Question 1.

(a) We will try and verify for a general component as per the hint.

So, we are trying to show that

$$(A^h \vec{x}^{(p)})_J = \lambda_p \vec{x}_J^{(p)}$$

Firstly, we take the $J=1$ case so

$A^h \vec{x}_1^{(p)} = \frac{1}{h^2}[-2\sin(p\pi h) + \sin(p\pi h 2)] =$

$= \frac{1}{h^2}[-2\sin(p\pi h) + 2\sin(p\pi h)\cos(p\pi h) = \frac{2\sin(p\pi h)}{h^2}[\cos(p\pi h) - 1] =$

$= \lambda_p x_1^{(p)}$

Because of the $\sin(2a) = 2\sin(a)\cos(a)$ identity

We take the $0 < J < m$ case which leads us to

$(A^h \vec{x}_J^{(p)}) = \frac{1}{h^2}[\sin(p\pi h(J-1)) - 2\sin(p\pi hJ) + \sin(p\pi h(J+1))]$

$= \frac{1}{h^2}[2\sin(p\pi hJ)\cos(-p\pi h) - 2\sin(p\pi hJ)] =$

$= \frac{2\sin(p\pi hJ)}{h^2}[\cos(p\pi h) - 1] = \lambda_p x_J^{(p)}$

Because of the identity $\sin(a-b) + \sin(a+b) = 2\sin(a)\cos(b)$

Finally, we take the $J=m$ case so

$A^h \vec{x}_m^{(p)} = \frac{1}{h^2}[\sin(p\pi h(m-1)) - 2\sin(p\pi hm)] =$

$= \frac{1}{h^2}[\sin(p\pi hm)\cos(-p\pi h) + \cos(p\pi hm)\sin(-p\pi h) - 2\sin(p\pi hm)] =$

$= \frac{1}{h^2}[\sin(p\pi hm)\cos(p\pi h) - \cos(p\pi hm)\sin(p\pi h) - 2\sin(p\pi hm)] =$

$= \frac{\sin(p\pi hm)}{h^2}[\cos(p\pi h) - \frac{\sin(p\pi h)}{\tan(p\pi hm)} - 2] =$

$= \frac{\sin(p\pi hm)}{h^2}[2\cos(p\pi h) - 2] = \frac{2\sin(p\pi hm)}{h^2}[\cos(p\pi h) - 1] = \lambda_p x_m^{(p)}$

Because we have that

$$\tan(p\pi h(m+1) - p\pi h) = \tan(p\pi - p\pi h) = \frac{\tan(p\pi) - \tan(p\pi h)}{\tan(p\pi)\tan(p\pi h) + 1}$$

$$= -\tan(p\pi h)$$

From $\tan(a-b) = \frac{\sin(a)\cos(b) - \cos(a)\sin(b)}{\sin(a)\sin(b) + \cos(a)\cos(b)} =$

$$= \frac{\tan(a) - \tan(b)}{\tan(a)\tan(b) + 1}$$

Therefore, we have shown that $x_j^{(p)} = \sin(p\pi j h)$ are the eigenvectors and $\lambda_p = \frac{2}{h^2}(\cos(p\pi h) - 1)$ are the eigenvalues of A^h.

(b) We need $\frac{d^2 u(x)}{dx^2} = \lambda u(x)$ according to our hint

Therefore, differentiating $u(x) = \sin(p\pi x)$ we get

$u'(x) = p\pi\cos(p\pi x)$ and $u''(x) = -(p\pi)^2 \sin(p\pi x) = -(p\pi)^2 u(x)$

Now we check the boundary conditions.

$u(0) = \sin(0) = 0$ and $u(1) = \sin(p\pi) = 0$ for $p = (1, 2, \ldots)$

In our situation our eigenvalue is $\lambda = -(p\pi)^2$ with $u(x)$ being being our eigen function

Discrete eigen-vectors/values:

$x_j^{(p)} = \sin(p\pi j h)$, $\lambda_p = \frac{2}{h^2}(\cos(p\pi h) - 1) = -\frac{4}{h^2}\sin^2\left(\frac{p\pi h}{2}\right)$

As $p, h \to 0$ we know that $\sin(x) \sim x$ so

$\lambda_p \sim -\frac{4}{h^2}\left(\frac{p\pi h}{2}\right)^2 = -(p\pi)^2$ and

$x_j^{(p)} \sim p\pi j h$

Continuous eigen-vectors/values.

$u^{(p)}(x) = \sin(p\pi x)$, $\lambda = -(p\pi)^2$

As $p \to 0$ we know that $\sin(x) \sim x$ so

$u^{(p)}(x) \sim p\pi x$ and $\lambda = -(p\pi)^2$

We see that our discrete and continuous eigenfunctions are similar but the eigenvalues are not.

As we take $p, h \to 0$ we see that both eigenfunctions and eigenvalues are similar.

Question 2.

