Assignment II

M693B
DR. BLOMGREN
MARCH 1, 2018

MATTEO POLIMENO

1 Problem 3.4.1 from Strikveda

1.1 Function

Here is the function that we used to plot the solution

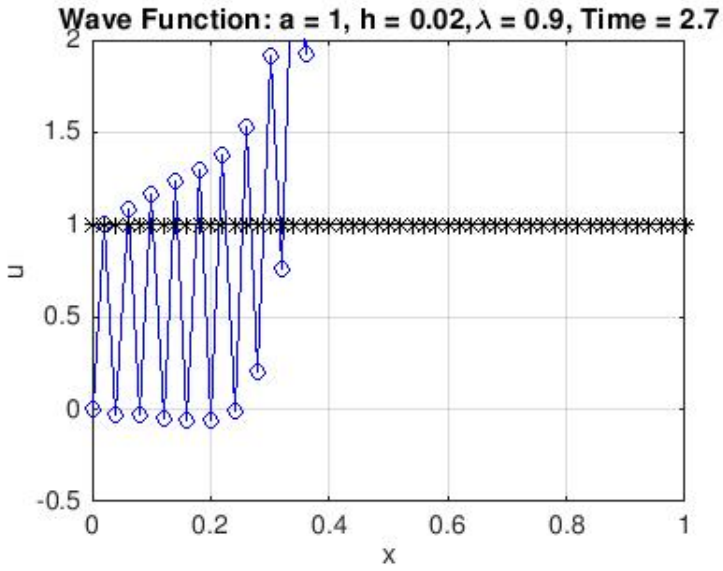
```
function J = f_sol(x)
if x <= 0
    J = 1;
else
    J = cos(2*pi*x);
end
```

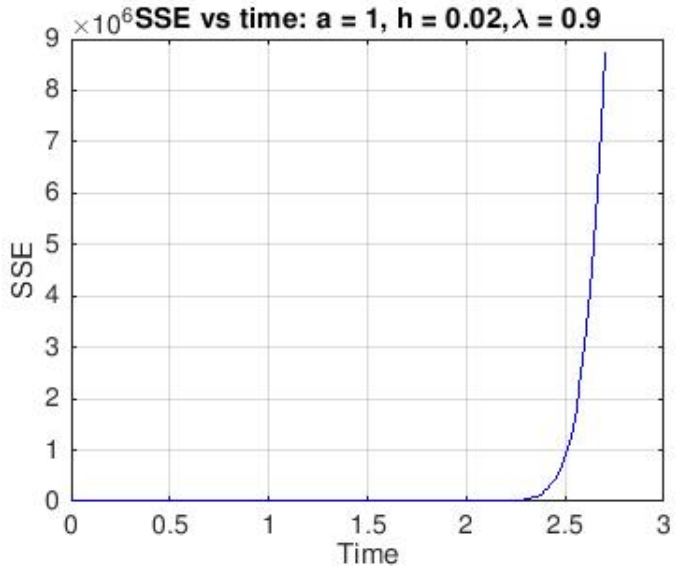
And now the codes and plots for part a, b, c and d

