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$\boldsymbol{\omega} = \frac{\mathbf{a}-\mathbf{b}}{\|\mathbf{a}-\mathbf{b}\|}$.

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

3. 对于二维的情况，我们设$\mu_1 = [\cos(\alpha) \quad sin(\alpha)]^\intercal, \mu_2 = [\cos(\alpha + \theta/2) \quad sin(\alpha+\theta/2)]^\intercal$.令$H_1 = I - 2\mu_1\mu_1^\intercal, H_2 = I - 2\mu_2\mu_2^\intercal$, 则$H_1 = \begin{bmatrix} -\cos(2\alpha) & \sin(2\alpha) \\ \sin(2\alpha) & \cos(2\alpha) \end{bmatrix}, H_2 = \begin{bmatrix} -\cos(2\alpha+\theta) & \sin(2\alpha+\theta) \\ \sin(2\alpha+\theta) & \cos(2\alpha+\theta) \end{bmatrix}$, 故$H_2H_1 = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$,对应于Givens矩阵。

综上， 1中的Givens矩阵可以表示成$U = \begin{bmatrix} -\cos(2\alpha) & \sin(2\alpha) \\ \sin(2\alpha) & \cos(2\alpha) \end{bmatrix} \begin{bmatrix} -\cos(2\alpha+\pi/4) & \sin(2\alpha+\pi/4) \\ \sin(2\alpha+\pi/4) & \cos(2\alpha+\pi/4) \end{bmatrix}$ 其中$\alpha \in [0, 2\pi)$中任意实数.

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$\begin{cases} \eta_i = \xi_i c e^{j\theta_1} + \xi_k s e^{j\theta_2} \\ \eta_k = -\xi_i s e^{-j\theta_2} + \xi_k c e^{-j\theta_1} \\ \eta_t = \xi_t (t \neq i, k) \end{cases}$，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$T = [\boldsymbol{\mu_1} \quad \boldsymbol{\mu_2} \quad ... \quad \boldsymbol{\mu_n}]$, 其中$\boldsymbol{\mu_1} = \boldsymbol{\eta}/|\boldsymbol{\eta}|$.

非常重要的一点是T是幺正矩阵，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η的单位向量为第一列，其他是单位向量，由于η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ξ, η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ξ'变成$\{|\xi'|, 0, 0, ..., 0\}$.理论上有$|\xi'| = |\eta'|$.

```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。

```c
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 1.0619
initial mod_xi is 16.8701+i0.0000, mod_eta is 12.7565+i0.0000
eta:
0.5135+i0.4352
0.2058+i0.6321
0.2664+i0.4020

final mod_xi is 1.0619+i0.0000, mod_eta is 1.0619+i0.0000
Gievens trans.
the trans. matrix(non-Unitary)
0.4835+i0.4099  1.0000+i0.0000  0.0000+i0.0000
0.1938+i0.5953  0.0000+i0.0000  1.0000+i0.0000
0.2509+i0.3786  0.0000+i0.0000  0.0000+i0.0000

mod 0.7734+i0.0000
mod 0.5872+i0.0000
the trans. matrix(Unitary).
0.4835+i0.4099  0.7734+i0.0000  0.0000+i0.0000
0.1938+i0.5953  -0.4366+i-0.2694  0.5872+i0.0000
0.2509+i0.3786  -0.3575+i-0.1037  -0.7800+i0.2162

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

xip and etap.
0.8527+i-0.2276
-0.3755+i0.2973
0.0797+i-0.3360

1.0619+i0.0000
0.0000+i-0.0000
0.0000+i0.0000

xip mod is 1.061864
U(Givens) is :
-0.2989+i-0.1147  0.6471+i0.5322  0.4183+i-0.1434
0.5730+i0.4857  -0.0515+i-0.0376  0.6557+i-0.0423
```

```
0.4738+i-0.3299  0.5101+i-0.1836  -0.1012+i0.6021

isUnitary? Udag*U is :
1.0000+i0.0000  -0.0000+i-0.0000  0.0000+i-0.0000
-0.0000+i0.0000  1.0000+i0.0000  -0.0000+i-0.0000
0.0000+i0.0000  -0.0000+i0.0000  1.0000+i0.0000

