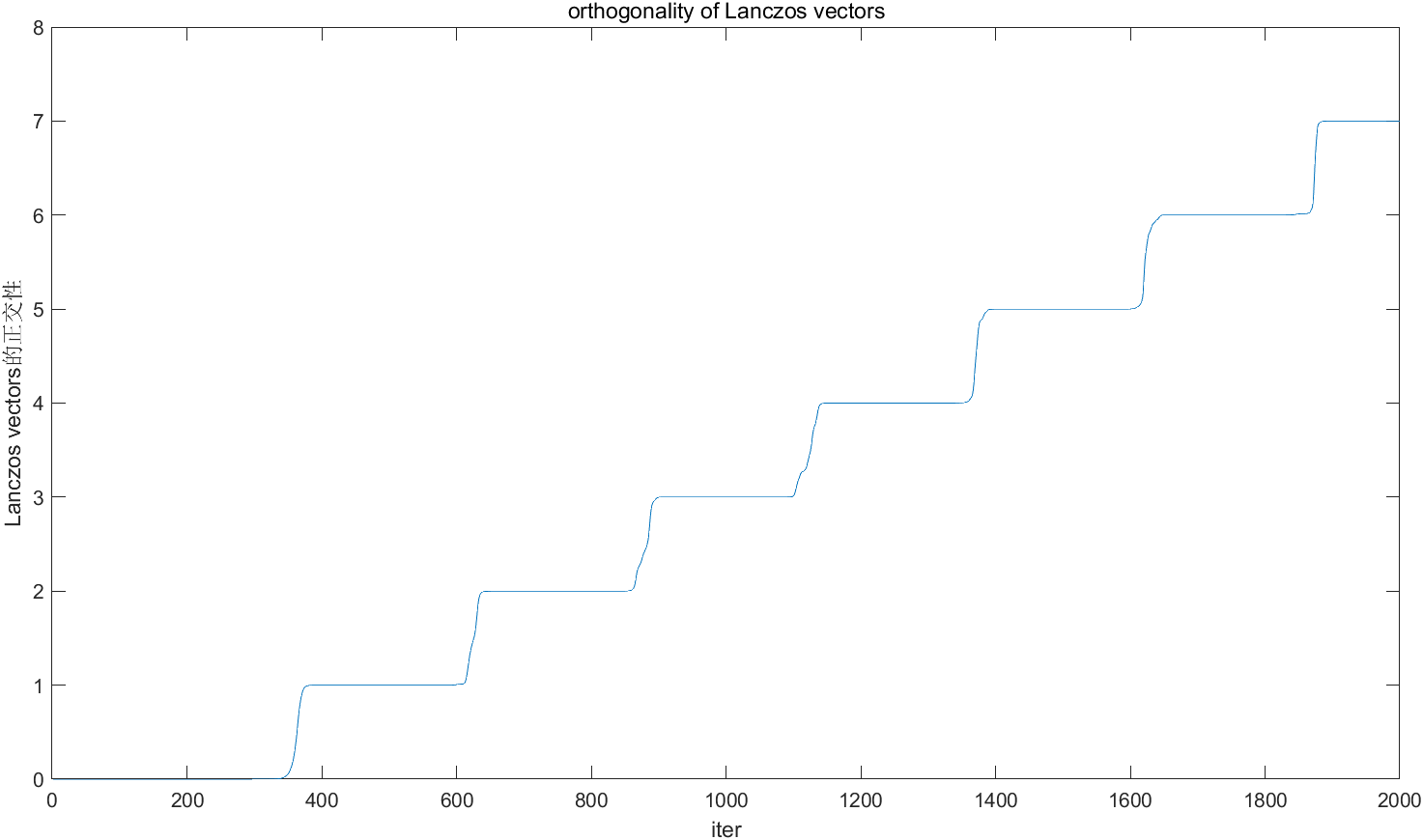
12月13日作业

## 吐槽与求助：这次的作业我基本把每道题都做了。但程序跑出来的结果实在让人费解，也不知道为什么！