Project 2

Q:

给定一个NXN的稀疏矩阵（N差不多20000000大小）A，NX1的向量b，试设计如下问题的算法:
(1) A为三对角主对角占优矩阵，设计高效LU算法求解Ax=b;
(2) 现假设A为5对角主对角占优矩阵，设计相应高效LU算法求解Ax=b;
(3) 若仅假设每行仅有5-6个元素非0，但元素分布无规律，能否设计高效LU算法求解Ax=b?（以上需限定在16GB内存的个人计算机上实现！）

A:

(1)

idea:

We solve this problem by chase method.

$$A=\begin{bmatrix} d_1 & e_1 & & & \\ c_1 & d_2 & e_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & d_{n-1} & e_{n-1} \\ & & & c_{n-1} & d_n \end{bmatrix}=\begin{bmatrix} \alpha_1 & & & & \\ \gamma_1 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & \gamma_{n-2} & \alpha_{n-1} & \\ & & & \gamma_{n-1} & \alpha_n \end{bmatrix}\begin{bmatrix} 1 & \beta_1 & & & \\ & 1 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & \beta_{n-1} \\ & & & & 1 \end{bmatrix}=LU$$

$\beta_1=e_1/d_1$
$\beta_i=e_i/(d_i-c_{i-1}\beta_{i-1})$ (i = 2, 3, ..., n-1)
$\alpha_1=d_1$
$\alpha_i=d_i-c_{n-1}\beta_{i-1}$ (i = 2, 3, ..., n)
$y_1=b_1/d_1$
$y_i=(b_i-c_{i-1}y_{i-1})/(d_i-c_{i-1}\beta_{i-1})$ (i = 2, 3, ..., n)
$x_n=y_n$
$x_i=y_i-\beta_i x_{i+1}$ (i=n-1, n-2, ..., 2, 1)

code:

LU.m:

```
% A is an n*n matrix
% b is an n*1 vector
function [L,U,x,y] = LU(A,b,n)
L = zeros(n);
U = eye(n);
c = diag(A,-1);
d = diag(A,0);
e = diag(A,1);
alpha = zeros(n,1);
beta = zeros(n-1,1);
gama = zeros(n-1,1);
beta(1) = e(1)/d(1);
alpha(1) = d(1);
for i=2:n-1
    beta(i) = e(i)/(d(i)-c(i-1)*beta(i-1));
end
for i=2:n
    alpha(i) = d(i)-c(i-1)*beta(i-1);
end
for i=1:n-1
    gama(i) = c(i);
end
L = diag(gama,-1)+diag(alpha,0);
U = U+diag(beta,1);
x = zeros(n,1);
y = zeros(n,1);     % To chase method, calculate y.
y(1) = b(1)/d(1);
for i=2:n
    y(i) = (b(i)-c(i-1)*y(i-1))/(d(i)-c(i-1)*beta(i-1));
end
x(n) = y(n);
for i=n-1:-1:1
    x(i) = y(i)-beta(i)*x(i+1);
end
```

In order to test its validity, let us use a simple example. (n = 4)

main.m:

```
format short
A = [2,-1,0,0;-1,3,-2,0;0,-2,4,-2;0,0,-3,5];
b = [6,1,0,1];
[L,U,x,y] = LU(A,b,4)
```

results:

```
A =

    2    -1     0     0
    -1    3    -2     0
    0    -2     4    -2
    0     0    -3     5


L =

    2.0000         0         0         0
-1.0000    2.5000         0         0
        0   -2.0000    2.4000         0
        0         0   -3.0000    2.5000


U =

    1.0000   -0.5000         0         0
        0    1.0000   -0.8000         0
        0         0    1.0000   -0.8333
        0         0         0    1.0000


x =

    5
    4
    3
    2


y =

    3.0000
    1.6000
    1.3333
    2.0000
```

(2)

Similarly, we use chase method.

LU.m:

```
% A is an n*n matrix
% b is an n*1 vector
function [x,y] = LU(A,f,n)
a = diag(A,-2);
a = [0 0 a']';
b = diag(A,-1);
b = [0 b']';
c = diag(A,0);
d = diag(A,1);
e = diag(A,2);
alpha = zeros(n,1);
beta = zeros(n,1);
gama = zeros(n,1);
q = zeros(n,1);
z = zeros(n,1);
alpha(1) = c(1);
beta(1) = d(1)/d(2);
gama(2) = b(2);
alpha(2) = c(2)-gama(2)*beta(1);
for i=3:n
    z(i) = alpha(i);
end
for i=3:n
    gama(i) = b(i)-z(i)*beta(i-2);
end
for i=3:n
    alpha(i) = c(i)-z(i)*q(i-2)-gama(i)*beta(i-1);
end
for i=2:n-1
    beta(i) = (d(i)-gama(i)*q(i-1))/alpha(i);
end
for i=1:n-2
    q(i) = e(i)/alpha(i);
end
y = zeros(n,0);
y(1) = f(1)/alpha(1);
y(2) = (f(2)-gama(2)*y(1))/alpha(2);
for i=3:n
    y(i) = (f(i)-z(i)*y(i-2)-gama(i)*y(i-1))/alpha(i);
end
x = y;
x(n-1) = y(n-1)-beta(n-1)*x(n);
for i=n-2:-1:1
    x(i) = y(i)-q(i)*x(i+2)-beta(i)*x(i+1);
end
```

In order to test its validity, let us use a simple example. (n = 4)

main.m:

```
format short
A = [2,-1,1,0;-1,3,-2,1;1,-2,4,-2;0,1,-3,5];
b = [6,1,0,1];
[x,y] = LU(A,b,4)
```

results:

```
x =

    1.8439    1.4694    0.8429    0.5429


y =

    3.0000    1.1429    0.5714    0.5429
```