Question

1. 分别用Givens和Householder变换写出幺正矩阵U使得$U\begin{bmatrix}1\\1\end{bmatrix}=\begin{bmatrix}\sqrt{2}\\0\end{bmatrix}$，并将你得到的Givens矩阵写为Householder矩阵的乘积。
2. 编写程序：
   随机生成$\mathbb{C}^n$中非零向量$\boldsymbol{\xi},\boldsymbol{\eta}$，要求$\|\boldsymbol{\xi}\|=\|\boldsymbol{\eta}\|$，分别用Givens变换和Householder变换计算幺正矩阵U,使$U\boldsymbol{\xi}=\boldsymbol{\eta}$。
   要求：
   - n 是函数参数,可以是任意正整数;
   - 非零向量$\boldsymbol{\xi},\boldsymbol{\eta}$作为函数参数;
   - 确认矩阵U是幺正矩阵;
   - 用函数实现, 例如:

     ```
     int Givens(int n, Complex* xi, Complex* yita, Complex* U)
       {
       }
     ```

     这里Complex也可以写为complex<double>。
   - 随机产生3组n, $\boldsymbol{\xi},\boldsymbol{\eta}$，调用函数计算相应的矩阵$U$，并验算你的结果。
   - 写文档详细介绍你的算法以及运行结果;
   - 如果取实数域中的向量,此题最多给9分;
   - 要求程序在linux下面可以运行;
   - 自作业发布之日起两周内交作业.

Question1

Note:题目中要求的是幺正矩阵，但由于涉及到的向量均属于$\mathbb{R}^2$空间（$\mathbb{C}^2$的子空间），所以实际要求的是正交矩阵。（所有的正交矩阵都是幺正的）

1. Givens矩阵

   Givens矩阵U可以表示为：$U=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}$，则$\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}=\begin{bmatrix}\cos\theta+\sin\theta\\ -\sin\theta+\cos\theta\end{bmatrix}=\begin{bmatrix}\sqrt{2}\\0\end{bmatrix}$，可以解得$\theta=\pi/4$.

综上，Givens矩阵$U=\frac{\sqrt{2}}{2}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}$.

2. Householder矩阵

   记$\boldsymbol{a}=\begin{bmatrix}1\\1\end{bmatrix}, \boldsymbol{b}=\begin{bmatrix}\sqrt{2}\\0\end{bmatrix}$，则$\mathbf{H}=\mathbf{I}-2\boldsymbol{\omega}\boldsymbol{\omega}^\intercal$, 其中$\boldsymbol{\omega}=\frac{\mathbf{a-b}}{\|\mathbf{a-b}\|}$.

   综上，Householder矩阵$U=\frac{\sqrt{2}}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$.

Question2

理论基础

Givens 变换

- 引理：有$x=(\xi_1,\xi_2,...,\xi_n)^\intercal\in\mathbb{C}^n$, 当$|\xi_i|^2+|\xi_k|^2\neq 0$时，令$c=\frac{|\xi_i|}{\sqrt{|\xi_i|^2+|\xi_k|^2}}, s=\frac{|\xi_k|}{\sqrt{|\xi_i|^2+|\xi_k|^2}}, \theta_1=-arg\xi_i, \theta_2=-arg\xi_k, \mathbf{y}=\mathbf{U_{ik}x}=(\eta_1,\eta_2,...,\eta_n)^\intercal$, 其中$\begin{cases}\eta_i=\xi_i ce^{j\theta_1}+\xi_k se^{j\theta_2}\\ \eta_k=-\xi_i se^{-j\theta_2}+\xi_k ce^{-j\theta_1}\\ \eta_t=\xi_t(t\neq i,k)\end{cases}$，则有$\eta_i=\sqrt{|\xi_i|^2+|\xi_k|^2}>0, \eta_k=0$。

  这个引理很好证明，直接带进去验算即可，这里就不展开去算了。

对于$\mathbb{C}^n$中的向量$\boldsymbol{\xi},\boldsymbol{\eta}$, 如果$\eta_n\neq 0$,那么我们可以得到线性无关集$\{\boldsymbol{\eta},\mathbf{e_2},\mathbf{e_3},...,\mathbf{e_n}\}$。使用上一次作业中施密特正交化可以得到一个坐标变换矩阵$T=[\boldsymbol{\mu_1}\quad\boldsymbol{\mu_2}\quad\dots\quad\boldsymbol{\mu_n}]$, 其中$\boldsymbol{\mu_1}=\boldsymbol{\eta}/|\boldsymbol{\eta}|$.