有的结果疯狂发散，有的结果和邵老师上课讲的相去甚远，有的出现一些意想不到但又想不清楚的规律。仔细想来，这学期以来，数值算法这门课就好像孤魂野鬼，既不知道自己在哪里，也不知道要学什么，没人疼没人爱，自己瞎搞，乱学一通，经常出现了问题也不知道找谁解决……唉，又崩溃啦。

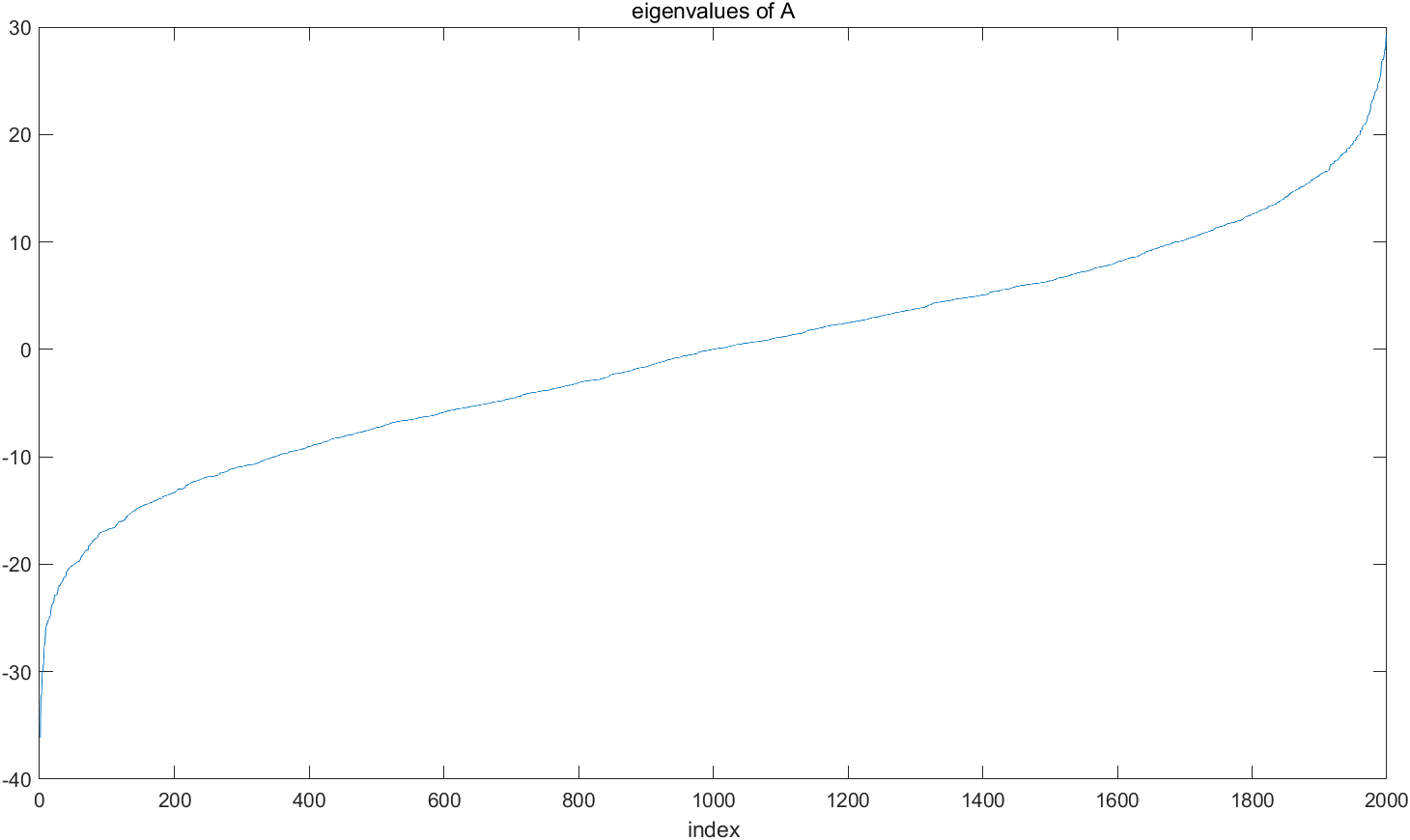
# 第一题

Lanczos向量正交性如下：