Uxi is :
0.5135+i0.4352
0.2058+i0.6321
0.2664+i0.4020

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

Householder trans.
U(Householder) is :
0.0150+i0.0040  0.6388+i0.6282  0.1956+i-0.3984
0.8669+i-0.2263  0.1783+i0.0476  0.0241+i0.4032
-0.0290+i0.4429  0.2219+i-0.3375  0.7729+i0.2063

isUnitary? Udag*U is :
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

Uxi is :
0.5135+i0.4352
0.2058+i0.6321
0.2664+i0.4020

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

[1] + Done
"/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm}
0<"/tmp/Microsoft-MIEngine-In-qk4uryjh.1kf" 1>"/tmp/
Microsoft-MIEngine-Out-tkxknbdi.2cd"
```

2. n=6

```
mod_define is 0.1753
initial mod_xi is 21.6119+i0.0000, mod_eta is 18.6258+i0.0000
eta:
0.0224+i0.0877
0.0081+i0.0872
0.0043+i0.0600
0.0375+i0.0256
0.0103+i0.0722
0.0490+i0.0385

final mod_xi is 0.1753+i0.0000, mod_eta is 0.1753+i0.0000
Gievens trans.
the trans. matrix(non-Unitary)
0.1281+i0.5002  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.0460+i0.4972  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.0247+i0.3422  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.2138+i0.1463  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000
0.0590+i0.4117  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000
0.2796+i0.2194  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000

mod 0.8564+i0.0000
mod 0.8124+i0.0000
mod 0.8699+i0.0000
mod 0.9038+i0.0000
mod 0.6497+i0.0000
the trans. matrix(Unitary).
0.1281+i0.5002  0.8564+i0.0000  0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000
0.0460+i0.4972  -0.2972+i-0.0475  0.8124+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i-0.0000
0.0247+i0.3422  -0.2036+i-0.0367  -0.2875+i-0.0058  0.8699+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.2138+i0.1463  -0.1174+i0.1030  -0.1386+i0.1671  -0.1314+i0.1651  0.9038+i0.0000  0.0000+i0.0000
0.0590+i0.4117  -0.2492+i-0.0271  -0.3481+i0.0175  -0.3380+i0.0238  -0.2200+i-0.2397  0.6497+i0.0000
0.2796+i0.2194  -0.1699+i0.1305  -0.2046+i0.2164  -0.1947+i0.2143  -0.2775+i-0.0181  -0.5494+i0.5254

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

```
xip and etap.
0.1291+i-0.0244
-0.0102+i-0.0207
0.0396+i-0.0513
0.0600+i-0.0171
0.0513+i0.0405
0.0227+i-0.0071

0.1753+i-0.0000
0.0000+i0.0000
-0.0000+i0.0000
-0.0000+i0.0000
0.0000+i0.0000
0.0000+i0.0000

xip mod is 0.175299
U(Givens) is :
0.0373+i0.4142  0.3138+i0.5658  0.3845+i0.2456  0.3516+i-0.1447  0.0284+i0.1808  -0.1324+i-0.0683
-0.4958+i-0.0842  0.3192+i-0.1268  0.2212+i0.0016  0.1839+i0.3727  0.4318+i-0.0811  0.3938+i0.2438
0.3703+i-0.0504  -0.6427+i0.0210  0.4446+i-0.0893  0.1653+i-0.0202  0.3838+i0.0433  0.1286+i0.2184
0.1443+i-0.2424  0.0711+i-0.0639  -0.5638+i0.1453  0.6019+i-0.2583  0.3363+i0.1695  -0.0217+i0.0307
0.4434+i-0.0925  0.1527+i0.0147  -0.0061+i0.3384  -0.1897+i0.3345  0.2664+i-0.0815  0.2584+i-0.6037
0.2192+i-0.3190  0.1030+i-0.0811  0.1961+i0.2126  0.2768+i-0.0376  -0.6363+i0.0071  0.4873+i0.1701

isUnitary? Udag*U is :
1.0000+i0.0000  -0.0000+i-0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  0.0000+i0.0000  0.0000+i-0.0000
-0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  -0.0000+i0.0000  0.0000+i0.0000
0.0000+i0.0000  0.0000+i-0.0000  1.0000+i0.0000  -0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i-0.0000
-0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  1.0000+i0.0000  0.0000+i-0.0000  -0.0000+i0.0000
0.0000+i-0.0000  -0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  -0.0000+i-0.0000
0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000  1.0000+i0.0000

Uxi is :
0.0224+i0.0877
0.0081+i0.0872
0.0043+i0.0600
0.0375+i0.0256
0.0103+i0.0722
0.0490+i0.0385

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

