HW15

HW #1. §6.6 # 3.

All the details are left as exercises.

- #3a. $L = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & -2/3 & 1 \end{bmatrix}$, $D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3/2 & 0 \\ 0 & 0 & 4/3 \end{bmatrix}$.
- #3c. $L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 1 & 0 & 0 \\ -1/4 & -3/11 & 1 & 0 \\ 0 & 0 & 11/25 & 1 \end{bmatrix}$, $D = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 11/4 & 0 & 0 \\ 0 & 0 & 50/11 & 0 \\ 0 & 0 & 0 & 78/25 \end{bmatrix}$.

HW #1. §6.6 # 5.

All the details are left as exercises.

- #5a. $L = \begin{bmatrix} 1.414213562373095 & 0 & 0 \\ -0.707106781186547 & 1.224744871391589 & 0 \\ 0 & -0.816496580927726 & 1.154700538379251 \end{bmatrix}.$
- #5c. $L = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0.5 & 1.658312395177700 & 0 & 0 \\ -0.5 & -0.452267016866645 & 2.132007163556104 & 0 \\ 0 & 0 & 0.938083151964686 & 1.766352173265569 \end{bmatrix}.$

HW #1. §6.6 # 14.

All the details are left as exercises.

- #14a. $L = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}$, $D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$.
- #14b. $L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & -1 & 1 \end{bmatrix}$, $D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

HW #1. §6.6 # 17.

The details are left as exercises.

When $-2 < \alpha < 3/2$ the matrix is positive definite.

HW #1. §6.6 # 24.

- (a) Yes. (Why?)
- (b) Not necessarily. Consider $\begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$.
- (c) Not necessarily. Consider $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ and $\begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}$.
- (d) Not necessarily. Consider $\begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$.
- (e) Not necessarily. Consider $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ and $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$.

HW #1. §6.6 # 25.

- (a) No; for example, consider $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
- (b) Yes, since $A = A^t$.
- (c) Yes, since $\mathbf{x}^t(A+B)\mathbf{x} = \mathbf{x}^t A\mathbf{x} + \mathbf{x}^t B\mathbf{x}$.
- (d) Yes, since $\mathbf{x}^t A^2 \mathbf{x} = \mathbf{x}^t A^t A\mathbf{x} = (A\mathbf{x})^t A\mathbf{x} \geq 0$, and because A is nonsingular, equality holds only if $\mathbf{x} = \mathbf{0}$.
- (e) No; for example, consider $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$.

HW #1. §6.6 # 32.

- (a)
$$D^{1/2}D^{1/2} = \begin{bmatrix} \sqrt{d_{11}} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sqrt{d_{nn}} \end{bmatrix} \begin{bmatrix} \sqrt{d_{11}} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sqrt{d_{nn}} \end{bmatrix} = \begin{bmatrix} d_{11} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & d_{nn} \end{bmatrix} = D.$$
- (b) We have
$$(L\hat{}\, D^{1/2})(L\hat{}\, D^{1/2})^t = L\hat{}\, D^{1/2} \ \ (D^{1/2})^t \ \ (L\hat{}\,)^t = L\hat{}\, D^{1/2} \ \ D^{1/2} \ \ (L\hat{}\,)^t = L\hat{}\, D \ \ (L\hat{}\,)^t = A.$$
Since $LL^t = A$ and the uniqueness of Choleski decomposition (can be easily derived from the algorithm), we have $L\hat{}\, D^{1/2} = L$.

HW #2.

The details are left as exercises.

The linear system is

$$\left(\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} - I\right)\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 - \frac{\alpha}{h^2} \\ f_2 \\ \vdots \\ f_{N-2} \\ f_{N-1} - \frac{\beta}{h^2} \end{bmatrix}$$

where $u_i = u(x_i)$ I is the identity matrix of size $(N-1) \times (N-1)$.

HW #3.

