OJ7:带限矩阵方程组求解

【问题描述】

考虑如下的带限矩阵方程组：

$$AX = Z \tag{1}$$

其中$A \in R^{n\times n}$为带限的三对角或五对角矩阵，若为三对角矩阵则有以下形式：

$$A = \begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix}$$

若为五对角矩阵则有以下形式：

$$A = \begin{bmatrix} c_1 & d_1 & e_1 & & & & & & \\ b_2 & c_2 & d_2 & e_2 & & & & & \\ a_3 & b_3 & c_3 & d_3 & e_3 & & & & \\ & a_4 & b_4 & c_4 & d_4 & e_4 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & a_{n-2} & b_{n-2} & c_{n-2} & d_{n-2} & e_{n-2} \\ & & & & & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\ & & & & & & a_n & b_n & c_n \end{bmatrix}$$

$Z = [z_1, z_2, \dots, z_m] \in R^{n\times m}$ 由m个列向量构成。现在给定非奇异矩阵A和矩阵Z，求解矩阵X。

【输入格式】

第1行输入p，表示矩阵A中存在非零元素的对角线的条数，p为3或5。

第2行输入n和m，表示矩阵A的维数和矩阵Z的维数。其中n不超过10000，m不超过500.

第3行到第p+2行，按照从矩阵A的最上方的对角线到最下方的对角线的顺序依次输入各对角线的元素值。

即对于三对角矩阵，第3行输入$c_1, c_2, \dots, c_{n-1}$，第4行输入 $b_1, b_2, \dots, b_n$，第5行输入$a_2, \dots, a_n$；

对于五对角矩阵， 第3行输入$e_1, e_2, \dots, e_{n-2}$，第4行输入$d_1, d_2, \dots, d_{n-1}$，第5行输入$c_1, c_2, \dots, c_n$，第6行输入$b_2, b_3, \dots, b_n$，第7行输入$a_3, a_4, \dots, a_n$。

最后m行每行输入n个浮点数，为别为矩阵Z的第m列向量zm的n个元素值，即

$$z_{1,1}, \dots, z_{1,n}$$

$$\dots$$

$$z_{m,1}, \dots, z_{m,n}$$

【输出格式】

输出共m行，每行为n个浮点数，分别为矩阵X的每一列的各个元素值，每个元素值结果四舍五入保留4位小数。

【输入样例】

```
3
3 2
44 62
44 43 30
3 34
27 63 53
14 52 19
```

3
3 2
44 62
44 43 30
3 34
27 63 53
14 52 19

【输出样例】

```
-0.9846 1.5983 -0.0447
0.7073 -0.3892 1.0744
```

-0.9846 1.5983 -0.0447
0.7073 -0.3892 1.0744

【时间、内存限制】

时间1000ms

内存1500KB

【提示】

计算带限矩阵LU分解后元素间的递推表达式。

【思路】

1. 矩阵A作用于X的第i个列向量，得到Z的第i个列向量。

$$\begin{aligned} AX &= Z \\ Ax_i &= z_i \end{aligned} \tag{2}$$

   因此可以每读入一个Z的列向量，计算出X对应的列向量，进行输出。

2. 对矩阵A进行LU分解：

$$A = LU \tag{3}$$

   LU分解如下：

$$
\begin{bmatrix}
b_1 & c_1 & & & \\
a_2 & b_2 & c_2 & & \\
& \ddots & \ddots & \ddots & \\
& & a_{n-1} & b_{n-1} & c_{n-1} \\
& & & a_n & b_n
\end{bmatrix}
=
\begin{bmatrix}
\beta_1 & & & & \\
a_2 & \beta_2 & & & \\
& \ddots & \ddots & & \\
& & a_{n-1} & \beta_{n-1} & \\
& & & a_n & \beta_n
\end{bmatrix}
\begin{bmatrix}
1 & \gamma_1 & & & \\
& 1 & \gamma_2 & & \\
& & \ddots & \ddots & \\
& & & 1 & \gamma_{n-1} \\
& & & & 1
\end{bmatrix}
\tag{4}
$$

递推公式为:

$$
\begin{gathered}
\beta_1 = b_1,\ \gamma_1 = \frac{c_1}{b_1} \\
\beta_i = b_i - a_i\gamma_{i-1},\ \gamma_i = \frac{c_i}{\beta_i}
\end{gathered}
\tag{5}
$$

因此只需将数组`b[n]`替换为`β[n]`，将数组`c[n-1]`替换为`γ[n-1]`，即可求出LU分解。

先通过前向回代，求解:

$$
Ly = z \tag{6}
$$

再通过后向回代，求解:

$$
Ux = y \tag{7}
$$

三对角矩阵的形式如下:

$$
\begin{bmatrix}
b_1 & c_1 & & & \\
a_2 & b_2 & c_2 & & \\
& \ddots & \ddots & \ddots & \\
& & a_{n-1} & b_{n-1} & c_{n-1} \\
& & & a_n & b_n
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n
\end{bmatrix}
=
\begin{bmatrix}
d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n
\end{bmatrix}
$$