```
Householder trans.
U(Householder) is :
0.7524+i0.1420  -0.2593+i0.2296  -0.0012+i0.2173  0.3691+i0.1149  -0.0969+i-0.1083  0.1290+i-0.2436
-0.1577+i-0.3084  0.4796+i0.0905  -0.2550+i0.1953  0.2000+i0.5353  0.0388+i-0.2112  0.4014+i-0.0699
0.0781+i-0.2028  -0.1662+i-0.2749  0.7846+i0.1481  -0.1664+i0.3176  0.1150+i-0.0703  0.2007+i0.1583
0.3856+i0.0276  0.3814+i-0.4255  -0.0392+i-0.3564  0.3559+i0.0672  0.1793+i0.1591  -0.1652+i0.4237
-0.1297+i0.0655  -0.0409+i0.2108  0.0814+i0.1074  0.2250+i-0.0828  0.8941+i0.1688  -0.0285+i-0.1685
0.0313+i0.2739  0.3483+i0.2115  0.2446+i-0.0742  0.0007+i-0.4548  -0.0880+i0.1465  0.6640+i0.1254

isUnitary? Udag*U is :
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+i0.0000
-0.0000+i-0.0000  1.0000+i0.0000  0.0000+i-0.0000  0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i0.0000
0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000
0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  1.0000+i0.0000  -0.0000+i-0.0000  0.0000+i-0.0000
-0.0000+i0.0000  0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000  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+i0.0000  0.0000+i-0.0000  1.0000+i0.0000

Uxi is :
0.0224+i0.0877
0.0081+i0.0872
0.0043+i0.0600
0.0375+i0.0256
0.0103+i0.0722
0.0490+i0.0385

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

[1] + Done                    "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-1vs4fekx.gel" 1>"/tmp/Microsoft-MIEngine-Out
-2cvtegql.v2d"
```

3. n=8

```
mod_define is 4.8124
initial mod_xi is 19.6159+i0.0000, mod_eta is 27.0237+i0.0000
eta:
1.3095+i0.5174
0.1088+i1.3967
0.4309+i1.5614
0.9748+i1.4310
1.6715+i0.6061
1.7423+i1.7051
0.4587+i1.3068
1.1874+i1.0771

final mod_xi is 4.8124+i0.0000, mod_eta is 4.8124+i0.0000
Gievens trans.
the trans. matrix(non-Unitary)
0.2721+i0.1075  1.0000+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
0.0226+i0.2902  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.0895+i0.3244  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.2026+i0.2974  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.3473+i0.1260  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.3620+i0.3543  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000
0.0953+i0.2716  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000
0.2467+i0.2238  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

