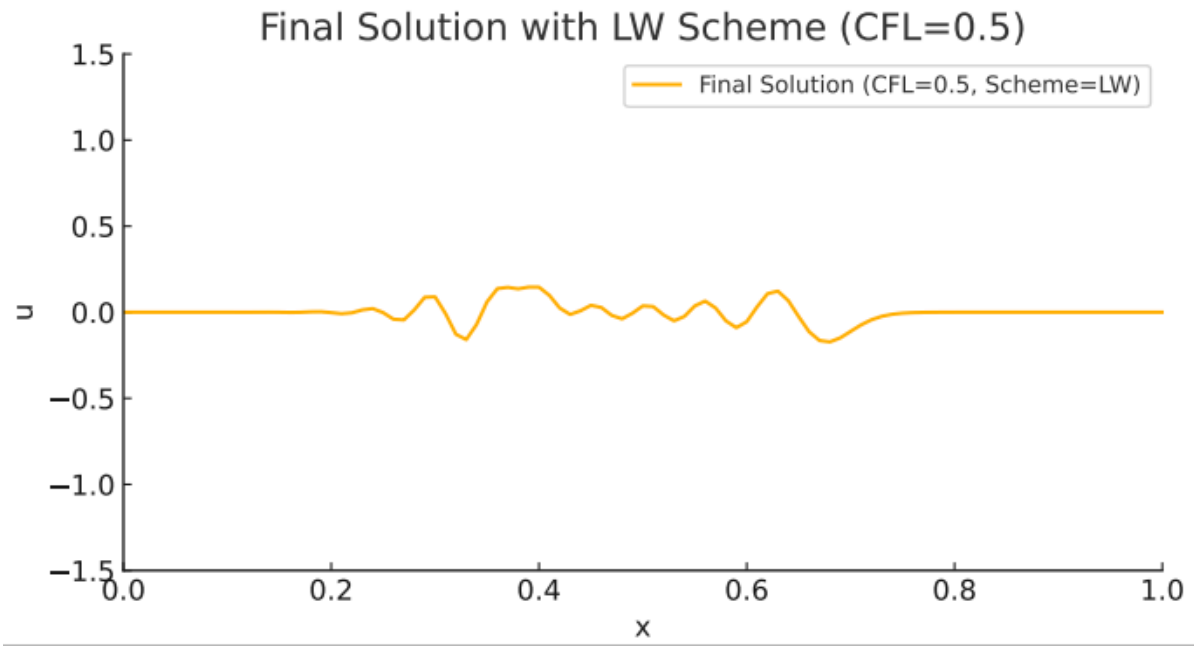


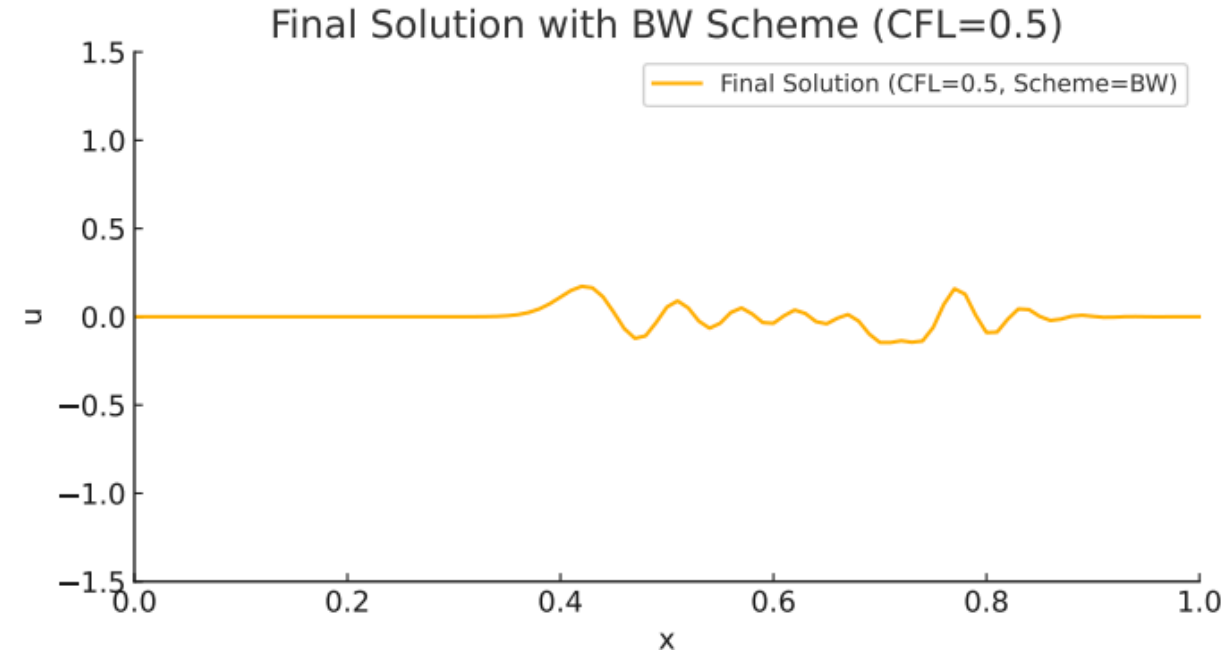
d) For 4th Condition

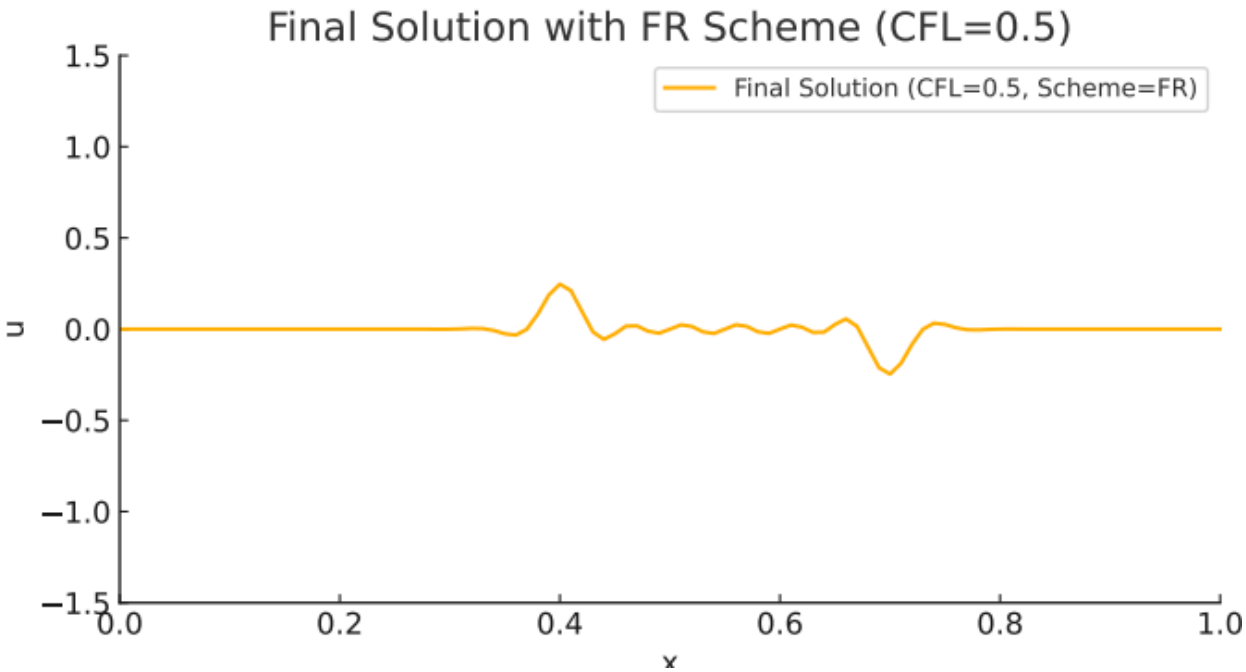


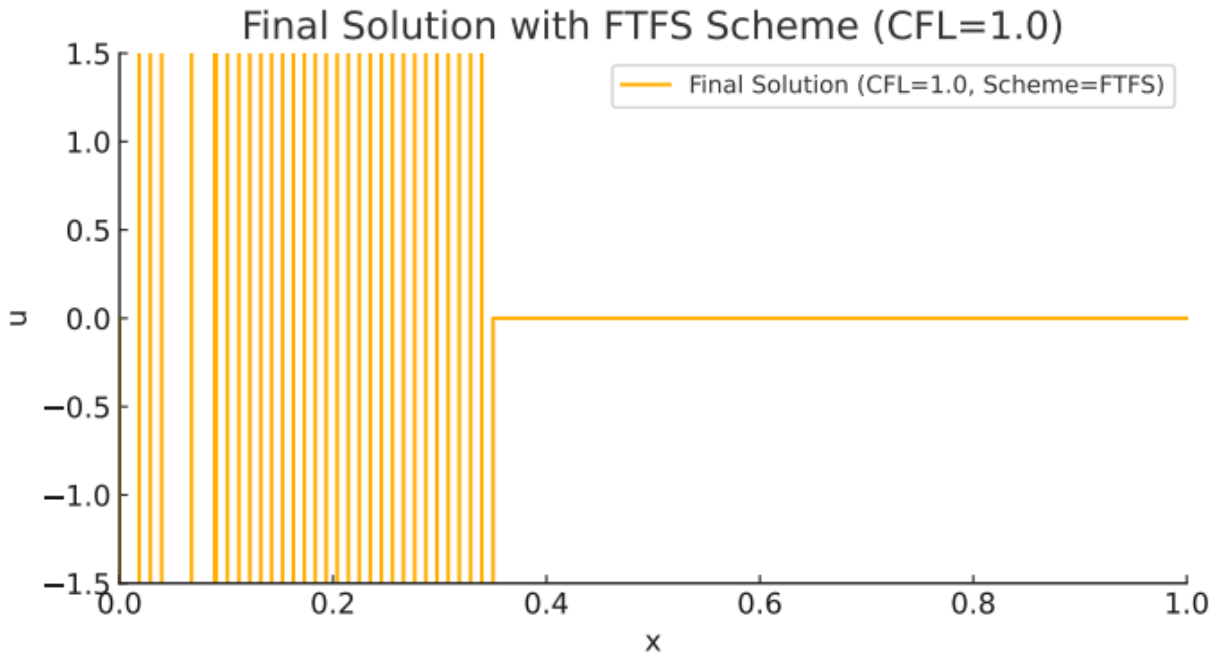

Final Solution with FTCS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

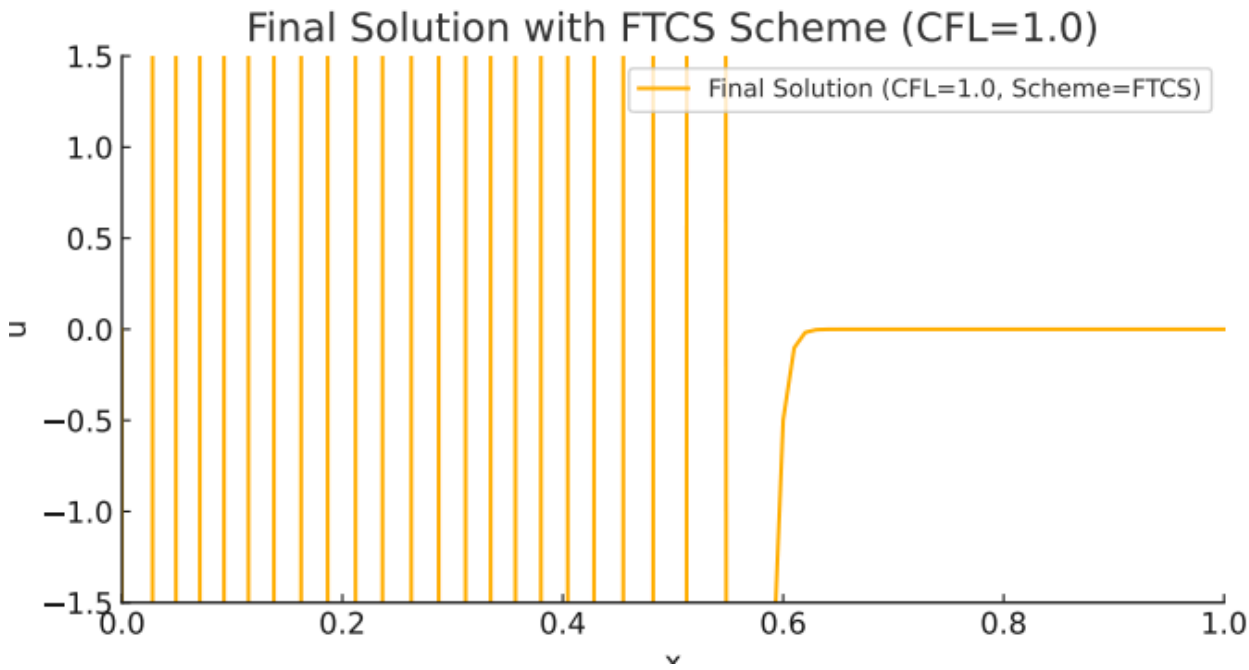

Final Solution with FTBS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

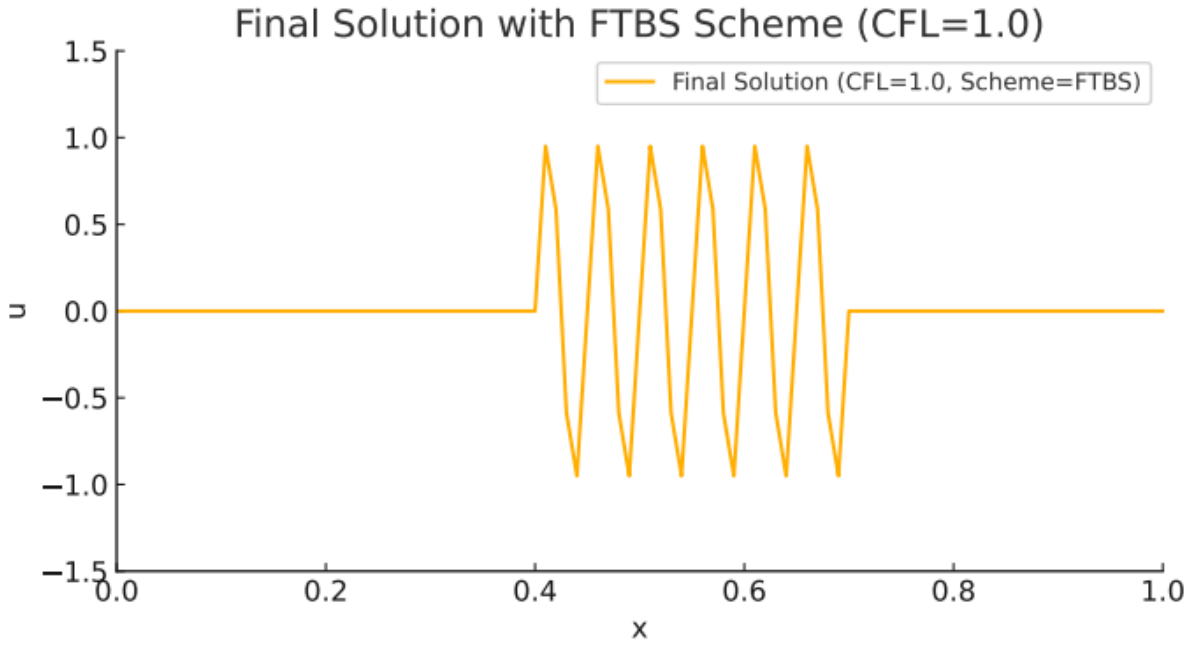

Final Solution with LW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

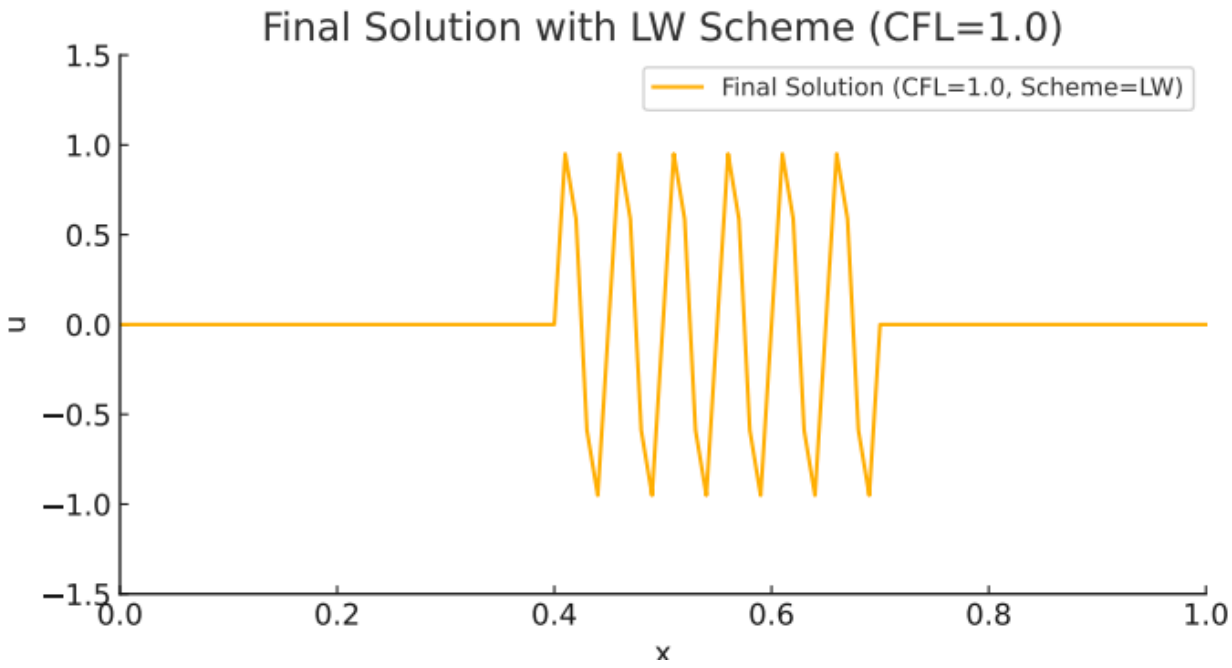

Final Solution with BW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

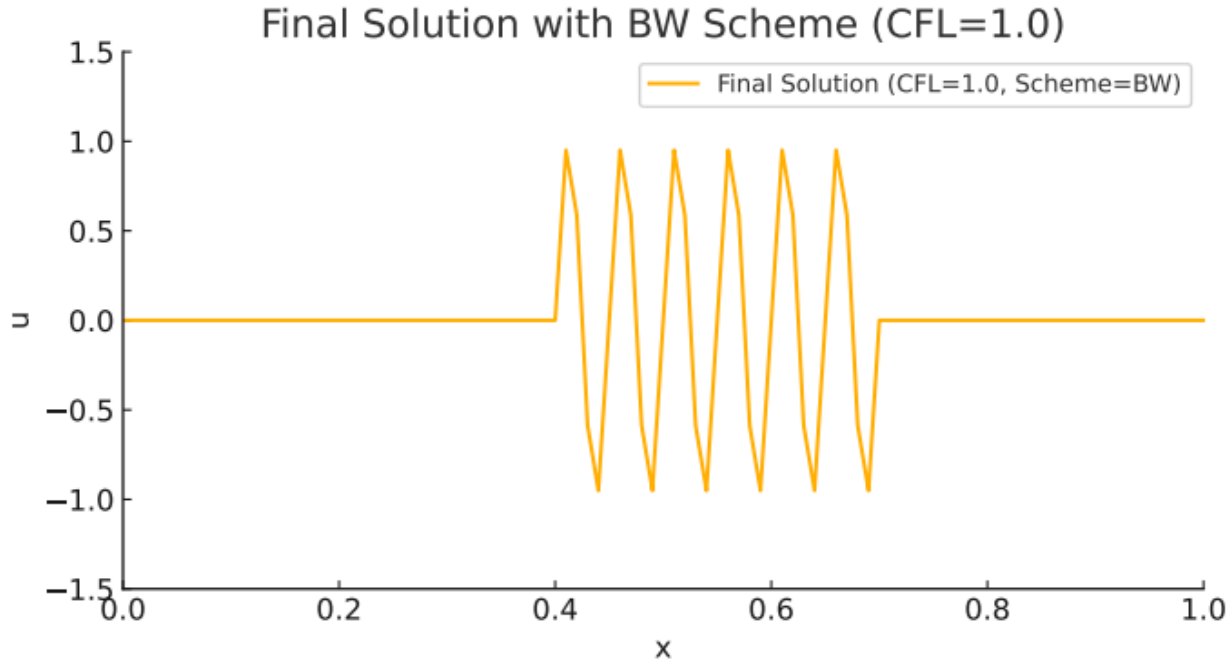

Final Solution with FR Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FR)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

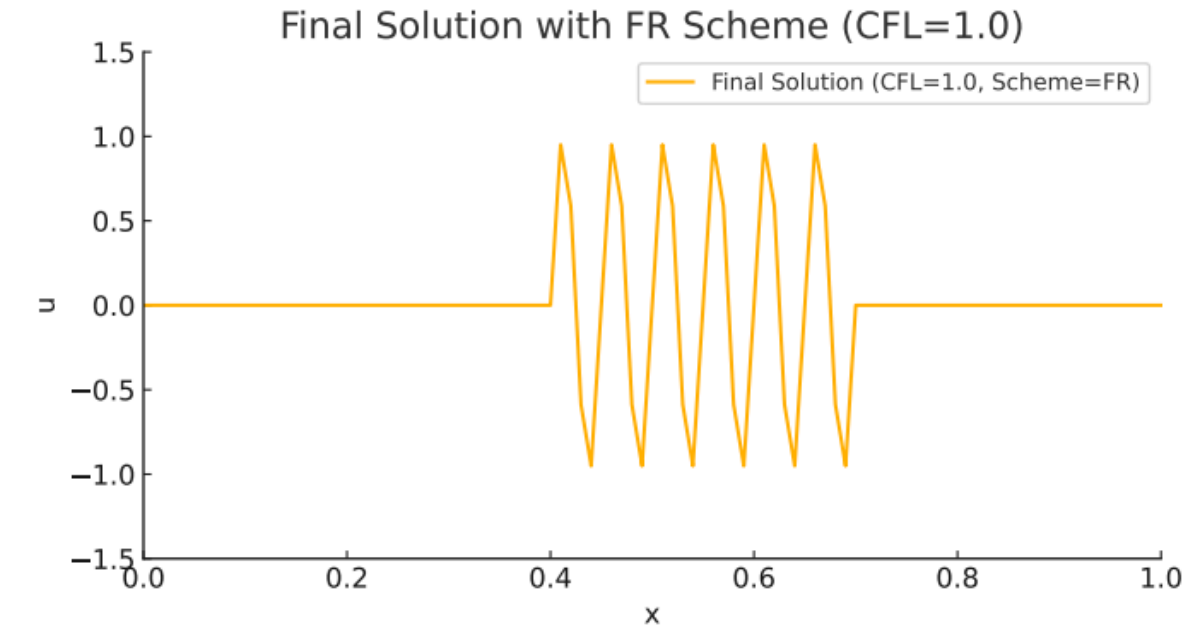

Final Solution with FTFS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTFS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