1.2 Matlab Code for Part a

```
clear all


lambda = .9;
h = 1/50;
xd = 0:h:1;
k = lambda*h;
p = length(xd);
td = 0:k:2.7;
q = length(td);


for i = 1:p
    u(1,i) = f_sol(xd(i));
end


for i = 1:q-1 %lax-friendrichs scheme
    u(i,1) = 1;
    for j = 1:p-2
        u(i+1,j+1) = -lambda*((u(i,j+2) - u(i,j))/2) + ((u(i,j+2) + u(i
            ,j))/2);
    end
end


for i = 1:q-1 %run leap frog

    for j = 2:p-2
        u(i+2,j+1) = -lambda*(u(i+1,j+2) - u(i+1,j)) + u(i,j+1);
```

```
30          u(i+1,p) = 2*u(i+1,p-1) - u(i+1,p-2);
31      end
32  end
33
34  for i = 1:q
35      v(i,1) = 0;
36  end
37
38
39  for i = 1:q
40      for j = 1:p
41          v(i,j) = F_341((xd(j) -  td(i)));
42      end
43  end
44
45  for i = 1:q
46      plot(xd,u(i,:),'b-o',xd,v(i,:),'k-*');
47      ylim([-0.5,2])
48      xlim([0,1])
49      title(['Wave Function: a = 1, h = ' num2str(h) ',\lambda = '
            num2str(lambda) ', Time = ' num2str(td(i))])
50      xlabel('x')
51      ylabel('u')
52      grid on
53      M(i) = getframe;
54  end
55
56
57  for i = 1:q
58      e(i,:) = abs((v(i,:)-u(i,:)));
59      err(i) = sum(e(i,:)).^2;
60  end
61
62  figure()
63  plot(td,err,'b-')
64  title(['SSE vs time: a = 1, h = ' num2str(h) ',\lambda = ' num2str(
        lambda)])
65  xlabel('Time')
66  ylabel('SSE')
67  grid on
```

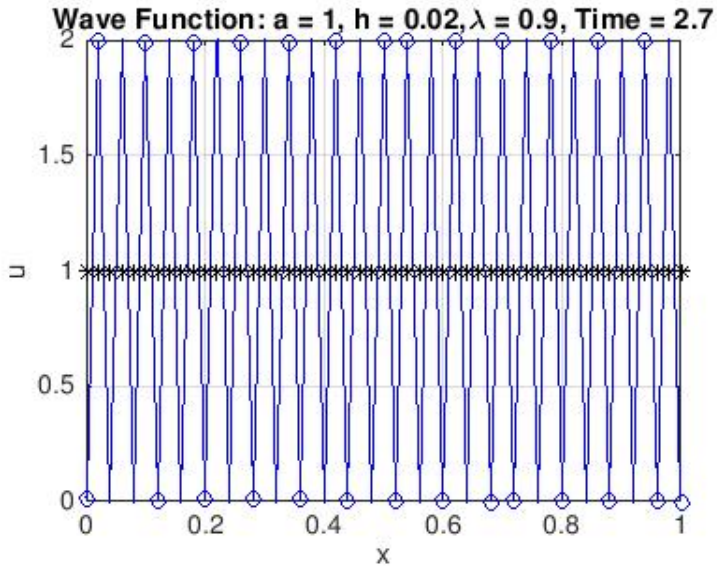

Wave Function: a = 1, h = 0.02, λ = 0.9, Time = 2.7
u
x
2
1.5
1
0.5
0
-0.5
0
0.2
0.4
0.6
0.8
1

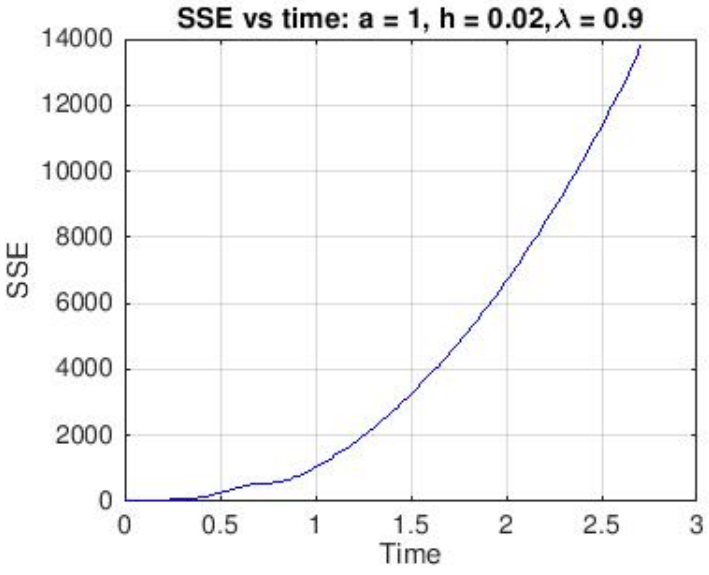

×10^6 SSE vs time: a = 1, h = 0.02, λ = 0.9
SSE
Time
9
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3