We have that $\frac{v_J^{n+1}-v_J^{n-1}}{2\Delta t} + a\frac{v_{J+1}^n - v_{J-1}^n}{2\Delta x} = 0$

We will plug in the exact solution so

$$\frac{u_J^{n+1}-u_J^{n-1}}{2\Delta t} + a\frac{u_{J+1}^n - u_{J-1}^n}{2\Delta x} = t_J^n$$

We will use Taylor Series

$$u_{J+1}^n = u_J^n + \Delta x\frac{\partial(u_J^n)}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2(u_J^n)}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3(u_J^n)}{\partial x^3} + \dots$$

$$u_{J-1}^n = u_J^n - \Delta x\frac{\partial(u_J^n)}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2(u_J^n)}{\partial x^2} - \frac{\Delta x^3}{6}\frac{\partial^3(u_J^n)}{\partial x^3} + \dots$$

So, $a\frac{u_{J+1}^n - u_{J-1}^n}{2\Delta x} = \frac{a}{2\Delta x}\left(2\Delta x\frac{\partial(u_J^n)}{\partial x} + \frac{\Delta x^3}{3}\frac{\partial^3(u_J^n)}{\partial x^3} + \dots\right) =$

$$= a\frac{\partial u_J^n}{\partial x} + \frac{a\Delta x^2}{6}\frac{\partial^3(u_J^n)}{\partial x^3} + \dots = a\frac{\partial u_J^n}{\partial x} + O(\Delta x^2)$$

Again we will use Taylor Series

$$u_J^{n+1} = u_J^n + \Delta t\frac{\partial(u_J^n)}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2(u_J^n)}{\partial t^2} + \frac{\Delta t^3}{6}\frac{\partial^3(u_J^n)}{\partial t^3} + \dots$$

$$u_J^{n-1} = u_J^n - \Delta t\frac{\partial(u_J^n)}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2(u_J^n)}{\partial t^2} - \frac{\Delta t^3}{6}\frac{\partial^3(u_J^n)}{\partial t^3} + \dots$$

So, $\frac{u_J^{n+1}-u_J^{n-1}}{2\Delta t} = \frac{1}{2\Delta t}\left(2\Delta t\frac{\partial(u_J^n)}{\partial t} + \frac{\Delta t^3}{3}\frac{\partial^3(u_J^n)}{\partial t^3} + \dots\right)$

$$= \frac{\partial(u_J^n)}{\partial t} + \frac{\Delta t^2}{6}\frac{\partial^3(u_J^n)}{\partial t^3} + \dots = \frac{\partial(u_J^n)}{\partial t} + O(\Delta t^2)$$

Therefore $t_J^n = O(\Delta x^2) + O(\Delta t^2)$

Question 3

We have that $\dfrac{V_J^{n+1} - V_J^n}{\Delta t} + \dfrac{a}{2}\left[\dfrac{V_{J+1}^n - V_{J-1}^n}{2\Delta x} + \dfrac{V_{J+1}^{n+1} - V_{J-1}^{n+1}}{2\Delta x}\right] = 0$

We plug in our actual error e_J^n in

$$\underbrace{\frac{e_J^{n+1} - e_J^n}{\Delta t}}_{①} + \frac{a}{2}\left[\underbrace{\frac{e_{J+1}^n - e_{J-1}^n}{2\Delta x}}_{②} + \underbrace{\frac{e_{J+1}^{n+1} - e_{J-1}^{n+1}}{2\Delta x}}_{③}\right] = t_J^n$$

We let $t_J^n = 0$ for simplicity and $e_J^n = \hat{e}_K^n \exp(ik_J\Delta x)$, k-th mode

① $\dfrac{e_J^{n+1} - e_J^n}{\Delta t} = \exp(ik_J\Delta x)\dfrac{(\hat{e}_K^{n+1} - \hat{e}_K^n)}{\Delta t}$

② $\dfrac{a}{2}\left(\dfrac{e_{J+1}^n - e_{J-1}^n}{2\Delta x}\right) = \dfrac{a\exp(ik_J\Delta x)\hat{e}_K^n}{4\Delta x}\left[\exp(ik\Delta x) - \exp(-ik\Delta x)\right]$

$$= \frac{a\exp(ik_J\Delta x)\hat{e}_K^n}{4\Delta x}\left[\cos(k\Delta x) + i\sin(k\Delta x) - (\cos(k\Delta x) - i\sin(k\Delta x))\right]$$

$$= \frac{a\exp(ik_J\Delta x)\hat{e}_K^n\, 2i\sin(k\Delta x)}{4\Delta x} = \frac{a\exp(ik_J\Delta x)\hat{e}_K^n}{2\Delta x} i\sin(k\Delta x)$$

③ Similarly, $\dfrac{a}{2}\left(\dfrac{e_{J+1}^{n+1} - e_{J-1}^{n+1}}{2\Delta x}\right) = \dfrac{a\exp(ik_J\Delta x)\hat{e}_K^{n+1}}{2\Delta x} i\sin(k\Delta x)$