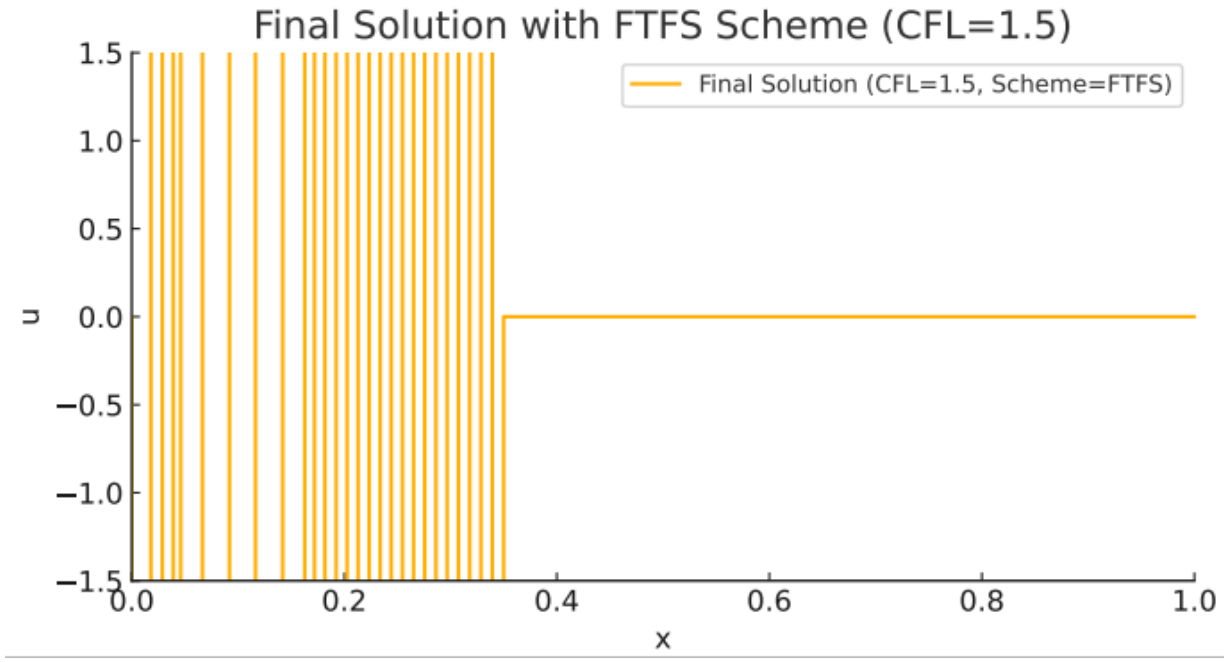

Final Solution with FTCS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

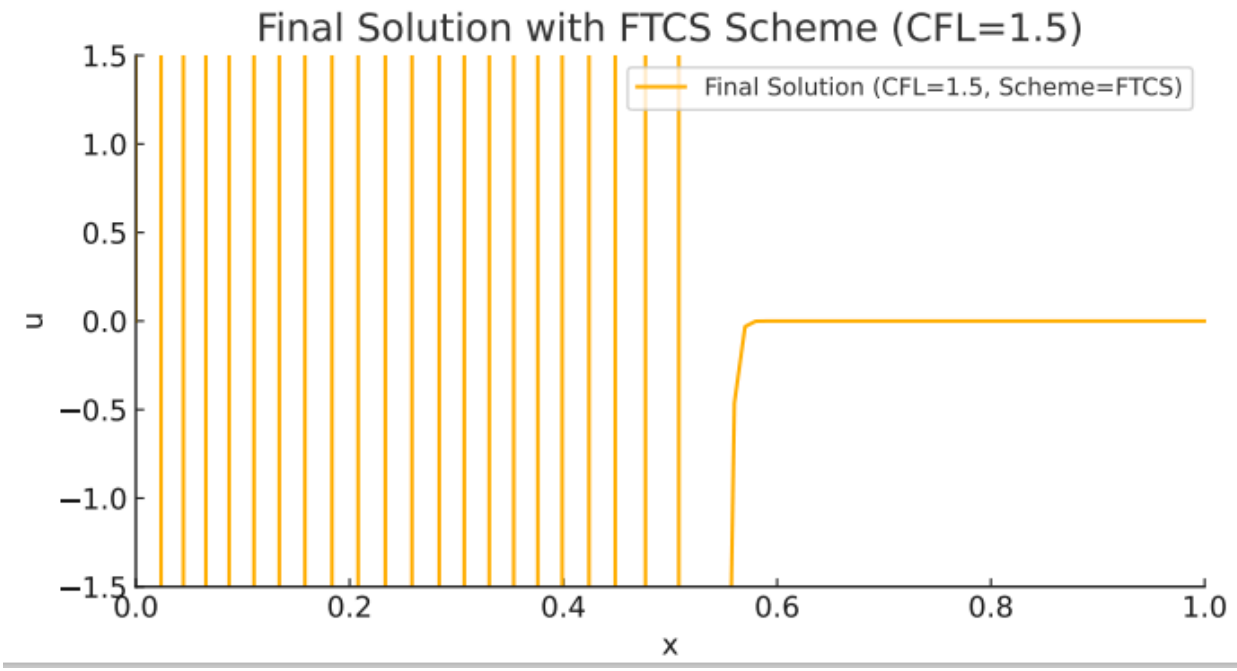

Final Solution with FTBS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

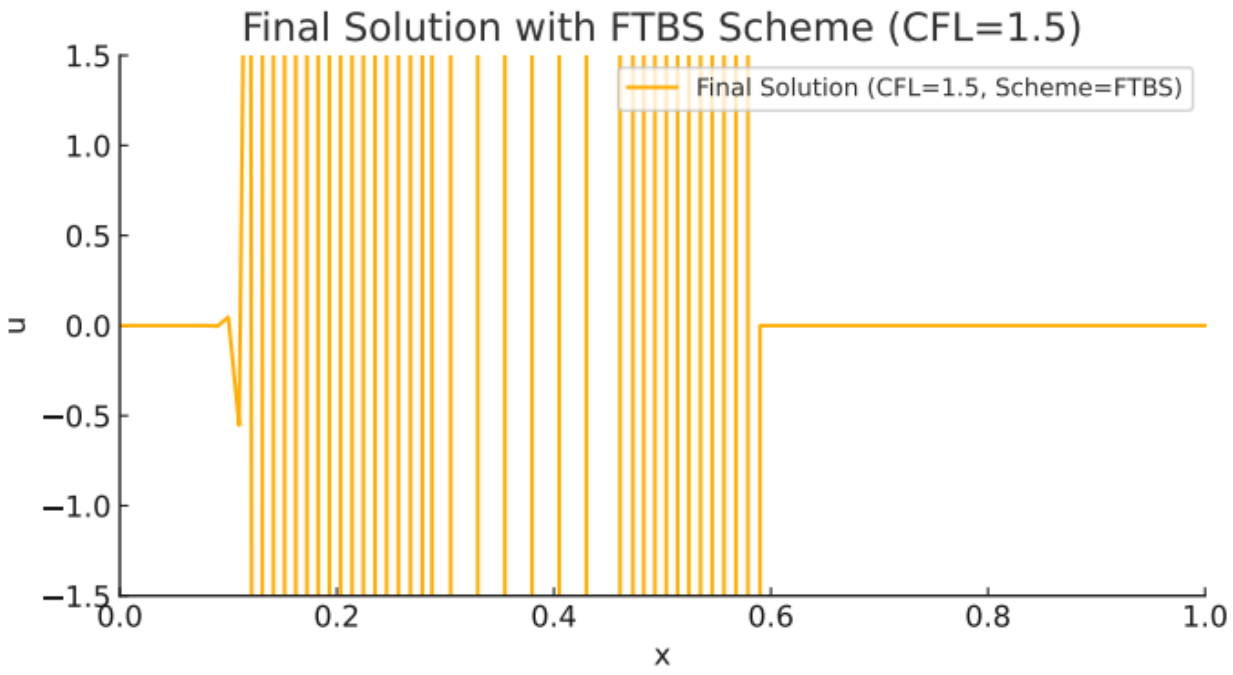

Final Solution with LW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

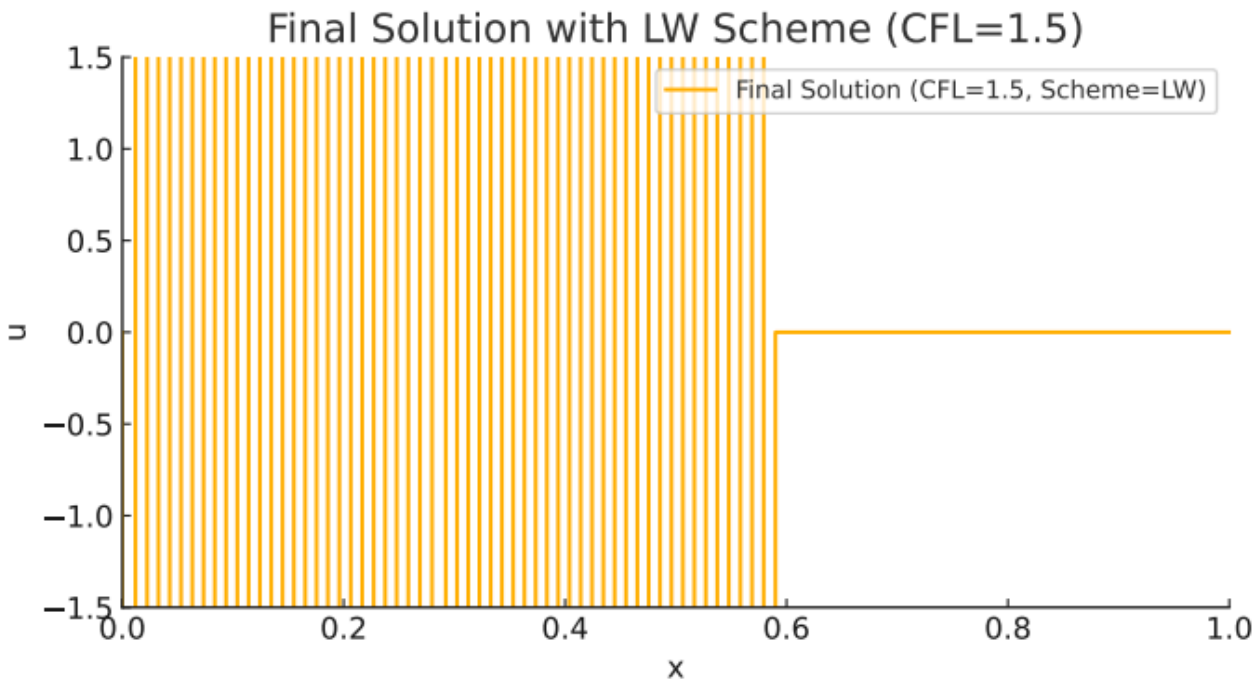

Final Solution with BW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

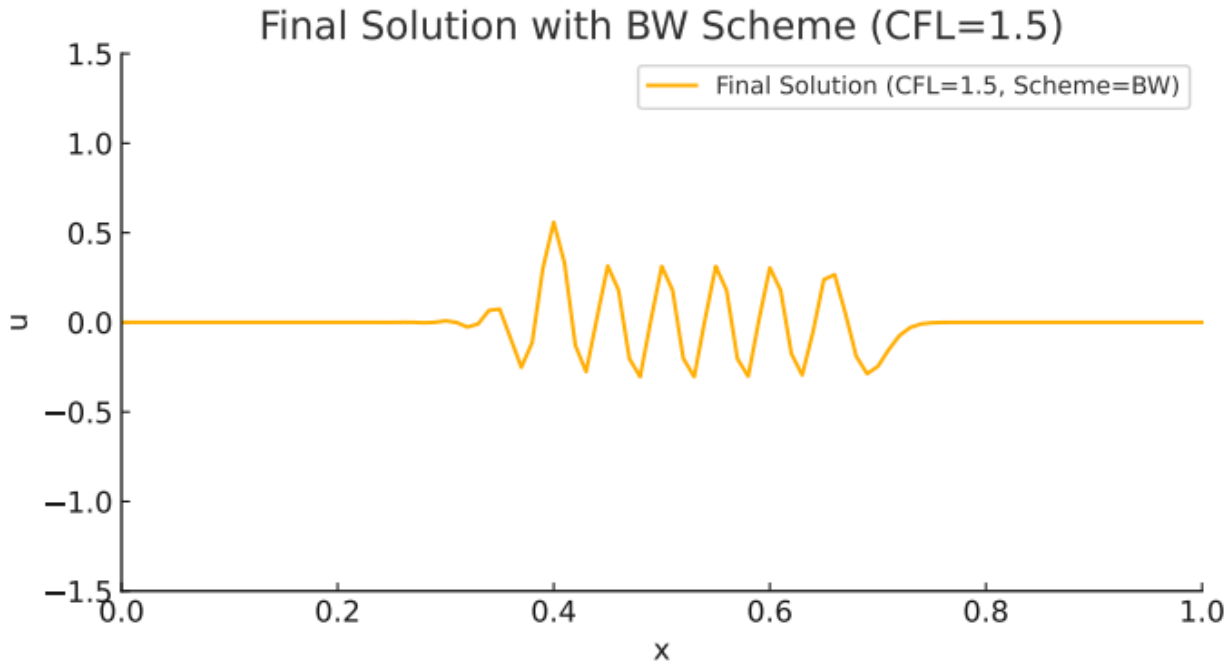

Final Solution with FR Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FR)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

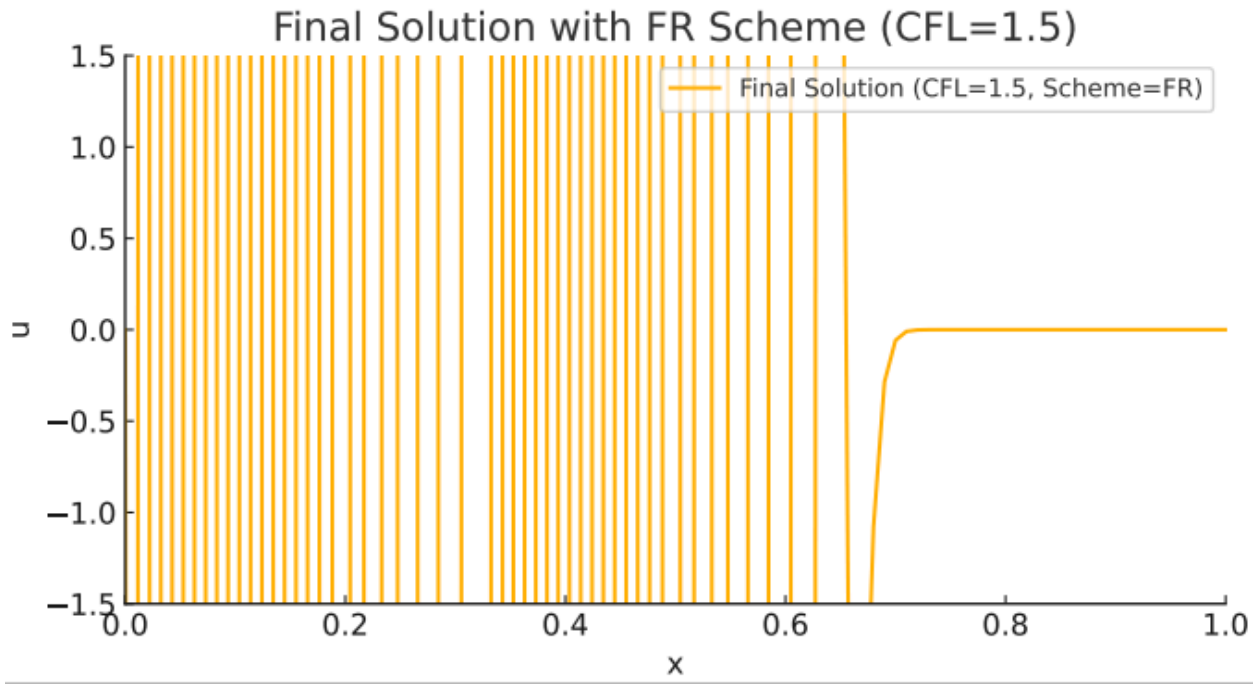

Final Solution with FTFS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTFS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0


