

由于**要求 2**的代码中我使用了 **rand()函数**生成随机矩阵，故每次运行得到的 A_{rand}、b_{rand} 和 x_{rand} 矩阵均不同，但都满足 $Ax = b$ 。

列主元消元法

.m 代码见 lab4_1.m,

对于**要求 1**，选取

$$A = \begin{bmatrix} 31 & -13 & 0 & 0 & 0 & -10 & 0 & 0 & 0 \\ -13 & 35 & -9 & 0 & -11 & 0 & 0 & 0 & 0 \\ 0 & -9 & 31 & -10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 79 & -30 & 0 & 0 & 0 & -9 \\ 0 & 0 & 0 & -30 & 57 & -7 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & -7 & 47 & -30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30 & 41 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 27 & -2 \\ 0 & 0 & 0 & -9 & 0 & 0 & 0 & -2 & 29 \end{bmatrix}$$

$$b = \begin{bmatrix} -15 \\ 27 \\ -23 \\ 0 \\ -20 \\ 12 \\ -7 \\ 7 \\ 10 \end{bmatrix}$$

输出 $Ax = b$ 的根值 x^* 可得

$$x^* = \begin{bmatrix} -0.5369 \\ 0.2446 \\ -0.7438 \\ -0.2258 \\ -0.4333 \\ 0.1535 \\ -0.0584 \\ 0.2004 \\ 0.2886 \end{bmatrix}$$

对于**要求 2**，随机生成 n 阶 ($n \geq 20$ ，具体值自定) 方阵 A ， $n \times 1$ 维非零向量

b 。设定 $n=20$ ，通过以下代码生成随机整数 n 阶方阵 A ， $n \times 1$ 维非零向量 b ：

```
syms xrand
n = 20;
```

```

min = -20;    % 产生随机数的最小值
max = 20;    % 产生随机数的最大值
Arand = round(min+(max-min)*rand(n,n));
brand = round(min+(max-min)*rand(n,1));
[xrand] = Column_Pivot_Elimination(Arand,brand);

```

运行代码，可得 Arand 方阵与 brand 矩阵分别为：

```

命令窗口
>> Arand

Arand =

    18    -6   -11    -8     9     0     9     3    17   -9   -19    13    -4   -9   -15    -7    13     9   -16   -19
     7     9    -9   -13     5     8    14     7    -4     2     2    14    -2     9    13    -8    -4    -6    -4    -3
    20    -6     7    -6    -6    19    -9    -6   -18    19    -8    -5   -10    -9    -7   -20     5     1     4    13
    11    15    -1   -12    17    -7     9     5    -6     9    18     4    11    16   -10     2    13     2    -2     5
    -7    -2     5     0   -15    14   -14    12     9    14    19    15    15    13    -6   -16    15   -14   -18     1
     6    -3   -11    16     9    10    13   -19    12    -3    -9    17    17    -4    -5   -14    17     2   -11    15
   -10   -11   -13     5     6    18   -14   -17     2    -1    12     7     2     0     2     5   -12     8    13   -16
    -8   -15    13   -16    13   -19     4    19     7     2    16   -12     4     8     2    14   -10    -3   -19    16
     7    -8    11    -4    -4    -6    -5     6    16    -9     4     6   -14    13    -4    19    16    13    15   -16
     1     9    17   -18    10     7    12   -11   -18    10    15   -17    16     4    -4     3     4     9   -17     1
    -4    11   -16     0    13    -9     0    -4    -8     0    18    -4    -2     3     1    20     0    -6     7   -14
     4     8   -13    -3    -7   -11     0   -15   -18     6     2     7   -12    -7     6     2     5    -2     0     2
    10   -20   -16    20     2     8    15    -9   -12    -8     9    17    16    -2    18     1    13    -5   -11   -20
     3    14     0    12    19     5    -6   -10     9   -14     3    12    11     9     9    -7     1    11     3    11
     2    17   -12    -1     2     4    -2    -7     9    -1   -19    -1    15    15    -4    -3   -12     9   -15    14
     3    11    16    16    -7     6    19   -14    15    -6    -2    10    -9     9    13     0    -2    -3     7    17
     0   -18   -16   -14     5   -18   -18    -6     3    12     6    -3     7   -19   -15   -17    -3     8     4    19
   -17    -5   -18    -4    -6    -6    19   -15   -17    11     1    19     7     7   -18    16    19    18   -18     0
     9     8     2    17    10    -2   -12    15    17     7    -5    20   -15    -2   -17   -17     5    11   -18    -9
    20     9    11    17    -3   -10     7   -16    12   -15    17    15    -4    -2   -13    -3     8     8   -14   -16

