Gauss 消去法

In [15]:

```
a = [
    [2, -1, 0, 0, 6],
    [-1, 3, -2, 0, 1],
    [0, -1, 2, -1, 0],
    [0, 0, -3, 5, 1]
]


def ZhuiGan(a):
    m = len(a)
    n = m + 1
    for i in range(m - 1):
        t = a[i][i] * 1.0
        for j in range(n):
            a[i][j] /= t
        t = a[i + 1][i] * 1.0
        for j in range(n):
            a[i + 1][j] -= a[i][j] * t
    return


def Showa(a):
    m = len(a)
    n = len(a[0])
    print '----------------------------------'
    for i in range(m):
        for j in range(n):
            print a[i][j],
        print
    print '----------------------------------'
    return


# a 为方程的增广矩阵
def Gauss(a):
    print '方程的增广矩阵'
    Showa(a)

    m = len(a)
    n = m + 1

    # 消元过程
    # 将 a 化为单位上三角矩阵
    for i in range(m):
        # 将该行的对角元系数化为1
        t = a[i][i] * 1.0
        if t != 0.0:
            for j in range(n):
                a[i][j] /= t
        # 利用该行消去对角元下方的元
        for k in range(i + 1, m):
            t = a[k][i] * 1.0
            if t != 0.0:
                for j in range(n):
                    a[k][j] -= a[i][j] * t
    print '消元结果'
    Showa(a)

    # 回代过程
```

```
    for i in range(m - 1, 0, -1):
        for k in range(i - 1, -1, -1):
            t = a[k][i] * 1.0
            if t != 0.0:
                for j in range(i, n):
                    a[k][j] -= a[i][j] * t
    print '回代结果'
    Showa(a)

    #得出解向量
    x = []
    for i in range(m):
        x.append(a[i][n - 1])
    print '解向量'
    print x
    return


Gauss(a)
```

```
方程的增广矩阵
------------------------------
2 -1 0 0 6
-1 3 -2 0 1
0 -1 2 -1 0
0 0 -3 5 1
------------------------------
消元结果
------------------------------
1.0 -0.5 0.0 0.0 3.0
0.0 1.0 -0.8 0.0 1.6
0.0 0.0 1.0 -0.833333333333 1.33333333333
0.0 0.0 0.0 1.0 2.0
------------------------------
回代结果
------------------------------
1.0 0.0 0.0 0.0 5.0
0.0 1.0 0.0 0.0 4.0
0.0 0.0 1.0 0.0 3.0
0.0 0.0 0.0 1.0 2.0
------------------------------
解向量
[5.0, 4.0, 3.0, 2.0]
```

P185 9

In [16]:

```
a = [
    [4, -2, 4, 8.7],
    [-2, 17, 10, 13.7],
    [4, 10, 9, -0.7]
]

Gauss(a)
```

```
方程的增广矩阵
--------------------------------
4 -2 4 8.7
-2 17 10 13.7
4 10 9 -0.7
--------------------------------
消元结果
--------------------------------
1.0 -0.5 1.0 2.175
0.0 1.0 0.75 1.128125
-0.0 -0.0 1.0 5.734375
--------------------------------
回代结果
--------------------------------
1.0 0.0 0.0 -5.145703125
0.0 1.0 0.0 -3.17265625
-0.0 -0.0 1.0 5.734375
--------------------------------
解向量
[-5.145703124999999, -3.1726562499999993, 5.734374999999999]
```

P185 10

In [17]:

```
a = [
    [5, 7, 9, 10, 1],
    [6, 8, 10, 9, 1],
    [7, 10, 8, 7, 1],
    [5, 7, 6, 5, 1]
]

Gauss(a)
```

```
方程的增广矩阵
--------------------------------
5 7 9 10 1
6 8 10 9 1
7 10 8 7 1
5 7 6 5 1
--------------------------------
消元结果
--------------------------------
1.0 1.4 1.8 2.0 0.2
-0.0 1.0 2.0 7.5 0.5
-0.0 -0.0 1.0 1.7 0.1
0.0 0.0 0.0 1.0 3.0
--------------------------------
回代结果
--------------------------------
1.0 0.0 0.0 0.0 20.0
-0.0 1.0 0.0 0.0 -12.0
-0.0 -0.0 1.0 0.0 -5.0
0.0 0.0 0.0 1.0 3.0
--------------------------------
解向量
[19.999999999999055, -11.99999999999943, -4.999999999999758, 2.9999999999998552]
```

Doolittle 分解法

In [25]:

```
a = [
    [2, -1, 0, 0],
    [-1, 3, -2, 0],
    [0, -1, 2, -1],
    [0, 0, -3, 5]
]
b = [6, 1, 0, 1]


# a 为方程的系数矩阵
def Doolittle(a, b):
    print '方程的系数矩阵'
    Showa(a)

    n = len(a)

    # 将 a 分解为 l 和 u
    # 初始化 l 和 u
    l = []
    u = []
    for i in range(n):
        l.append([])
        u.append([])
        for j in range(n):
            l[i].append(0.0)
            u[i].append(0.0)
    Showa(l)
    Showa(u)
    # 计算 l 和 u
    for i in range(n):
        for j in range(i):
            l[i][j] = a[i][i] * 1.0
            for k in range(j):
                l[i][j] -= l[i][k] * u[k][j]
            if u[j][j] != 0.0:
                l[i][j] /= u[j][j]
    for i in range(n):
        for j in range(i, n):
            u[i][j] = a[i][j] * 1.0
            for k in range(i):
                u[i][j] -= l[i][k] * u[k][j]
    Showa(l)
    Showa(u)
    return


Doolittle(a, b)
```

```
方程的系数矩阵
------------------------------
2 -1 0 0
-1 3 -2 0
0 -1 2 -1
0 0 -3 5
------------------------------
------------------------------
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
```

```
0.0 0.0 0.0 0.0
--------------------------------
--------------------------------
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
--------------------------------
--------------------------------
0.0 0.0 0.0 0.0
3.0 0.0 0.0 0.0
2.0 2.0 0.0 0.0
5.0 5.0 5.0 0.0
--------------------------------
--------------------------------
2.0 -1.0 0.0 0.0
0.0 6.0 -2.0 0.0
0.0 0.0 6.0 -1.0
0.0 0.0 0.0 10.0
--------------------------------
```

In [ ]:

```
1
```