Final Solution with FTCS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

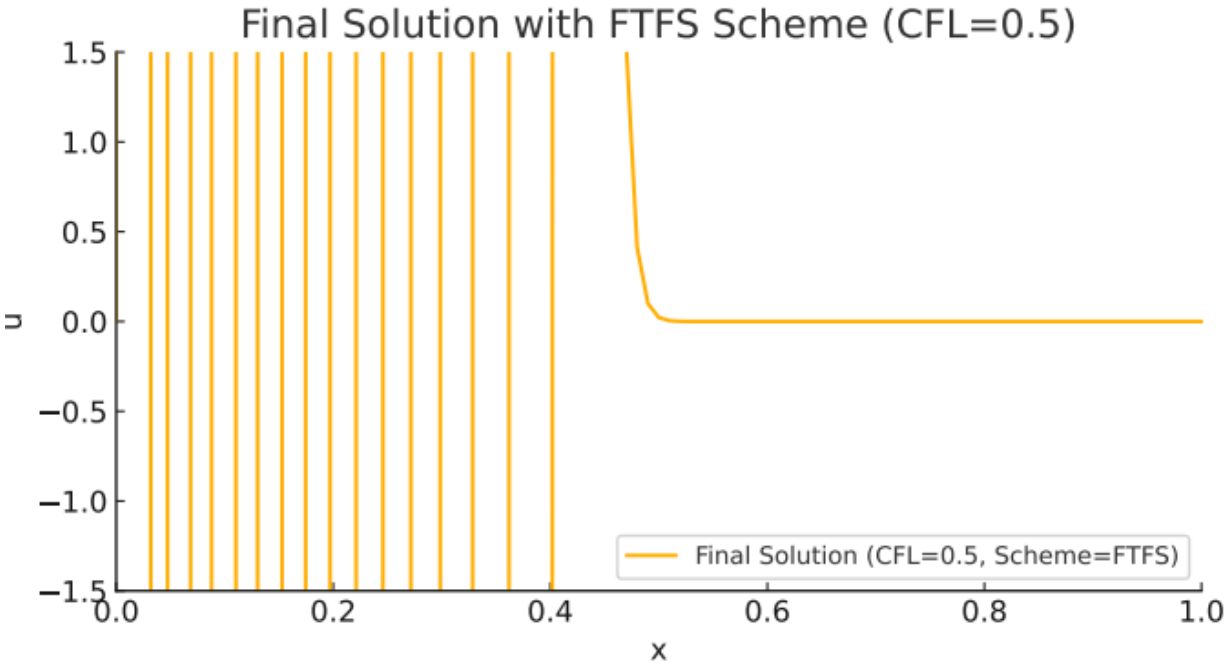

Final Solution with FTBS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

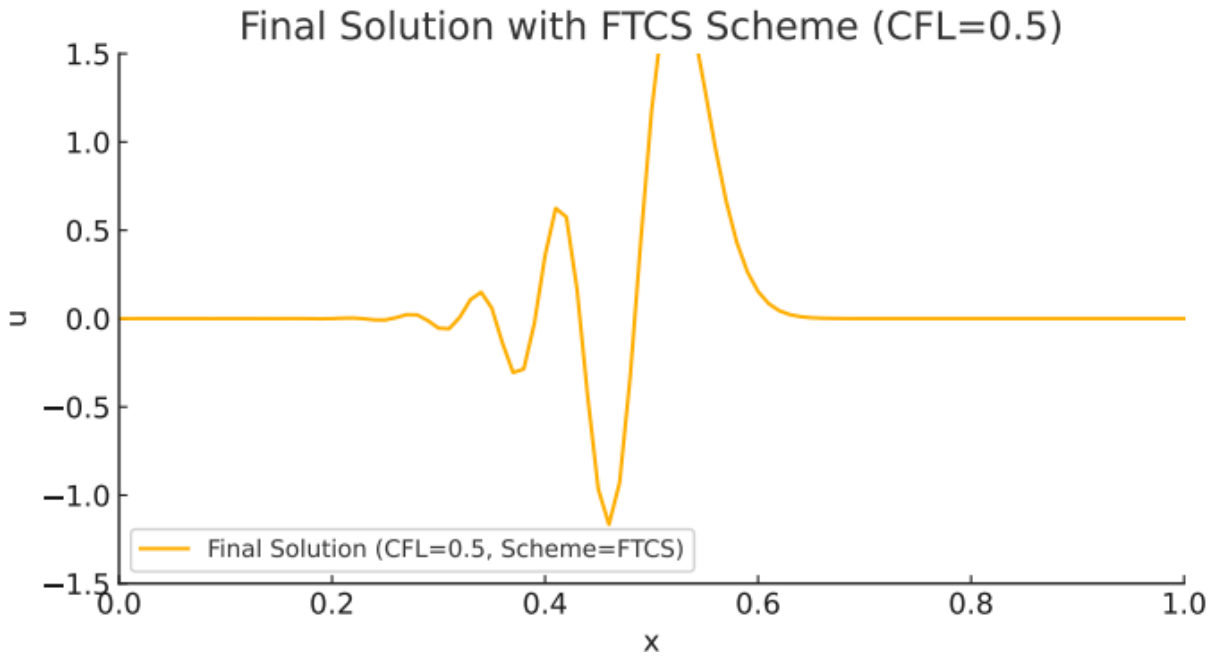

Final Solution with LW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

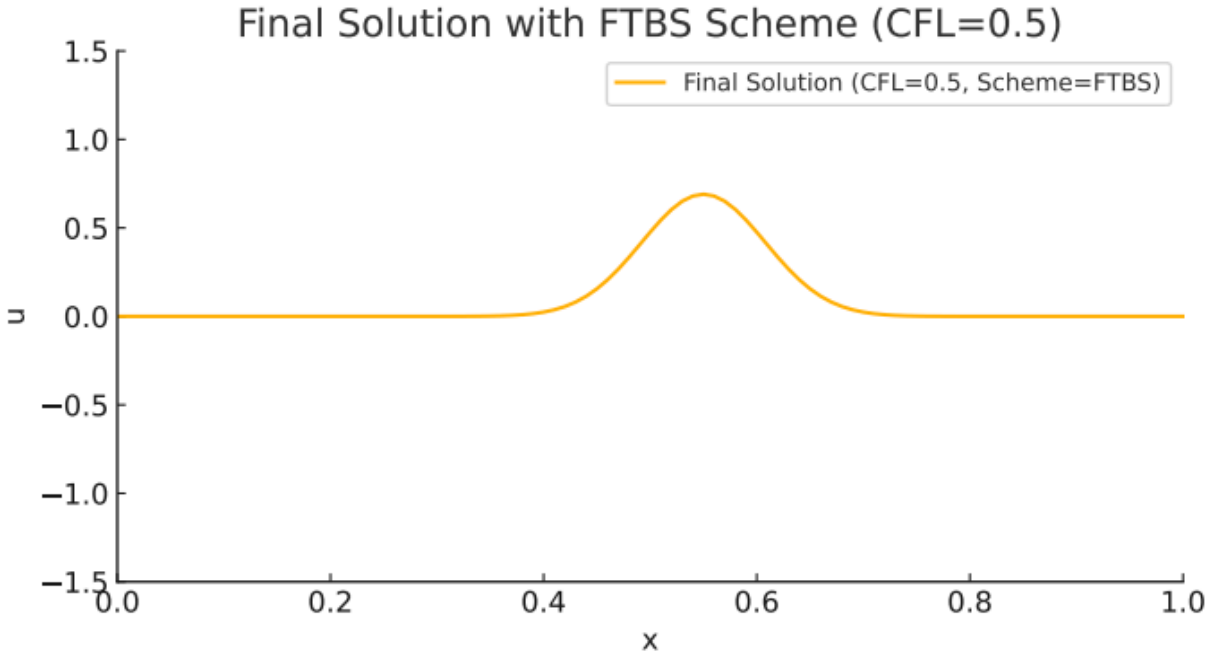

Final Solution with BW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

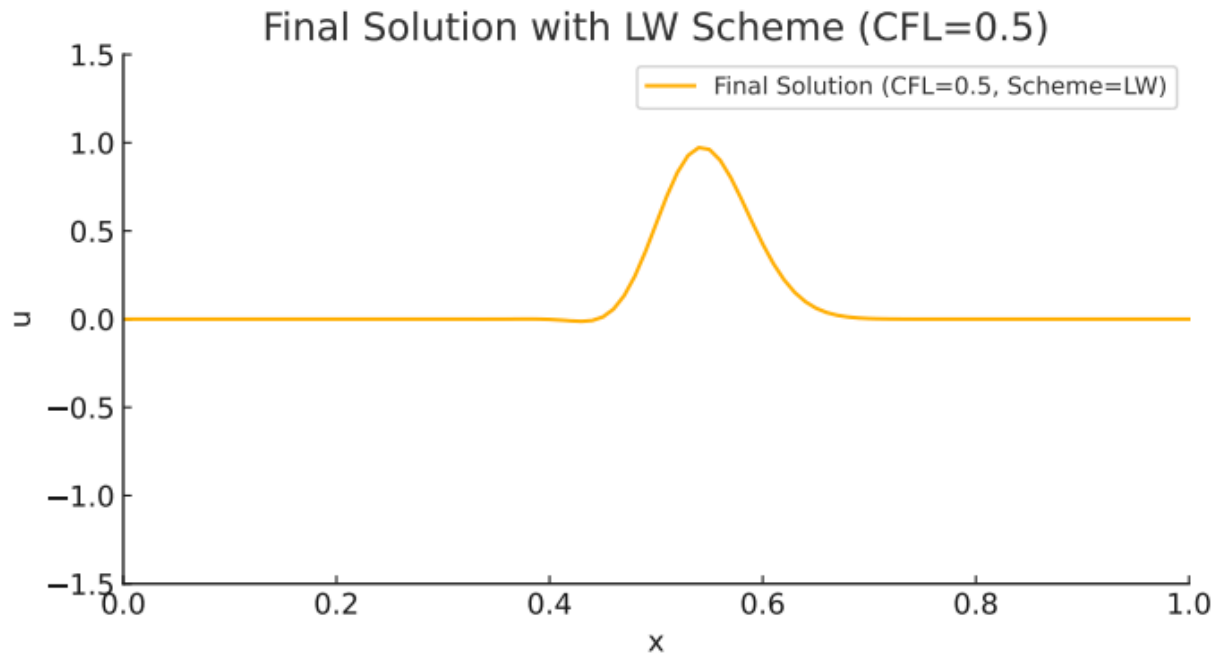


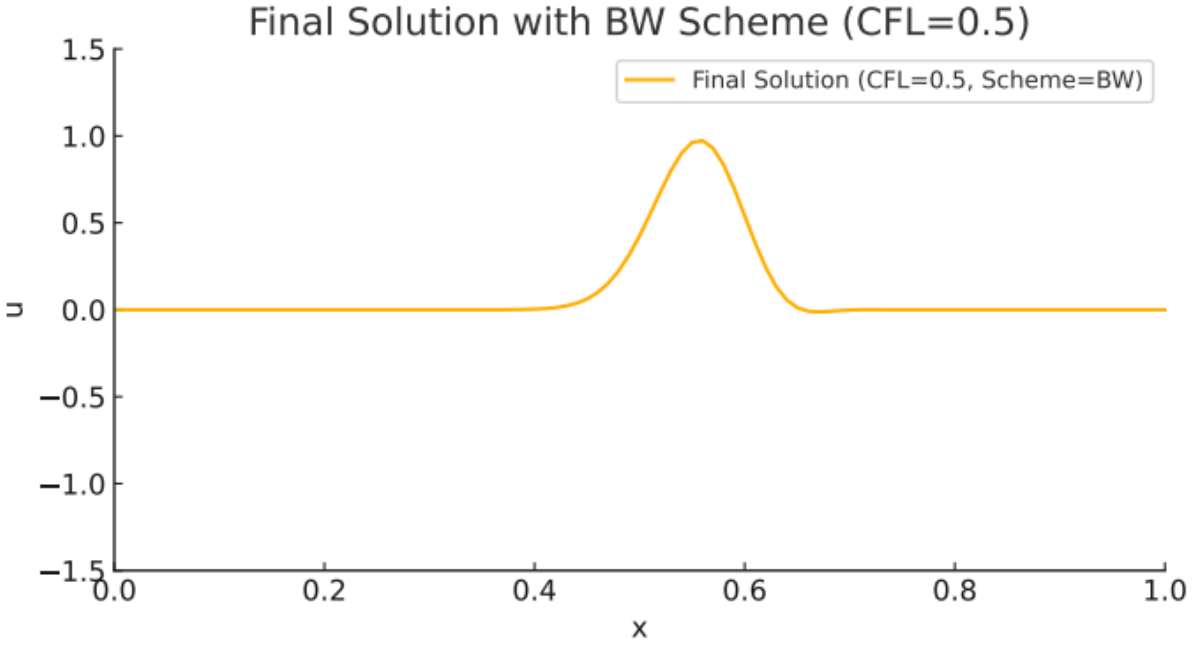
e) For 5th Condition

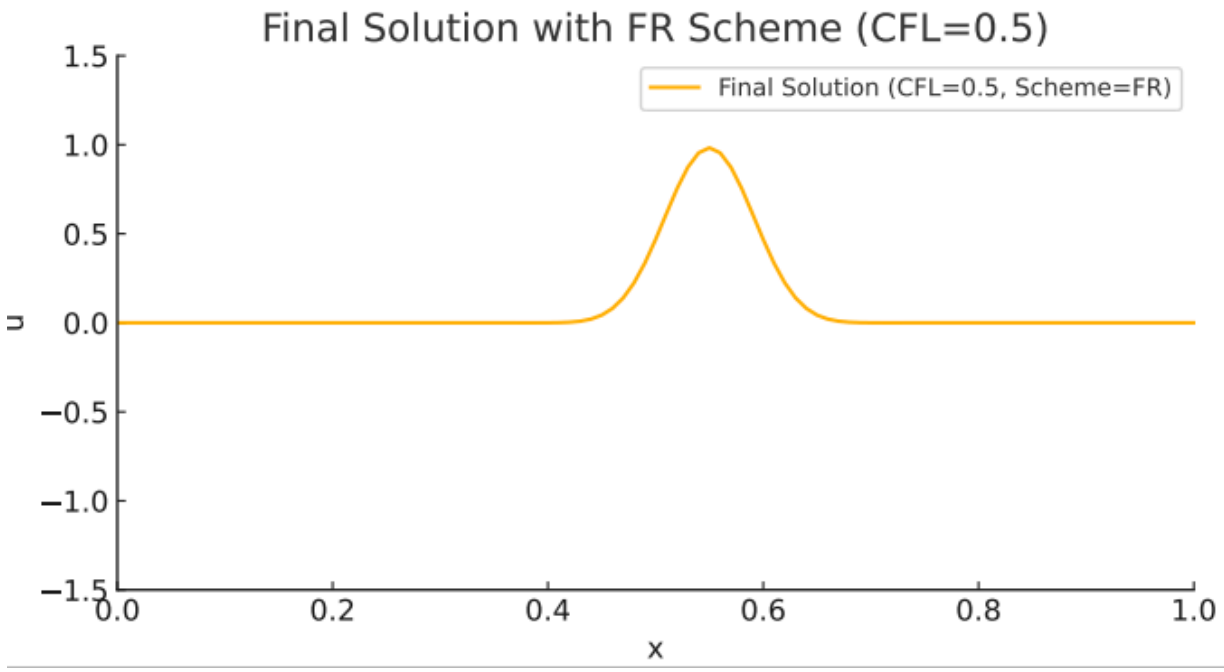


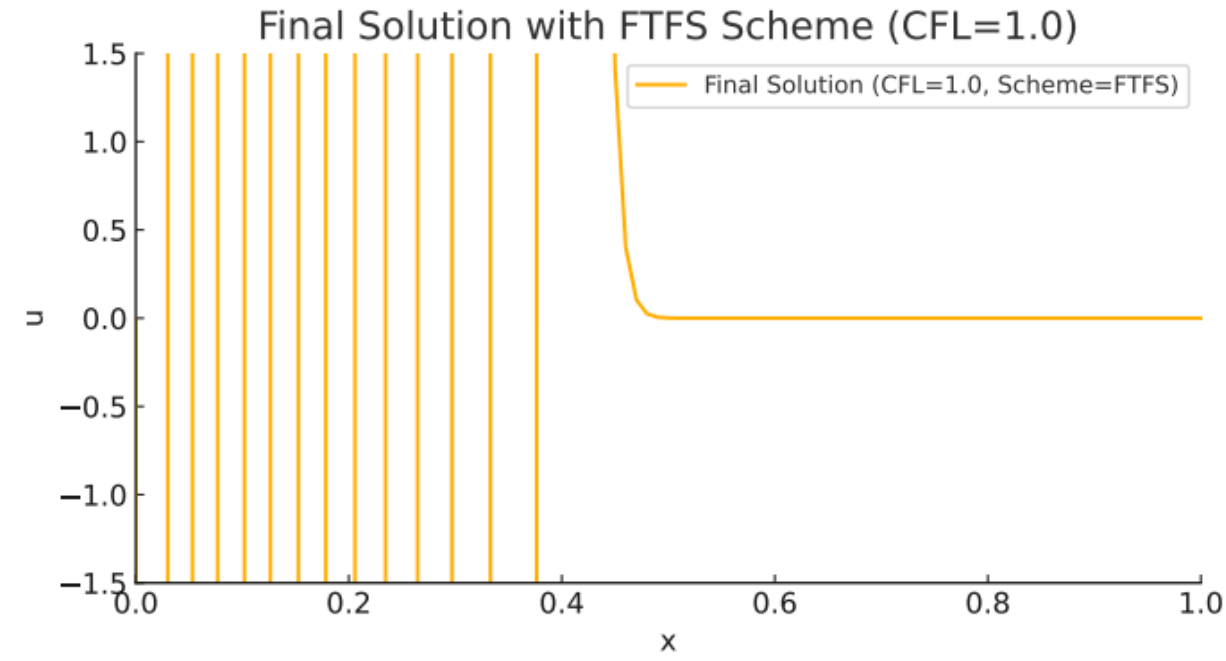

