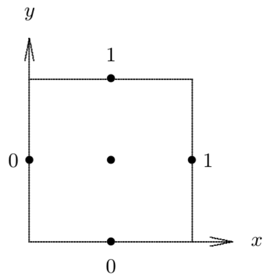
Scientific Computing Homework #5 & #6

李阳 11935018

May 31, 2020

Problem 1. *Consider a finite difference solution of the Poisson equation $u_{xx} + u_{yy} = x + y$ on the unit square using the boundary conditions and mesh points shown in the drawing. Use a second-order accurate, centered finite difference scheme to compute the approximate value of the solution at the center of the square.*



Solution.

$$\frac{u_{2,1} - 2u_{1,1} + u_{0,1}}{h_x^2} + \frac{u_{1,2} - 2u_{1,1} + u_{1,0}}{h_y^2} = x_1 + y_1$$

$$\Rightarrow u_{1,1} = \frac{u_{2,1} + u_{0,1} + u_{1,2} + u_{1,0} - h^2(x_1 + y_1)}{4}$$

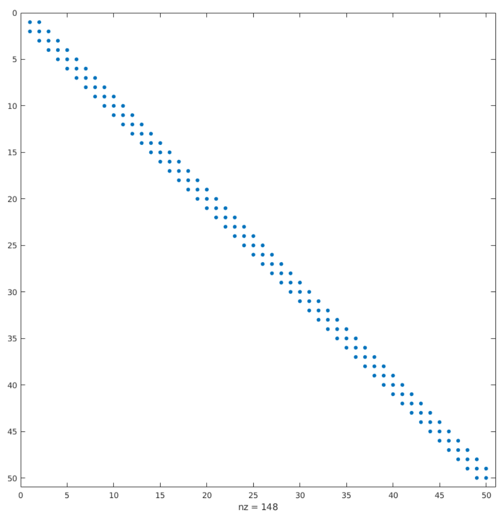
$$\Rightarrow u_{1,1} = \frac{7}{16}.$$

□

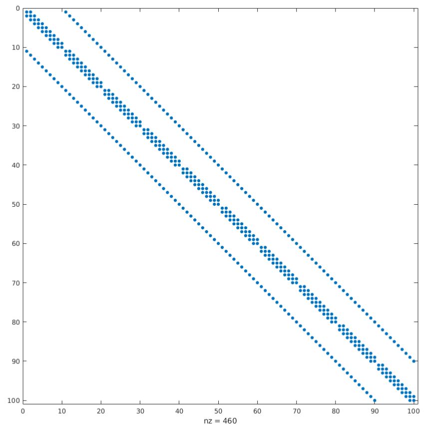
Problem 2. *Draw pictures to illustrate the nonzero pattern of the matrix resulting from a finite difference discretization of the Laplace equation on a d-dimensional grid, with k grid points in each dimension, for $d = 1, 2,$ and 3, as described at the end of Section 11.3.1. Use a value of k that is large enough to show the general pattern clearly. In each case, what are the numerical values of the nonzero entries?*

Solution.

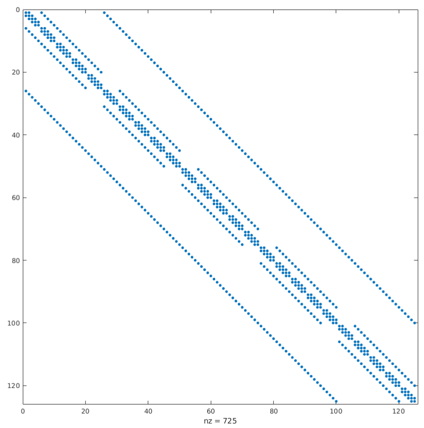
- $d = 1, k = 50$. The diagonal entries are -2, and all other nonzero entries are 1.



- $d = 2, k = 10$. The diagonal entries are -4, and all other nonzero entries are 1.



- $d = 3, k = 5$. The diagonal entries are -8, and all other nonzero entries are 1.



