MATH 612

Numerical Methods for Partial Differential Equations

Spring Term 2004

Instructor: Georg Dolzmann

Solutions to Homework Set 9

Problem 1: Show that the row-sum norm,

$$\|A\|_\infty = \max_{\alpha=1,\dots,N} \sum_{\beta=1}^N |a_{\alpha\beta}|$$

is the operator norm of the $N \times N$ matrix A associated with the $\|\cdot\|_\infty$ norm on $\mathbb{R}^N$, that is,

$$\|A\|_\infty = \max_{u\neq 0} \frac{\|Au\|_\infty}{\|u\|_\infty}.$$

Solution: We first compute that

$$\|Au\|_\infty = \max_{\alpha=1,\dots,N} \Big|\sum_{\beta=1}^N a_{\alpha\beta}u_\beta\Big| \leq \max_{\alpha=1,\dots,N} \sum_{\beta=1}^N |a_{\alpha\beta}|\,|u_\beta| \leq \max_{\alpha=1,\dots,N} \sum_{\beta=1}^N |a_{\alpha\beta}|\,\|u\|_\infty$$

and hence

$$\|A\|_\infty = \max_{u\neq 0} \frac{\|Au\|_\infty}{\|u\|_\infty} \leq \max_{\alpha=1,\dots,N} \sum_{\beta=1}^N |a_{\alpha\beta}|$$

We only need to show that we can have equality in this inequality. To see this, assume that $A \neq 0$ (there is nothing to show if $A = 0$) and let α be an index of a row with maximal sum. Define the vector

$$u_\beta = \begin{cases} 0 & \text{if } a_{\alpha\beta} = 0, \\ 1 & \text{if } a_{\alpha\beta} > 0, \\ -1 & \text{if } a_{\alpha\beta} < 0. \end{cases}$$

Then $\|u\|_\infty = 1$ (by assumption, $u \neq 0$) and

$$\|Au\|_\infty = \max_{\alpha=1,\dots,N} \Big|\sum_{\beta=1}^N a_{\alpha\beta}u_\beta\Big| = \max_{\alpha=1,\dots,N} \Big|\sum_{\beta=1}^N |a_{\alpha\beta}|\,|u_\beta|\Big| = \max_{\alpha=1,\dots,N} \sum_{\beta=1}^N |a_{\alpha\beta}|,$$

and hence

$$\|A\|_\infty \geq \max_{\alpha=1,\dots,N} \sum_{\beta=1}^N |a_{\alpha\beta}|.$$

This establishes the assertion.

Problem 2: Let $\Omega = [0,1] \times [0,1]$ be the unit square in the plane. Consider the Dirichlet problem

$$-\Delta u = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega$$

with

$$f(x,y) = \sin(\pi x)\sin(\pi y) + \sin(\pi x)\sin(2\pi y).$$

Compute the approximate solution by the finite difference discretization based on the five-point-star with $h = 1/5,\ 1/10,\ 1/20,\ 1/40$. Here $\Delta x = \Delta y = h$, i.e., we use the same spacing in the x and in the y direction. What is the experimental rate of convergence in L^∞ and in L^2? You may use that the exact solution is given by

$$u(x,y) = \frac{1}{2\pi^2}\sin(\pi x)\sin(\pi y) + \frac{1}{5\pi^2}\sin(\pi x)\sin(2\pi y).$$

For $J = 5$, use the command `spy` to view and print the structure of the matrix in the linear system. Plot the solution and the error with $J = 20$.

Solution: Let $x_0 = 0, x_1, \ldots, x_J = 1$ be the grid points on the x-axis. We use the lexicographic ordering of the points, i.e., we order them as

$$(h,h),\ (2h,h), \ldots, ((J-1)h, h),\ (h,2h), \ldots ((J-1)h, (J-1)h).$$

We store the full system matrix; a better implementation would use the sparse matrix features in MATLAB. The structure of the matrix is

$$L_h = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & & \ddots & & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix}$$

where $-I$ is the negative of the $(J-1)\times(J-1)$ identity matrix, and

$$T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & & \ddots & & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}$$