Final Solution with FTCS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTCS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

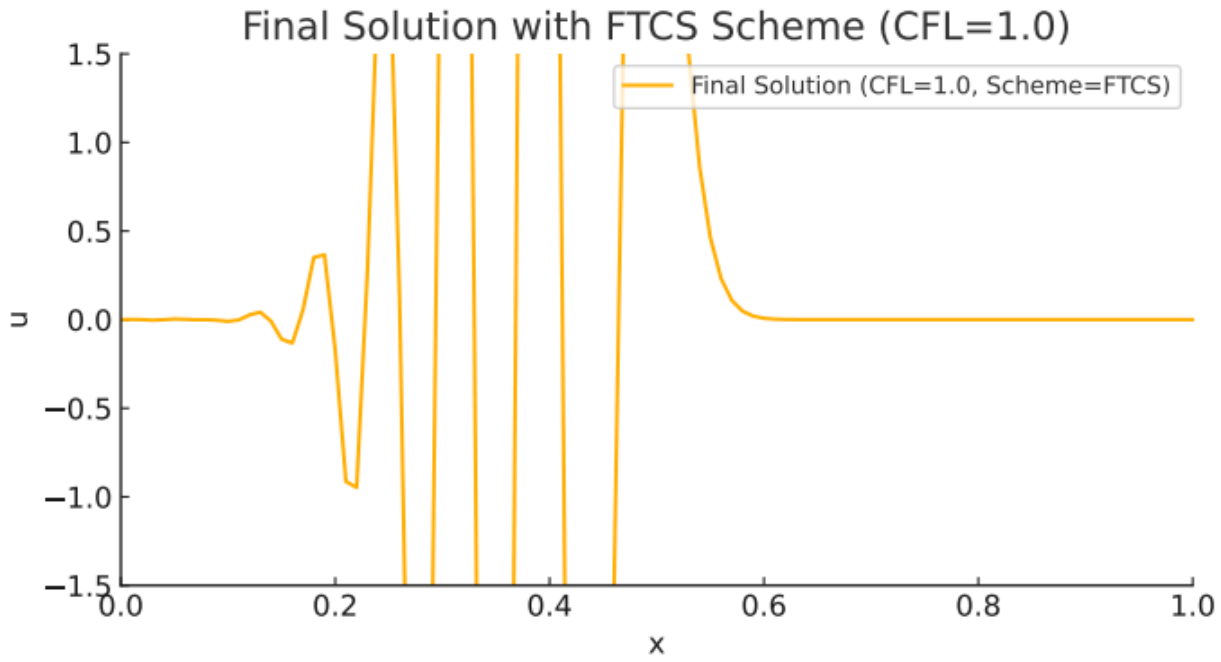

Final Solution with FTBS Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FTBS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

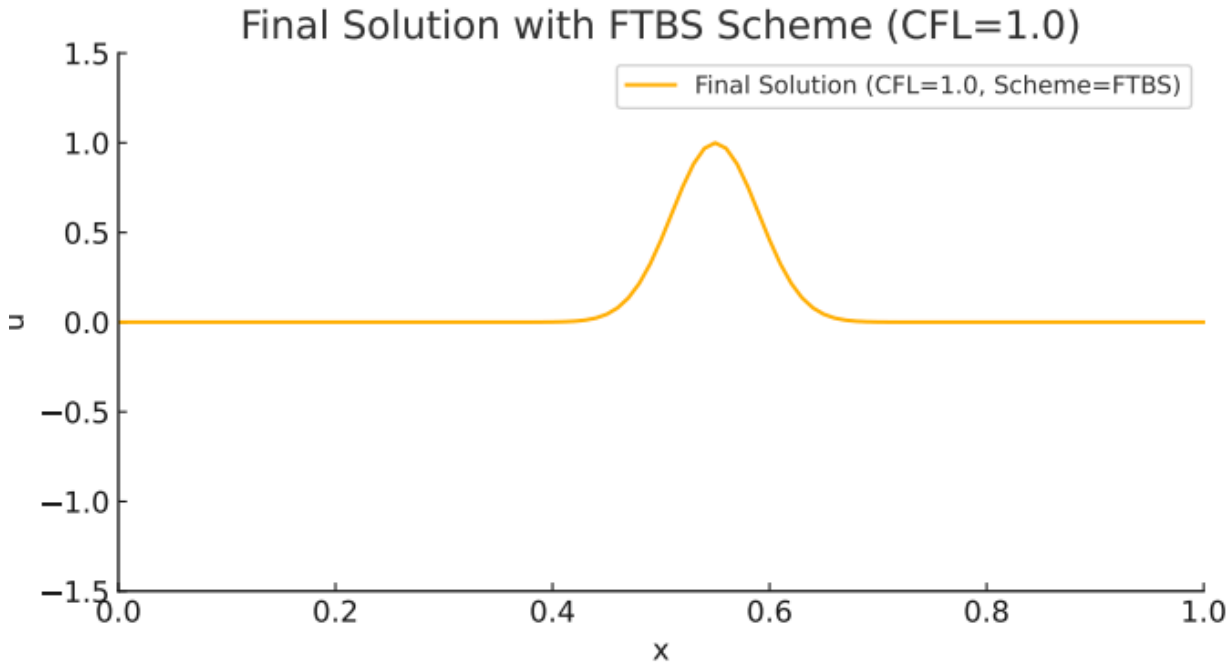

Final Solution with LW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=LW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

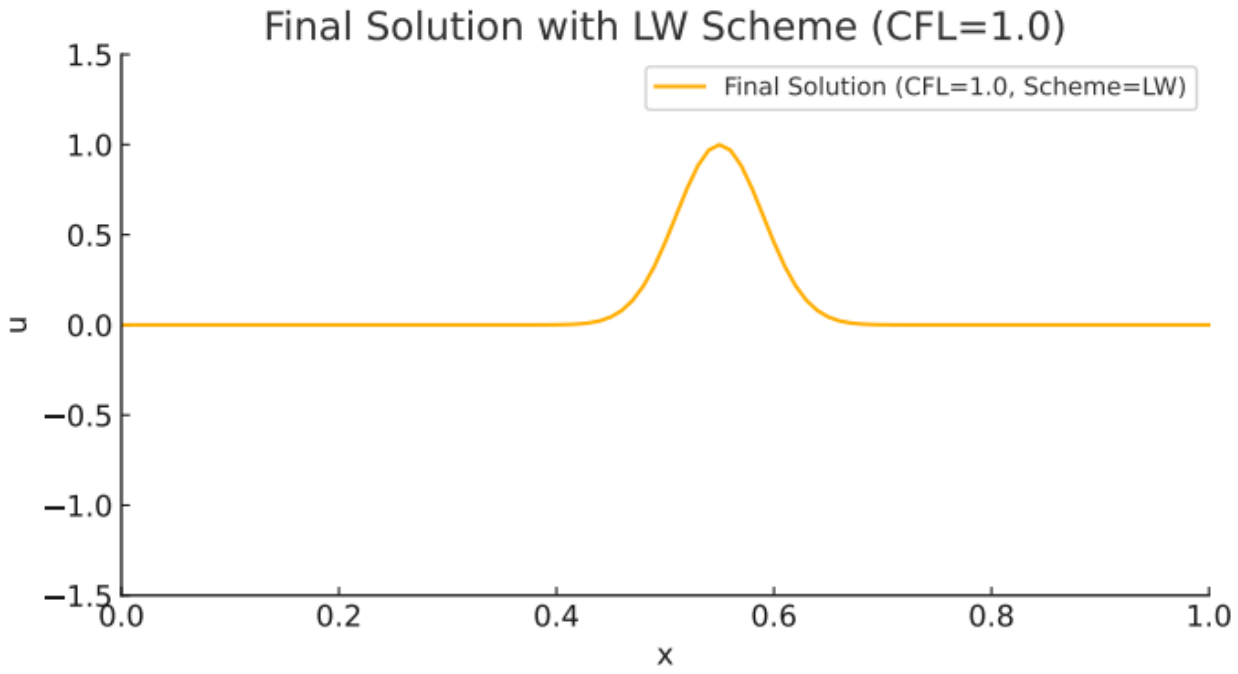

Final Solution with BW Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=BW)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