1.3 Matlab Code for Part b

```
clear all


lambda = .9;
h = 1/50;
xd = -4:h:1;
k = lambda*h;
p = length(xd);
td = 0:k:2.7;
q = length(td);


for i = 1:p
    u(1,i) = f_sol(xd(i));
end


for i = 1:q-1 %lax-friendrichs scheme
    u(i,1) = 1;
    for j = 1:p-2
        u(i+1,j+1) = -lambda*((u(i,j+2) - u(i,j))/2) + ((u(i,j+2) + u(i
            ,j))/2);
    end
end


for i = 1:q-1 %run leap frog

    for j = 2:p-2
        u(i+2,j+1) = -lambda*(u(i+1,j+2) - u(i+1,j)) + u(i,j+1);
        u(i+1,p) = 0;
    end
end


for i = 1:q
    v(i,1) = 0;
end


for i = 1:q
    for j = 1:p
        v(i,j) = F_341((xd(j) -  td(i)));
    end
```

```
44 end
45
46 for i = 1:q
47     plot(xd,u(i,:),'b-o',xd,v(i,:),'k-*');
48     ylim([0,2])
49     xlim([0,1])
50     title(['Wave Function: a = 1, h = ' num2str(h) ',\lambda = '
           num2str(lambda) ', Time = ' num2str(td(i))])
51     xlabel('x')
52     ylabel('u')
53     grid on
54     M(i) = getframe;
55 end
56
57
58 for i = 1:q
59     e(i,:) = abs((v(i,:)-u(i,:)));
60     err(i) = sum(e(i,:)).^2;
61 end
62
63 figure()
64 plot(td,err,'b-')
65 title(['SSE vs time: a = 1, h = ' num2str(h) ',\lambda = ' num2str(
       lambda)])
66 xlabel('Time')
67 ylabel('SSE')
68 grid on
```

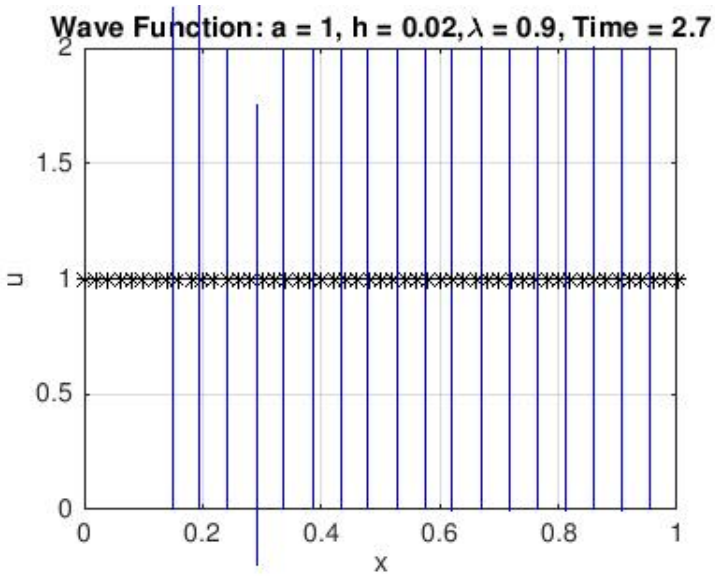

Wave Function: a = 1, h = 0.02, λ = 0.9, Time = 2.7
u
x
0
0.2
0.4
0.6
0.8
1
0.5
1.5
2