非常重要的一点是T是么正矩阵，即$T^\dagger T=TT^\dagger=I$.

这样就有$\boldsymbol{\xi'}=T^\dagger\boldsymbol{\xi}, \boldsymbol{\eta'}=T^\dagger\boldsymbol{\eta}=(|\boldsymbol{\eta}|,0,0,...,0)^\intercal$，于是我们可以通过引理，进行最多n-1次（如果$\boldsymbol{\eta}$含有0分量的话，那一次就不用转了）的Givens旋转使$\boldsymbol{\xi'}$变成$\boldsymbol{\eta'}$。即$U'\boldsymbol{\xi'}=\boldsymbol{\eta'}$.

通过坐标变换可以得到$TU'T^\dagger T\boldsymbol{\xi'}=T\boldsymbol{\eta'}$, 即$TU'T^\dagger\boldsymbol{\xi}=\boldsymbol{\eta}$;这样我们就得到了$U=TU'T^\dagger$.

Householder 变换

非零向量$\boldsymbol{\xi},\boldsymbol{\eta}$满足$\boldsymbol{\xi}\cdot\boldsymbol{\xi}=\boldsymbol{\eta}\cdot\boldsymbol{\eta}$, 如果$\boldsymbol{\eta}=e^{i\theta}\boldsymbol{\xi}$, 令$U=e^{i\theta}I$, 则$U\boldsymbol{\xi}=\boldsymbol{\eta}$.

否则，令$\boldsymbol{\xi}^\dagger\boldsymbol{\eta}=e^{i\theta}|\boldsymbol{\xi}^\dagger\boldsymbol{\eta}|$, 定义$\boldsymbol{\omega}=\frac{e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta}}{|e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta}|}$.

则$e^{i\theta}(I-2\boldsymbol{\omega}\boldsymbol{\omega}^\dagger)\boldsymbol{\xi}=e^{i\theta}\boldsymbol{\xi}-(e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta})\frac{2e^{i\theta}(e^{-i\theta}\boldsymbol{\xi}^\dagger-\boldsymbol{\eta}^\dagger)\boldsymbol{\xi}}{(e^{-i\theta}\boldsymbol{\xi}^\dagger-\boldsymbol{\eta}^\dagger)(e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta})}=e^{i\theta}\boldsymbol{\xi}-(e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta})\frac{2\boldsymbol{\xi}^\dagger\boldsymbol{\xi}-2e^{i\theta}\boldsymbol{\eta}^\dagger\boldsymbol{\xi}}{2\boldsymbol{\xi}^\dagger\boldsymbol{\xi}-e^{-i\theta}\boldsymbol{\xi}^\dagger\boldsymbol{\eta}-e^{i\theta}\boldsymbol{\eta}^\dagger\boldsymbol{\xi}}=\boldsymbol{\eta}$。
综上$U=e^{i\theta}(I-\boldsymbol{\omega}\boldsymbol{\omega}^\dagger)$,其中$\boldsymbol{\xi}^\dagger\boldsymbol{\eta}=e^{i\theta}|\boldsymbol{\xi}^\dagger\boldsymbol{\eta}|,\boldsymbol{\omega}=\frac{e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta}}{|e^{i\theta}\boldsymbol{\xi}-\boldsymbol{\eta}|}$.

orthogonalization.h

我们以一个n*n大小的一维数组存储基，将其视为二维的;每一列代表一个基。

首先，我们要把这个数组第一列归一化，需要引入中间变量以存储向量的二范数.double 类型就可以。

```
for(int i = 0; i < n; i++){
    length += conj(T[i*n])*T(i*n); //Don't forget to re-init length
}
for(int i = 0; i < n; i++){
    T[i*n] = T[i*n]/length;
}
```