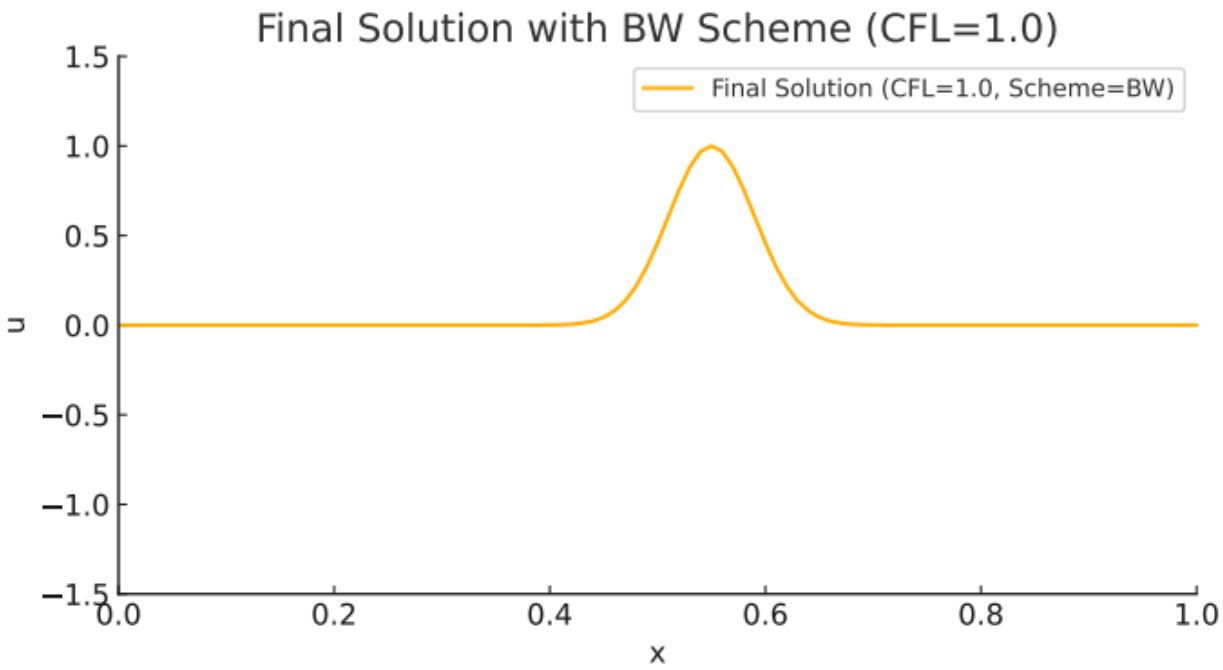

Final Solution with FR Scheme (CFL=0.5)
Final Solution (CFL=0.5, Scheme=FR)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

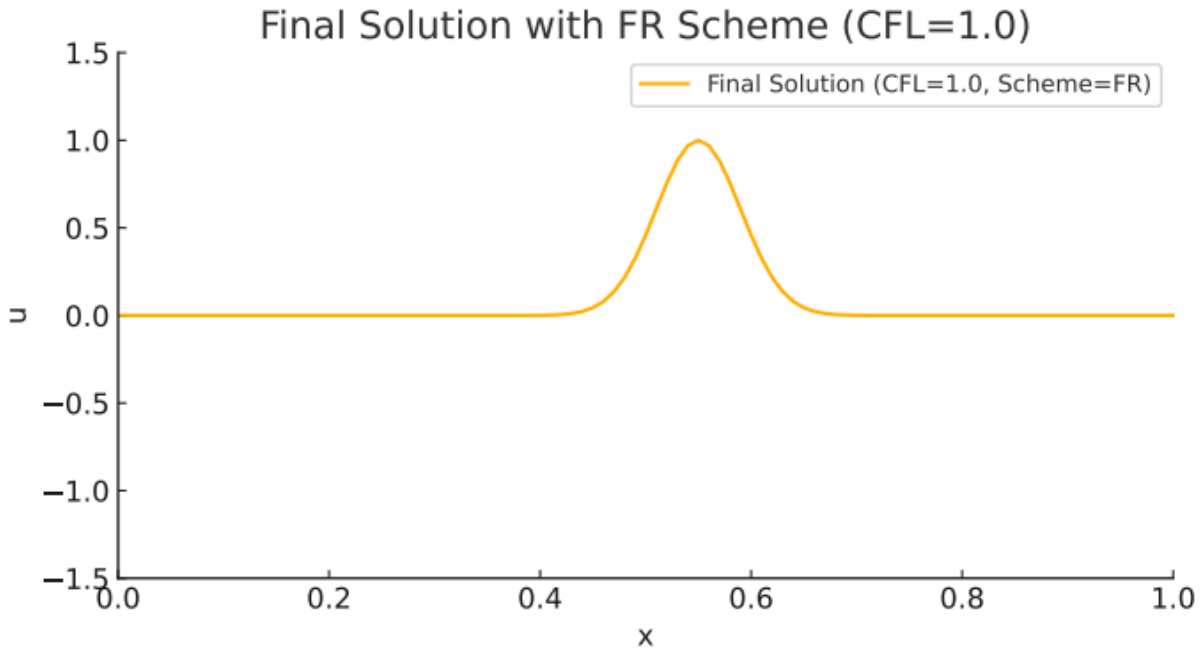

Final Solution with FTFS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTFS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

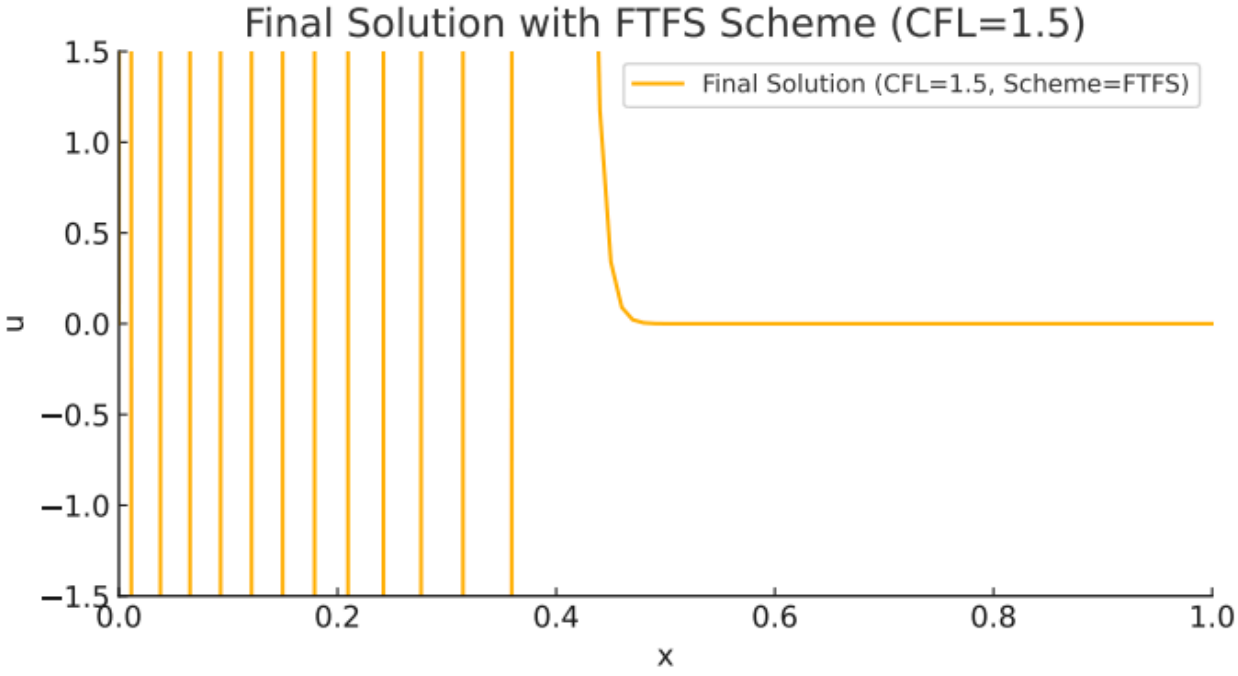

Final Solution with FTCS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTCS)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

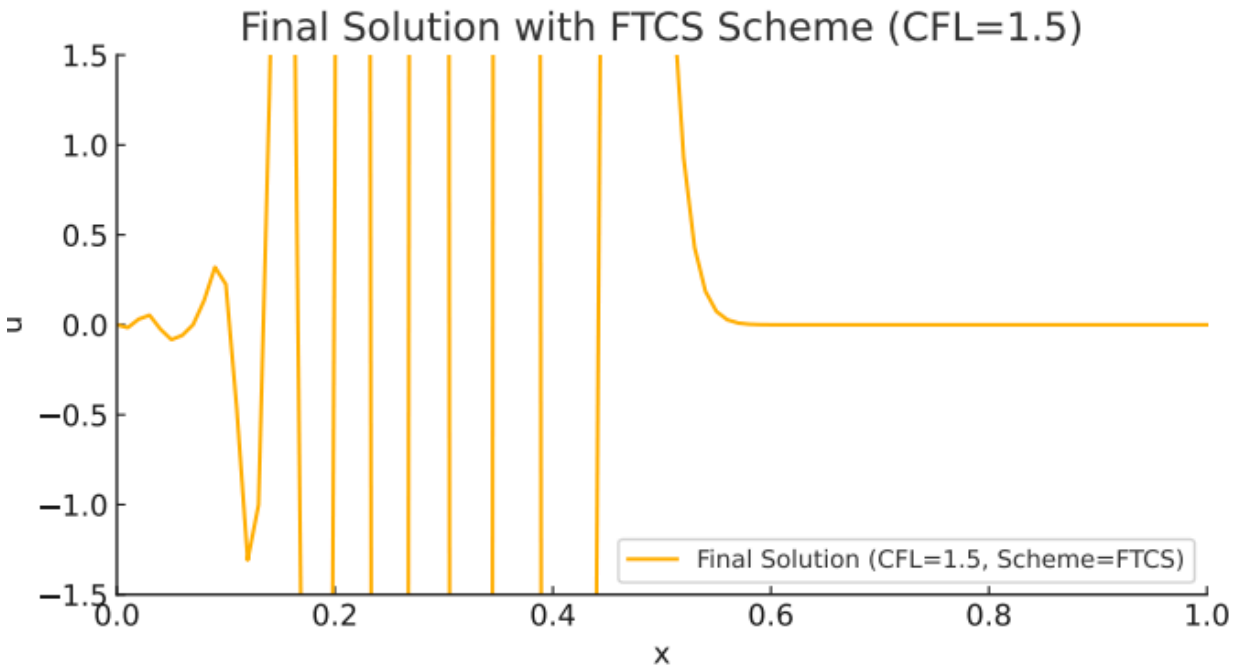

Final Solution with FTBS Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FTBS)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