Run the following code. Then you will find the result

$$\log 2 \frac{abs(error(100) - error(200))}{abs(error(200) - error(400))} = 1.9998 \approx 2$$

where $error(N) = \max_i |u^e(x_i) - u_i^h|$.

```
%Gaussian elimination
function err = ge(N)

%Set up exact solution
h = 1/N;
x = 0:h:1;
u = exp(x);

%Set up entries and variables
n = N-1;
A = ones(1,n  )*(-2/h^2-1);
B = ones(1,n-1)/h^2;
C = ones(1,n-1)/h^2;
b = zeros(1,n);
b(1) = -1/h^2;
b(n) = -e/h^2;

%gaussian elimination
for i = 1:n-1
    A(i+1) = A(i+1)-C(i)*B(i)/A(i);
    b(i+1) = b(i+1)-b(i)*B(i)/A(i);
end

z = zeros(1,n);
z(n) = b(n)/A(n);
for i = n-1:-1:1
    z(i) = (b(i)-C(i)*z(i+1))/A(i);
end

%Compute error
err = norm(z-u(2:n+1), inf);

end
```

```matlab
%Crout
function err = crout(N)

%Set up exact solution
h = 1/N;
x = 0:h:1;
u = exp(x);

%Set up entries and variables
n = N-1;
A = ones(1,n  )*(-2/h^2-1);
B = ones(1,n-1)/h^2;
C = ones(1,n-1)/h^2;
D = zeros(1,n  );
E = zeros(1,n-1);
F = zeros(1,n-1);
b = zeros(1,n  );
b(1) = -1/h^2;
b(n) = -e/h^2;

%Solve Lz=b
z = zeros(1,n);
D(1) = A(1);
F(1) = C(1)/D(1);
z(1) = b(1)/D(1);
for i = 1:n-2
  E(i) = B(i);
  D(i+1) = A(i+1) - E(i)*F(i);
  F(i+1) = C(i+1)/D(i+1);
  z(i+1) = (b(i+1)-E(i)*z(i)) / D(i+1);
end
E(n-1) = B(n-1);
D(n) = A(n)-E(n-1)*F(n-1);
z(n) = (b(n)-E(n-1)*z(n-1))/D(n);

%Solve Ux=z
x = zeros(1,n);
x(n) = z(n);
for i = n-1:-1:1
  x(i) = z(i) - F(i)*x(i+1);
end

%Compute error
err = norm(x-u(2:N), inf);

end
```

HW #4.

- For 2D case, let $u_h = (u_{1,1}, \dots, u_{1,N-1}, \dots \dots, u_{N-1,1}, \dots, u_{N-1,N-1})$. Then

$$A = \frac{1}{h^2}\begin{bmatrix} -4 & 1 & 0 & \overset{(N-3)\text{ 0s}}{\cdots} & 0 & 1 & & & \\ 1 & \ddots & \ddots & & & & \ddots & & \\ 0 & \ddots & \ddots & \ddots & & & & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & & & & 1 \\ 0 & & & \ddots & \ddots & \ddots & & & 0 \\ 1 & & & & \ddots & \ddots & \ddots & & \vdots \\ & \ddots & & & & \ddots & \ddots & \ddots & 0 \\ & & \ddots & & & & \ddots & \ddots & 1 \\ & & & 1 & 0 & \cdots & 0 & 1 & -4 \end{bmatrix}. \text{ (Why?)}$$

  The number of operations are (Why?)