下面进行其他基的正交化。指标将从1开始，到n截止。

我们需要一个中间变量来存储中间向量：$\mathbf{t_i}-\sum_{j=0}^{i-1}\frac{\mathbf{t_j^\dagger}\cdot\mathbf{t_i}}{\mathbf{t_j^\dagger}\cdot\mathbf{t_j}}\mathbf{t_j}$.
如果前面已经归一化了，那么分母为1,可以省略。

Givens.h

我们首先需要生成坐标变换矩阵，即以$\boldsymbol{\eta}$的单位向量为第一列，其他是单位向量，由于$\boldsymbol{\eta}$不含零元，故他们一定线性无关。虽然和原理的构造有些出入，但原理没有任何问题。这个矩阵还不是坐标变换矩阵，还需要对其进行幺正化，及施密特正交化。我们用“orthogonalization.h”来实现。

下面需要调用检验幺正矩阵的函数;该矩阵与其dagger的乘积为单位阵就可以检验;

```
//is unitary?
void m_isUnitary(double complex *A, int n){
    double complex *U;
    U = (double complex *)malloc(n*n*sizeof(double complex));
    for(int i = 0; i < n*n; i++){
        U[i] = 0+0*I;
    }
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            for(int k = 0; k < n; k++){
                U[i*n+j] += conj(A[k*n+i])*A[k*n+j];
            }
        }
    }
    m_cprint(U, n, n);
    free(U);
}
```

然后，我们需要对$\boldsymbol{\xi}, \boldsymbol{\eta}$进行坐标变换，直接右乘变换矩阵(T的dagger)。（定义个矩阵乘法是不错的）

我们需要n-1次Givens旋转，从最后开始，将$\boldsymbol{\xi}'$变成$\{|\boldsymbol{\xi}'|, 0, 0, ..., 0\}$.理论上有$|\boldsymbol{\xi}'| = |\boldsymbol{\eta}'|$.

```c
//Givens 旋转
double complex c, s, z1, z2;
double theta1, theta2;
double complex *Q;
Q = (double complex *)malloc(D*D*sizeof(double complex));
for(int i = 0; i < D; i++){
    Up[i*D+i] = 1+0*I;
}
for(int i = D-1; i > 0; i--){
    c = cabs(xip[i-1])/sqrt(cabs(xip[i-1])*cabs(xip[i-1]) \
    +cabs(xip[i])*cabs(xip[i]));
    s = cabs(xip[i])/sqrt(cabs(xip[i-1])*cabs(xip[i-1])+ \
    cabs(xip[i])*cabs(xip[i]));
    theta1 = -carg(xip[i-1]);
    theta2 = -carg(xip[i]);
    z1 = cos(theta1)+sin(theta1)*I;
    z2 = cos(theta2)+sin(theta2)*I;
    for(int i = 0; i < D*D; i++){
        Q[i] = 0;
    }
    for(int i = 0; i < D; i++){
        Q[i*D+i] = 1+0*I;
    }
    Q[(i-1)*D+i-1] = c*z1;
    Q[(i-1)*D+i] = s*z2;
    Q[(i)*D+i-1] = -s*conj(z2);
    Q[(i)*D+i] = c*conj(z1);
    Up = m_mul(Q, Up, D, D, D);

    xip[i-1] = sqrt(cabs(xip[i-1])*cabs(xip[i-1])+ \
    cabs(xip[i])*cabs(xip[i]));
    xip[i] = 0;
}
U = m_mul(T, Up, D, D, D);
U = m_mul(U, Tdag, D, D, D);
```