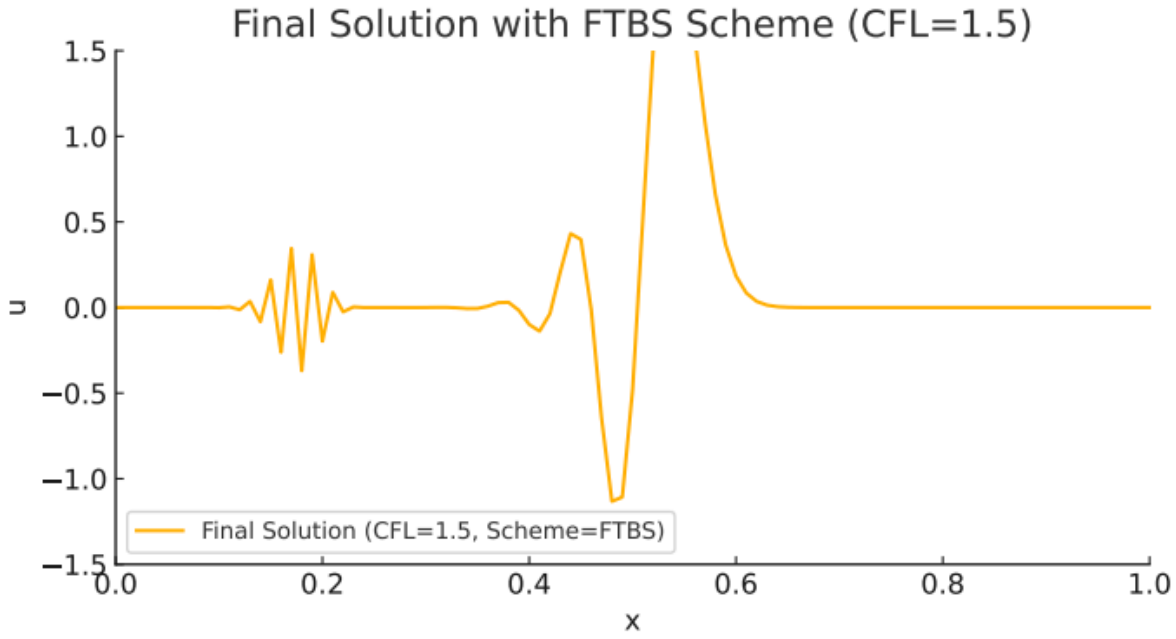

Final Solution with LW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

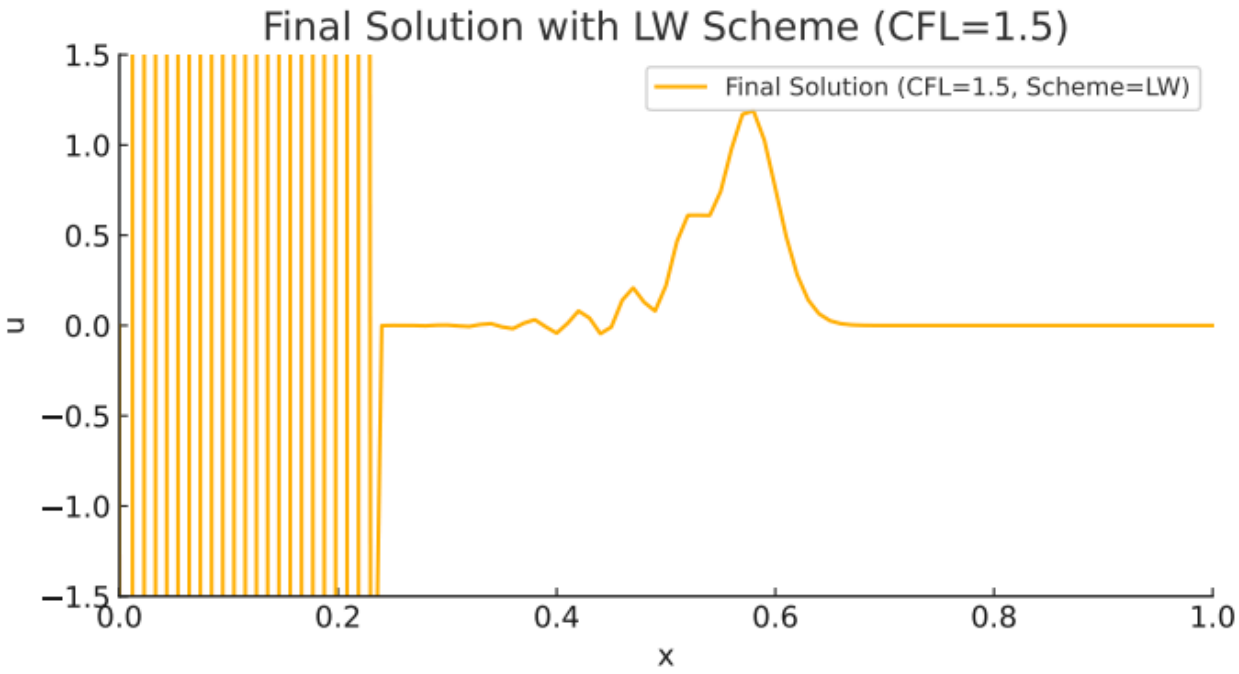

Final Solution with BW Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

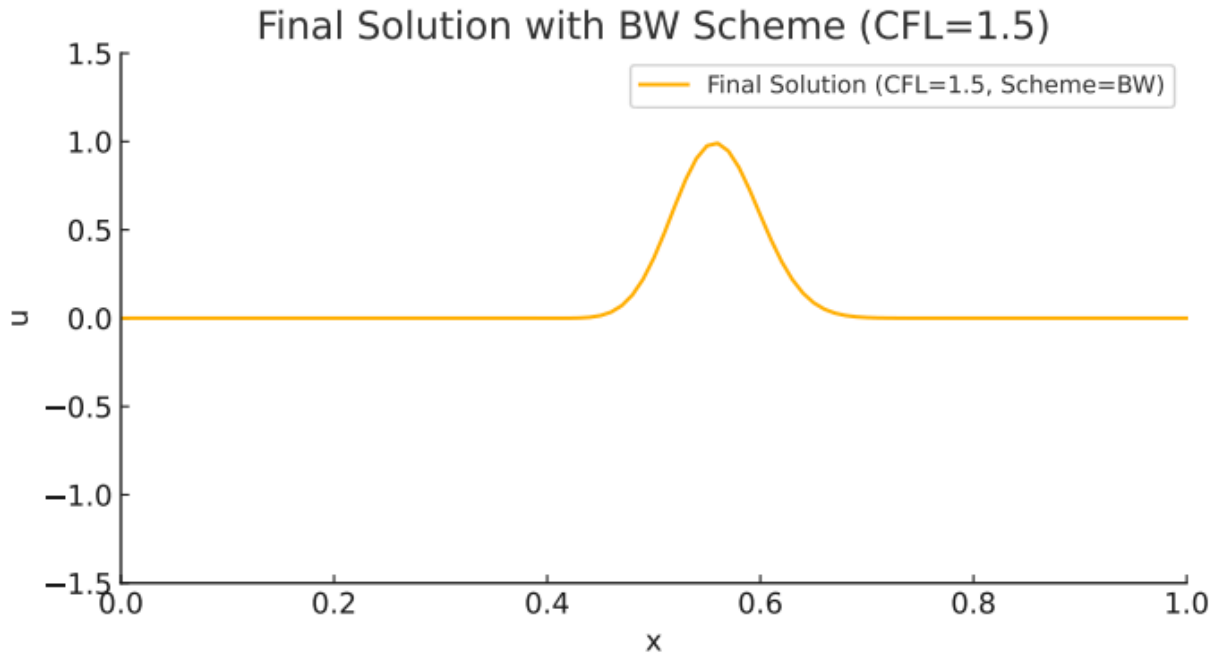

Final Solution with FR Scheme (CFL=1.0)
Final Solution (CFL=1.0, Scheme=FR)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

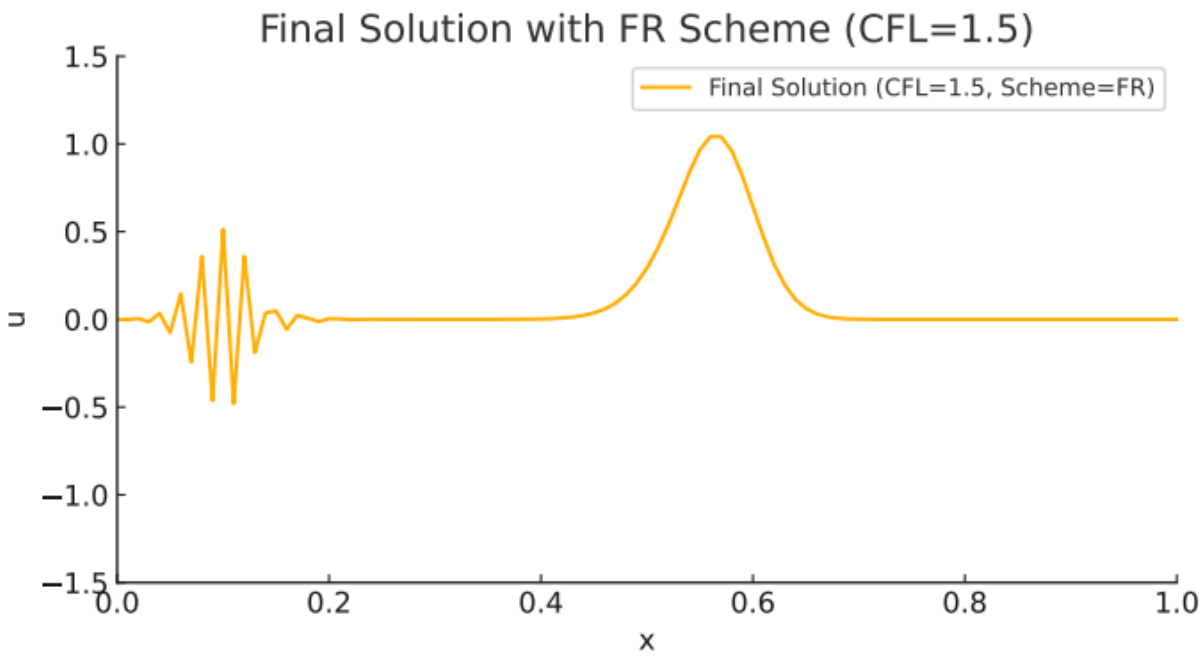

Final Solution with FTFS Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=FTFS)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0


Final Solution with FTCS Scheme (CFL=1.5)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x
Final Solution (CFL=1.5, Scheme=FTCS)


Final Solution with FTBS Scheme (CFL=1.5)
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x
Final Solution (CFL=1.5, Scheme=FTBS)

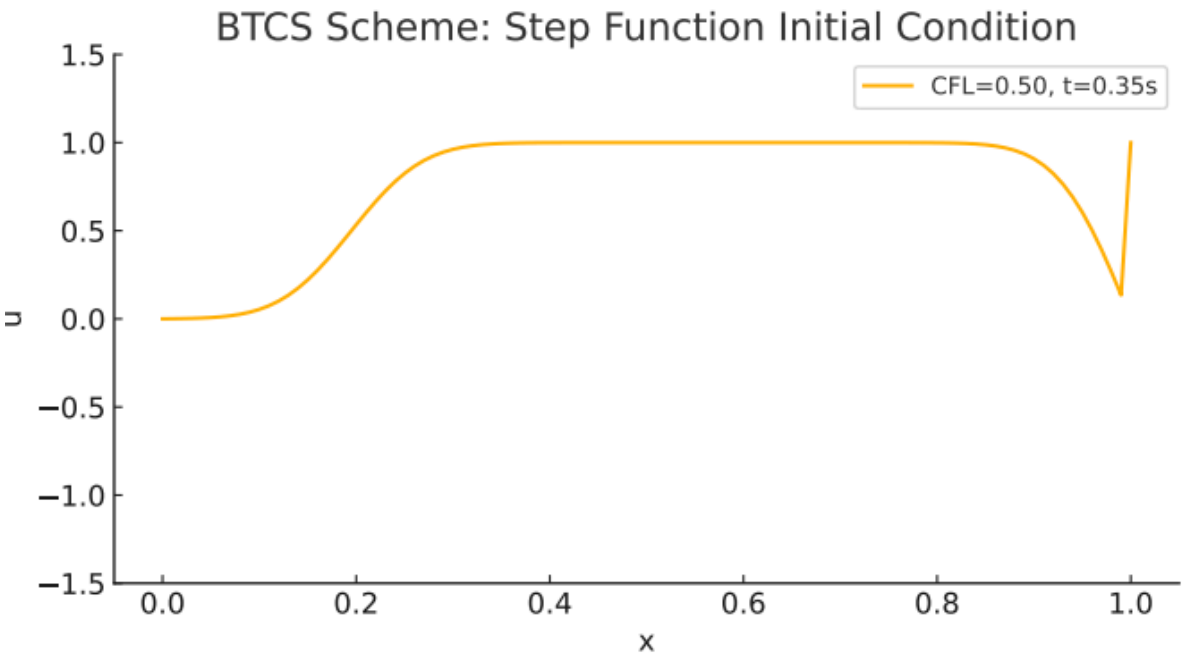

Final Solution with LW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=LW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

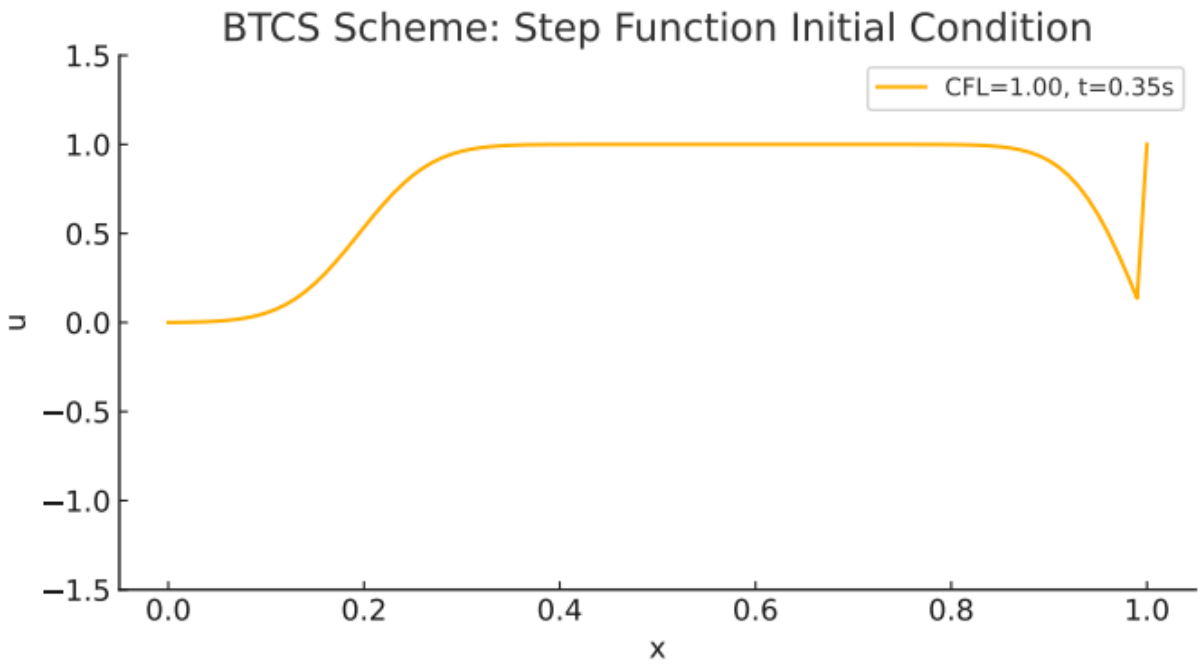

Final Solution with BW Scheme (CFL=1.5)
Final Solution (CFL=1.5, Scheme=BW)
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

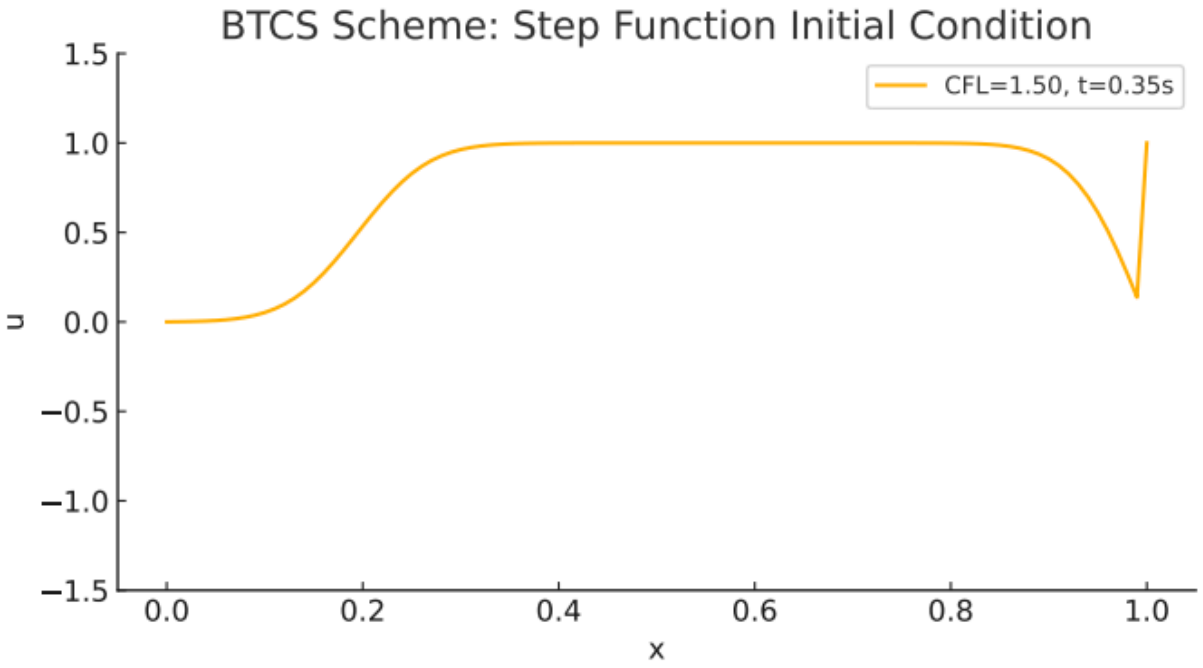


For Explicit Method:

