# Study of First order Upwind method and Fourth order Runge-Kutta method for solving inviscid Burger's equation

Homework-5

Course: MEEN 489/689 Special topic: Computational Fluid Dynamics

Prepared by,

Yatharth Kishor Vaishnani

327005677

Date: 11/26/2018

# 1 First Order Explicit Upwind Method

## 1.1 Visualization of Wave-Propagation

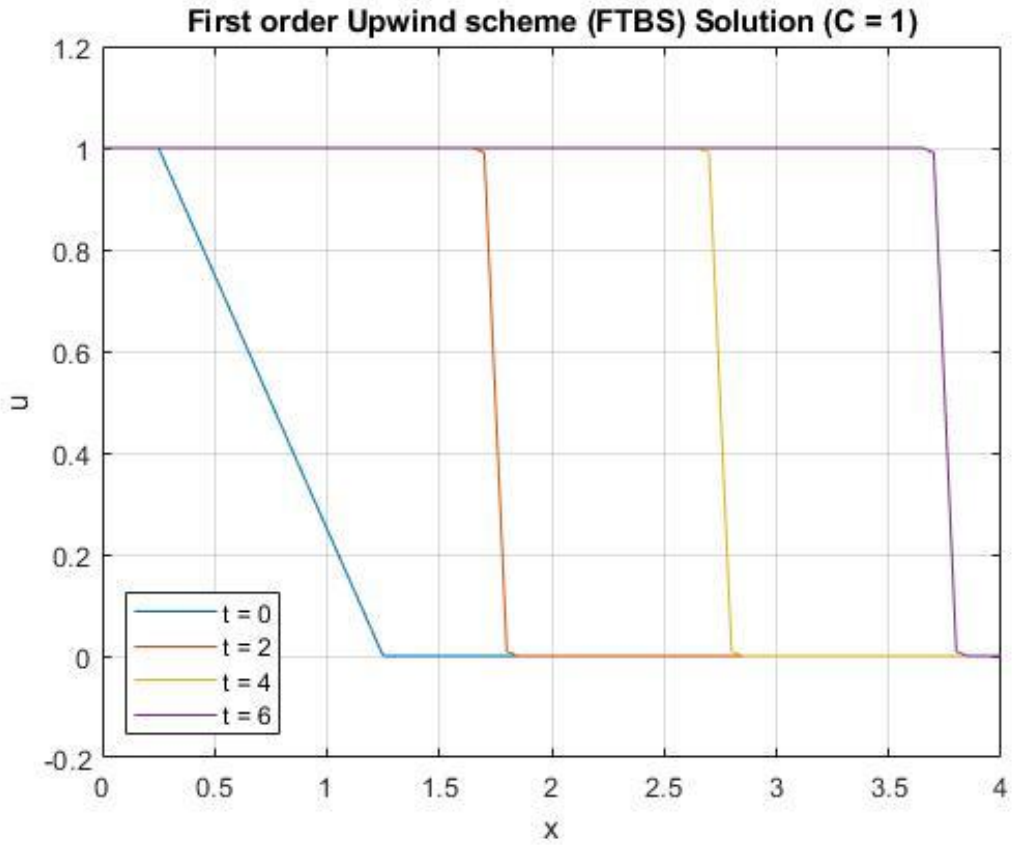


*Figure 1 First order upwind scheme solution at C = 1*

The first order explicit upwind scheme implements the forward marching in time and backward marching in space, both in first order discretization. The results obtained from this scheme is shown in the Fig. (1) for the value of courant number, 1 and for different time values (t = 0, 2, 4 and 6 seconds). Here, for the simplicity of the code, in both the methods, the courant number is defined as $C = {dt}/{dx'}$ as the value of $u_{max}$ = 1. From the Fig. (1), we can say that the results are quite acceptable when compared with the analytical solution of the Burger's equation.