$\square$

Problem 3. *Prove that the Jacobi iterative method for solving the linear system $A\mathbf{x} = \mathbf{b}$ converges if the matrix A is diagonally dominant by rows. (Hint: Use the ∞-norm.)*

Proof. Recall *Jacobi method*:

$$\mathbf{x}^{(k+1)} = -D^{-1}(L+U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b},$$

where D is a diagonal matrix with the same diagonal entries as A, and L and U are the strict lower and upper triangular portions of A, respectively. To prove the convergence of Jacobi method, we need to show that the spectral radius

$$\rho(D^{-1}(L+U)) < 1.$$

We have

$$\|D^{-1}(L+U)\|_\infty = \max_{1 \le i \le n} \sum_{j=1, j \ne i}^{n} |a_{ij}/a_{ii}|.$$

Diagonal dominance means that

$$|a_{ii}| > \sum_{j=1, j \ne i}^{n} |a_{ij}| \quad (1 \le i \le n).$$

We then conclude that

$$\|D^{-1}(L+U)\|_\infty < 1,$$

and therefore

$$\rho(D^{-1}(L+U)) \le \|D^{-1}(L+U)\|_\infty < 1,$$

which completes the proof. $\square$

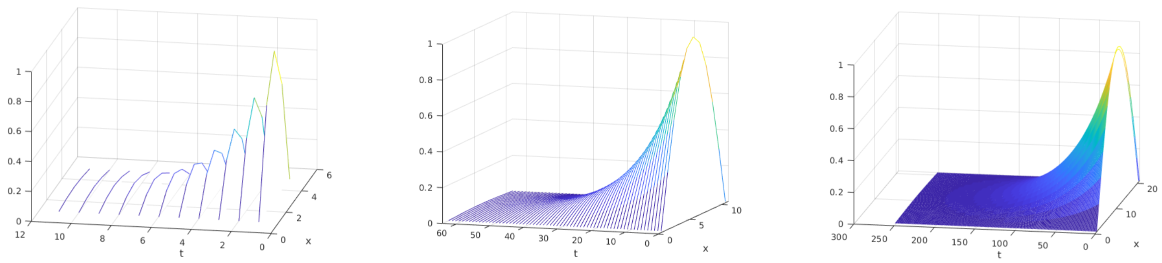
Problem 4. *Implement the following scheme:*

$$-\mu U_{j-1}^{n+1} + (1+2\mu)U_j^{n+1} - \mu U_{j+1}^{n+1} = U_j^n, \quad \forall j = 1, 2, \ldots, (N-1).$$

Solution. Modify the given code to obtain

```
% Solve U_t = \nu U_{xx}
% Set up parameters
N = 21;% number of grid points
a = 0; b = 1; nu = 1.0; T = 0.5;
h = (b-a)/(N-1); x = linspace(a,b,N); % space discretization
tau = 0.8*h*h/nu; % time step
mu = nu*tau/h/h;
NT = ceil(T/tau); uh = zeros(N,NT+1);
uh(:,1)  = sin(pi*x);  % u_0 = sin(\pi x);
main = (1+2*mu)*sparse(ones(N-2,1));
off  = -mu*sparse(ones(N-3,1));
A = diag(main) + diag(off,1) + diag(off,-1);
for  n = 1:NT
     uh(2:N-1,n+1) = A\uh(2:N-1,n);
end
waterfall(uh'); xlabel('x'); ylabel('t');
```

The result is shown in the following figures.



(a) $N = 5$ (b) $N = 11$ (c) $N = 21$

Compare with the explicit scheme, we find that the above implicit scheme is unconditionally stable, i.e., we don't need the restriction $\Delta t = \mathcal{O}(\Delta x^2)$ to ensure convergence. □

Problem 5. *Implement the Total Variation Diminishing(TVD) scheme.*

Solution. We have the following method for conservation law

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[\mathcal{F}(Q_i^n, Q_{i+1}^n) - \mathcal{F}(Q_{i-1}^n, Q_i^n)\right],$$

where $\mathcal{F}(Q_i^n, Q_{i+1}^n) \approx F_{i+\frac{1}{2}}^n = h(Q_{i+\frac{1}{2}}^-, Q_{i+\frac{1}{2}}^+)$. For a TVD scheme, we require that the numerical flux function $h(\cdot,\cdot)$ satisfies

- Lipschitz continuous;
- monotone;
- $h(a, a) = a$.