  - LU: N^4
  - FS: N^3
  - BS: N^3

- For 3D case, let $u_h = (u_{1,1,1}, \dots, u_{1,1,N-1}, u_{1,2,1}, \dots, u_{1,2,N-1}, \dots \dots)$. Then

$$A = \frac{1}{h^2}\begin{bmatrix} -6 & 1 & 0 & \overset{(N-3)\text{ 0s}}{\cdots} & 0 & 1 & 0 & \overset{(N-1)^2-3\text{ 0s}}{\cdots} & 0 & 1 & & & \\ 1 & \ddots & & \ddots & & & & & & & \ddots & & \\ 0 & & & & & & & & & & & & \\ \vdots & \ddots & & \ddots & & \ddots & & & & & & \ddots & \\ 0 & & & & & & & & & & & & \\ 1 & & & \ddots & & \ddots & & \ddots & & & & & 1 \\ 0 & & & & & & & & & & & & 0 \\ \vdots & & & & & \ddots & & \ddots & & \ddots & & & \vdots \\ 0 & & & & & & & & & & & & 0 \\ 1 & & & & & & & \ddots & & \ddots & \ddots & & 1 \\ & & & & & & & & & & & & 0 \\ & \ddots & & & & & & & & \ddots & \ddots & \ddots & \vdots \\ & & & & & & & & & & & & 0 \\ & & & \ddots & & & & & & & \ddots & \ddots & 1 \\ & & & & & 1 & & 0\dots0 & & 1 & 0\dots0 & 1 & -6 \end{bmatrix}. \text{ (Why?)}$$

The number of operations are (Why?)

- LU: N^7
- FS: N^5
- BS: N^5

HW #5. §7.3 #7a8a.

Run the following code. Then you will find the solution.

- #7a. $\mathbf{x}^{(6)} = (0.0353510682848767, -0.2367886265953429, 0.6577589535616284)^t$.
- #8a. $\mathbf{x}^{(6)} = (1.4478163499871399, -0.8358173037765774, -0.0447996184842250)^t$.

```
format long
n = 3;
Nmax = 100;
%#7a
%A = [3 -1 1; 3 6 2; 3 3 7];
%b = [1 0 4]';
%x0 = [0 0 0]';
%tol = 10^-3;
%#8a
A = [4 1 -1; -1 3 1; 2 2 5];
b = [5 -4 1]';
x0 = [0 0 0]';
tol = 10^-3;

x = zeros(3,1);
k = 1;
while (k <= Nmax)
    for i = 1:n
        x(i) = (-A(i,1:i-1)*x(1:i-1)-A(i,i+1:n)*x0(i+1:n)+b(i))/A(i,i);
    end
    if (norm(x-x0, inf) < tol)
        k
        x
        printf("The procedure was succesful.\n");
        break;
    end
    k = k+1;
    x0 = x;
end

if (k > Nmax)
    printf("Maxumum number of iterations exceeded.\n");
end
```

HW #5. §7.3 #9.

All the details are left as exercises.

- #9a.

$$T_j = \begin{bmatrix} 0 & 1/2 & -1/2 \\ -1 & 0 & -1 \\ 1/2 & 1/2 & 0 \end{bmatrix} \Rightarrow \rho(T_j) = \frac{\sqrt{5}}{2} > 1.$$

- #9c.

$$T_g = \begin{bmatrix} 0 & 1/2 & -1/2 \\ 0 & -1/2 & -1/2 \\ 0 & 0 & -1/2 \end{bmatrix} \Rightarrow \rho(T_g) = \frac{1}{2}.$$

HW #5. §7.3 #10.

All the details are left as exercises.

- #10a.

$$T_j = \begin{bmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{bmatrix} \Rightarrow \rho(T_j) = 0.$$

- #10c.

$$T_g = \begin{bmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{bmatrix} \Rightarrow \rho(T_g) = 2.$$

HW #5. §7.3 #17.

The matrix $T_j = (t_{ik})$ has entries given by

$$t_{ik} = \begin{cases} 0, & i = k \text{ for } 1 \le i, k \le n \\ -\frac{a_{ik}}{a_{ii}}, & i \ne k \text{ for } 1 \le i, k \le n. \end{cases}$$

Since A is strictly diagonally dominant,

$$\|T_j\|_\infty = \max_{1\le i\le n} \sum_{\substack{k=1 \\ k\ne i}}^{n} \left| \frac{a_{ik}}{a_{ii}} \right| < 1.$$