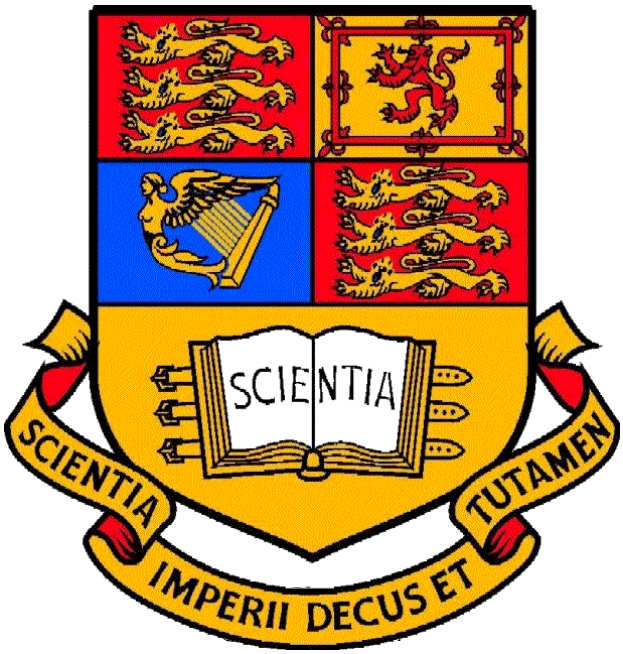
Computational Fluid Dynamics

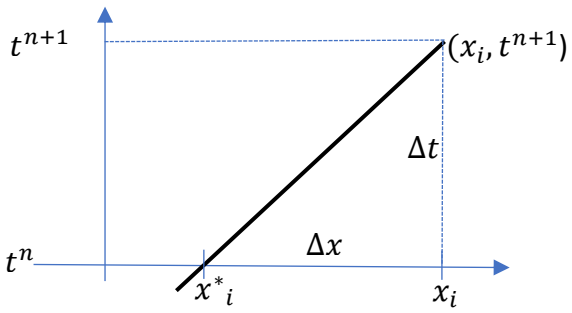
Coursework 1



Student: Mr Xerxes Chong Xian

Submission Date: 25th November 2019

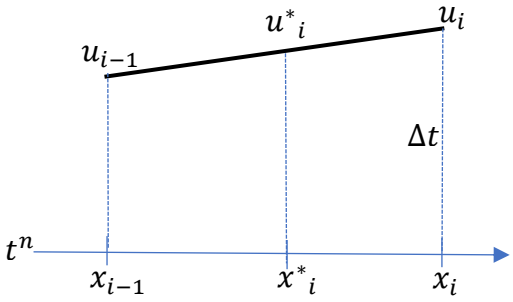
Question 1



Starting off by graphically representing the Semi-Lagrangian method, where the black line on the graph represents the characteristic line of the problem. The assumption is that along the characteristic line, the value of the solution $u(x,t)$ is constant, hence $u(x^*_i, t^n) = u(x_i, t^{n+1})$. a is the speed of propagation of the solution. The gradient of the characteristic line is therefore the speed of propagation $a = \frac{\Delta x}{\Delta t}$. From the graph above, the following relation between x^*_i and x_i can be deduced:

$$
\begin{aligned}
x^*_i &= x_i - \Delta x \\
&= x_i - [\text{Change in x as } t^n \rightarrow t^{n+1}] \\
&= x_i - \left[\frac{\Delta x}{\Delta t}\,\Delta t\right] \\
&= x_i - a\Delta t
\end{aligned}
$$

Question 2



We will use linear interpolation to find the value of $u(x^*_i, t^n)$ at the departure point. Graphically representing the interpolation, the following equation is obtained:

$$
\begin{aligned}
u^*_i &\approx u_{i-1} + \left[\frac{Change\ in\ u\ as\ x_{i-1} \rightarrow x_i}{Change\ in\ x\ from\ x_{i-1} \rightarrow x_i}\right] [Change\ in\ x\ from\ x_{i-1} \rightarrow x^*_i] \\
&= u_{i-1} + \left[\frac{u_i - u_{i-1}}{x_i - x_{i-1}}\right] [x^*_i - x_{i-1}] \\
&= u_{i-1} + \left[\frac{x^*_i - x_{i-1}}{x_i - x_{i-1}}\right] [u_i - u_{i-1}]
\end{aligned}
$$