Householder.h

这个比较简单，直接代公式就可以了。

```
double complex * m_Householder \
(double complex *xi, double complex *eta, int D){
    double complex *U;
    U = (double complex *)malloc(D*D*sizeof(double complex));
    for(int i = 0; i < D*D; i++){
        U[i] = 0+0*I;
    }

    double complex z, mod;
    double complex *omega;
    omega = (double complex *)malloc(D*sizeof(double complex));
    z = m_ipro(xi, eta, D);
    z = z/cabs(z);
    for(int i = 0; i < D; i++){
        omega[i] = z*xi[i]-eta[i];
    }
    mod = csqrt(m_ipro(omega, omega, D));
    for(int i = 0; i < D; i++){
        omega[i] = omega[i]/mod;
    }
    for(int i = 0; i < D; i++){
        for(int j = 0; j < D; j++){
            U[i*D+j] = z*((i==j)-2*omega[i]*conj(omega[j]));
        }
    }

    //check
    printf("U(Householder) is :\n");
    m_cprint(U, D, D);

    printf("Uxi is :\n");
    m_cprint(m_mul(U, xi, D, D, 1), D, 1);
    printf("well, a great success. its equ. to eta.\n\n");

    free(omega);
    free(U);
    return U;
}
```

main fuction

程序不会出现零元的情况，因为我使用的随机数生成器是（0, 1）
首先，我们需要生成两个二范数相同的位于$\mathbb{C}^n$空间的向量。可以采用类似于归一化的方法，我们将其二范数设定为一个随机的数值。
为了方便，我们一般取0-10之间的随机数。
n的生成采用宏定义的方式。

我们还需要一个打印函数，用来打印矩阵。

定义完$\boldsymbol{\xi}, \boldsymbol{\eta}$;之后，我们需要对其进行“等模处理”，这里可能引入计算机的舍入误差。目前我还没有好的办法避免。
定义两个向量的内积是方便的。

```
//inner product :c = <a, b>
double complex m_ipro(double complex *a, double complex *b, int n){
    double complex c;
    c = 0 + 0*I;

    for(int i = 0; i < n; i++){
        c += conj(a[i])*b[i];
    }

    return c;
}
```

现在就可以调用Givens、Householder变换函数了。

Results

取 n = 3,6,8;得到的结果如下：

1. n=3

```
mod_define is 9.6487
initial mod_xi is 15.1327+i0.0000, mod_eta is 15.3890+i0.0000
eta:
4.5230+i0.6455
6.1070+i3.8381
0.5979+i4.4542

final mod_xi is 9.6487+i0.0000, mod_eta is 9.6487+i0.0000
Gievens trans.
the trans. matrix(non-Unitary)
0.4688+i0.0669  1.0000+i0.0000  0.0000+i0.0000
0.6329+i0.3978  0.0000+i0.0000  1.0000+i0.0000
0.0620+i0.4616  0.0000+i0.0000  0.0000+i0.0000

mod 0.8808+i0.0000
mod 0.5288+i0.0000
the trans. matrix(Unitary).
0.4688+i0.0669  0.8808+i0.0000  0.0000+i0.0000
0.6329+i0.3978  -0.3671+i-0.1636  0.5288+i0.0000
0.0620+i0.4616  -0.0680+i-0.2410  -0.5432+i-0.6521

check the T(trans. matrix) is Unitary.
1.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000
-0.0000+i0.0000  1.0000+i0.0000  0.0000+i-0.0000
-0.0000+i-0.0000  0.0000+i0.0000  1.0000+i0.0000

xip and etap.
7.3178+i0.6835
-0.7511+i2.3962
0.0569+i5.7246

9.6487+i0.0000
-0.0000+i0.0000
-0.0000+i-0.0000

xip mod is 9.648735
U(Givens) is :
-0.1646+i-0.3170  0.0207+i-0.2463  0.6153+i0.6578
0.3909+i-0.3661  0.7190+i0.1770  0.2785+i-0.2954
```

```
-0.3930+i0.6566  0.6208+i-0.0715  0.0199+i0.1532

Uxi is :
4.5230+i0.6455
6.1070+i3.8381
0.5979+i4.4542

well, a great success. its equ. to eta.

Householder trans.
U(Householder) is :
0.7080+i-0.0661  -0.1572+i-0.0247  0.6780+i-0.0968
-0.1499+i0.0534  0.9084+i-0.0849  0.3487+i-0.1437
0.6842+i-0.0304  0.3693+i0.0767  -0.6208+i0.0580

Uxi is :
4.5230+i0.6455
6.1070+i3.8381
0.5979+i4.4542

well, a great success. its equ. to eta.
```