```

```

命令窗口
>> brand

brand =

     0
     3
    11
   -17
     6
     1
   -13
    18
     4
    -2
    18
     6
    -2
    14
     1
     2
     7
    -5
   -10
     3

```

最终计算出根值xrand*:

```

命令窗口
>> xrand

xrand =

   -5.6938
   -3.9020
   -7.8247
    1.3693
    0.6485
   -0.3058
    3.2665
   -2.4068
   -0.6349
   -8.0480
    4.3011
  -13.2404
  -11.7195
   12.5598
   -0.1539
  -13.8674
    5.5820
    1.6214
   -2.7892
   -0.1656

```

LU 分解法

.m 代码见 lab4_2.m

对于要求 1, 选取

$$A = \begin{bmatrix} 30 & 33 & -43 & -11 & -38 & -29 & 37 & 28 & 23 \\ -480 & -523 & 644 & 128 & 621 & 480 & -618 & -489 & -329 \\ 60 & 266 & -1862 & -1991 & 464 & 546 & -968 & -1567 & 1652 \\ 540 & 624 & -782 & 290 & -893 & 123 & 567 & 5 & -122 \\ -450 & -675 & 2245 & 2326 & -1512 & 1230 & -822 & 129 & -189 \\ -300 & -120 & -1114 & -1295 & 1946 & 302 & -376 & -1540 & -609 \\ 1080 & 998 & 508 & 2460 & -1628 & -1358 & 2896 & 2828 & -2002 \\ -1080 & -1408 & 3340 & 2267 & 21 & -1202 & 866 & -2690 & -1351 \\ -300 & -435 & 1594 & 1685 & 340 & 2279 & -27 & 2917 & -2336 \end{bmatrix}$$

对矩阵 A 进行 LU 分解后，可得 L 矩阵和 U 矩阵：

```
命令行窗口
>> L

L =

    1.0000         0         0         0         0         0         0         0
   -16.0000     1.0000         0         0         0         0         0         0
    2.0000    40.0000     1.0000         0         0         0         0         0
   18.0000     6.0000    -0.1667     1.0000         0         0         0         0
  -15.0000   -36.0000    -0.9020     0.5064     1.0000         0         0         0
 -10.0000    42.0000     0.6569     0.8377    -0.8569     1.0000         0         0
   36.0000   -38.0000    -1.4020     1.2546    -0.4255    -0.0824     1.0000         0
  -36.0000   -44.0000    -0.9020    -0.3714     0.5938    11.1449   -82.7594     1.0000         0
 -10.0000   -21.0000    -0.8088     0.6868    -0.3057   -12.5006    34.6013   -0.3711     1.0000

fx >>
```

输出 $Ax = b$ 的根值 x^* 可得

$$x^* = (1.0e + 05) * \begin{bmatrix} -3.5200 \\ 3.4565 \\ 0.4425 \\ -0.1662 \\ -0.1403 \\ -0.0444 \\ 0.0290 \\ 0.0483 \\ -0.0134 \end{bmatrix}$$