Question 3

$$u^*_i = u_{i-1} + \left[\frac{x^*_i - x_{i-1}}{x_i - x_{i-1}}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + \left[\frac{(x_i - a\Delta t) - x_{i-1}}{x_i - x_{i-1}}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + \left[\frac{x_i - x_{i-1} - a\Delta t}{\Delta x}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + \left[\frac{\Delta x - a\Delta t}{\Delta x}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + \left[1 - \frac{a\Delta t}{\Delta x}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + [1 - \lambda][u_i - u_{i-1}] \qquad where\ \ Courant\ Number, \lambda = \frac{a\Delta t}{\Delta x}$$

$$u_i = \left[\frac{1}{\lambda - 1}\right][\lambda u_{i-1} - u^*_i] \qquad Solving \text{ for } u_i \ yields\ an\ upwind\ scheme$$

Starting from the linear interpolated form of u^*_i from Question 2, we can substitute the definition for x^*_i as defined by the Semi-Lagrangian method and obtain an equation involving the Courant Number, λ. Re-arranging equation to solve for u_i, it can be shown that at any given time, the value of u_i obtained from the linear interpolation, is a function of the value of u_{i-1} and the departure point u^*_i. These means that u_i is dependent on values upwind of it and thus, is equivalent to the upwind scheme.

Question 4

The Courant-Friedrichs-Lewy (CFL) condition states that the domain of dependence of the numerical scheme must contain the domain of dependence of the physical solution.
Given a speed of propagation, a, the real solution will move a distance of $a\Delta t$. In the same time, the numerical scheme can move a distance of Δx. Applying the CFD condition, this indicates that:

$$a\Delta t \leq \Delta x$$

$$x_i - x^*_i \leq \Delta x$$

$$x_i - \Delta x \leq x^*_i$$

$$x^*_i \geq x_i - \Delta x$$

$$x^*_i \geq x_{i-1}$$

This assumption is only valid assuming the intervals between each special point x_i is a constant Δx.

Question 5

5.1

Assuming the CFL condition has been satisfied, this indicates:

$$u^*_i = u_{i-1} + \left[\frac{x^*_i - x_{i-1}}{x_i - x_{i-1}}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + \left[\frac{\beta}{\Delta x}\right][\Delta u] \quad \text{where } \beta = x^*_i - x_{i-1} \text{ and } \Delta u = u_i - u_{i-1}$$

$$= u_{i-1} + 0 \qquad if\ x^*_i = x_{i-1}$$

$$= u_{i-1}$$

$$\therefore u^*_i \geq u_{i-1}$$

If $\frac{\beta}{\Delta x}$ is equal to 1, $u^*_i = u_i$ where $x^*_i = x_i$. Since $x^*_i \not> x_i$ as defined by the upwind nature of the Semi-Lagrangian method due to the usage of the stencil points (x_{i-1}, x_i), $\frac{\beta}{\Delta x} \not> 1$ and $u^*_i \not> u_i$. Hence the range of u is:

$$u_{i-1} \leq u^*_i \leq u_i$$

5.2

From Question 3

$$u_i = \left[\frac{1}{\lambda - 1}\right][\lambda u_{i-1} - u^*_i]$$

$$u^*_i = \lambda u_{i-1} + u_i(1 - \lambda)$$

$$u^{n+1}_i = \lambda u^n_{i-1} + u^n_i(1 - \lambda)$$

Using the triangle inequality $|a + b| \leq |a| + |b|$,

$$|u^{n+1}_i| \leq |\lambda u^n_{i-1}| + |u^n_i(1 - \lambda)|$$

$$|u^{n+1}_i| \leq |\lambda||u^n_{i-1}| + |(1 - \lambda)||u^n_i|$$

If $|u^n_{max}| = max_j|u^n_j|$

$$|u^{n+1}_i| \leq |\lambda||u^n_{max}| + |(1 - \lambda)||u^n_{max}|$$

$$\leq [|\lambda| + |(1 - \lambda)|]|u^n_{max}|$$

Since by the CFL condition, $\lambda \leq 1$

$$|u^{n+1}_i| \leq [\lambda + 1 - \lambda]|u^n_{max}|$$

$$|u^{n+1}_i| \leq [1]|u^n_{max}|$$

$$|u^{n+1}_i| \leq |u^n_{max}|$$

5.3

$$|u^{n+1}{}_i| \leq |u^n{}_{max}|$$

$$\frac{|u^{n+1}{}_i|}{|u^n{}_{max}|} \leq 1$$

This indicates that across all values of x at time t^{n+1}, it will always be less than or equal to the maximum value of u at t^n, this indicates that the solution is stable and the values are not "blowing up" across all grid points as time progresses.

Question 6

Assuming the CFL condition is not satisfied, this indicates $\lambda > 1$. Hence starting from the equation in Question 3

$$u^*{}_i = \lambda u_{i-1} + u_i(1-\lambda)$$

Using the triangle inequality $|a + b| \leq |a| + |b|$,

$$|u^{n+1}{}_i| \leq |\lambda u^n{}_{i-1}| + |u^n{}_i(1-\lambda)|$$

$$|u^{n+1}{}_i| \leq |\lambda||u^n{}_{i-1}| + |(1-\lambda)||u^n{}_i|$$

If $|u^n{}_{max}| = max_j|u^n{}_j|$

$$\begin{aligned} |u^{n+1}{}_i| &\leq |\lambda||u^n{}_{max}| + |(1-\lambda)||u^n{}_{max}| \\ &\leq [|\lambda| + |(1-\lambda)|]|u^n{}_{max}| \\ &\leq [\lambda + (\lambda - 1)]|u^n{}_{max}| \quad \text{since } (1-\lambda) < 1 \\ &\leq [2\lambda - 1]|u^n{}_{max}| \end{aligned}$$

$$\frac{|u^{n+1}{}_i|}{|u^n{}_{max}|} \leq \alpha \qquad where\ \alpha = 2\lambda - 1 > 1\ since\ \lambda > 1$$

There exists a case where $\frac{|u^{n+1}{}_i|}{|u^n{}_{max}|} = \alpha$. Since $\alpha > 1$, this indicates that $|u^{n+1}{}_i| > |u^n{}_{max}|$. Similar to question 6, this indicates the possibility that across all grid points at time t^{n+1}, the value of the solution u will be greater than the maximum value of u at the previous time step t^n, this indicates that the solution is unstable as the solution will "blow up" across all grid points as time progresses.

Question 7

When the Courant Number, $\lambda = \frac{a\Delta t}{\Delta x} = 1$

$$\begin{aligned} x^*{}_i &= x_i - a\Delta t \\ &= x_i - \Delta x \\ &= x_{i-1} \end{aligned}$$

Substituting this into the linear interpolated value of u

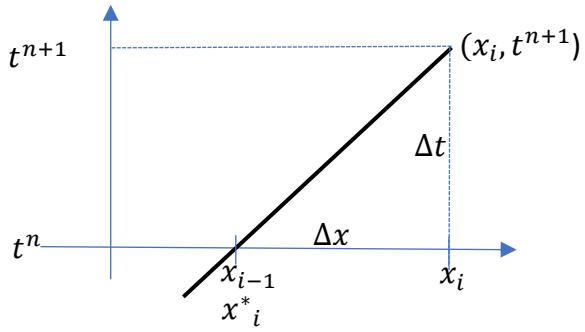
$$u^*_i = u_{i-1} + \left[\frac{x^*_i - x_{i-1}}{x_i - x_{i-1}}\right][u_i - u_{i-1}]$$

$$= u_{i-1} + \left[\frac{x_{i-1} - x_{i-1}}{x_i - x_{i-1}}\right][u_i - u_{i-1}]$$