2. n=6

```
mod_define is 4.5932
initial mod_xi is 19.2303+i0.0000, mod_eta is 20.7809+i0.0000
eta:
1.4314+i1.1961
0.9254+i1.7430
1.0507+i1.4768
0.7484+i0.7723
1.6388+i0.0325
2.1668+i1.3786

final mod_xi is 4.5932+i0.0000, mod_eta is 4.5932+i0.0000
Gievens trans.
the trans. matrix(non-Unitary)
0.3116+i0.2604  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.2015+i0.3795  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.2288+i0.3215  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.1629+i0.1681  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.
0.3568+i0.0071  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.
0.4717+i0.3001  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.

mod 0.9138+i0.0000
mod 0.8826+i0.0000
mod 0.8721+i0.0000
mod 0.9430+i0.0000
mod 0.8429+i0.0000
the trans. matrix(Unitary).
0.3116+i0.2604  0.9138+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000  -0.000
0.2015+i0.3795  -0.1768+i-0.0720  0.8826+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i
0.2288+i0.3215  -0.1696+i-0.0445  -0.2281+i0.0299  0.8721+i0.0000  0.0000+i0.0000  0.0000+
0.1629+i0.1681  -0.1035+i-0.0109  -0.1311+i0.0379  -0.1610+i0.0245  0.9430+i0.0000  -0.000
0.3568+i0.0071  -0.1237+i0.0993  -0.1012+i0.1818  -0.1479+i0.1994  -0.1271+i0.1261  0.8429
0.4717+i0.3001  -0.2464+i0.0321  -0.2835+i0.1608  -0.3603+i0.1463  -0.2729+i0.0652  -0.459

check the T(trans. matrix) is Unitary.
1.0000+i0.0000  0.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  0.0000+i-0.0000  0.0000+
0.0000+i-0.0000  1.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.000
0.0000+i0.0000  -0.0000+i-0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i
0.0000+i0.0000  -0.0000+i-0.0000  0.0000+i-0.0000  1.0000+i0.0000  0.0000+i-0.0000  0.000
0.0000+i0.0000  -0.0000+i-0.0000  0.0000+i-0.0000  0.0000+i0.0000  1.0000+i0.0000  -0.0000
```

```
0.0000+i-0.0000  -0.0000+i0.0000  0.0000+i-0.0000  0.0000+i0.0000  -0.0000+i0.0000  1.0000

xip and etap.
3.8323+i0.5632
0.2314+i-0.9559
-0.1180+i0.2288
-0.4668+i0.9798
0.6104+i1.7331
-0.4617+i0.5416

4.5932+i0.0000
-0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i-0.0000

xip mod is 4.593195
U(Givens) is :
-0.0249+i0.4583  -0.0166+i-0.0437  -0.0676+i-0.3010  0.3792+i-0.4514  -0.3442+i-0.0786  -0
-0.6030+i0.0221  0.3169+i-0.1071  0.2326+i-0.0422  0.0652+i-0.1064  0.1956+i0.2857  0.5670
0.3183+i-0.0213  -0.6086+i-0.0602  0.4441+i-0.0307  0.3565+i0.1488  0.2120+i0.1329  0.3023
0.1848+i-0.0409  0.2274+i-0.0708  -0.6504+i-0.0505  0.5065+i0.3095  0.2942+i0.0264  0.1920
0.1561+i-0.2427  0.1700+i-0.3207  0.1186+i-0.2261  0.1945+i-0.2928  0.1378+i0.5863  -0.422
0.4066+i-0.1976  0.4889+i-0.2907  0.3430+i-0.2060  -0.1024+i-0.0354  0.1162+i-0.4789  0.12

Uxi is :
1.4314+i1.1961
0.9254+i1.7430
1.0507+i1.4768
0.7484+i0.7723
1.6388+i0.0325
2.1668+i1.3786

well, a great success. its equ. to eta.

Householder trans.
U(Householder) is :
0.6327+i-0.0930  0.0070+i-0.1067  0.2027+i-0.2258  0.4767+i0.1042  -0.0018+i-0.3422  -0.36
```

```
0.0374+i0.1002  0.9580+i-0.1408  -0.0781+i-0.0448  0.0006+i-0.1447  -0.0992+i0.0218  0.014
0.2591+i0.1579  -0.0619+i0.0653  0.7366+i-0.1083  -0.2526+i-0.3239  -0.1829+i0.2226  0.220
0.4265+i-0.2369  0.0422+i0.1385  -0.1487+i0.3829  0.3360+i-0.0494  0.1660+i0.4325  0.4724+
0.0967+i0.3283  -0.1012+i0.0076  -0.2392+i-0.1605  0.0345+i-0.4620  0.6679+i-0.0981  0.022
-0.3395+i0.1306  -0.0168+i-0.1066  0.1495+i-0.2672  0.4923+i-0.0030  -0.0776+i-0.3365  0.6

Uxi is :
1.4314+i1.1961
0.9254+i1.7430
1.0507+i1.4768
0.7484+i0.7723
1.6388+i0.0325
2.1668+i1.3786

well, a great success. its equ. to eta.
```