Plugging everything into our equation, multiplying by Δt and setting $R = \dfrac{a\Delta t}{\Delta x}$

We get

$$\cancel{\exp(ik_J\Delta x)}\left[\hat{e}_K^{n+1} - \hat{e}_K^n + \frac{R}{2} i\sin(k\Delta x)\left(\hat{e}_K^n + \hat{e}_K^{n+1}\right)\right] = 0$$

Multiplying by 2 throughout

$$\hat{e}_K^{n+1}(2 + iR\sin(k\Delta x)) - \hat{e}_K^n(2 - iR\sin(k\Delta x)) = 0$$

So, $S(K) = \dfrac{\hat{e}_K^{n+1}}{\hat{e}_K^n} = \dfrac{2 - iR\sin(k\Delta x)}{2 + iR\sin(k\Delta x)}$

With $|S(K)| = \sqrt{\dfrac{4 + R^2\sin^2(k\Delta x)}{4 + R^2\sin^2(k\Delta x)}} = 1$

Therefore, it is numerically Stable, Unconditionally.

Question 4

We claim that $u(x,y,t) = f(x-at, y-bt)$ is our general solution for

$$U_t + aU_x + bU_y = 0$$

Set $s_1 = x - at$ and $s_2 = y - bt$ so

$$U_t = \frac{\partial f}{\partial s_1}\frac{\partial s_1}{\partial t} + \frac{\partial f}{\partial s_2}\frac{\partial s_2}{\partial t} = -a\frac{\partial f}{\partial s_1} - b\frac{\partial f}{\partial s_2}$$

$$aU_x = a\frac{\partial f}{\partial s_1}\frac{\partial s_1}{\partial x} + a\frac{\partial f}{\partial s_2}\frac{\partial s_2}{\partial x} = a\frac{\partial f}{\partial s_1}$$

$$bU_y = b\frac{\partial f}{\partial s_1}\frac{\partial s_1}{\partial y} + b\frac{\partial f}{\partial s_2}\frac{\partial s_2}{\partial y} = b\frac{\partial f}{\partial s_2}$$

Therefore,

$$U_t + aU_x + bU_y = -a\frac{\partial f}{\partial s_1} - b\frac{\partial f}{\partial s_2} + a\frac{\partial f}{\partial s_1} + b\frac{\partial f}{\partial s_2} = 0$$

So, our claim is correct and $u(x,y,t) = f(x-at, y-bt)$ is a general solution to the 2D linear advection equation.

Question 5

We have that $\frac{Y_J^{n+1}-Y_J^n}{\Delta t} - \frac{Y_{J-1}^n - Y_J^{n+1} - Y_J^n + Y_{J+1}^n}{\Delta x^2} = 0$

We plug in our actual error e_J^n so

$$\underset{①}{\frac{e_J^{n+1}-e_J^n}{\Delta t}} - \underset{②}{\frac{e_{J-1}^n - e_J^{n+1} - e_J^n + e_{J+1}^n}{\Delta x^2}} = t_J^n$$

We let $t_J^n = 0$ for simplicity and $e_J^n = \hat{e}_k^n \exp(ik J\Delta x)$ the k-th mode

① $\frac{e_J^{n+1}-e_J^n}{\Delta t} = \exp(ikJ\Delta x) \frac{(\hat{e}_K^{n+1} - \hat{e}_K^n)}{\Delta t}$

② $-\frac{e_{J-1}^n - e_J^{n+1} - e_J^n + e_{J+1}^n}{\Delta x^2} = -\frac{\exp(ikJ\Delta x)}{\Delta x^2}\left[\hat{e}_K^n \exp(-ik\Delta x) - \hat{e}_K^{n+1} - \hat{e}_K^n + \hat{e}_K^n \exp(ik\Delta x)\right]$

$$= -\frac{\exp(ikJ\Delta x)}{\Delta x^2}\left[\hat{e}_K^n\left(2\cos(k\Delta x) - 1\right) - \hat{e}_K^{n+1}\right]$$

Plugging everything into our equation and multiplying by Δt we have

$$\cancel{\exp(ikJ\Delta x)}\left[\hat{e}_K^{n+1} - \hat{e}_K^n - \frac{\Delta t}{\Delta x^2}\left[\hat{e}_K^n\left(2\cos(k\Delta x)-1\right) - \hat{e}_K^{n+1}\right]\right] = 0$$