Here we take

$$h(a, b) = 0.5(f(a) + f(b) - \alpha(b-a)), \text{ with } \alpha = \max_u |f'(u)|.$$

For Burger's equation,

$$f(u) = \frac{u^2}{2}.$$

Modify the given code `convection_fvm.m`, we obtain the following main part of the code:

```
h1 = 0.5*(0.5*un(i).^2+0.5*un(i+1).^2-norm(un,inf)*(un(i+1)-un(i)));
h2 = 0.5*(0.5*un(i-1).^2+0.5*un(i).^2-norm(un,inf)*(un(i)-un(i-1)));
u(i+1) = un(i) - (dt/dx)*(h1-h2);
```

□

Problem 6. *Solve Poisson equation on the unit square.*

Solution. We take the solution to be

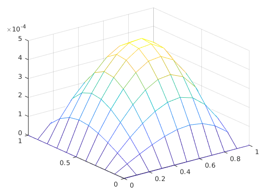
$$u(x, y) = (x^2 - x^4)(y^4 - y^2),$$

and therefore the right-hand side is given by

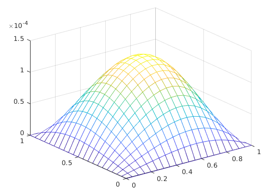
$$f(x, y) = 2\left[(1 - 6x^2)y^2(1 - y^2) + (1 - 6y^2)x^2(1 - x^2)\right].$$

Modify the given code, and we obtain the following numerical result. Note that the convergence rate is obtained by the formula

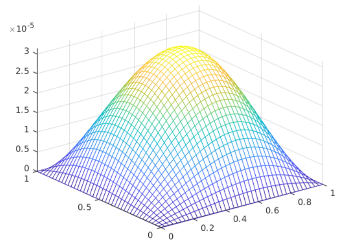
$$p \approx \frac{\log(\|\mathbf{e}_h\|_\infty) - \log(\|\mathbf{e}_{2h}\|_\infty)}{\log 2}.$$



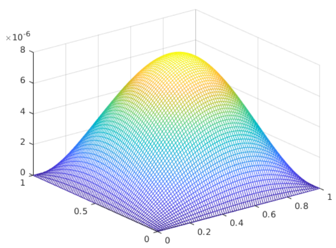
(d) $N = 10$



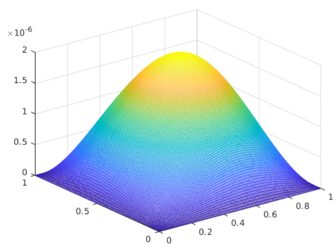
(e) $N = 20$



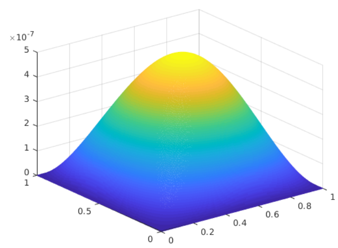
(f) $N = 40$



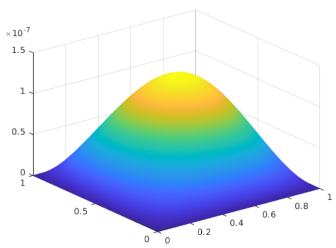
(g) $N = 80$



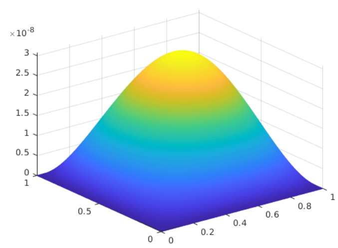
(h) $N = 160$



(i) $N = 320$



(j) $N = 640$



(k) $N = 1280$

n	$\|e^h\|_\infty$	convergence rate
10	4.99e-4	
20	1.26e-4	2.00
40	3.14e-5	2.00
80	7.87e-6	2.00
160	1.97e-6	2.00
320	4.92e-7	2.00
640	1.23e-7	2.00
1280	3.07e-8	2.00