3. n=8

```
mod_define is 8.4728
initial mod_xi is 19.6048+i0.0000, mod_eta is 24.0815+i0.0000
eta:
0.9097+i3.4596
0.2581+i1.6037
0.6119+i3.1144
1.2634+i0.3744
2.0143+i1.7647
2.7307+i2.8168
1.8058+i2.5302
0.5399+i3.4678

final mod_xi is 8.4728+i0.0000, mod_eta is 8.4728+i0.0000
Gievens trans.
the trans. matrix(non-Unitary)
0.1074+i0.4083  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.0305+i0.1893  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.0722+i0.3676  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.1491+i0.0442  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.
0.2377+i0.2083  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.
0.3223+i0.3324  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.2131+i0.2986  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.0637+i0.4093  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.

mod 0.9065+i0.0000
mod 0.9774+i0.0000
mod 0.9062+i0.0000
mod 0.9811+i0.0000
mod 0.9160+i0.0000
mod 0.7669+i0.0000
mod 0.7486+i0.0000
the trans. matrix(Unitary).
0.1074+i0.4083  0.9065+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.
0.0305+i0.1893  -0.0889+i-0.0087  0.9774+i0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000  -0.0
0.0722+i0.3676  -0.1741+i-0.0110  -0.0894+i0.0031  0.9062+i0.0000  0.0000+i0.0000  0.0000+
0.1491+i0.0442  -0.0376+i0.0619  -0.0161+i0.0335  -0.0380+i0.0726  0.9811+i0.0000  0.0000+
0.2377+i0.2083  -0.1220+i0.0824  -0.0581+i0.0481  -0.1318+i0.1017  -0.0706+i-0.0325  0.916
0.3223+i0.3324  -0.1879+i0.1058  -0.0906+i0.0633  -0.2045+i0.1328  -0.0992+i-0.0559  -0.25
0.2131+i0.2986  -0.1598+i0.0606  -0.0785+i0.0389  -0.1759+i0.0798  -0.0711+i-0.0555  -0.19
```

```
0.0637+i0.4093  -0.1919+i-0.0198  -0.0989+i-0.0005  -0.2180+i-0.0086  -0.0436+i-0.0920  -0

check the T(trans. matrix) is Unitary.
1.0000+i0.0000  0.0000+i-0.0000  0.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  0.0000+
0.0000+i0.0000  1.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000  -0.000
0.0000+i0.0000  -0.0000+i-0.0000  1.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000  -0.0
0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000  1.0000+i0.0000  -0.0000+i-0.0000  -0.00
0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  1.0000+i0.0000  -0.0000
0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  1.0000
0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000  -0.00