So, $\hat{e}_K^{n+1}\left(1 + \frac{\Delta t}{\Delta x^2}\right) - \hat{e}_K^n\left(1 + \frac{\Delta t}{\Delta x^2}\left(2\cos(k\Delta x)-1\right)\right) = 0$

$$S(k) = \frac{\hat{e}_K^{n+1}}{\hat{e}_K^n} = \frac{1 + \frac{\Delta t}{\Delta x^2}\left(2\cos(k\Delta x)-1\right)}{1 + \frac{\Delta t}{\Delta x^2}}$$

$$= \frac{1 + R\left(2\cos(k\Delta x)-1\right)}{1+R}$$

In order to be stable we need $|S(k)| \leq 1$ so

$$-1 \leq \left|\frac{1+R(2\cos(k\Delta x)-1)}{1+R}\right| \leq 1 \quad \text{since } 1+R>0$$

$$-1 \leq \frac{|1+R(2\cos(k\Delta x)-1)|}{1+R} \leq 1 \quad \text{with } 2\cos(k\Delta x) - 1 \in [1,-3]$$

Our first case

$$-1 \leq \frac{1+R(2\cos(k\Delta x)-1)}{1+R}$$

$$-1-R \leq 1-R+R(2\cos(k\Delta x))$$

$$-2 \leq R(2\cos(k\Delta x)) \Rightarrow R^{-1} \geq -\cos(k\Delta x)$$ with $\cos \in [-1,1]$

So, $R^{-1} \geq 1$

Second Case

$$\frac{1+R(2\cos(k\Delta x)-1)}{1+R} \leq 1$$

$$1-R+R2\cos(k\Delta x) \leq 1+R$$

$$R2\cos(k\Delta x) \leq 2R$$

$\cos(k\Delta x) \leq 1$ which is true always.

Therefore, our only condition is $R^{-1} \geq 1$ which becomes

$$\frac{\Delta x^2}{\Delta t} \geq 1 \Rightarrow \Delta t \leq \Delta x^2 \Rightarrow 0 \leq \Delta t \leq \Delta x^2$$

since our timestep has to be ≥ 0.

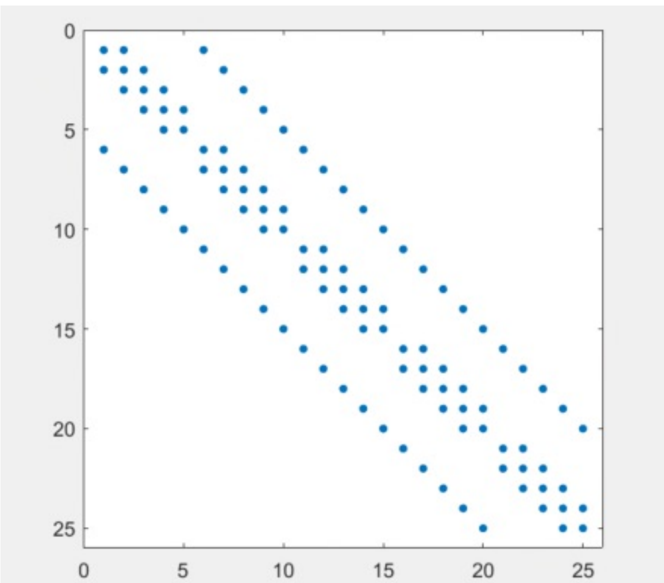
So, our method is conditionally stable for

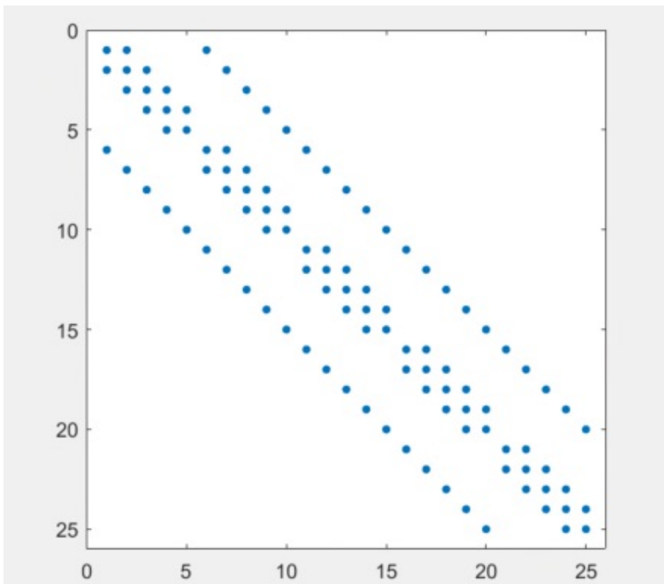
$$0 \leq \Delta t \leq \Delta x^2$$

Question 6(a)

```
function A=build_laplace_2D(k)

%Arrays of 1's that we will use.
e1=ones(k^2,1);
e2=ones(k,1);

%Creating our block matrix T.
D1= spdiags([e2,-4*e2,e2], -1:1, k, k);
%Making a repmat k^2 by k^2 matrix and keeping only its three central diagonals.
J=repmat(D1,k,k);
full(J)
x1=diag(diag(J,-1),-1);
x2=diag(diag(J));
x3=diag(diag(J,1),1);
full(x1)
full(x2)
full(x3)
X=x1+x2+x3;

%Our I matrices on our off diagonals take the form.
D2=spdiags([e1 e1], [-k k], k^2, k^2);

%Adding the two matrices and multiplying them by 1/(h^2)
A=D2+X;
h = 1/(k+1);
A = A/h^2;
full(A)
spy(A)
end
```