Therefore, the convergence rate of the proposed numerical scheme is 2. □

Problem 7. *Run the code* `main_FD5Newton.m` *to calculate the convergence rate of Newton's method.*

Solution. The convergence rate is calculated by

$$p \approx \frac{\log e_{n+2} - \log e_{n+1}}{\log e_{n+1} - \log e_n}.$$

The numerical result is as follows.

k	$\|\boldsymbol{e}_k\|_\infty$	convergence rate
1	41.15	
2	1.14	
3	2.87e-3	1.67
4	1.80e-8	2.01
5	6.20e-14	1.05

Therefore, the convergence rate of Newton's method is 2. □

Problem 8. *Modify the code* `demo_elliptic_fem_p1.m` *to solve the following diffusion equation*

$$\nabla \cdot (\kappa(x,y)\nabla u) = f(x,y) \text{ in } \Omega,$$

with

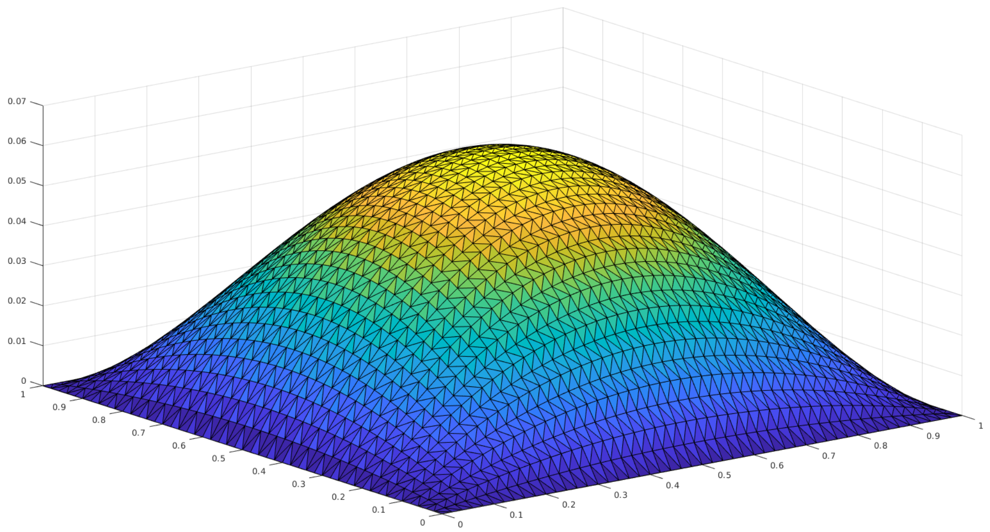
- *nonconstant coefficient* $\kappa(x,y) = 1 + xy^2$*;*
- *analytic solution* $u(x,y) = xy(1-x)(1-y), u = 0$ *on the boundary;*
- *right hand side*

$$f(x,y) = -y^3 + y^4 + 4y^3x - 4y^4x + 2y - 2y^2 - 2x^2y + 6x^2y^2 + 2x^3y - 6x^3y^2 + 2x - 2x^2.$$

Solution. The following shows a modification of `demo_elliptic_fem_p1.m`.

```
fun_D = @(x,y) 1+x.*y.^2;
fun_r = @(x,y) zeros(size(x));                        % reaction term
fun_f = @(x,y) -y.^3+y.^4+4*y.*y.*y.*x-4*y.*y.*y.*y.*x+2*y ...
    - 2*y.^2-2*x.*x.*y+6*x.*x.*y.*y+2*x.*x.*x.*y ...
    - 6*x.*x.*x.*y.*y+2*x-2*x.^2;
fun_u = @(x,y) x.*y.*(1-x).*(1-y);          %  Analyt
fun_g = @(x,y) x.*y.*(1-x).*(1-y);          %  Dirichlet boundary
```

Run the code and we obtain the following result:



□