Assignment VII

M693B
DR. BLOMGREN
MAY 9, 2018

MATTEO POLIMENO

1 Problem 10.4.2 from Strikverda

In this problem we have to numerically solve the heat equation

$$u_t = u_{xx}.$$

Initial data is given to be

$$u(0,x) = \begin{cases} 1 & \text{if } |x| \le \frac{1}{2} \\ \frac{1}{2} & \text{if } |x| = \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

The scheme was run over the interval $-1 \le x \le 1$ for $0 \le t \le 1/2$. Here is the code used

```
clear all
tic
h = 1/80;
b = 1;
mu=10;
tmax = .5;


k = mu*h^2;
td = 0:k:tmax;
q = length(td);
xd = -1:h:1;
p = length(xd);
L = 0:500;
u0 = @(x) (abs(x) < 0.5)+(abs(x) == 0.5)/2;
u = zeros(p,1);
for i = 1:p %initial conditions
    u(i,1) = u0(xd(i));
end

w = zeros(q,p);

for ii = 1:q %calculates exact solution
    for jj = 1:p
        S1 = ((((-1).^L)./(pi*(2*L+1))) + 2./((pi^2)*((2*L+1).^2)))...
            .*cos(pi*(2*L+1)*xd(jj)).*(exp(1).^(-(pi^2)*((2*L+1).^2)*td
                (ii)));
        S2 = (cos(2*pi*(2*L+1)*xd(jj))./((pi^2)*((2*L+1).^2)))...
            .*(exp(1).^(-4*(pi^2)*((2*L+1).^2)*td(ii)));
        w(ii,jj) = (3/8) + sum(S1) + sum(S2);
    end
end

```

```
alpha = -(b/2)*mu;
beta = b+b*mu;
mbeta = b-b*mu;

T = zeros(p,p);



T(1,1) = beta; %creates T matrix
T(1,2) = 0;
for i=2:p-1
    T(i,i-1) = alpha;
    T(i,i) = beta;
    T(i,i+1) = alpha;
end
T(p,p) = beta;
T(p,p-1) = 0;

B = zeros(p,p);

B(1,1) = mbeta; %creates B matrix
B(1,2) = -alpha;
for i=2:p-1
    B(i,i-1) = -alpha;
    B(i,i) = mbeta;
    B(i,i+1) = -alpha;
end
B(p,p) = mbeta;
B(p,p-1) = -alpha;


v = zeros(p,1);
v(1) = w(1,1); %sets initial boundary conditions
for i = 2:p-1
    v(i) = u(i,1);
end
v(p) = w(1,p);

for i = 2:q
    T(1,1) = ((1-mu)*v(1)+mu/2*v(2))/w(i,1); %changes boundary values
    T(p,p) = (mu/2*v(p-1)+(1-mu)*v(p))/w(i,p);
    v = T\(B*v); %calculates solution vector
    plot(xd,v,'b-o',xd,w(i,:),'k');
    ylim([0,1.1])
    xlim([-1,1])
```

```
    title(['Heat Equation: h = ' num2str(h) ',\mu = ' num2str(mu) ',
        Time = ' num2str(td(i))])
    xlabel('x')
    ylabel('u')
    grid on
    M(i) = getframe;
end

supnorm = max(abs(w(q,:)-v));

L2norm = norm(w(q,:)-v)

time = toc
```