## 1.2 Effect of Courant Number value on Stability

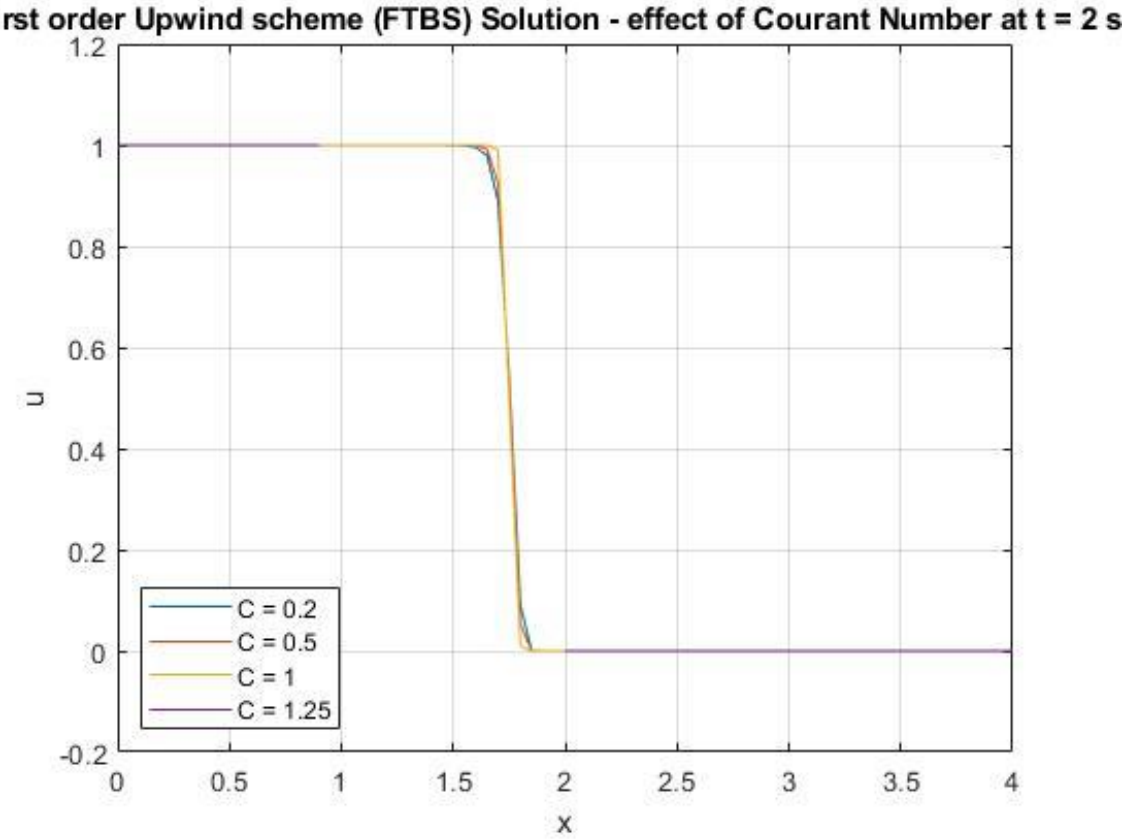


*Figure 2 First order Upwind scheme results at time = 2 second for different Courant number values*

The value of Courant number can be changed by changing the time step for the iteration. The effect of the same is shown graphically in the Fig. (2). From the Fig. (2), we can say that the results are most accurate when C = 1. As the value of Courant number decreases, the results have more dissipation compared to the analytical solution of wave propagation. From the theoretical stability analysis of the first order upwind scheme, the method is conditionally stable for $C \leq 1$. From the Fig. (2), it can be seen that the results for C = 1.25 is not being calculated as the numerical scheme becomes unstable.