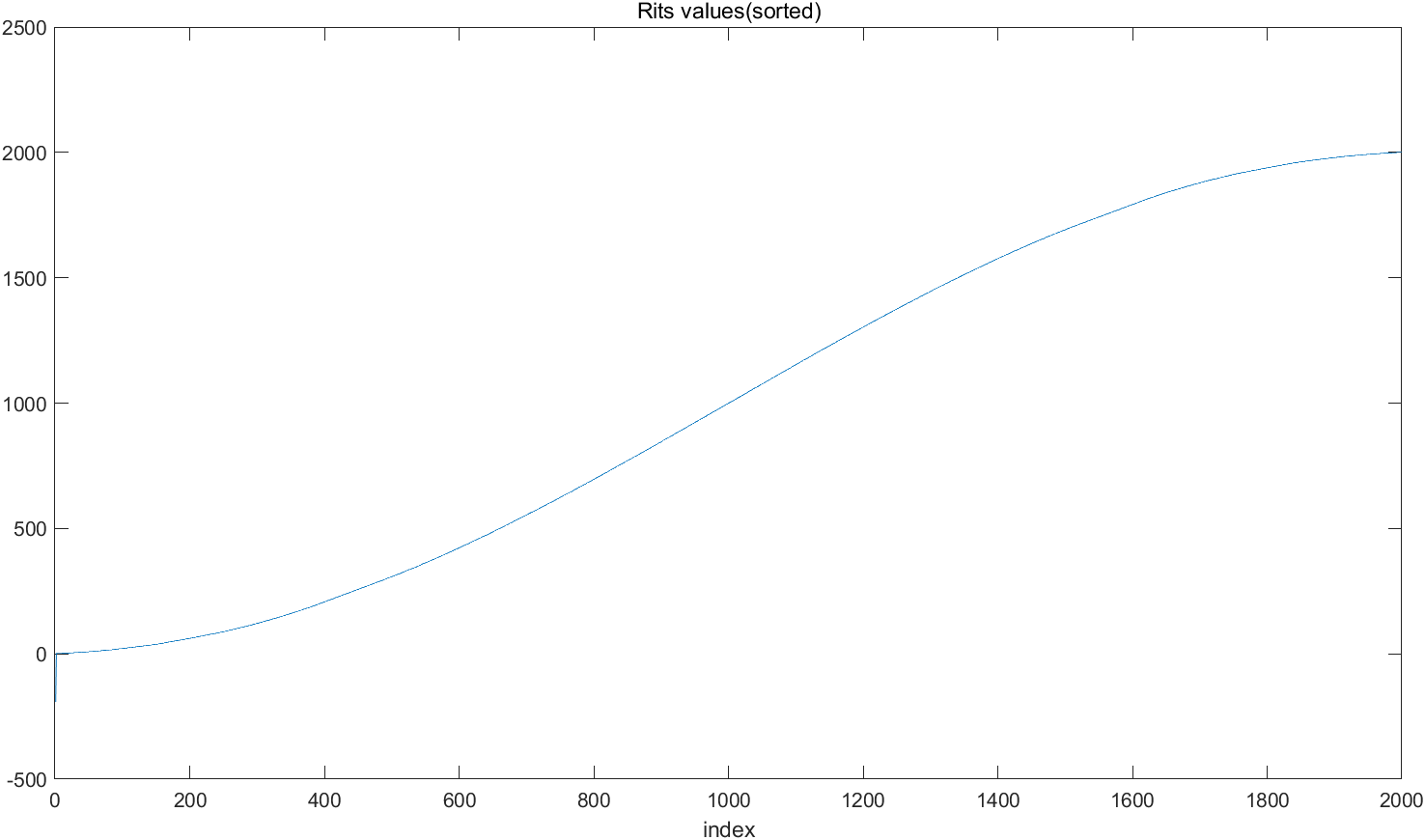


Q：请问助教，为什么会呈现“阶梯式”的失去正交性啊？而且每个阶梯的“宽度”还基本一致？