$$= u_{i-1}$$

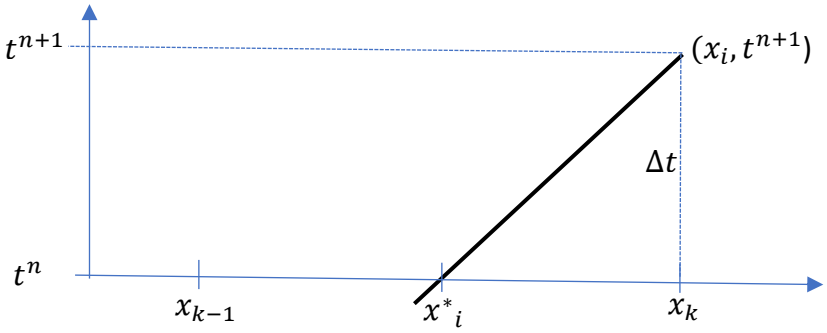
$$\therefore u^*_i = u^{n+1}{}_i = u^n{}_{i-1}$$

The departure point coincides with the previous grid point.



This indicates that with known initial starting conditions, there the linear interpolation performed will yield an exact value that is known was from the previous timestep, hence there is method corresponds to the exact solution. The numerical scheme is propagating at the exact speed as the physical solution as $a\Delta t = \Delta x$, the domain of dependence of the numerical scheme is exactly the domain of dependence of the physical solution. Hence there the method has zero diffusion and dispersion error.

Question 8



Given the new stencil between which to perform the linear interpolation (x_{k-1}, x_k) and defining $x_i = i\Delta x$, then the numerical domain of dependence now contains the departure point.

Now $x^*_i = x_i - a\Delta t = i\Delta x - a\Delta t$,

$$u^*_i = u_{k-1} + \left[\frac{x^*_i - x_{k-1}}{x_k - x_{k-1}}\right][u_k - u_{k-1}]$$

$$= u_{k-1} + \left[\frac{i\Delta x - a\Delta t - (k-1)\Delta x}{k\Delta x - (k-1)\Delta x}\right][u_k - u_{k-1}]$$

$$= u_{k-1} + \left[\frac{i\Delta x - \lambda\Delta x - (k-1)\Delta x}{\Delta x}\right][u_k - u_{k-1}]$$

$$= u_{k-1} + [i - \lambda - k + 1][u_k - u_{k-1}]$$

$$= -(i-k)u_{k-1} + (i-k+1)u_k - \lambda(u_k - u_{k-1})$$

Question 9

The Semi-Lagrangian method is such that the numerical domain of dependence has been selected between (x_{k-1}, x_k) and will always include the physical domain of dependence. For Courant Numbers greater than 1, the physical domain of dependence falls outside the numerical domain of dependence (i.e. the departure point lies outside the numerical domain of dependence). By performing interpolation using the surrounding grid point values to obtain the value of the departure point, this ensures that the physical domain of dependence falls within the numerical domain of dependence. Hence the scheme by way of its interpolation, automatically satisfies the condition for stability, allowing any value of the Courant Number to be used.

Question 10

Given the inequality relationship defined in Question 10, the following relationship between i, k and λ can be obtained:

$$x_{k-1} \leq x_i - a\Delta t \leq x_k$$

$$(k-1)\Delta x \leq x_i - a\Delta t \leq k\Delta x$$

$$(k-1) \leq \frac{x_i - a\Delta t}{\Delta x} \leq k$$

$$(k-1) \leq \frac{i\Delta x - a\Delta t}{\Delta x} \leq k$$

$$(k-1) \leq i - \lambda \leq k$$

$$k \leq i - \lambda + 1 \leq k$$

From this relation, the value of k can be evaluated at each value of i using the floor function of MatLab.
To account for the periodic boundary conditions, values of $k \leq 1$ are adjusted by adding 100, as seen in Table 1.
For example:

$k = -5$ This value of subscript is unreadable by MatLab

$k = k + 100$

$= 95$

$\therefore u_k = u_{-5} = u_{95}$

u_{98}	u_{99}	u_{100}	u_{101}	u_{102}	u_{103}	u_{104}
u_{-2}	u_{-1}	u_0	u_1	u_2	u_3	u_4