# 2 Fourth Order Runge-Kutta Method

## 2.1 Visualization of Wave-Propagation

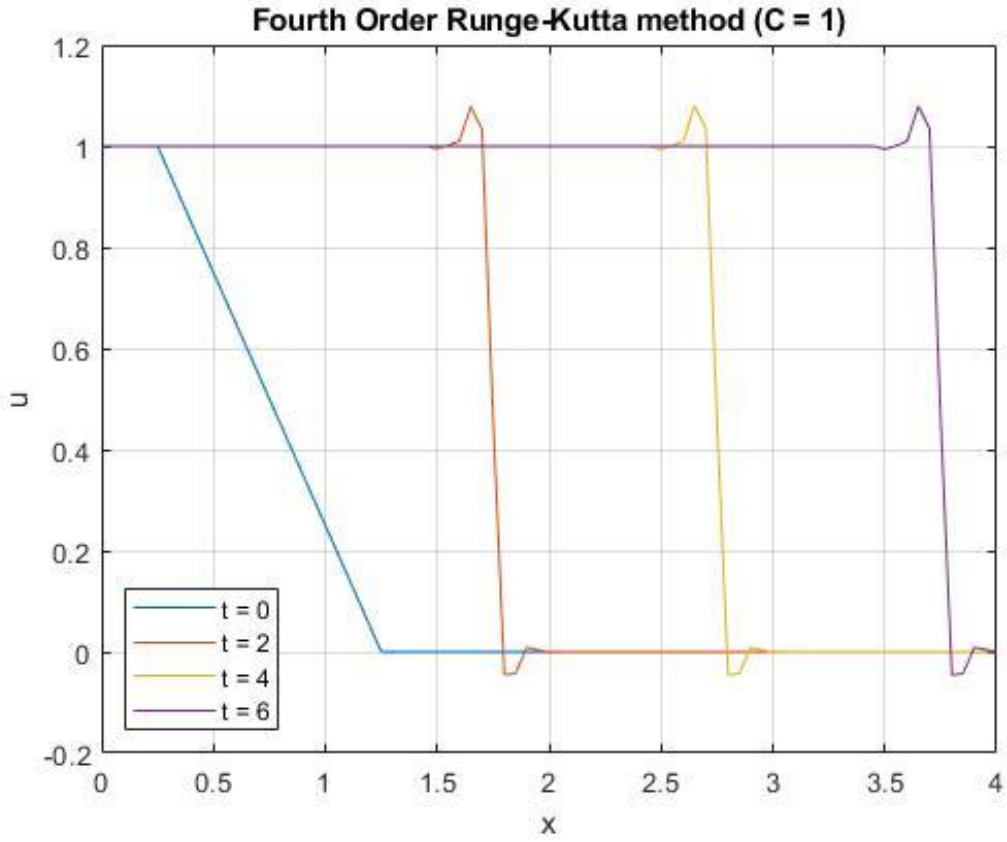


*Figure 3 Fourth order Runge-Kutta method with artificial dissipation (e = 0.1) with C = 1*

The results obtained from the fourth order Runge-Kutta method is presented in the Fig. (3). Here, the results are obtained by adding a fourth order artificial dissipative term in the velocity calculations. The Runge-Kutta method uses the first order central difference scheme for calculating the $\frac{\partial E}{\partial x}$ term in the discretization, which cancel out the dissipative terms (even terms) and the results obtained will have major ringing and will not be acceptable compared to the analytical solution of wave propagation. To avoid this, an artificial dissipative term is added to the calculation of velocity, which removes the ringing. Compared to the results obtained from First order upwind scheme, the R-K method does have slight ringing in the results. Here, in the Fig. (3) and (4), the results are obtained with the value of co-efficient of artificial dissipation e = 0.1.