需要将三对角阵分解成如下的形式:

$$
\begin{bmatrix}
b_1 & c_1 & & & \\
a_2 & b_2 & c_2 & & \\
& \ddots & \ddots & \ddots & \\
& & a_{n-1} & b_{n-1} & c_{n-1} \\
& & & a_n & b_n
\end{bmatrix}
=
\begin{bmatrix}
\alpha_1 & & & & \\
\gamma_2 & \alpha_2 & & & \\
& \ddots & \ddots & & \\
& & \gamma_{n-1} & \alpha_{n-1} & \\
& & & \gamma_n & \alpha_n
\end{bmatrix}
\begin{bmatrix}
1 & \beta_1 & & & \\
& 1 & \beta_2 & & \\
& & \ddots & \ddots & \\
& & & 1 & \beta_{n-1} \\
& & & & 1
\end{bmatrix}
$$

求解Ax=d等价于求解Ly=d和Ux=y.

计算$\alpha_i, \beta_i, \gamma_i$的公式:

$\alpha_1 = b_1,\ \beta_1 = c_1/b_1$

$\gamma_i = a_i,\ \alpha_i = b_i - a_i\beta_{i-1},\ i = 2, 3, \cdots n;$

$\beta_i = c_i/\alpha_i$

上述方法参见教程: [三对角矩阵的LU分解]

3. 对于五对角矩阵，可以采取大致相同的思路，是不过稍微复杂一点:

   LU分解如下:

$$\begin{bmatrix} c_1 & d_1 & e_1 & & & & & & \\ b_2 & c_2 & d_2 & e_2 & & & & & \\ a_3 & b_3 & c_3 & d_3 & e_3 & & & & \\ & a_4 & b_4 & c_4 & d_4 & e_4 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & a_{n-2} & b_{n-2} & c_{n-2} & d_{n-2} & e_{n-2} \\ & & & & & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\ & & & & & & a_n & b_n & c_n \end{bmatrix} = \tag{8}$$

$$A = \begin{bmatrix} c_1 & d_1 & e_1 & & & & & & \\ b_2 & c_2 & d_2 & e_2 & & & & & \\ a_3 & b_3 & c_3 & d_3 & e_3 & & & & \\ & a_4 & b_4 & c_4 & d_4 & e_4 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & a_{n-2} & b_{n-2} & c_{n-2} & d_{n-2} & e_{n-2} \\ & & & & & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\ & & & & & & a_n & b_n & c_n \end{bmatrix}$$

$$L = \begin{bmatrix} \alpha_1 & & & & & \\ \gamma_2 & \alpha_2 & & & & \\ z_3 & \gamma_3 & \alpha_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & z_{n-1} & \gamma_{n-1} & \alpha_{n-1} & \\ & & & z_n & \gamma_n & \alpha_n \end{bmatrix},$$

$$
U=\begin{bmatrix}
1 & \beta_1 & q_1 & & & \\
 & 1 & \beta_2 & q_2 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & 1 & \beta_{n-2} & q_{n-2} \\
 & & & & 1 & \beta_{n-1} \\
 & & & & & 1
\end{bmatrix},
$$

则矩阵L和矩阵U的计算公式为：

$$
\begin{cases}
(1)\alpha_1 = c_1 \\
(2)\beta_1 = d_1/d_2 \\
(3)\gamma_2 = b_2 \\
(4)\alpha_2 = c_2 - \gamma_2\beta_1 \\
(5)z_i = \alpha_i, & i = 3,4,...,n \\
(6)\gamma_i = b_i - z_i\beta_{i-2}, & i = 3,4,...,n \\
(7)\alpha_i = c_i - z_i q_{i-2} - \gamma_i\beta_{i-1}, & i = 3,4,...,n \\
(8)\beta_i = (d_i - \gamma_i q_{i-1})/\alpha_i, & i = 2,3,...,n-1 \\
(9)q_i = e_i/\alpha_i, & i = 1,2,...,n-2
\end{cases}
$$

~~按(1)==>(9)的顺序依次进行计算~~

2.4 矩阵X、Y的计算公式

$$
\text{矩阵}Y:\begin{cases}
y_1 = f_1/\alpha_1 \\
y_2 = (f_2 - \gamma_2 y_1)/\alpha_2 \\
y_i = (f_i - z_i y_{i-2} - \gamma_i y_{i-1})/\alpha_i, & i = 3,4,...,n
\end{cases}
$$

$$
\text{矩阵}X:\begin{cases}
x_n = y_n \\
x_{n-1} = y_{n-1} - \beta_{n-1}x_n \\
x_i = y_i - q_i x_{i+2} - \beta_i x_{i+1}, & i = n-2,n-3,...,1
\end{cases}
$$