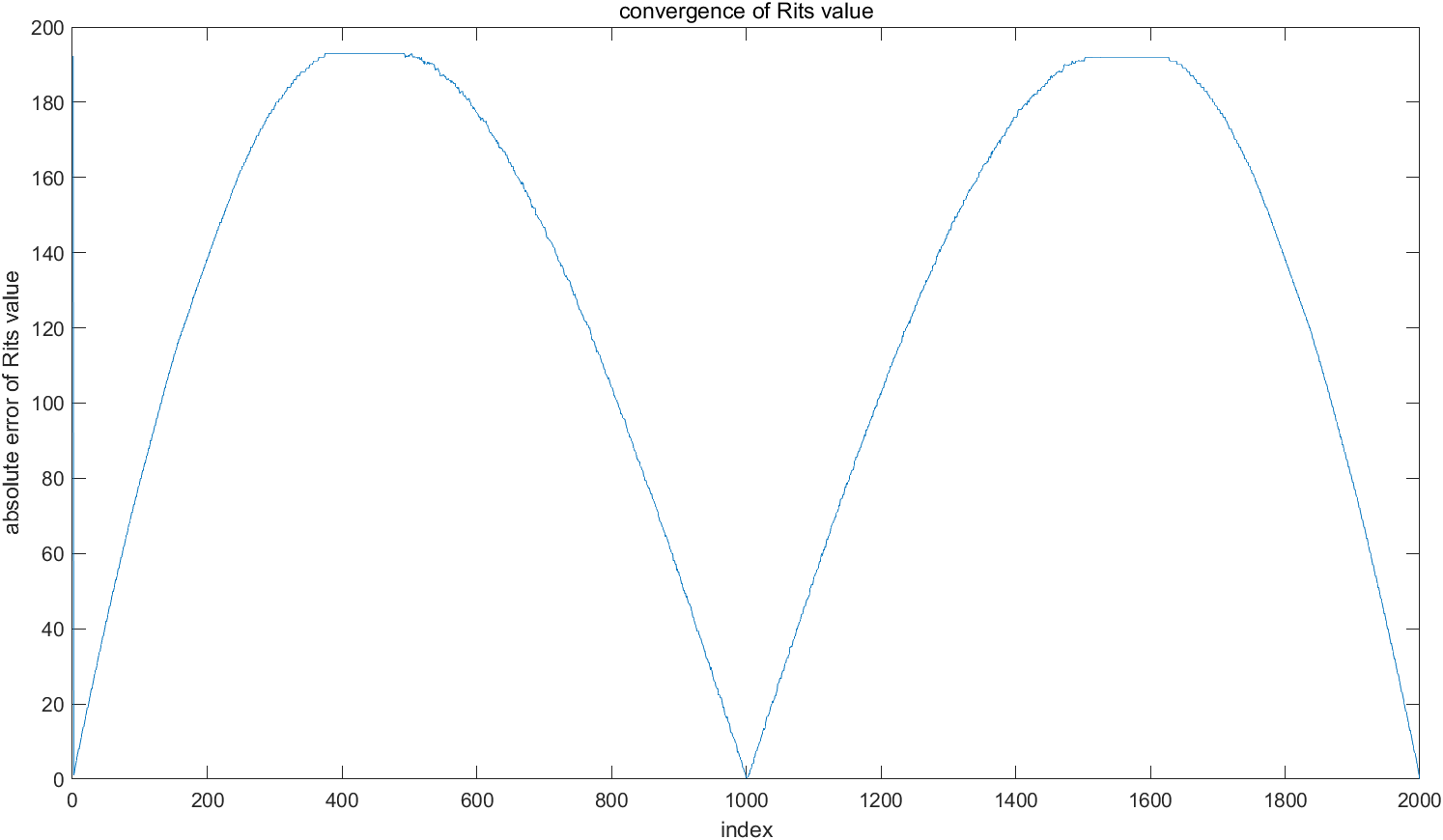
原矩阵的特征值（输入时就已经从小到大排序）



Rits值（计算完T后也进行了一个重排序使得其特征值从小到大）如下：