mod 0.9562+i0.0000
mod 0.9525+i0.0000
mod 0.9292+i0.0000
mod 0.9051+i0.0000
mod 0.8760+i0.0000
mod 0.6560+i0.0000
mod 0.7567+i0.0000
the trans. matrix(Unitary).
0.2721+i0.1075  0.9562+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
0.0226+i0.2902  -0.0391+i-0.0800  0.9525+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
0.0895+i0.3244  -0.0620+i-0.0823  -0.1104+i0.0214  0.9292+i0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000
0.2026+i0.2974  -0.0911+i-0.0618  -0.1043+i0.0598  -0.1487+i0.0507  0.9051+i0.0000  0.0000+i0.0000  -0.0000+i0.0000  -0.0000+i0.0000
0.3473+i0.1260  -0.1130+i0.0032  -0.0510+i0.1125  -0.0934+i0.1315  -0.1663+i0.1199  0.8760+i0.0000  0.0000+i0.0000  -0.0000+i0.0000
0.3620+i0.3543  -0.1429+i-0.0601  -0.1275+i0.1114  -0.1912+i0.1112  -0.2756+i0.0553  -0.3314+i-0.1507  0.6560+i0.0000  0.0000+i0.0000
0.0953+i0.2716  -0.0577+i-0.0666  -0.0930+i0.0247  -0.1254+i0.0086  -0.1543+i-0.0411  -0.1309+i-0.1601  -0.4425+i-0.2184  0.7567+i0.0000
0.2467+i0.2238  -0.0954+i-0.0360  -0.0810+i0.0764  -0.1229+i0.0778  -0.1797+i0.0432  -0.2215+i-0.0908  -0.5708+i0.0216  -0.5748+i0.3114

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

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。

```
xip and etap.
3.8763+i-0.0797
0.7886+i1.5243
0.1723+i0.6113
1.3542+i-0.5845
-0.6605+i-0.5089
0.2950+i1.1027
0.3793+i-0.3040
-0.1578+i-0.5858

4.8124+i0.0000
-0.0000+i0.0000
-0.0000+i-0.0000
0.0000+i-0.0000
-0.0000+i0.0000
-0.0000+i-0.0000
-0.0000+i-0.0000
0.0000+i-0.0000

xip mod is 4.812382
U(Givens) is :
0.1857+i-0.4013  0.1097+i-0.0425  0.4111+i0.3936  -0.3147+i0.2872  -0.0246+i-0.2550  -0.0310+i0.3457  -0.1081+i0.2902  -0.0499+i0.0266
-0.6369+i0.1112  0.2772+i-0.0830  0.1100+i0.3396  -0.0305+i-0.1779  0.4732+i0.0401  0.1593+i0.1372  -0.0344+i-0.0705  0.2496+i0.0031
0.2114+i0.0900  -0.7553+i0.0027  0.1416+i0.2016  0.1389+i-0.0081  0.2726+i0.2254  0.2104+i0.2282  0.1092+i-0.0034  0.2133+i0.1336
0.2449+i0.0181  0.2327+i-0.0763  -0.5703+i-0.0253  0.0689+i0.2699  0.1488+i-0.2252  0.5071+i0.1975  0.1843+i0.1616  0.2130+i-0.0417
0.2148+i-0.1320  0.1480+i-0.2034  0.1408+i-0.1336  -0.6272+i-0.1339  0.0163+i0.3798  0.2687+i-0.2104  0.2880+i-0.2308  0.1488+i0.0697
0.3431+i-0.0429  0.3001+i-0.1698  0.2516+i-0.0869  0.3971+i-0.0211  -0.0841+i0.1135  -0.0461+i0.1488  -0.3609+i-0.3385  0.4898+i-0.0603
0.1856+i0.0644  0.1956+i-0.0110  0.1500+i0.0190  0.2076+i0.0891  0.3116+i0.2257  0.0874+i0.2439  0.1862+i-0.2741  -0.6439+i-0.3310
0.2244+i-0.0367  0.1930+i-0.1191  0.1632+i-0.0634  0.2604+i-0.0237  0.4416+i0.0580  -0.3158+i-0.3678  0.3102+i0.4888  0.1006+i0.1389

isUnitary? Udag*U is :
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+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000
0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  -0.0000+i-0.0000  0.0000+i0.0000  0.0000+i0.0000  -0.0000+i-0.0000  0.0000+i0.0000
-0.0000+i-0.0000  0.0000+i-0.0000  1.0000+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  -0.0000+i0.0000  -0.0000+i-0.0000  1.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  -0.0000+i0.0000  -0.0000+i-0.0000
-0.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i-0.0000  -0.0000+i0.0000  0.0000+i0.0000
0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000
-0.0000+i-0.0000  -0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000  0.0000+i-0.0000  1.0000+i0.0000  0.0000+i0.0000
-0.0000+i0.0000  0.0000+i0.0000  0.0000+i-0.0000  -0.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  0.0000+i-0.0000  1.0000+i0.0000

Uxi is :
1.3095+i0.5174
0.1088+i1.3967
0.4309+i1.5614
0.9748+i1.4310
1.6715+i0.6061
1.7423+i1.7051
0.4587+i1.3068
1.1874+i1.0771

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