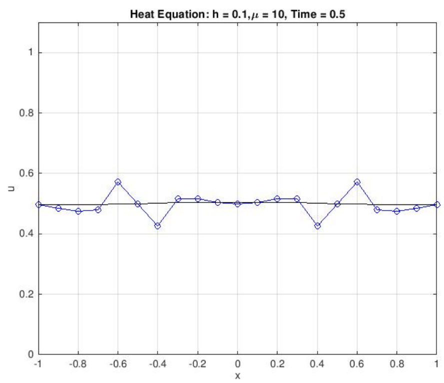

Figure 1: Analytical solution vs. scheme

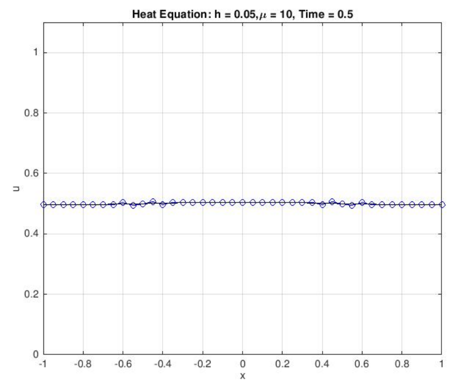

Figure 2: Analytical solution vs. scheme

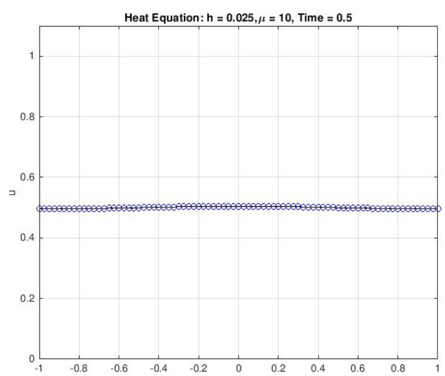

Figure 3: Analytical solution vs. scheme

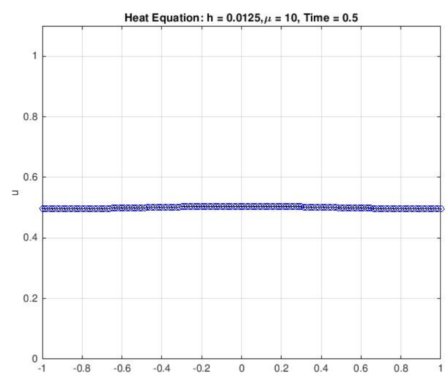

Figure 4: Analytical solution vs. scheme

Thus, we can see that for $h = 1/10$ and $h = 1/20$ there is still some instability, whereas for $h = 140$ and $h = 1/80$, the scheme is very useful.

L2 Norm and Supremum Norm		
h value	Supremum Norm	L2 Norm
1/10	0.0054	0.0154
1/20	0.0090	0.0359
1/40	0.0092	0.0502

2 Problem 13.5.1 from Strikwerda

In this problem, we solve the Poisson equation

$$u_{xx} + u_{yy} = -2\cos(x)\sin(y)$$

using the SOR method. Here is the code implemented

```
% Solve the Poisson Equation using SOR method
 clear; close all;
L = 1;
dx = .1;
dy = dx;
h= dx;
n = round(L/dx);
m = n;
x = linspace(0,L,n)';
y = linspace(0,L,m)';
T_0 = zeros(n,m);

bb = 2/(1 + pi*h);

% Boundary Condition cosxsiny
% At x= 0;
 for i =1:n
     T_0(i,1) = cos(x(1))*sin(y(i));
% At x = 1
     T_0(i,m) = cos(x(m)).*sin(y(i));
end
% At y = 0;
 for j =1:m
     T_0(1,j) = cos(x(j))*sin(y(1));
% At y = 1;
     T_0(m,j) = cos(x(j))*sin(y(m));
end
```

```
T= T_0;

S = zeros(n,m);

for i = 1:n
    for j=1:m
           S(i,j)= -2*cos(x(i))*sin(y(j));
     end
end


max_step = 1000;


for l=1:max_step
 for i=2:n-1
      % Boundary Conditions

      for j=2:m-1
          T(i,j)=bb*0.25*(T(i+1,j)+...
          T(i,j+1)+T(i-1,j)+T(i,j-1)-h^2*S(i,j))+(1.0-bb)*T(i,j);
          % Boundary Conditions

      end
 end
 % find residual
 res=0;
 for i=2:n-1
      for j=2:m-1
          res=res+abs(T(i+1,j)+...
          T(i,j+1)+T(i-1,j)+T(i,j-1)-4*T(i,j))/h^2 - S(i,j);
      end

 end
 if rem(l,2) == 0
      figure(1);
      surf(x,y,T);
      title([' SOR for Poisson Equation, at iteration ',num2str(l),' with'...
          ' h =',num2str(h)]);
      xlabel('x')
      ylabel('y')
      zlabel('z')
      axis([ 0 1 0 1 0 1]);
      getframe(gcf);
 end
```

```
 l,res/((m-2)*(n-2)) % Print iteration and residual
 if (res/((m-2)*(n-2)) < 0.001)
     break
 end
 disp('Norm: ')
 Lnorm = (norm(T)^2*h^2)
end
```

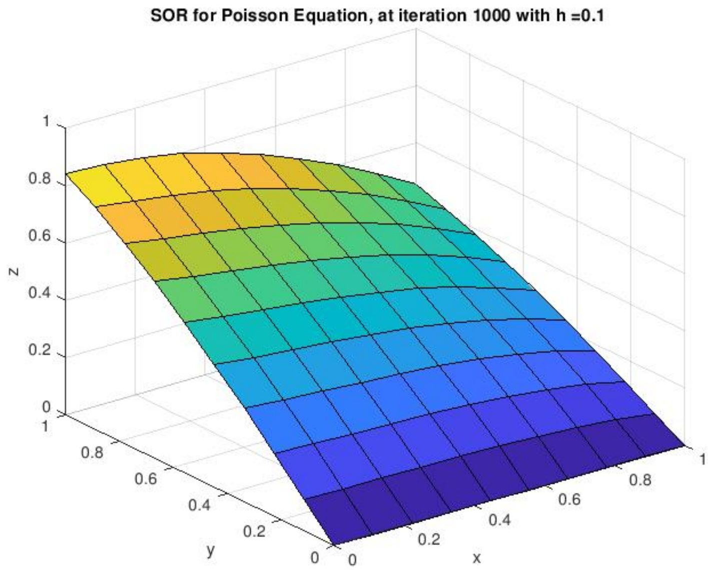

Figure 5: Solution to the Poisson equation

We display our results for 1000 iterations