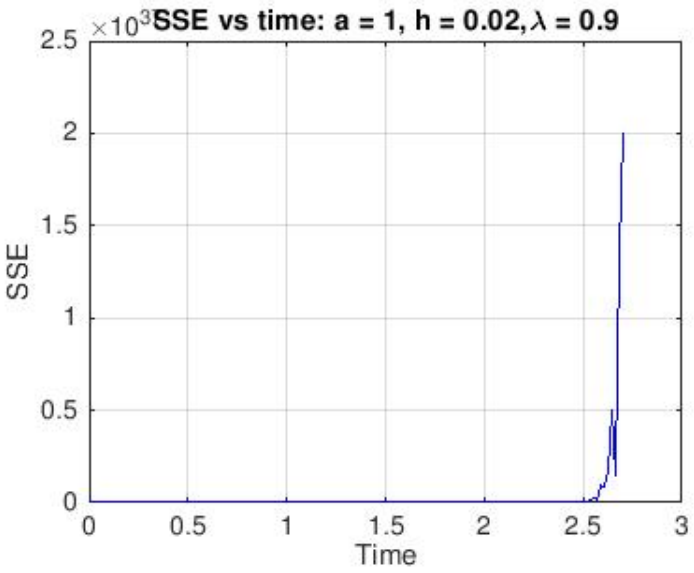

SSE vs time: a = 1, h = 0.02, λ = 0.9
SSE
Time
0
2000
4000
6000
8000
10000
12000
14000
0.5
1
1.5
2
2.5
3

1.4 Matlab Code for Part c

```
clear all


lambda = .9;
h = 1/50;
xd = 0:h:1;
k = lambda*h;
p = length(xd);
td = 0:k:2.7;
q = length(td);


for i = 1:p
    u(1,i) = f_sol(xd(i));
end


for i = 1:q-1 %lax-friendrichs scheme
    u(i,1) = u(i,2) - 2*u(i,3); %left boundary condition
    for j = 1:p-2
        u(i+1,j+1) = -lambda*((u(i,j+2) - u(i,j))/2) + ((u(i,j+2) + u(i
            ,j))/2);
    end
end


for i = 1:q-1 %run leap frog
    u(i+1,1) = 2*u(i+1,2) - u(i+1,3); %left boundary condition
    for j = 2:p-2
        u(i+2,j+1) = -lambda*(u(i+1,j+2) - u(i+1,j)) + u(i,j+1);
        u(i+1,p) = u(i,p-1); %right boundary condition
    end
end

for i = 1:q
    v(i,1) = 0;
end


for i = 1:q
    for j = 1:p
        v(i,j) = F_341((xd(j) -  td(i)));
    end
end
```

```
44
45 for i = 1:q
46     plot(xd,u(i,:),'b-o',xd,v(i,:),'k-*');
47     ylim([0,2])
48     xlim([0,1])
49     title(['Wave Function: a = 1, h = ' num2str(h) ',\lambda = '
           num2str(lambda) ', Time = ' num2str(td(i))])
50     xlabel('x')
51     ylabel('u')
52     grid on
53     M(i) = getframe;
54 end
55
56
57 for i = 1:q
58     e(i,:) = abs((v(i,:)-u(i,:)));
59     err(i) = sum(e(i,:)).^2;
60 end
61
62 figure()
63 plot(td,err,'b-')
64 title(['SSE vs time: a = 1, h = ' num2str(h) ',\lambda = ' num2str(
       lambda)])
65 xlabel('Time')
66 ylabel('SSE')
67 grid on
```