上述的方法参见教程：[五对角追赶发求解线性方程组]。

【代码（C）】

```c
#include <stdio.h>
#include <stdlib.h>
// #include <string.h>

int main()
{
    int p, n, m;
    scanf("%d%d%d", &p, &n, &m);
    double z[n]; // z为m个列向量，每个列向量n维,所有列向量共用一个数组
```

```c
    if (p == 3)
    {
        // 三对角矩阵
        double c[n - 1], b[n], a[n - 1];
        for (int i = 0; i < n - 1; i++)
        {
            scanf("%lf", &c[i]);
        }
        for (int i = 0; i < n; i++)
        {
            scanf("%lf", &b[i]);
        }
        for (int i = 0; i < n - 1; i++)
        {
            scanf("%lf", &a[i]);
        }
        // 首先数组a不变，替换数组b和c，使之成为A的LU分解
        // b[0]=b[0];
        c[0] = c[0] / b[0];
        for (int i = 1; i < n - 1; i++)
        {
            b[i] = b[i] - a[i - 1] * c[i - 1];
            c[i] = c[i] / b[i];
        }
        b[n - 1] = b[n - 1] - a[n - 2] * c[n - 2];
        // 每读入一个z的列向量，就计算一次，输出x的列向量

        for (int i = 0; i < m; i++)
        {
            for (int j = 0; j < n; j++)
            {
                scanf("%lf", &z[j]);
            }
            // 操作时x，y共用一个数组，节省内存
            // 求解Ly=z
            z[0] = z[0] / b[0];
            for (int j = 1; j < n; j++)
            {
                z[j] = (z[j] - a[j - 1] * z[j - 1]) / b[j];
            }
            // 求解Ux=y
            // z[n-1]=z[n-1];
            for (int j = n - 2; j >= 0; j--)
            {
                z[j] = z[j] - c[j] * z[j + 1];
            }
            for (int j = 0; j < n; j++)
            {
                printf("%.4lf ", z[j]);
            }
            printf("\n");
        }
    }
    else if (p == 5)
    {
        // 五对角矩阵
        double e[n - 2], d[n - 1], c[n], b[n - 1], a[n - 2];
        for (int i = 0; i < n - 2; i++)
        {
```

```
            scanf("%lf", &e[i]);
        }
        for (int i = 0; i < n - 1; i++)
        {
            scanf("%lf", &d[i]);
        }
        for (int i = 0; i < n; i++)
        {
            scanf("%lf", &c[i]);
        }
        for (int i = 0; i < n - 1; i++)
        {
            scanf("%lf", &b[i]);
        }
        for (int i = 0; i < n - 2; i++)
        {
            scanf("%lf", &a[i]);
        }
        // 进行LU分解
        // b[0]=b[0]
        // c[0]=c[0];
        d[0] = d[0] / c[0];
        e[0] = e[0] / c[0];
        b[1] = b[1] - a[0] * d[0];
        c[1] = c[1] - b[0] * d[0];
        d[1] = (d[1] - b[0] * e[0]) / c[1];
        e[1] = e[1] / c[1];
        for (int i = 2; i < n - 2; i++)
        {
            b[i] = b[i] - a[i - 1] * d[i - 1];
            c[i] = c[i] - a[i - 2] * e[i - 2] - b[i - 1] * d[i - 1];
            d[i] = (d[i] - b[i - 1] * e[i - 1]) / c[i];
            e[i] = e[i] / c[i];
        }
        b[n - 2] = b[n - 2] - a[n - 3] * d[n - 3];
        c[n - 2] = c[n - 2] - a[n - 4] * e[n - 4] - b[n - 3] * d[n - 3];
        d[n - 2] = (d[n - 2] - b[n - 3] * e[n - 3]) / c[n - 2];
        c[n - 1] = c[n - 1] - a[n - 3] * e[n - 3] - b[n - 2] * d[n - 2];

        for (int i = 0; i < m; i++)
        {
            for (int j = 0; j < n; j++)
            {
                scanf("%lf", &z[j]);
            }
            // 主体操作部分
            // 求解Ly=z
            z[0] = z[0] / c[0];
            z[1] = (z[1] - b[0] * z[0]) / c[1];
            for (int j = 2; j < n; j++)
            {
                z[j] = (z[j] - b[j - 1] * z[j - 1] - a[j - 2] * z[j - 2]) / c[j];
            }
            // 求解Ux=y
            // z[n-1]=z[n-1];
            z[n - 2] = z[n - 2] - d[n - 2] * z[n - 1];
            for (int j = n - 3; j >= 0; j--)
            {
                z[j] = z[j] - d[j] * z[j + 1] - e[j] * z[j + 2];
```

```
            }
            for (int j = 0; j < n; j++)
            {
                printf("%.4lf ", z[j]);
            }
            printf("\n");
        }
    }
    return 0;
}
```

最后附上作者的通过图片：

提交详情（带限矩阵方程组求解）

提交者: 2322022010597　　创建时间: 2023-12-12 19:42:23

运行结果　　**分数** 100.00

#	状态	时间	内存
1	Accepted	0 ms	788 KB
2	Accepted	0 ms	760 KB
3	Accepted	0 ms	760 KB
4	Accepted	0 ms	764 KB
5	Accepted	40 ms	792 KB
6	Accepted	84 ms	1068 KB
7	Accepted	416 ms	916 KB
8	Accepted	204 ms	788 KB
9	Accepted	168 ms	820 KB
10	Accepted	444 ms	1224 KB