Table 1. Visualisation of periodic boundary conditions for MatLab implementation

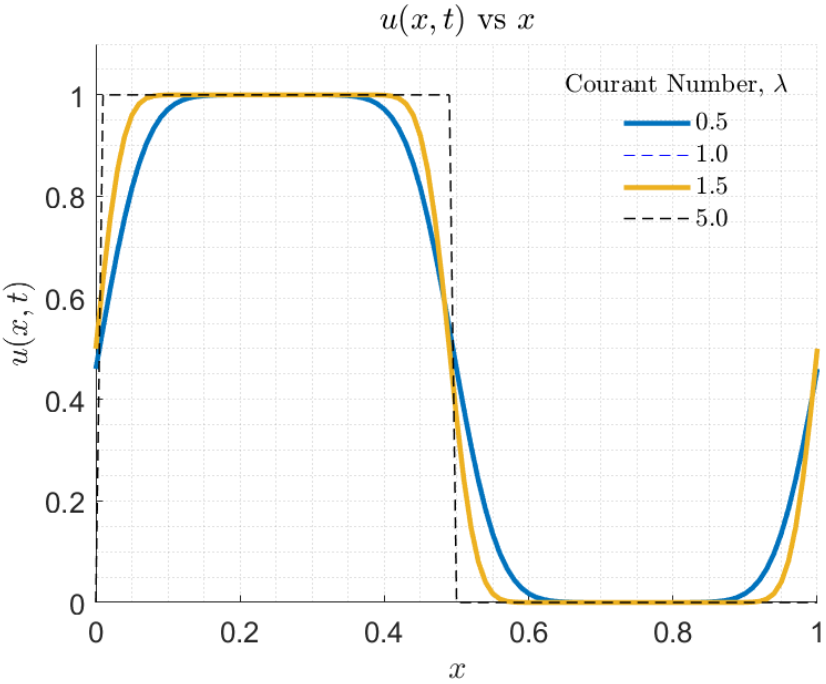


Figure 1. Graph of solutions at various Courant Numbers

From the graph above and further testing of several Courant Numbers, it is concluded that from the given Courant Numbers, the integer values of 1.0 and 5.0, the solution will be identical to the square wave seen in the graph (the black and blue dashed lines have the exact same values). The lower the non-integers values, the less the graph approximates a square wave, which is the analytical solution. The lowest non-integer value of the Courant Number at 0.5 has the highest amount of diffusion error. This is confirmed by the graph below, where for phase angles not equal to zero, a Courant Number of 0.5, denoted by σ, will give the largest amount of diffusion error, ϵ_D