Our code at k=5 which agrees with kron



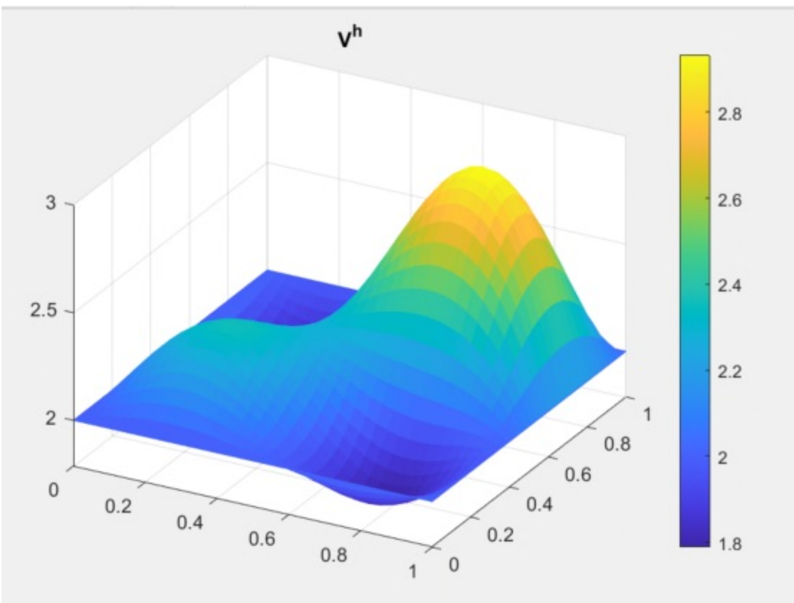


Question 6(b)

Approximate Plot

```
%Question 6(b)
function v=Question_6b(k)
%Firstly, we will recal our Laplace Matrix A from Q6(a)
A=build_laplace_2D(k);
%We set our h and the ranges of x and y.
h=1/(k+1);
x=(0+h:h:1-h);
y=(0+h:h:1-h);
[X,Y]= meshgrid(x,y);
%Now we set our RHS of the linear system
RHS=20*(Y.^4-Y.^2).*(6*X.^2-1)+20*(X.^4-X.^2).*(6*Y.^2-1)-8*pi*sin(2*pi*X).*sin↙
(2*pi*Y);
%We use reshape the lexicographically ordered vector into a matrix
f=reshape(RHS.',[],1);
%We solve our Linear System with
v=A\f;
v=reshape(v,[k,k]);

v=2+v;

v = [v 2*ones(k,1)];
v = [2*ones(k,1) v];
v = [2*ones(1,k+2)
    v];
v = [v
    2*ones(1,k+2)];

%We set the x and y ranges again
x=(0:h:1);
y=(0:h:1);
[X,Y]=meshgrid(x,y);

%We finally plot our mesh
s=mesh(X,Y,v)
title('V^h');
s.FaceColor='flat';
colorbar;

end
```