对于要求 2，随机生成 n 阶 ($n \geq 20$ ，具体值自定) 方阵 A ， $n \times 1$ 维非零向量

b 。设定 $n=20$ ，通过以下代码生成随机整数 n 阶方阵 A ， $n \times 1$ 维非零向量 b ：

```
syms xrand
n = 20;
min = -1000; % 产生随机数的最小值
max = 1000; % 产生随机数的最大值
Arand = round(min+(max-min)*rand(n,n));
brand = round(min+(max-min)*rand(n,1));
[xrand] = Column_Pivot_Elimination(Arand,brand);
```

运行代码，可得 Arand 方阵为：

lab-
实

命令窗口

>> Arand

Arand =

列 1 至 15

308	-956	533	-50	725	492	707	45	622	-146	-791	-30	956	-337	770
315	617	503	611	793	-765	-204	-205	-723	911	-60	676	-813	-93	680
-678	-642	-722	62	-622	18	-769	-42	764	448	-562	-718	323	475	-764
-135	-669	-301	-545	321	-662	-839	988	847	162	845	464	206	20	-179
10	-637	-697	419	882	662	-279	209	-974	81	-359	382	-52	-235	-760
-249	383	-7	-703	951	856	658	890	-246	411	715	-931	-287	811	144
-39	-572	617	316	-784	-561	-571	-19	-664	-990	-480	-22	-49	931	899
-315	-404	266	268	-642	767	582	-124	80	565	756	943	342	237	-487
554	537	377	-541	493	-224	309	545	-797	854	-623	-775	919	-736	980
-232	2	279	-636	-901	-235	-948	488	-921	-983	518	486	-822	237	-300
423	819	459	-667	-857	-457	572	-114	866	649	-937	277	595	-234	-583
-38	-884	720	-701	-22	736	845	-894	943	535	285	188	182	982	332
458	-126	254	-595	700	483	-15	-824	-278	994	134	-3	824	-426	947
875	145	-639	910	994	-104	668	596	288	-545	-247	136	-798	412	245
35	130	147	-968	-991	419	-737	311	-864	839	-575	-147	-413	70	-873
806	648	-673	915	85	889	520	-935	-584	284	584	-848	-897	-614	-253
-564	-748	812	-949	723	-652	851	114	-921	-789	-709	-419	8	379	-667
746	-400	-845	942	818	-511	665	440	-61	-464	-22	123	537	-899	-537
-835	-996	-323	-405	691	282	-481	-779	-700	528	-974	267	-434	-631	-896
-69	902	161	50	758	617	-574	-567	983	611	-627	862	-549	-909	804

详细...

选择文件以...

< > 页

UTF-8

脚本

行

列 16 至 20

587	151	796	765	-749
-254	685	-6	-960	-271
664	-1	543	-516	352
508	-122	-879	532	-248
244	-702	-475	-314	727
-212	-943	302	238	-416
-281	513	-733	-94	-733
-822	592	277	-980	345
-317	-413	-230	198	-595
97	-770	531	203	737
-79	-250	306	299	502
291	658	-237	-315	-161
27	684	-400	-13	-1000
629	330	-320	404	-701
-806	920	838	776	-452
-73	886	-67	-990	745
180	-775	-115	-803	203
-626	297	-92	300	-358
223	-38	891	528	-431
-896	-867	-562	976	-129

详细...

选择文件以...

< > 页 >

UTF-8

脚本

行

brand 矩阵为:



对矩阵 A_{rand} 进行 LU 分解后, 可得 L_{rand} 矩阵和 U_{rand} 矩阵:

[illegible]

最终计算出根值 $xrand^*$:

```
>> xrand

xrand =

    -0.8012
     1.2560
     0.4547
     0.0894
     1.3081
     0.4162
    -1.0854
     0.3274
    -0.3165
    -0.7803
    -0.1985
     0.2546
     1.2111
     0.2870
     0.1895
     0.8163
     1.6621
     0.1375
     0.3232
     0.9921

> fx >>
```