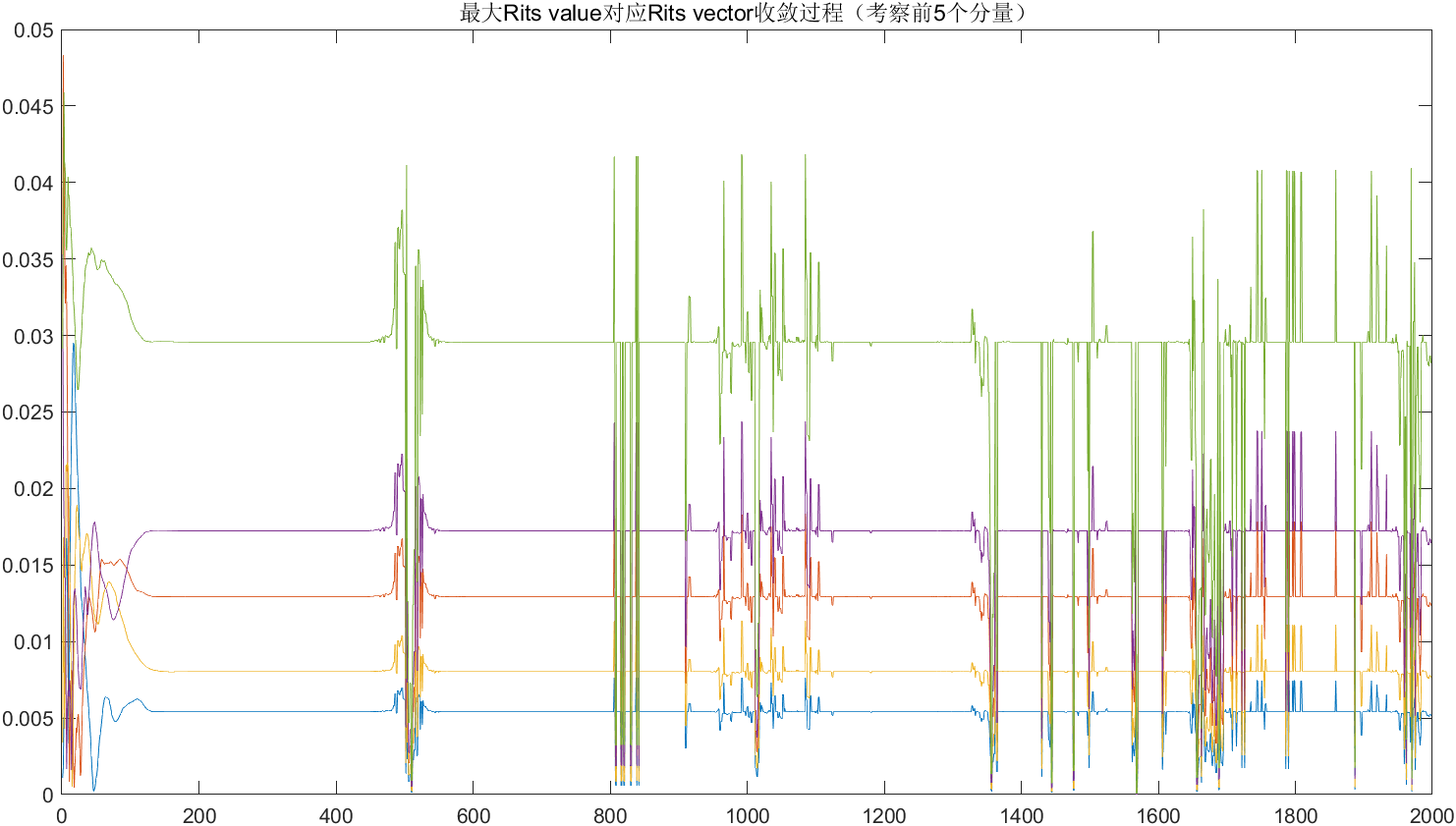


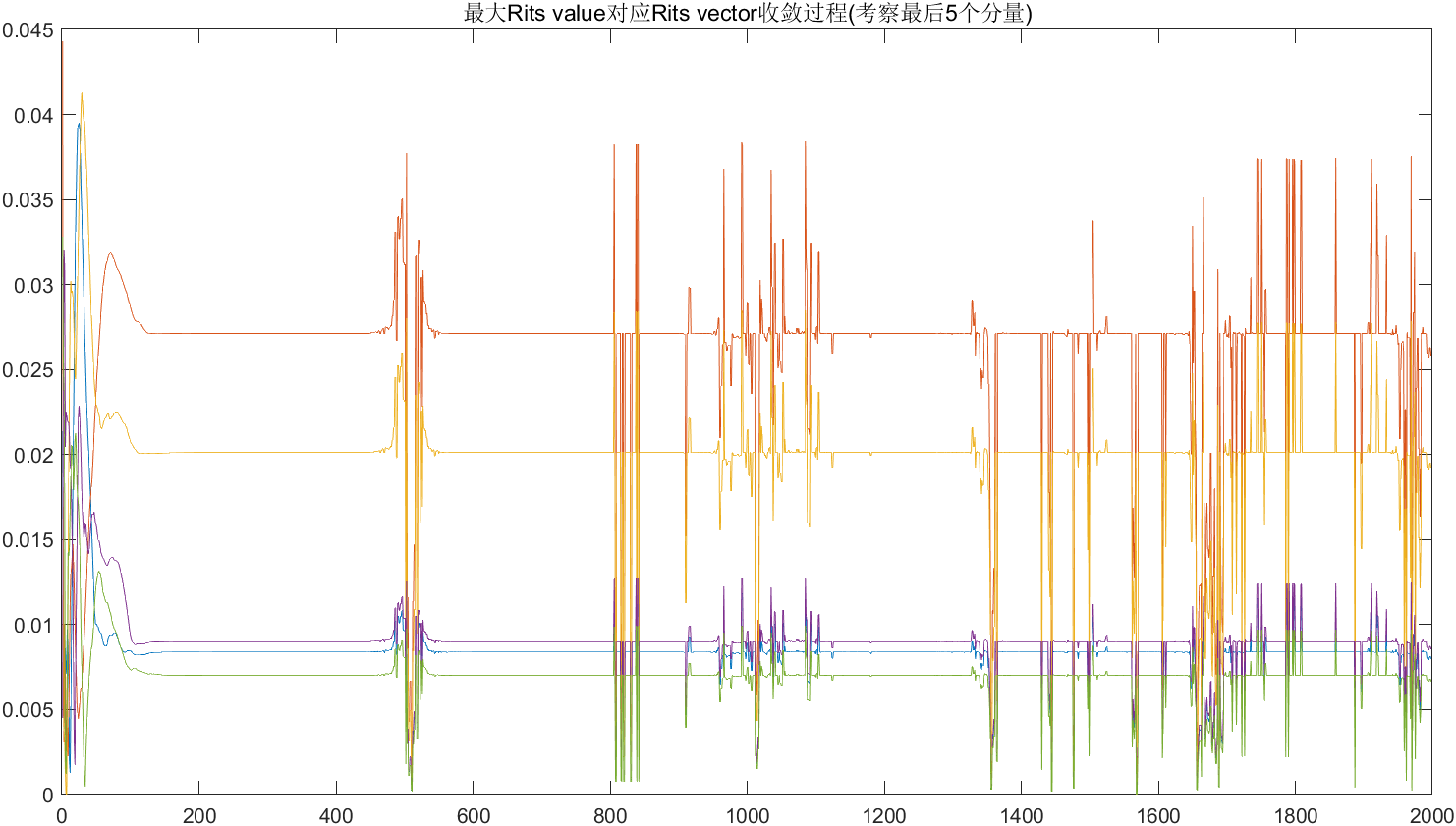
Rits值的收敛性如下：

