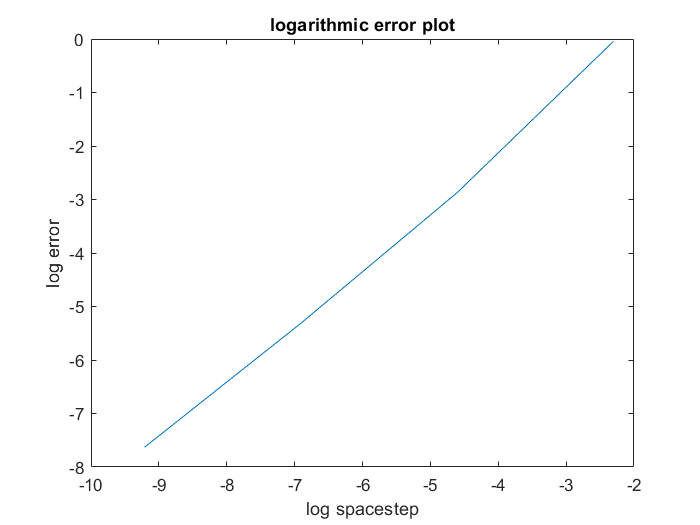
Acknowledgement: Thanks to the 16.90 TAs, I have identified the initial bug in my code. That was caused by not paying enough attention to MATLAB syntax. The initial statement was exact\_u=zeros(Nx+1) and was generating a square matrix! I modified that statement into exact\_u=zeros(1,Nx+1); and everything worked fine.

I have used the following code to generate the log-log plot:

  
As expected the slope is 1 so the method is first order accurate in space.

My code:

conv\_velocity=1;

diffusivity=0.1;

c=1/(1-exp(conv\_velocity/diffusivity));

%define Dirichlet boundary conditions

U\_left\_bc=1;U\_right\_bc=0;

dxs=[1/10, 1/100, 1/1000, 1/10000];

errs=zeros(1,4);

%errs=[1.5759,10.1571,31.6732,100.0159] values I found after running code

for j=1:4

dx=dxs(j);

Nx=round(1/dx+1);

A=zeros(Nx-1,Nx-1);

for i=1:Nx-1

if i==1

A(i,i)=2\*diffusivity/(dx^2);

A(i,i+1)=conv\_velocity/(2\*dx)-diffusivity/(dx^2);

elseif i==Nx-1

A(i,i-1)=-(conv\_velocity/(2\*dx))-(diffusivity/(dx^2));

A(i,i)=2\*diffusivity/(dx^2);

else

A(i,i-1)=-(conv\_velocity/(2\*dx))-(diffusivity/(dx^2));

A(i,i)=2\*diffusivity/(dx^2);

A(i,i+1)=conv\_velocity/(2\*dx)-diffusivity/(dx^2);

end

end

b=zeros(Nx-1,1);

b(1,1)=U\_left\_bc\*(conv\_velocity/(2\*dx)+diffusivity/(dx^2));

b(Nx-1,1)=-U\_right\_bc\*(conv\_velocity/(2\*dx)-diffusivity/(dx^2));

U\_solution=A\b;

exact\_u=zeros(1,Nx+1);

for i=1:length(exact\_u)

exact\_u(i)=c\*exp(i\*dx\*conv\_velocity/diffusivity)+(1-c);

end

errs(j)=norm(exact\_u-[1,transpose(U\_solution),0])/norm(exact\_u);

end

disp(size(exact\_u))

disp(size(U\_solution))

%figure(1);

%plot(linspace(0,1,Nx+1),[1,transpose(U\_solution),0],'-',linspace(0,1,Nx+1),exact\_u,'-')

%legend('State convection','exact');

%title('Convection');

%xlabel('location');

%ylabel('State');

disp(errs)

figure(2);

plot(log(dxs),log(errs),'-')

title('logarithmic error plot');

xlabel('log spacestep');

ylabel('log error');