We plot the structure of the matrix for $J = 5$ in Figure 1. The solution and the error for $J = 20$ are shown in Figures 2 and 3. We summarize the results for the error in Table 1. Here are the commands that we use to call the solution routine:

```
  clear all;
  close all;

  f =inline('sin(pi*x).*sin(pi*y)+sin(pi*x).*sin(2*pi*y)','x','y');
  ux=inline('1/2/pi^2*sin(pi*x).*sin(pi*y)+1/5/pi^2*sin(pi*x).*sin(2*pi*y)','x','y');
%
```

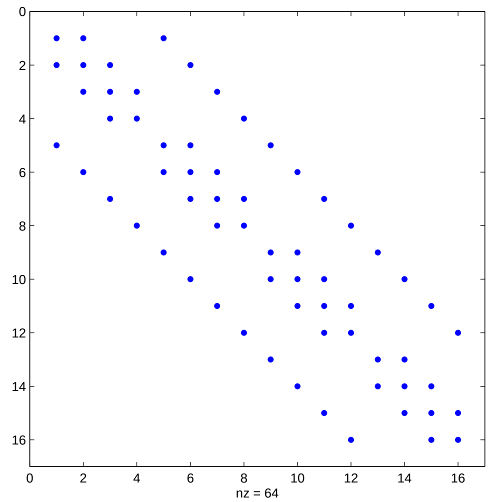

FIGURE 1. The block structure of the discretization of the Laplacian on the unit square.

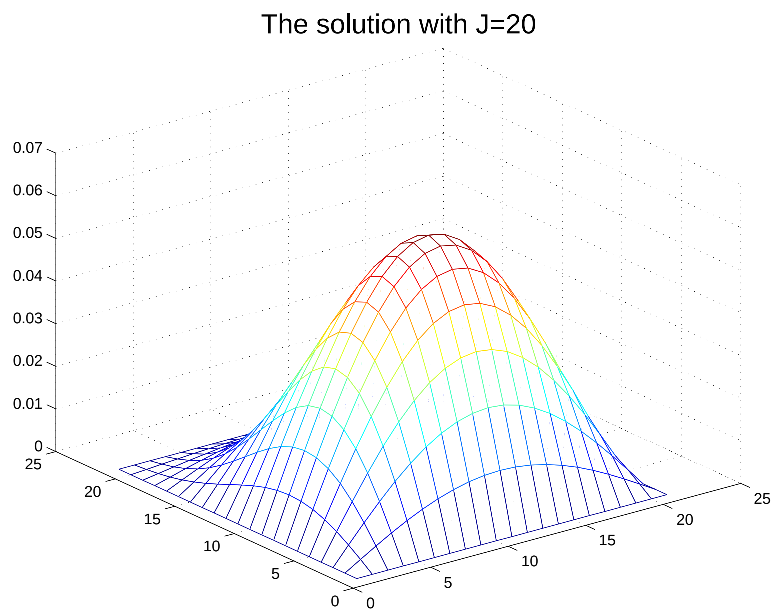

FIGURE 2. The solution of the Poisson equation with the five point star on the unit square and the error for $h = 1/20$.

```
% solve the problem and find the errors
%
  jvals=[5 10 20 40];
  l2_error=zeros(length(jvals),1);
  linf_error=zeros(length(jvals),1);
```