## 2.2 Effect of Courant Number value on Stability

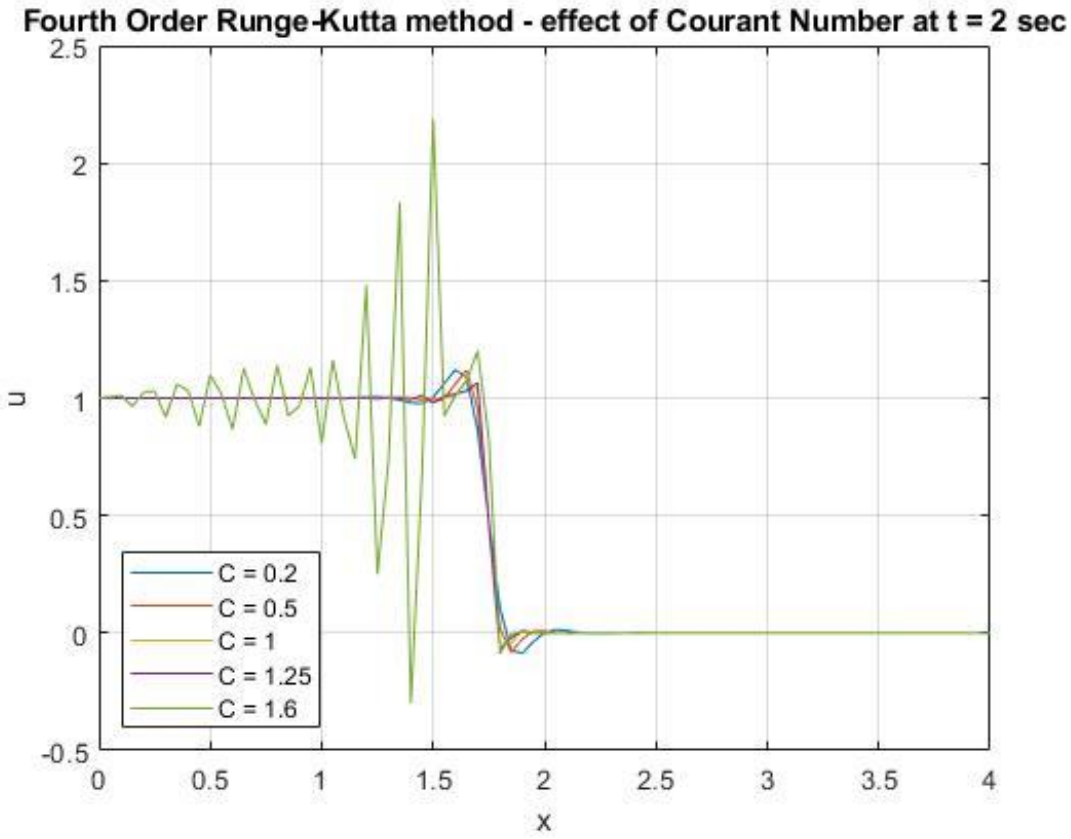


*Figure 4 Fourth order Runge-Kutta method results at time = 2 seconds with different Courant Number and e = 0.1 for fourth order artificial dissipation*

Figure (4) shows the results obtained at different values of Courant number by changing the time step. In the Runge-Kutta method, From the Fig. (4), we can say that as the value of the Courant number increases, the ringing in the results reduces. From the theoretical analysis for the stability of the fourth order Runge-Kutta method is (2√2 = 2.828). But from the numerical calculation for the Burger's equation, the results get more oscillation for C = 1.6.

As we look closely, the increase in the value of Courant number, for the Runge-Kutta method, the amount of ringing is being reduced. This can be seen in the Fig. (5).