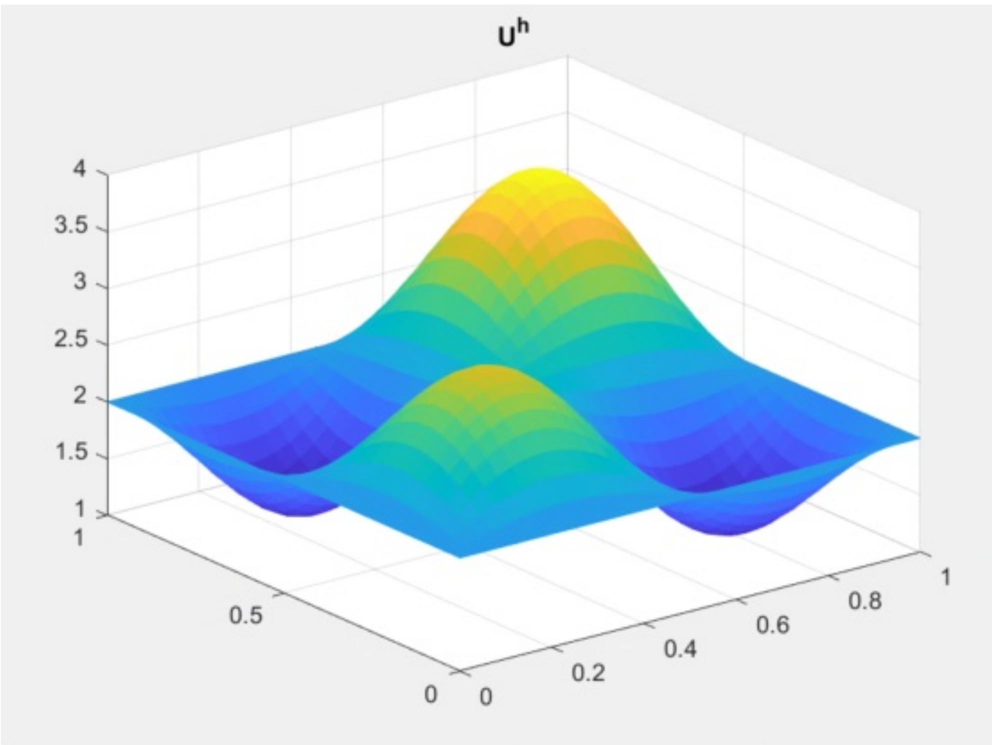
This gives us for k=32



Exact Plot

```
function E=Exact_Solution(k)
%We will follow the assignment hints to produce the exact mesh.
%Introduction of h, and x and y ranges.

h=1/(k+1);
y=(0:h:1);
x=(0:h:1);
[X,Y]=meshgrid(x,y);

%Set exact solution which is given.
u=sin(2*pi*X).*sin(2*pi*Y)+10*(X.^4-X.^2).*(Y.^4-Y.^2)+2;

%Mesh of the exact solution.
r=mesh(X,Y,u);
title('U^h');
r.FaceColor='Flat';
end
```

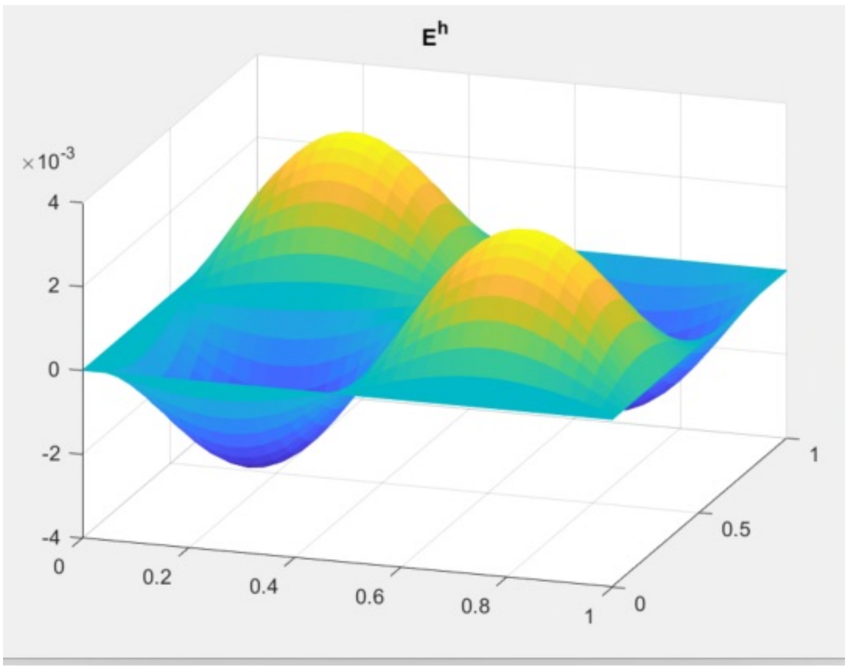
Which gives us



Error Plot

```
%Error Function
function D=error_find(k)
%Introduction of h and the range of x and y.
h=1/(k+1);
x=(0+h:h:1-h);
y=(0+h:h:1-h);
[X,Y]=meshgrid(x,y);
%The RHS of our linear system
RHS=20*(Y.^4-Y.^2).*(6*X.^2-1)+20*(X.^4-X.^2).*(6*Y.^2-1)-8*pi^2*sin(2*pi*X).*sin
(2*pi*Y);
%Reshaping the lexicographically ordered vector into a matrix
f=reshape(RHS.',[],1);
%Recall the Laplace Matrix A
A=build_laplace_2D(k);
%Solve the linear system
v=A\f;
%Introduce our boundary conditions adding two rows and collumns.
v = 2+reshape(v,[k, k]);
v = [v 2*ones(k,1)];
v = [2*ones(k,1) v];
v = [2*ones(1,k+2)
v];
v = [v
2*ones(1,k+2)];
%Introduce the exact solution.
x = (0:h:1);
y = (0:h:1);
[X,Y] = meshgrid(x,y);
u=sin(2*pi*X).*sin(2*pi*Y)+10*(X.^4-X.^2).*(Y.^4-Y.^2)+2;
%Taking the difference between the Exact and Approximate
D=minus(u,v);
x=(0:h:1);
y=(0:h:1);
[X,Y]=meshgrid(x,y);
%Mesh of the Error.
L=mesh(X,Y,D);
%L.FaceColor='Flat';
end
```