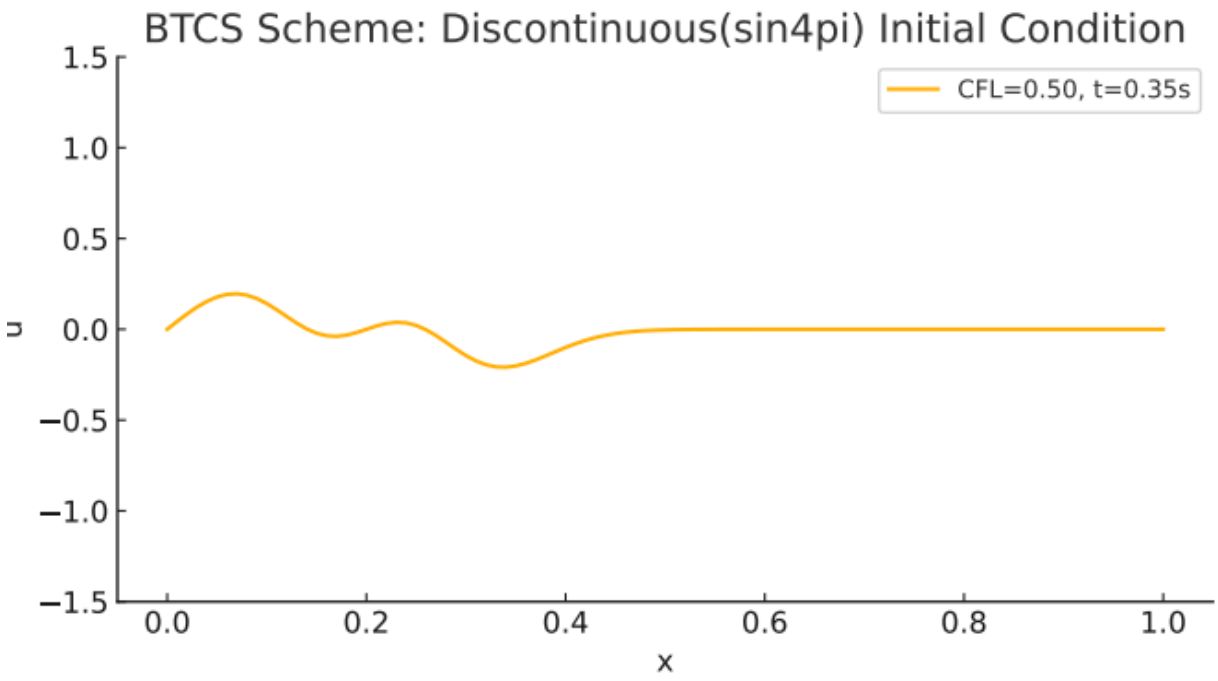
a) For 1st Condition

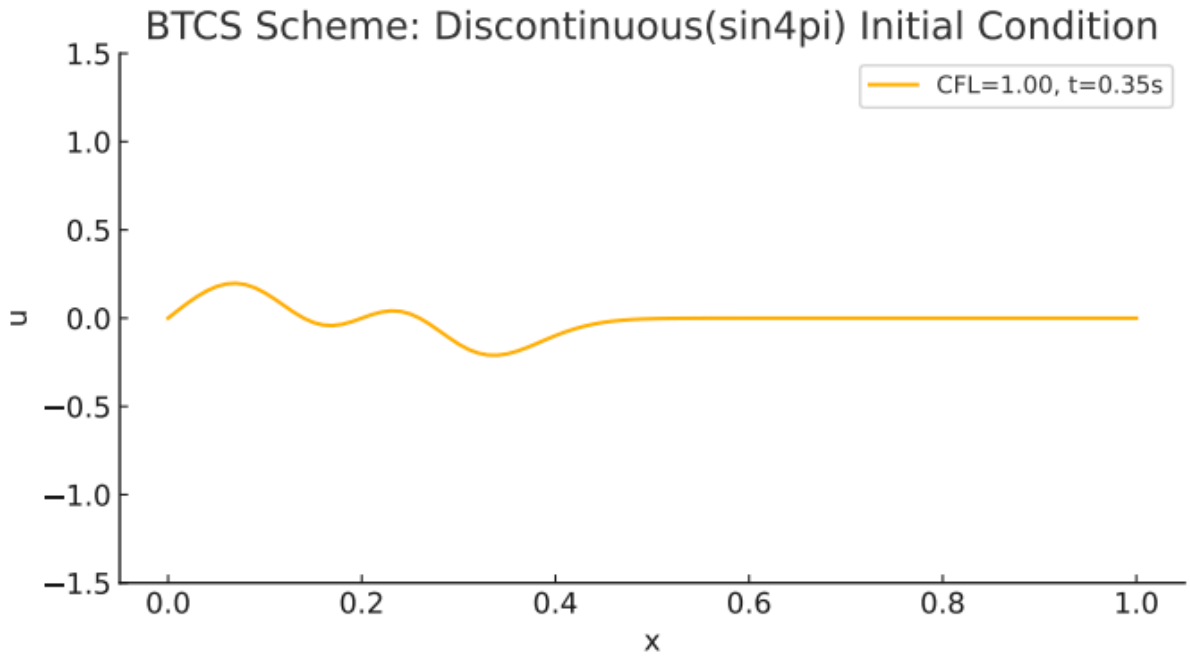


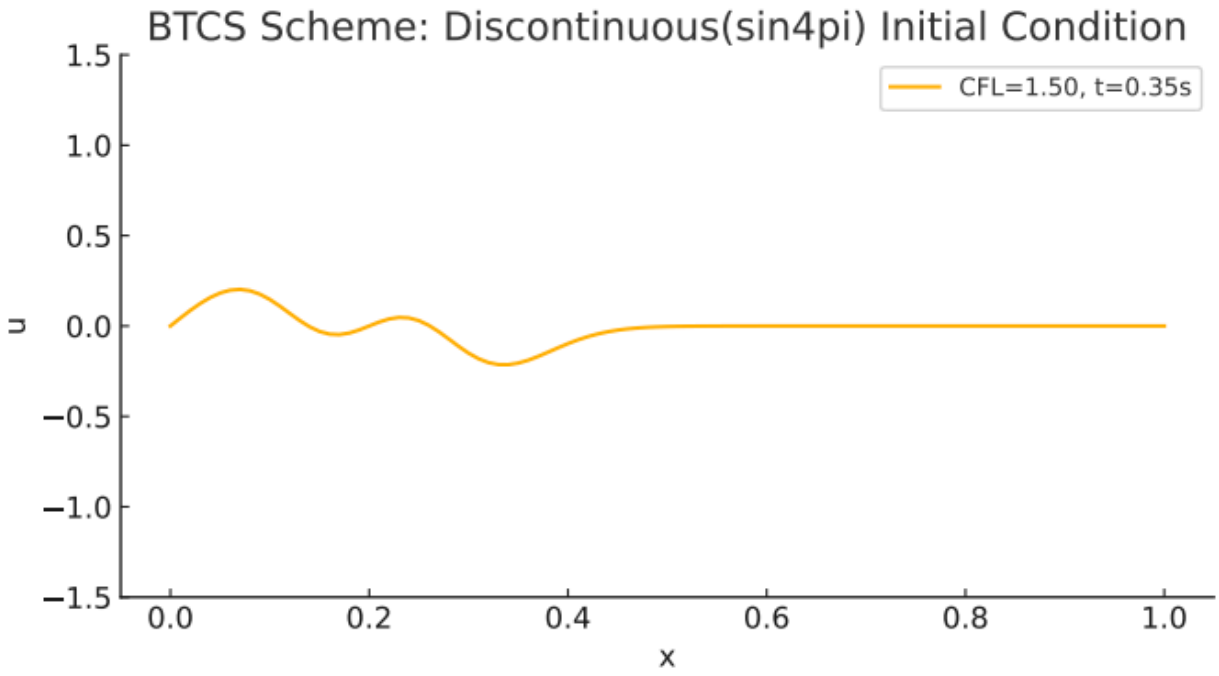




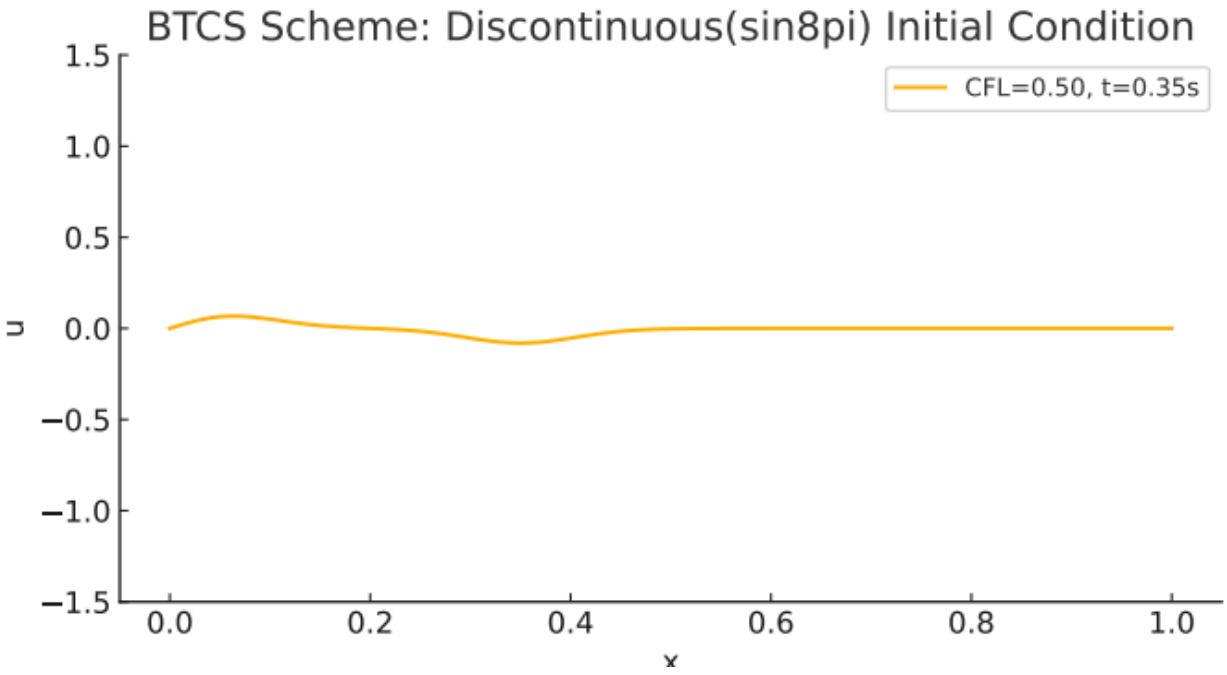
b) For 2nd Condition EAT

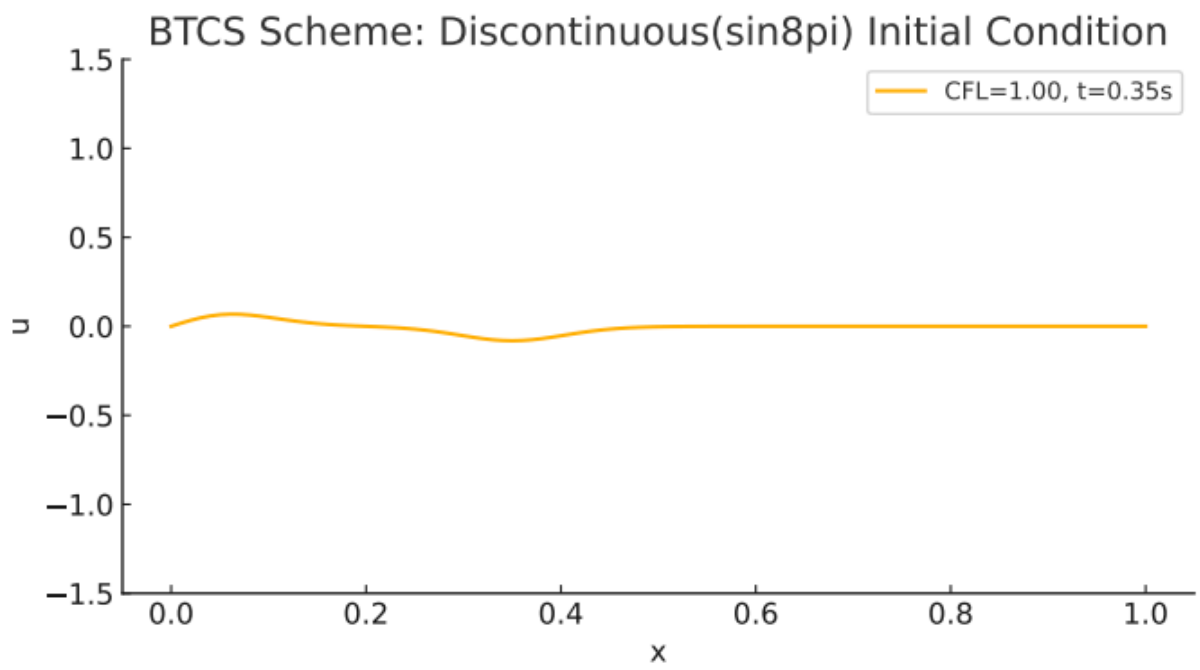

BTCS Scheme: Discontinuous(sin4pi) Initial Condition
CFL=0.50, t=0.35s
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

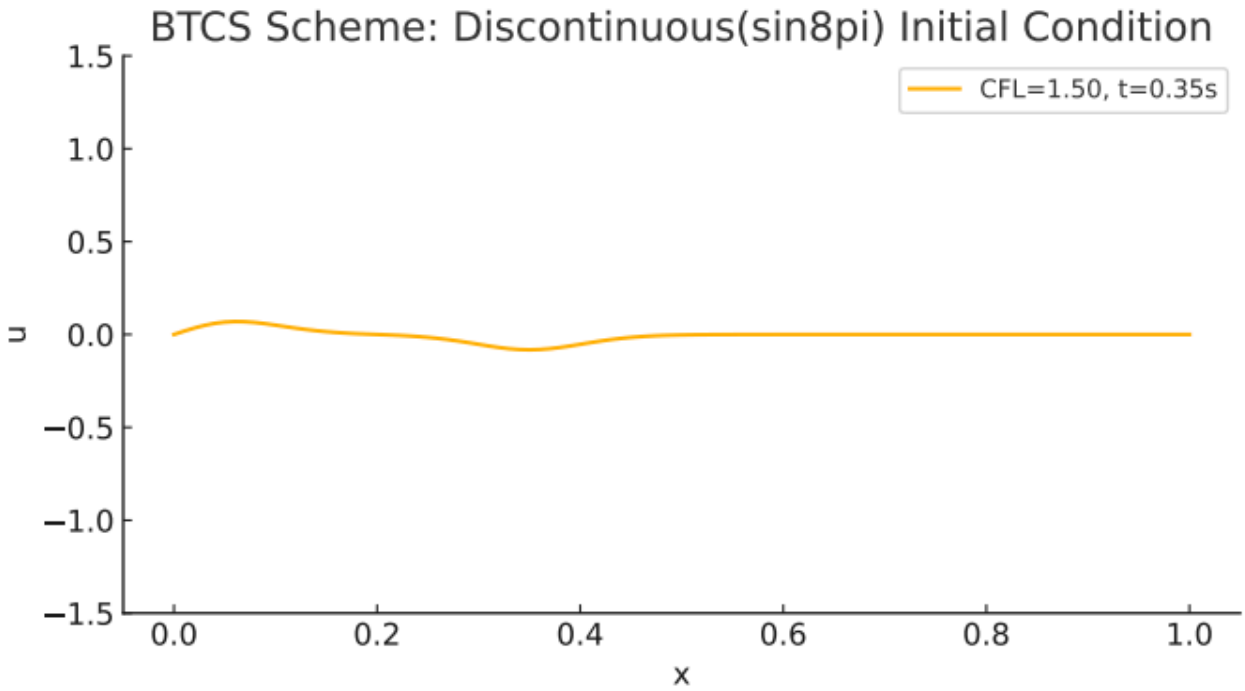

BTCS Scheme: Discontinuous(sin4pi) Initial Condition
CFL=1.00, t=0.35s
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x