0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000  -0.00

xip and etap.
6.9140+i-0.9002
0.2573+i0.2244
1.8234+i-0.2790
2.2521+i-1.5158
0.0727+i0.4644
-0.2487+i-0.6831
-2.5016+i0.0884
-2.2567+i0.4204

8.4728+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000

xip mod is 8.472806
U(Givens) is :
0.1347+i0.2020  0.3355+i0.1983  0.3591+i0.4801  -0.0252+i-0.0510  -0.1276+i0.1882  -0.3983
-0.8095+i0.0026  0.1086+i0.0427  0.1882+i0.0389  0.0764+i0.0848  0.1626+i0.0916  0.2880+i0
0.2260+i-0.0027  -0.8150+i0.0600  0.2553+i0.1164  0.0731+i0.0295  0.1110+i0.0801  0.1213+i
0.0425+i-0.0837  0.0569+i-0.0549  -0.7074+i0.0203  0.0294+i-0.0248  0.1518+i0.1641  -0.089
0.1496+i-0.1183  0.1517+i-0.0531  0.0249+i0.0569  -0.8210+i0.0122  0.0949+i-0.0924  0.2488
0.2325+i-0.1549  0.2278+i-0.0595  0.0296+i0.0861  0.3495+i0.0464  -0.6136+i-0.0702  0.4462
```

```
0.2005+i-0.0938  0.1855+i-0.0196  0.0130+i0.0710  0.2683+i0.0778  0.4426+i0.0106  0.4650+i
0.2498+i0.0069  0.1971+i0.0742  -0.0229+i0.0782  0.2313+i0.2144  0.4412+i0.2350  -0.1576+i

Uxi is :
0.9097+i3.4596
0.2581+i1.6037
0.6119+i3.1144
1.2634+i0.3744
2.0143+i1.7647
2.7307+i2.8168
1.8058+i2.5302
0.5399+i3.4678

well, a great success. its equ. to eta.

Householder trans.
U(Householder) is :
0.9802+i0.1276  -0.0229+i0.0483  -0.0632+i0.0430  0.0220+i-0.0029  -0.0088+i-0.0282  0.024
-0.0098+i-0.0526  0.7463+i0.0972  -0.2794+i-0.2168  0.0439+i0.0926  0.1118+i-0.0791  0.406
-0.0501+i-0.0578  -0.3256+i0.1381  0.4903+i0.0638  0.1212+i0.0824  0.0806+i-0.1783  0.4940
0.0205+i0.0084  0.0662+i-0.0783  0.1382+i-0.0486  0.9495+i0.1236  0.0025+i0.0567  -0.0877+
-0.0158+i0.0250  0.0878+i0.1051  0.0322+i0.1930  0.0169+i-0.0541  0.9165+i0.1193  -0.1693+
0.0025+i0.0879  0.3905+i0.1142  0.3989+i0.4232  -0.0479+i-0.1616  -0.2016+i0.1001  0.3282+
0.0136+i0.0455  0.2199+i0.0051  0.2674+i0.1653  -0.0476+i-0.0777  -0.0913+i0.0804  -0.3585
0.0337+i-0.0423  -0.1380+i-0.2085  -0.0193+i-0.3569  -0.0416+i0.0948  0.1344+i0.0330  0.27

Uxi is :
0.9097+i3.4596
0.2581+i1.6037
0.6119+i3.1144
1.2634+i0.3744
2.0143+i1.7647
2.7307+i2.8168
1.8058+i2.5302
0.5399+i3.4678

well, a great success. its equ. to eta.
```

由于结果较长，无法显示所有矩阵元，可以自行编译README.md文件查看。

由于程序交叉过多，设及相对路径，建议直接将压缩包解压作为工作空间，必要的程序不好找。或者直接从**github**克隆。

github仓库地址:https://github.com/jdw-heaven/Linear_Algebra.git

至于**linux**系统上运行的话，应该和环境有关，**linux**系统**bug**比较多。不过在我的系统上运行地很好。