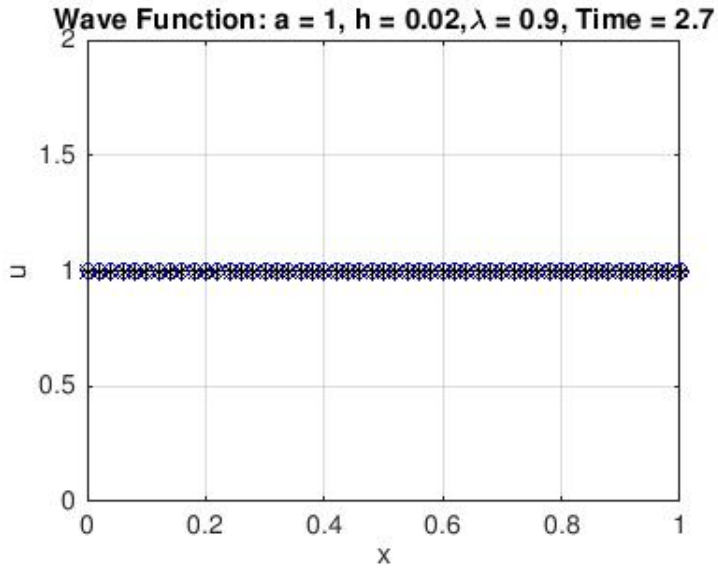


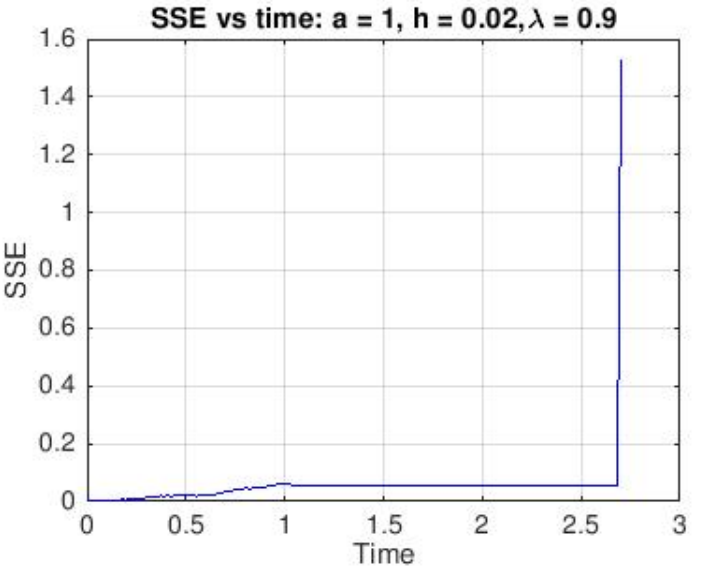

SSE vs time: a = 1, h = 0.02,λ = 0.9
×10^3
SSE
Time
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3

1.5 Matlab Code for Part d

```
clear all


lambda = .9;
h = 1/50;
xd = -4:h:1;
k = lambda*h;
p = length(xd);
td = 0:k:2.7;
q = length(td);


for i = 1:p
    u(1,i) = f_sol(xd(i));
end


for i = 1:q-1 %lax-friendrichs scheme
    u(i,1) = 1;
    for j = 1:p-2
        u(i+1,j+1) = -lambda*((u(i,j+2) - u(i,j))/2) + ((u(i,j+2) + u(i
            ,j))/2);
    end
end


for i = 1:q-1 %run leap frog

    for j = 2:p-2
        u(i+2,j+1) = -lambda*(u(i+1,j+2) - u(i+1,j)) + u(i,j+1);
        u(i+1,p) = u(i,p-1);
    end
end


for i = 1:q
    v(i,1) = 1;
end


for i = 1:q
    for j = 1:p
        v(i,j) = F_341((xd(j) -  td(i)));
    end
```

```
end

for i = 1:q
    plot(xd,u(i,:),'b-o',xd,v(i,:),'k-*');
    ylim([0,2])
    xlim([0,1])
    title(['Wave Function: a = 1, h = ' num2str(h) ',\lambda = '
        num2str(lambda) ', Time = ' num2str(td(i))])
    xlabel('x')
    ylabel('u')
    grid on
    M(i) = getframe;
end


for i = 1:q
    e(i,:) = abs((v(i,:)-u(i,:)));
    err(i) = sum(e(i,:)).^2;
end

figure()
plot(td,err,'b-')
title(['SSE vs time: a = 1, h = ' num2str(h) ',\lambda = ' num2str(
    lambda)])
xlabel('Time')
ylabel('SSE')
grid on
```


Wave Function: a = 1, h = 0.02, λ = 0.9, Time = 2.7
u
x
0
0.5
1
1.5
2
0.2
0.4
0.6
0.8


SSE vs time: a = 1, h = 0.02, λ = 0.9
SSE
Time
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0.5
1.5
2
2.5
3

As expected, only d gives good results, while all the other schemes blow up.