```
Householder trans.
U(Householder) is :
0.5325+i0.0110  -0.0814+i-0.0500  0.1991+i-0.4812  0.3541+i0.2183  -0.2655+i0.1314  0.2846+i-0.0555  0.2939+i0.0251  0.0134+i0.0630
-0.0833+i0.0466  0.9803+i0.0202  -0.0147+i-0.1054  0.0850+i0.0019  -0.0332+i0.0506  0.0444+i-0.0392  0.0544+i-0.0260  0.0089+i0.0097
0.1791+i0.4889  -0.0190+i0.1047  0.4197+i0.0086  0.0833+i-0.4559  0.2438+i0.2225  -0.1728+i-0.2730  -0.0929+i-0.3152  0.0599+i-0.0394
0.3627+i-0.2036  0.0850+i0.0016  0.0645+i0.4590  0.6296+i0.0130  0.1443+i-0.2206  -0.1933+i0.1711  -0.2368+i0.1134  -0.0387+i-0.0422
-0.2599+i-0.1422  -0.0311+i-0.0519  0.2528+i-0.2122  0.1352+i0.2264  0.8121+i0.0167  0.1763+i0.0521  0.1579+i0.1001  -0.0109+i0.0393
0.2821+i0.0672  0.0428+i0.0410  -0.1839+i0.2657  -0.1861+i-0.1789  0.1782+i-0.0448  0.8199+i0.0169  -0.1749+i-0.0538  0.0001+i-0.0399
0.2947+i-0.0130  0.0533+i0.0282  -0.1057+i0.3112  -0.2319+i-0.1231  0.1618+i-0.0936  -0.1770+i0.0466  0.8137+i0.0167  -0.0110+i-0.0391
0.0160+i-0.0624  0.0093+i-0.0093  0.0582+i0.0419  -0.0404+i0.0406  -0.0092+i-0.0397  -0.0015+i0.0399  -0.0126+i0.0386  0.9909+i0.0204

isUnitary? Udag*U is :
1.0000+i0.0000  0.0000+i-0.0000  0.0000+i0.0000  0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000  -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  0.0000+i-0.0000  -0.0000+i0.0000  -0.0000+i0.0000
0.0000+i-0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  0.0000+i-0.0000  -0.0000+i-0.0000  -0.0000+i-0.0000
0.0000+i-0.0000  0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000  0.0000+i-0.0000  0.0000+i0.0000  0.0000+i-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+i0.0000  0.0000+i0.0000  0.0000+i-0.0000  0.0000+i0.0000
-0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  0.0000+i-0.0000  0.0000+i-0.0000  1.0000+i0.0000  -0.0000+i0.0000  -0.0000+i-0.0000
-0.0000+i0.0000  -0.0000+i-0.0000  -0.0000+i0.0000  0.0000+i0.0000  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  -0.0000+i0.0000  0.0000+i0.0000  0.0000+i0.0000  -0.0000+i0.0000  0.0000+i0.0000  1.0000+i0.0000

Uxi is :
1.3095+i0.5174
0.1088+i1.3967
0.4309+i1.5614
0.9748+i1.4310
1.6715+i0.6061
1.7423+i1.7051
0.4587+i1.3068
1.1874+i1.0771

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

[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-dqdhhklc.4cp" 1>"/tmp/Microsoft-MIEngine-Out
-jmvgfhdn.gzo"
heaven@heaven:~/Desktop/doc/Linear_Algebra$
```

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。