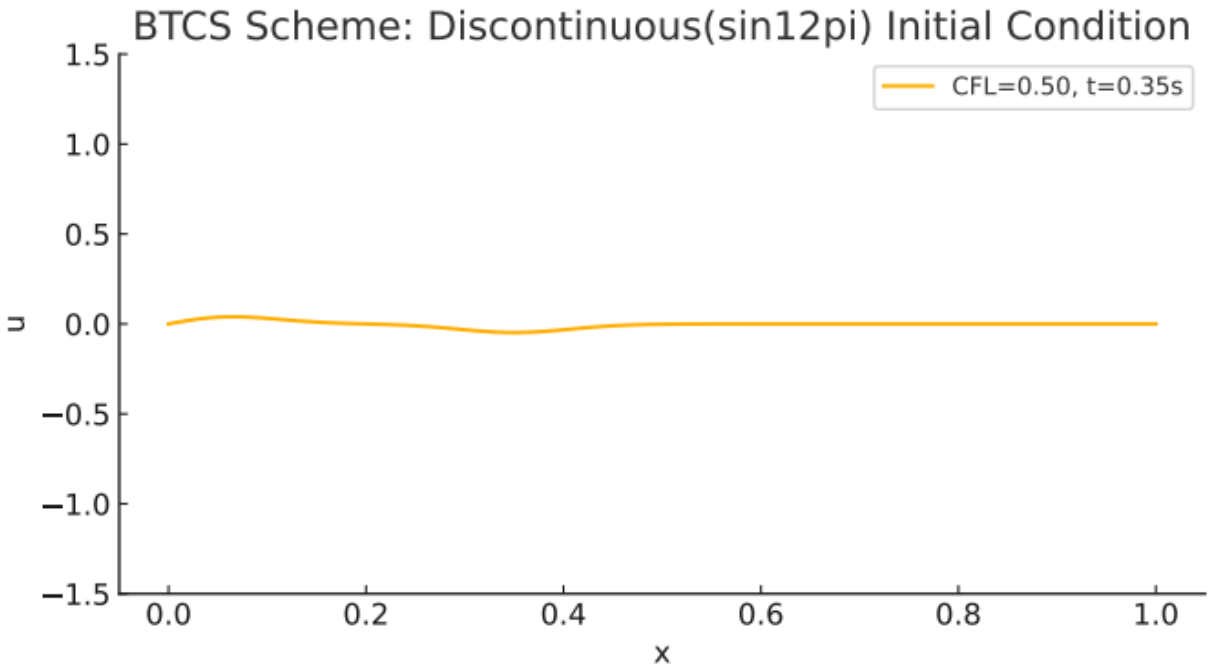
c) For 3rd Condition

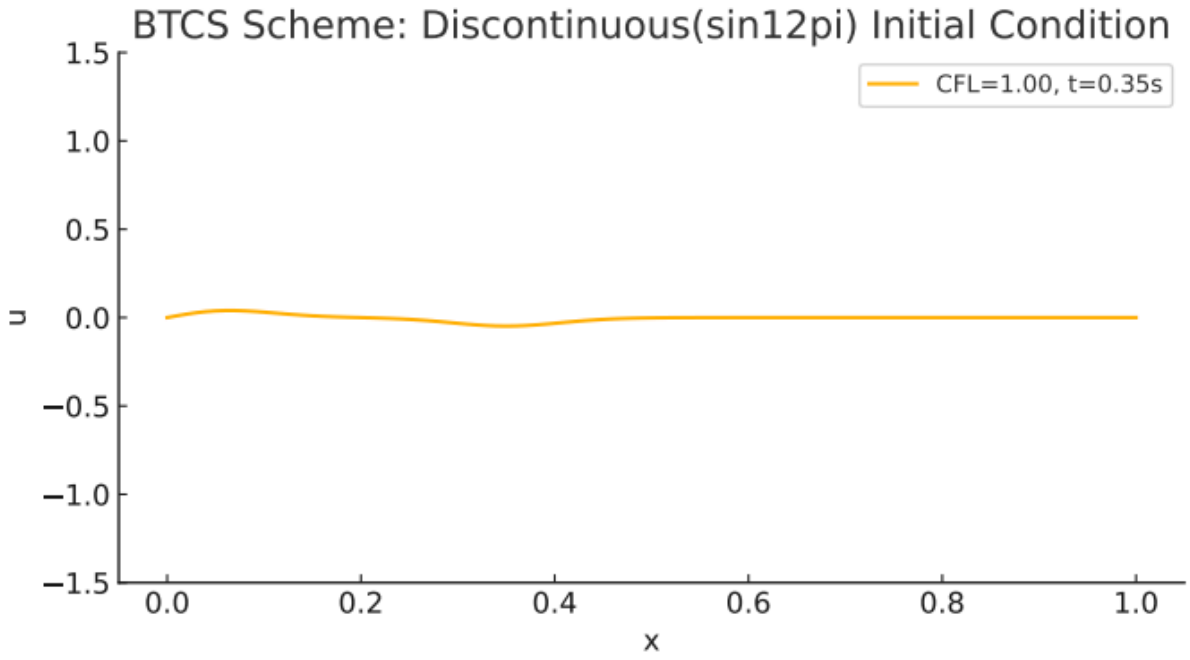


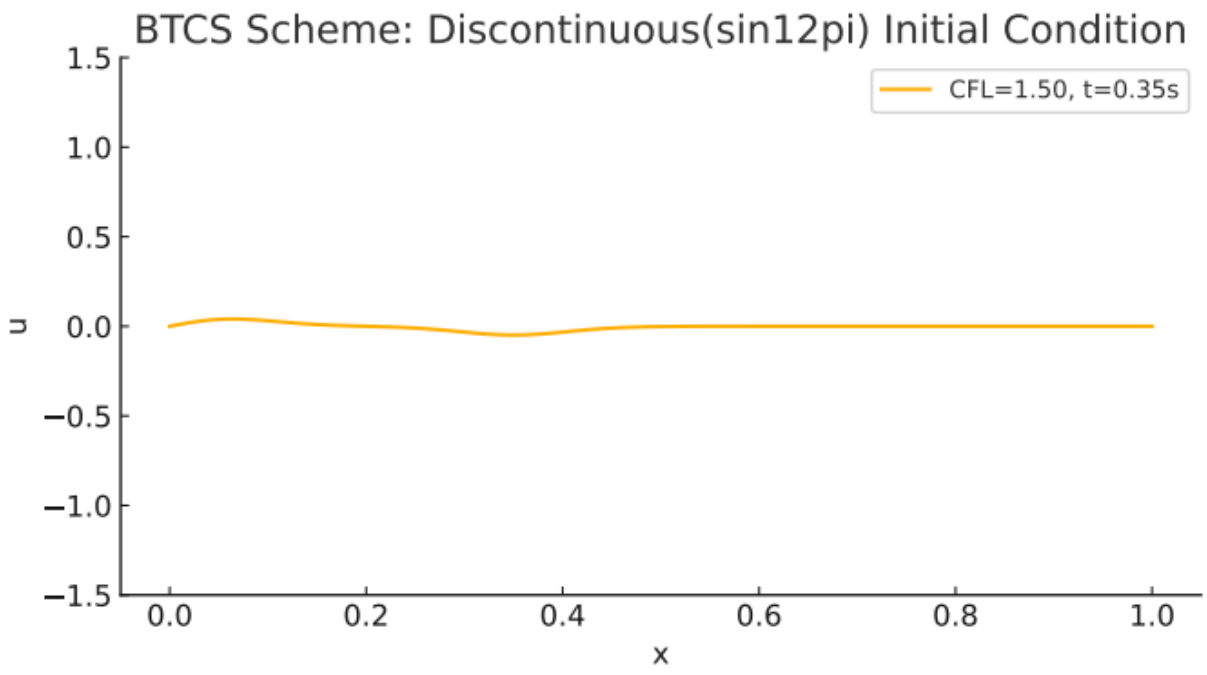




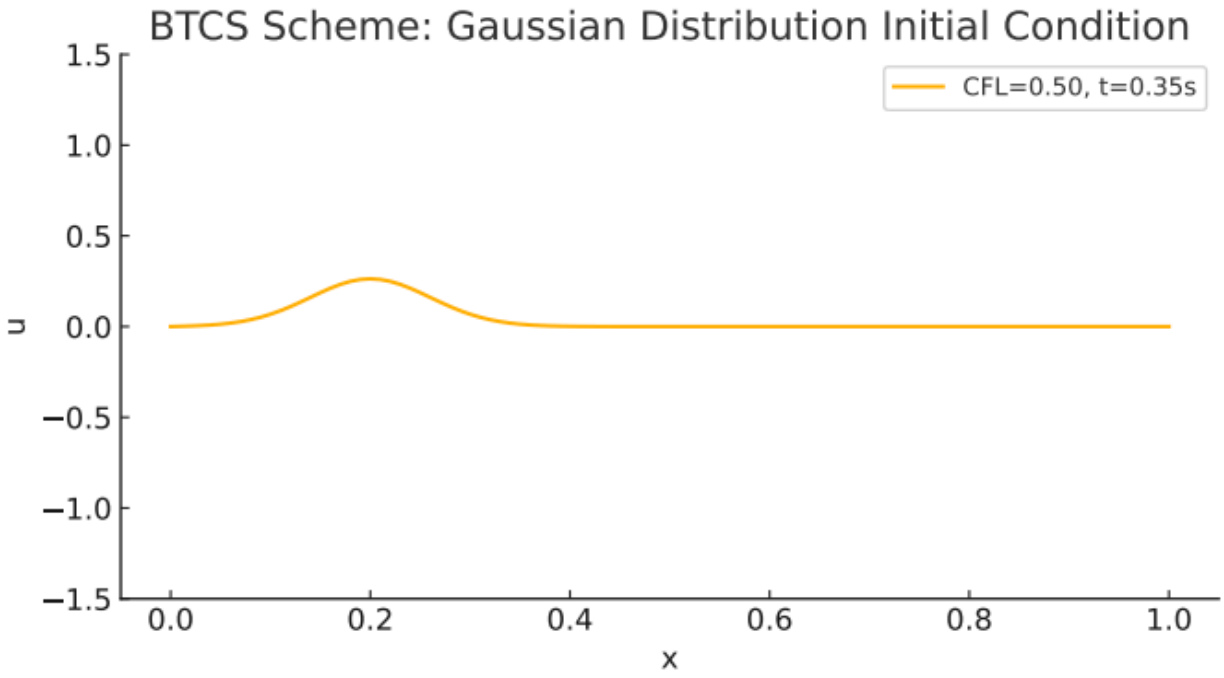
d) For 4th Condition

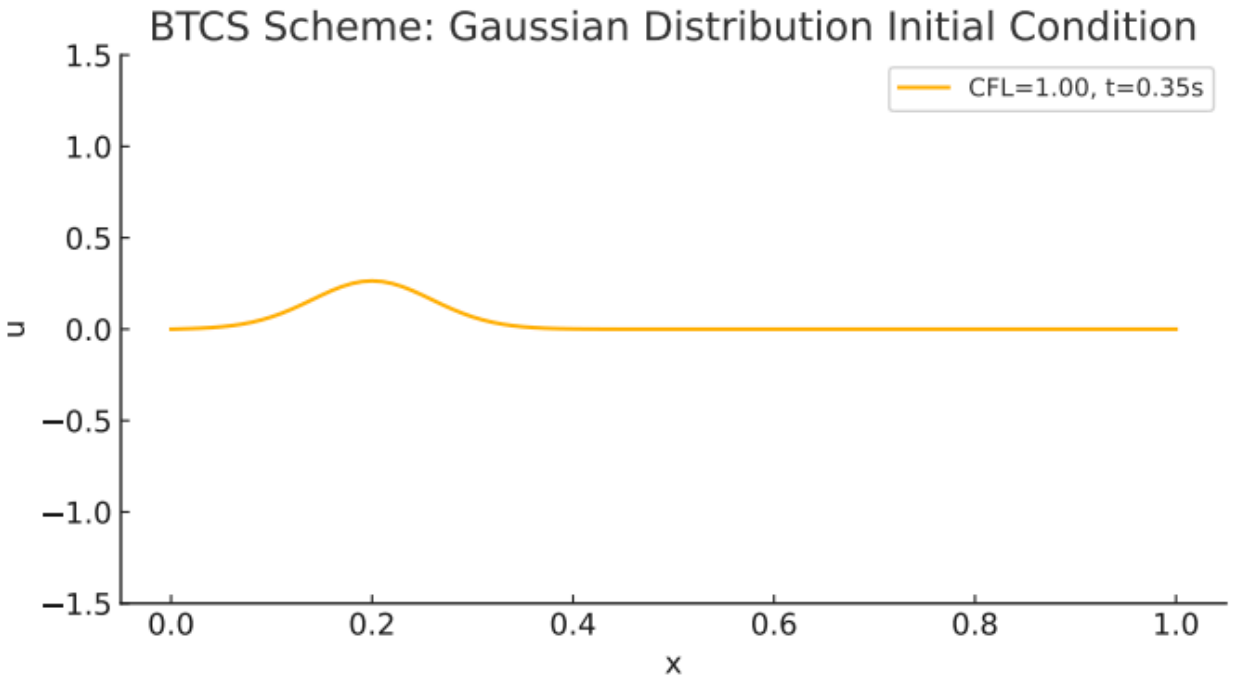

BTCS Scheme: Discontinuous(sin12pi) Initial Condition
CFL=0.50, t=0.35s
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0

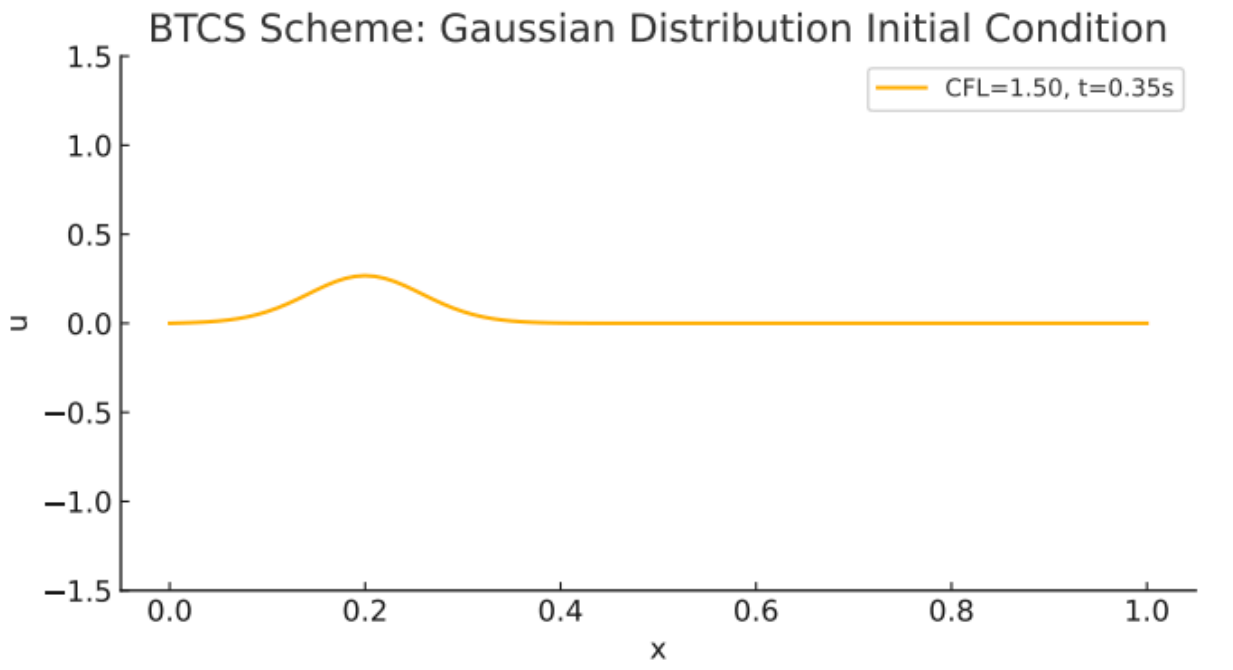

BTCS Scheme: Discontinuous(sin12pi) Initial Condition
CFL=1.00, t=0.35s
u
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.2
0.4
0.6
0.8
1.0



e) For 5th Condition




BTCS Scheme: Gaussian Distribution Initial Condition
CFL=1.00, t=0.35s
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x


BTCS Scheme: Gaussian Distribution Initial Condition
CFL=1.50, t=0.35s
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
u
0.0
0.2
0.4
0.6
0.8
1.0
x

Critical observations

For 1st condition

	FTFS	FTCS	FTBS	LW	BW	FR	BTCS
v=0.5	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1.5	Unstable	Unstable	Unstable	Unstable	Stable	Unstable	Stable

For 2nd condition

	FTFS	FTCS	FTBS	LW	BW	FR	BTCS
v=0.5	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1.5	Unstable	Unstable	Unstable	Unstable	Stable	Unstable	Stable

For 3rd condition

	FTFS	FTCS	FTBS	LW	BW	FR	BTCS
v=0.5	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1.5	Unstable	Unstable	Unstable	Unstable	Stable	Unstable	Stable

For 4th condition

	FTFS	FTCS	FTBS	LW	BW	FR	BTCS
v=0.5	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1.5	Unstable	Unstable	Unstable	Unstable	Stable	Unstable	Stable

For 5th condition

	FTFS	FTCS	FTBS	LW	BW	FR	BTCS
v=0.5	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1	Unstable	Unstable	Stable	Stable	Stable	Stable	Stable
v=1.5	Unstable	Unstable	Unstable	Unstable	Stable	Stable	Stable

From the above tabulated data, the following conclusions can be deduced

1) FTFS and FTCS, both schemes cannot be used to solve the Advection Equation.
2) FTBS is sensitive to Courant number. For the system to be stable, v<1.
3) Lax Wandroff and Fromm Schemes are more sensitive to Courant number than Beam Warming and BTCS scheme.
4) Implicit method(BTCS) is found to be better at solving Advection Equation