Which gives us



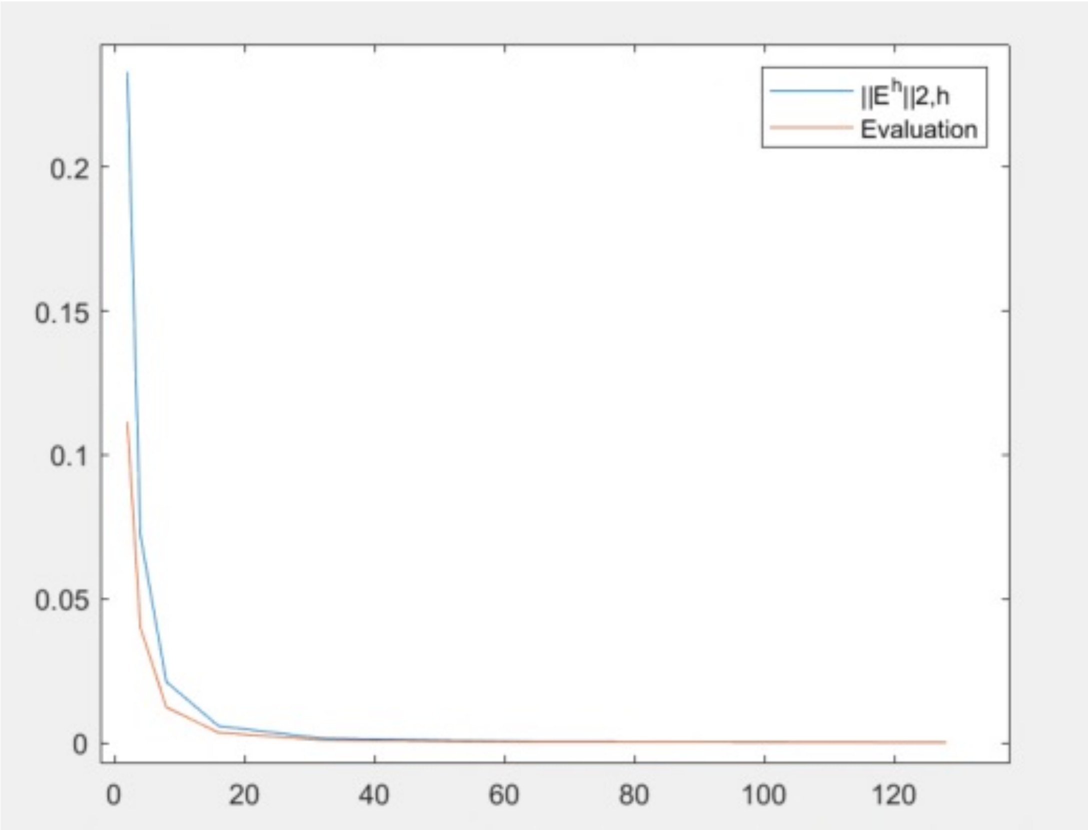
Question 6(c)

```
function c= Question_6c(k)
D=error_find(k);
D2=D.^2;
ALLD2=sum(D2,'all');
SQRTALLD2=ALLD2^(1/2);
h=1/(k+1);
Final=SQRTALLD2*h;

%After running our function these are our values"
Val=[2;4;8;16;32;64;128];
E2H=[0.2329;0.0721;0.0211;0.0058;0.0015;3.9449e-04;1.0012e-04];
Ratio=[0;0.3096;0.2926;0.2749;0.2586;0.2630;0.2538];
h2val=[0.1111;0.04;0.0123;0.0035;9.1809e-04;2.3716e-04;6.0840e-05];

%Our table
T=table(Val,E2H,Ratio)

%Our plot of our ||E^2||h against the
figure
plot(Val,E2H);
hold on
plot(Val,h2val);
legend('||E^2||h','Evaluation')
hold off

end
```

Our table is then

Val	E2H	Ratio
2	0.2329	0
4	0.0721	0.3096
8	0.0211	0.2926
16	0.0058	0.2749
32	0.0015	0.2586
64	0.00039449	0.263
128	0.00010012	0.2538

Also, we plot the ||E^2||_h against our evaluation to see its behaviour.



We see that both converge with a similar rate based on both the plot and the tables since the ratio converges to 0.25