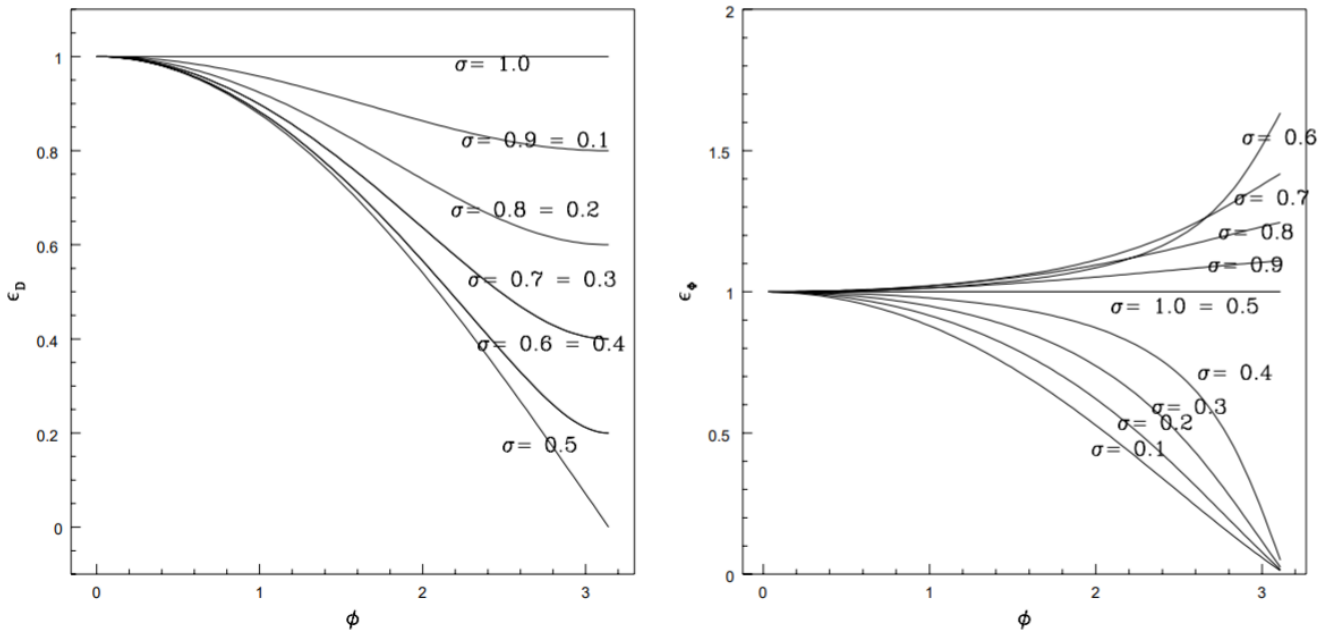


Figure 2. Graphs representing the diffusion errors (left) and the dispersion errors (right) of the linear advection equation for various Courant Numbers and phase angles

Question 11

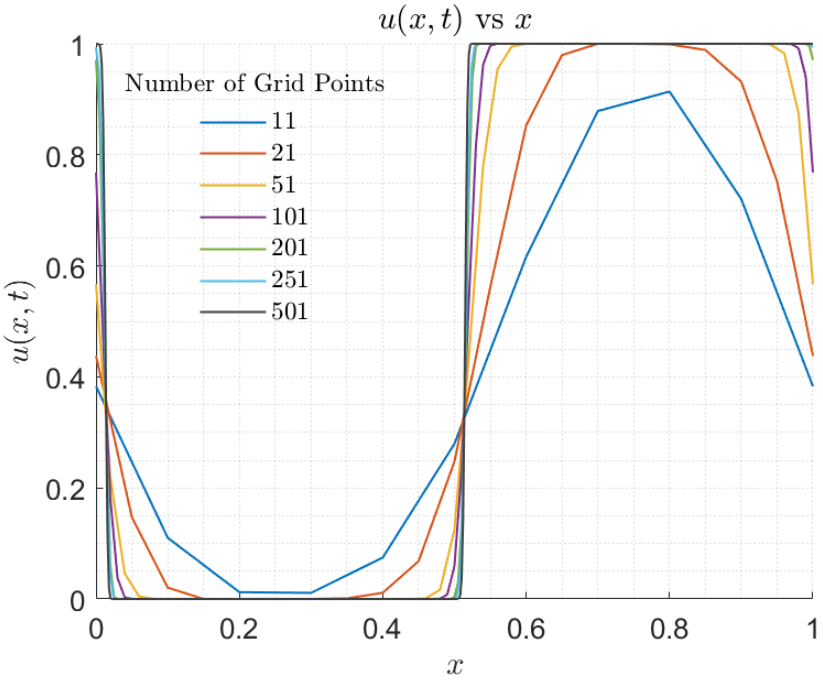


Figure 3. Graph of solution at various number of grid points

With an increasing number of grid points, the solution better approximates the square wave. At a lower number of grid points, the effect of the sine terms dominates and the effect reduces with increasing grid points.