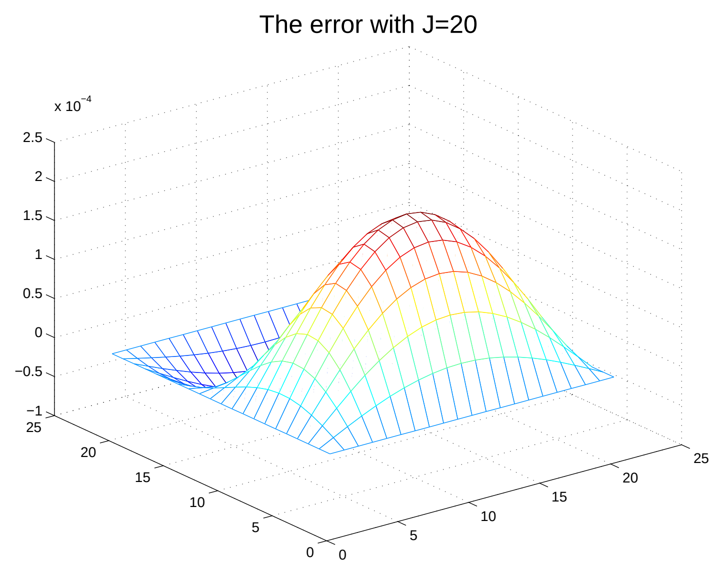

FIGURE 3. The error in the approximation of the Poisson equation with the five point star on the unit square for $h = 1/20$.

J	L^2-error	EOC	L^∞-error	EOC
5	0.0014756		0.0031323	
10	0.0003558	2.0519889	0.0008860	1.8217649
20	0.0000881	2.0127579	0.0002196	2.0123569
40	0.0000219	2.0031746	0.0000548	2.0022269

TABLE 1. The L^2 error and the L^∞ error in the calculation of the solution of Poisson's equation with the five-point star.

```
for j=1:length(jvals)
  [u,e2,einf]=sl_est(jvals(j),f,ux);
  l2_error(j)=e2;
  linf_error(j)=einf;
end;

conv_l2=zeros(length(jvals)-1,1);
conv_linf=zeros(length(jvals)-1,1);

for j=1:length(jvals)-1,
 conv_l2(j)  = -log(l2_error(j+1)/l2_error(j))/log(2);
 conv_linf(j)= -log(linf_error(j+1)/linf_error(j))/log(2);
end
```

The function `sl_est` calls the solver for the Poisson equation and evaluates the L^2 and the L^∞ error:

```
%
```

```
% solve the Poisson equation and compute
% the error estimates
%
  function [u,e2,einf] = sl_est(J,f,ux);
%
% solve the equation
%
  u=sl(J,f);
%
% find the error
%
  x=0:1/J:1;
  y=0:1/J:1;
  [X,Y]=meshgrid(x,y);

  eh=u-ux(X,Y);
  figure
  mesh(u);
  title(sprintf('The solution with J=%i',J),'FontSize',18);
  figure
  mesh(eh);
  title(sprintf('The error with J=%i',J),'FontSize',18);
  e2=(1/J)*sqrt(sum(sum(eh(2:J,2:J).^2)));
  einf=max(max(abs(eh)));
```

The equation is solved with the following code:

```
%
% solution of -Delta u = f on [0,1]^2
%
% J = 1/h = number of points on the axes
% f=f(x,y)= right-hand side of the equation
%
  function u = sl(J,f)
%
% generate the block structure of the matrix
%
  I=eye(J-1);
  T=4*eye(J-1);
  T=T+diag(-ones(J-2,1),1);
  T=T+diag(-ones(J-2,1),-1);
%
% define the big system matrix
%
  N=(J-1)*(J-1);
  L=zeros(N,N);
%
% fill in the block structure
% the first and the last line
%
```

```
  L(1:J-1,1:2*(J-1))=[T,-I];
  L(N-(J-1)+1:N,N-2*(J-1)+1:N)=[-I,T];
%
% the regular blocks with three entries
%
  for i=2:J-2,
    L((i-1)*(J-1)+1:i*(J-1),(i-2)*(J-1)+1:(i+1)*(J-1))=[-I,T,-I];
  end
%
% correct scaling of L
%
  h=1/J;
  L=L/h^2;
  spy(L)
%
% evaluation of f(x,y)=f(i*h,j*j)
%
  b=zeros(N,1);
  for i=1:J-1,
    for j=1:J-1
      b((j-1)*(J-1)+i)=f(i*h,j*h);
    end
  end

  v=L\b;
%
% produce a matrix as output
% in a format compatible with meshgrid
% fill in zeros on the boundary by
% substituting i->i+1, j->j+1
%
  u=zeros(J+1);
  for j=1:J-1,
    for i=1:J-1
      u(j+1,i+1)=v((j-1)*(J-1)+i);
    end
  end
```