Question 7 (a)

```
function P=Parabolic(k)

%Given values
eta=2;
alpha=5;
to=0.05;

%Spacial Info
dx=30/(k+1);
x=(-10:dx:20);

%Given function u at time to
u=(1/(to^(1/2)))*exp((-((x-alpha*to).^2))/(4*eta*to));
u=reshape(u,[k+2,1]);

%2 Different Timesteps.
Dtd= (dx^2)/(2*eta);
Dta=(dx)/(alpha);

%(1) EXPLICIT METHOD

%Given timestep
Dt1=(0.4750)*min(Dta,Dtd);

%New timestep
Dtnew=((1-to)/(ceil((1-to)/(Dt1))));

%Now we create our matrix with spdiags like in Q6.

e=ones(k+2,1);
Matr1= spdiags([e*(((eta*Dtnew)/(dx^2))+((alpha*Dtnew)/(dx))),...;
    e*(1-((2*eta*Dtnew)/(dx^2))-((alpha*Dtnew)/(dx))),...;
    e*((eta*Dtnew)/(dx^2))], -1:1, k+2, k+2);

%Preparing for our loop
T=1;
tI=0.05;
UL=u;

%Loop
while tI<1
    T=T+1;
    tI=tI+Dtnew;
    Ur=UL(:,end);
    UL=Matr1*Ur;
end

%Exact Plot
figure
EXACT=exp(-(((x-alpha).^2)/(4*eta)));
```

```
plot(x,EXACT);
hold on

%Explicit Plot at t=3
plot(x,UL)
hold on

%(2) SEMI-IMPLICIT METHOD

%Given and New Timesteps
Dt2=0.95*Dta;
Dt2new=(1-to)/(ceil((1-to)/(Dt2)));

%Now we have two matrices for which we use spdiags again.
Matr21=spdiags([-e*((eta*Dt2)/(dx^2)),e*(1+((2*eta*Dt2)/(dx^2))),...;
    e*((-eta*Dt2)/(dx^2))],-1:1, k+2, k+2);
Matr22=spdiags([e*((alpha*Dt2)/(dx)),e*(1-((alpha*Dt2)/(dx)))], -1:0, k+2, k+2);

%Preparing for our loop
UR=u;
TR=0.05;

%Our loop
while TR<1
   TR= TR+Dt2;
   UR=Matr22*UR;
   UR=Matr21\UR;
end

%Implicit Plot
plot(x,UR)
hold off
end
```

Our plot is then for k=321

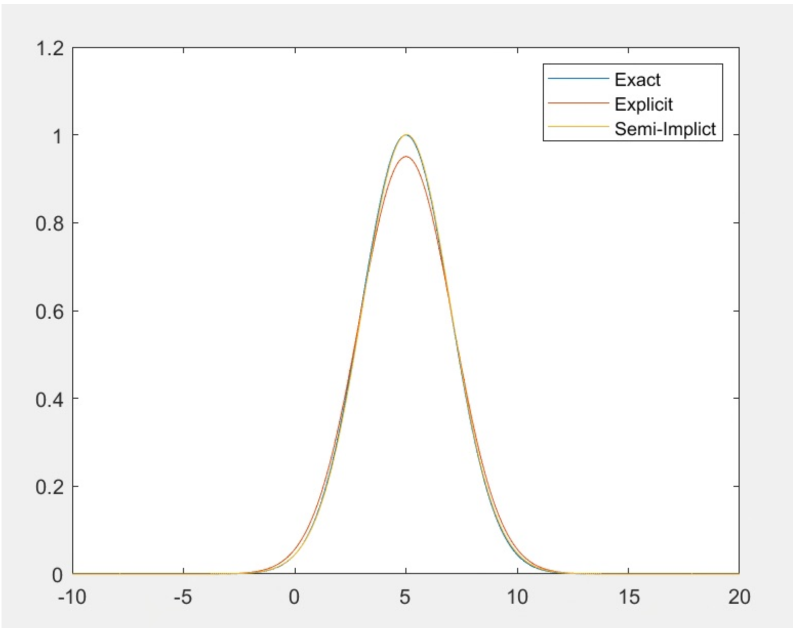


Figure 7a: Exact,Semi Implicit and Explicit at t=1

As we can see our plots are almost the same!

Question 7(b)

Our table looks like this:

k	Explicit Steps	Semi-Implicit Steps	Explicit Runtime	Semi-Implicit Runtime	Δt_{diff}	Δt_{adv}	$\Delta t_{Explicit}$	$\Delta t_{Implicit}$
11	4	2	0.0034	0.0021	1.5625	0.5000	0.2375	0.4750
21	8	4	0.0022	0.0015	0.4649	0.2727	0.1187	0.2375
41	17	8	0.0011	0.0011	0.1276	0.1429	0.0594	0.1357
81	60	14	0.0018	0.0064	0.0335	0.0732	0.0158	0.0679
161	235	27	0.0057	0.0013	0.0086	0.0370	0.0041	0.0339
321	923	54	0.0277	0.0023	0.0022	0.0186	0.0010	0.0176
641	3664	108	0.1221	0.0083	5.4590e-04	0.0093	2.5928e-04	0.00880

To obtain the Semi-Implicit and Explicit Information 2 codes were run. For the Explicit Method sufficiently more steps are needed to be taken as k increases whilst the exact opposite is noticed for the Semi-Implicit Method. Thus, our difference in runtimes can be explained since there is a noticeable difference between the explicit and semi-implicit runtimes. The explicit ones tend to increase more rapidly than the semi-implicit as k increases. Therefore, the semi-implicit method is more efficient because of its fewer steps and runtime.