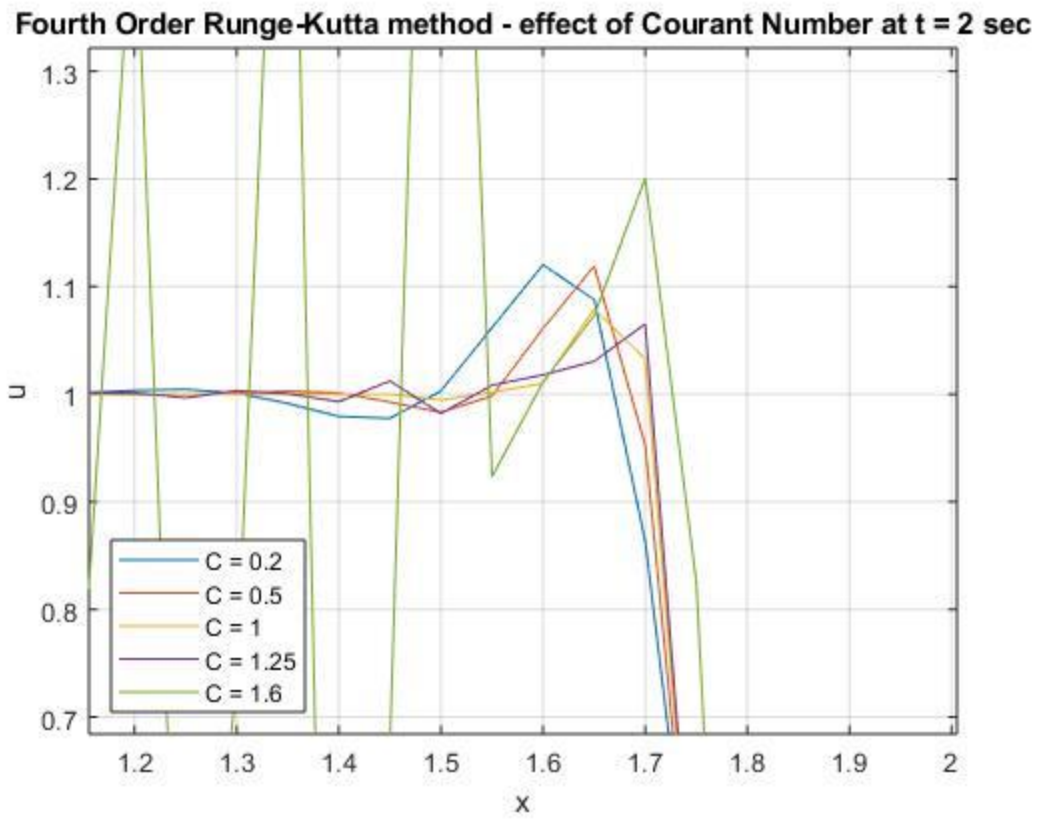


*Figure 5 Zoomed-in part of the Fig. (4) at the point x = 1.6 m and u = 1 m/s.*

# 3 Appendices

## 3.1 First order explicit Upwind scheme

```
clear
L = 4;
dx = 0.05;
x = (0:dx:L);
N = (L/dx)+1;
T = 6;

k = 1;
for dt = [0.01,0.025,0.04,0.05]
    unitstep1 = x<0.25;
    unitstep2 = x<=1.25;
    u(:,1) = unitstep1 + ((1.25 - x).*(unitstep2 - unitstep1));
    figure(k)
    plot(x,u)
    hold on
    n = (T/dt);
    C = dt/dx;
    for i = 1:n
        E(:,1) = (u(:,1).^2)/2;
        for j = 2:N
        u1(j,1) = u(j,1) - C*(E(j,1) - E(j-1,1));
        end
        u = u1;
        u(1,1) = 1;
        u(N,1) = 0;

        time = dt*i;
        if time == 2 || time == 4 || time == 6
            plot(x,u)
            hold on
        end
    end
    title(['First order Upwind scheme (FTBS) Solution (C =
  ',num2str(C),')'])
    grid on
    xlabel('x')
    ylabel('u')
    ylim([-0.2,1.2])
    legend({'t = 0','t = 2','t = 4','t = 6'},'Location','southwest')
    hold off
    k = k + 1;
end
```

## 3.2 Stability of Upwind scheme

```
clear
L = 4;
dx = 0.05;
x = (0:dx:L);
N = (L/dx)+1;
T = 6;

for dt = [0.01,0.025,0.05,0.0625]
    unitstep1 = x<0.25;
    unitstep2 = x<=1.25;
    u(:,1) = unitstep1 + ((1.25 - x).*(unitstep2 - unitstep1));

    n = (T/dt);
    C = dt/dx;
    for i = 1:n
        E(:,1) = (u(:,1).^2)/2;
        for j = 2:N
        u1(j,1) = u(j,1) - C*(E(j,1) - E(j-1,1));
        end
        u = u1;
        u(1,1) = 1;
        u(N,1) = 0;

        time = dt*i;
        if time == 2
            figure(1)
            plot(x,u)
            hold on
        end
    end
end
title('First order Upwind scheme (FTBS) Solution - effect of Courant
  Number at t = 2 sec')
grid on
xlabel('x')
ylabel('u')
ylim([-0.2,1.2])
legend({'C = 0.2','C = 0.5','C = 1','C =
  1.25'},'Location','southwest')
hold off
```