MatLab Code Q10

```
clear
clc
%this script is for solving Question 10 of the CFD coursework
Time =1;
%Constants--------
n=101; %Number of grid points
DeltaX=(1-0)/(n-1); %Range of x from 0 to 1 divided by number of intervals
a=0.5;
lambda=[0.5 1.0 1.5 5.0]; %The Courant-Number
x=[0:1/(n-1):1]; %Discretise x over 101 grid points for 0<x<1
for Q=1:length(lambda)
    DeltaT=lambda(Q)*DeltaX/a;
    tarray=0:DeltaT:Time; %Discretise an array of time value given delta t
    u=zeros([length(tarray),length(x)]); %Create empty matrix of u values
    for r=1:length(x) %Populate u matrix with initial conditions t=0
        if x(r)<=0.5
            u(1,r)=0;
        else
            u(1,r)=1;
        end
    end
    u(1,length(x))=u(1,1); %Enforce periodic condition
    p=2; %Start while loop at the 2nd grid point in time value to enter the while loop
    while p<=length(tarray)
        for i=1:length(x)
            k=floor(i-lambda(Q)+1); %calculate the value of k based on the inequality
relationship
            if k>1
                u(p,i)=-(i-k)*u(p-1,k-1) + (i-k+1)*u(p-1,k)-lambda(Q)*(u(p-1,k)-u(p-1,k-
1));
            else
                k=k+100; %to account for periodic conditions when k<=1 due to presence of
u_(k-1) terms
                u(p,i)=-(i-(k-100))*u(p-1,k-1) + (i-(k-100)+1)*u(p-1,k)-lambda(Q)*(u(p-
1,k)-u(p-1,k-1));
            end
        end
        u(p,length(x))=u(p,1); %Enforcing Periodic conditions
        p=p+1;
    end
U(Q,:)=u(length(tarray),:);
end
hold on
ylim([0 1.1]);
grid minor
plot(x,U(1,:),'r','color',[0, 0.4470, 0.7410],'LineWidth',1.5)
plot(x,U(2,:),'b--')
plot(x,U(3,:),'g','color',[0.9290, 0.6940, 0.1250],'LineWidth',1.5)
plot(x,U(4,:),'k--','LineWidth',0.75)
% legend('Lambda = 0.5','Lambda = 1.0','Lambda = 1.5','Lambda = 5.0');
 legend('0.5','1.0','1.5','5.0');
set(gca,'FontSize',13)
set(legend,'Interpreter','latex','location','northeast','box','off')
title(legend,'Courant Number, $\lambda$')
title('$u(x,t)$ vs $x$','interpreter','latex');
xlabel('$x$','interpreter','latex');
ylabel('$u(x,t)$','interpreter','latex');
```

MatLab Code Q11

```
clear
clc
%this script is for solving Question 11 of the CFD coursework
Time =1;
%Constants--------
n=501; %Vary the number of grid points here
DeltaX=1/(n-1);
DeltaT=0.1;
tarray=0:DeltaT:Time; %Discretise an array of time value given DeltaT
x=[0:DeltaX:1]; %Discretise x over n grid points for 0<x<1
u=zeros([length(tarray),length(x)]); %Create empty matrix of u values
for r=1:length(x) %Populate u matrix with initial conditions t=0
    if x(r)<=0.5
        u(1,r)=0;
    else
        u(1,r)=1;
    end
end
u(1,length(x))=u(1,1); %Enforce periodic condition
p=2; %Start while loop at the 2nd grid point in time value to enter the while loop
while p<=length(tarray)
    a=10*sin(tarray(p));
    lambda=a*DeltaT/DeltaX; %Courant-Number
        for i=1:length(x)
            k=floor(i-lambda+1); %calculate the value of k at each spacial grid point i
            if k>1
                u(p,i)=-(i-k)*u(p-1,k-1) + (i-k+1)*u(p-1,k)-lambda*(u(p-1,k)-u(p-1,k-1));
            else
                k=k+(n-1);
                u(p,i)=-(i-(k-(n-1)))*u(p-1,k-1) + (i-(k-(n-1))+1)*u(p-1,k)-lambda*(u(p-
1,k)-u(p-1,k-1));
            end
        end
    p=p+1;
end
hold on
grid minor
plot(x,u(length(tarray),:),'color',[0.25, 0.25, 0.25],'LineWidth',1.0)
%Graph Content
legend('11','21','51','101','201','251','501');
set(gca,'FontSize',13)
set(legend,'Interpreter','latex','location','northwest','box','off')
title(legend,'Number of Grid Points')
title('$u(x,t)$ vs $x$','interpreter','latex');
xlabel('$x$','interpreter','latex');
ylabel('$u(x,t)$','interpreter','latex');
```