## 3.3 Fourth order Runge-Kutta method

```
clear
L = 4;
dx = 0.05;
x = (0:dx:L);
N = (L/dx)+1;
T = 6;

k = 1;
for dt = [0.01,0.025,0.04,0.05]
    unitstep1 = x<0.25;
    unitstep2 = x<=1.25;
    u(:,1) = unitstep1 + ((1.25 - x).*(unitstep2 - unitstep1));
    u1 = u;
    figure(k)
    plot(x,u)
    hold on
    n = (T/dt);
    C = dt/dx;
    for i = 1:n
        for l = 1:4
            Ex = Exi(u1,N,dx);
            u1(:,1) = u(:,1) - (1/(5-l))*dt*(Ex(:,1));
        end
        u1(:,1) = u1(:,1) + Damp(u,N,0,0.1);
        u = u1;
        u(1,1) = 1;
        u(N,1) = 0;

        time = dt*i;
        if time == 2 || time == 4 || time == 6
            plot(x,u)
            hold on
        end
    end
    title(['Fourth Order Runge-Kutta method (C = ',num2str(C),')'])
    grid on
    xlabel('x')
    ylabel('u')
    legend({'t = 0','t = 2','t = 4','t = 6'},'Location','southwest')
    hold off
    k = k + 1;
end
```

## 3.4 Calculation of $\frac{\partial E}{\partial x}$

```
%Ex calculator
function [Ex] = Exi(u1,N,dx)

for i = 2:N-1
    Ex(i,1) = ((u1(i+1,1).^2) - (u1(i-1,1).^2))/(4*dx);
end
Ex(N,1) = 0;
```

## 3.5 Calculation of artificial dissipation

```
%D calculator
function [D] = Damp(u1,N,e1,e2)

for i = 3:N-2
    D(i,1) = e1*(u1(i+1,1) - 2*u1(i,1) + u1(i-1,1)) - e2*(u1(i+2,1) -
4*u1(i+1,1) + 6*u1(i,1) - 4*u1(i-1,1) + u1(i-2,1));
end

D(1,1) = e1*(u1(2,1) - 2*u1(1,1) + u1(1,1)) - e2*(u1(3,1) - 4*u1(2,1)
+ 6*u1(1,1) - 4*u1(1,1) + u1(1,1));
D(2,1) = e1*(u1(3,1) - 2*u1(2,1) + u1(1,1)) - e2*(u1(4,1) - 4*u1(3,1)
+ 6*u1(2,1) - 4*u1(1,1) + u1(1,1));
D(N-1,1) = e1*(u1(N,1) - 2*u1(N-1,1) + u1(N-2,1)) - e2*(u1(N,1) -
4*u1(N,1) + 6*u1(N-1,1) - 4*u1(N-2,1) + u1(N-3,1));
D(N,1) = e1*(u1(N,1) - 2*u1(N,1) + u1(N-1,1)) - e2*(u1(N,1) -
4*u1(N,1) + 6*u1(N,1) - 4*u1(N-1,1) + u1(N-2,1));
end
```

## 3.6 Stability of Runge-Kutta method

```
clear
L = 4;
dx = 0.05;
x = (0:dx:L);
N = (L/dx)+1;
T = 6;

for dt = [0.01,0.025,0.05,0.0625,0.08]
    unitstep1 = x<0.25;
    unitstep2 = x<=1.25;
    u(:,1) = unitstep1 + ((1.25 - x).*(unitstep2 - unitstep1));
    u1 = u;
    n = (T/dt);
    C = dt/dx;
    for i = 1:n
        for l = 1:4
            Ex = Exi(u1,N,dx);
```

```
            u1(:,1) = u(:,1) - (1/(5-l))*dt*(Ex(:,1));
        end
        u1(:,1) = u1(:,1) + Damp(u,N,0,0.1);
        u = u1;
        u(1,1) = 1;
        u(N,1) = 0;

        time = dt*i;
        if time == 2
            figure(1)
            plot(x,u)
            hold on
        end
    end
end
grid on
title('Fourth Order Runge-Kutta method - effect of Courant Number at t
= 2 sec')
xlabel('x')
ylabel('u')
legend({'C = 0.2','C = 0.5','C = 1','C = 1.25','C =
1.6'},'Location','southwest')
hold off
```