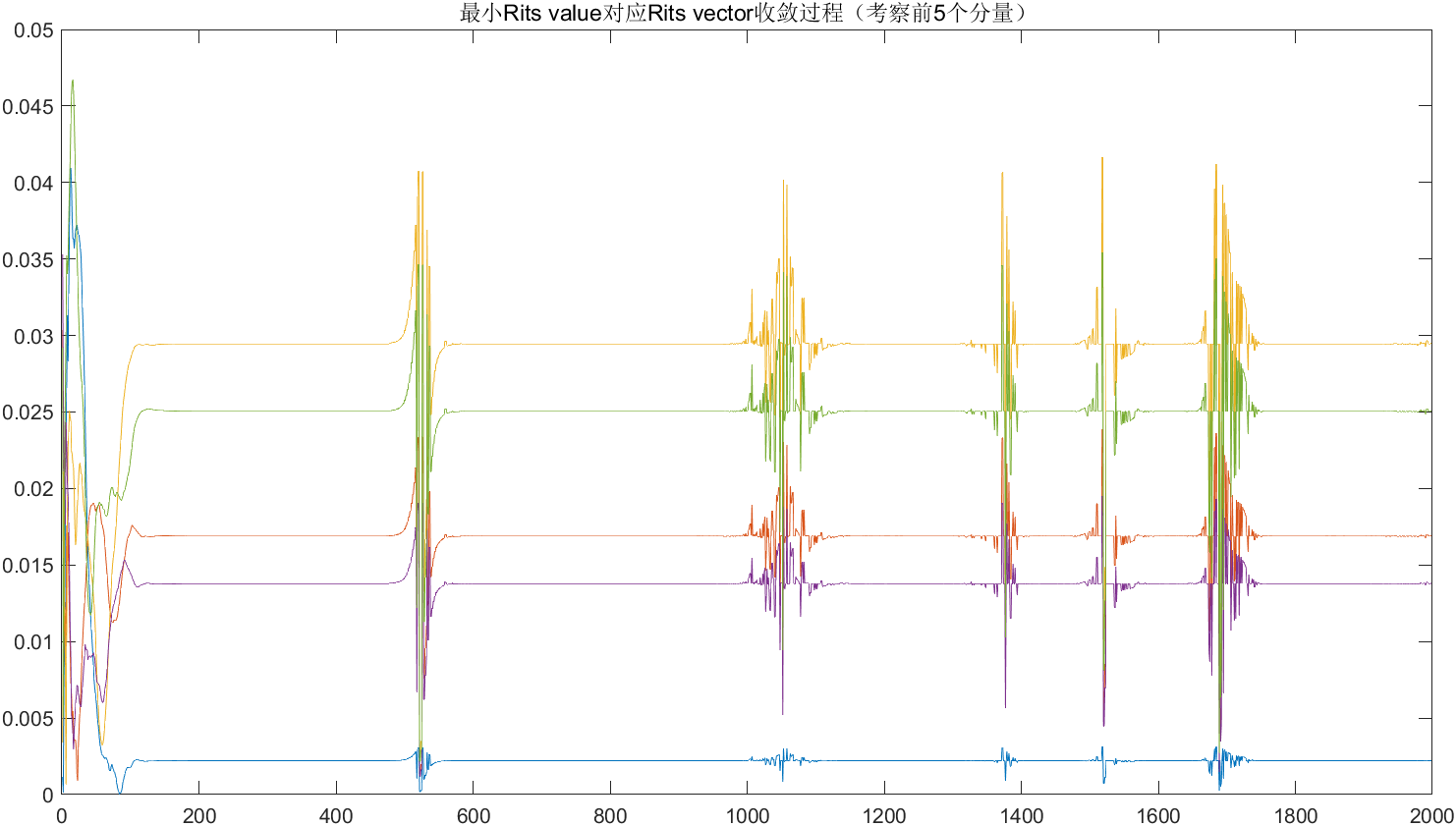


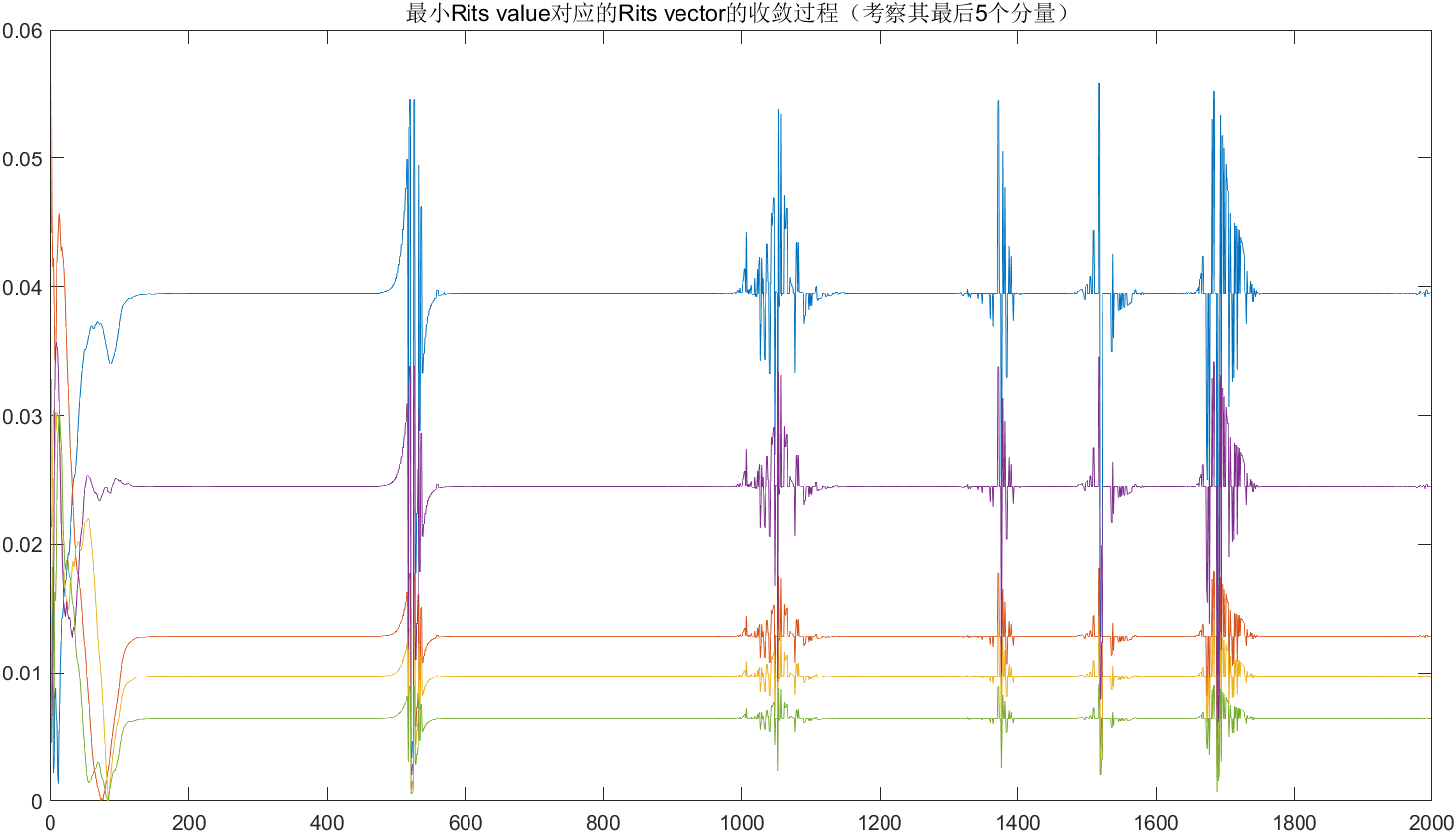
Q：请问助教，我用了大量的例子去测试，发现都是中间和两端收敛性较好，可以算的准。但是按照邵老师上课讲的，Lanczos算法应该时两端算的准，中间都算不准，不知道这是什么原因呀？

Rits vector收敛性：





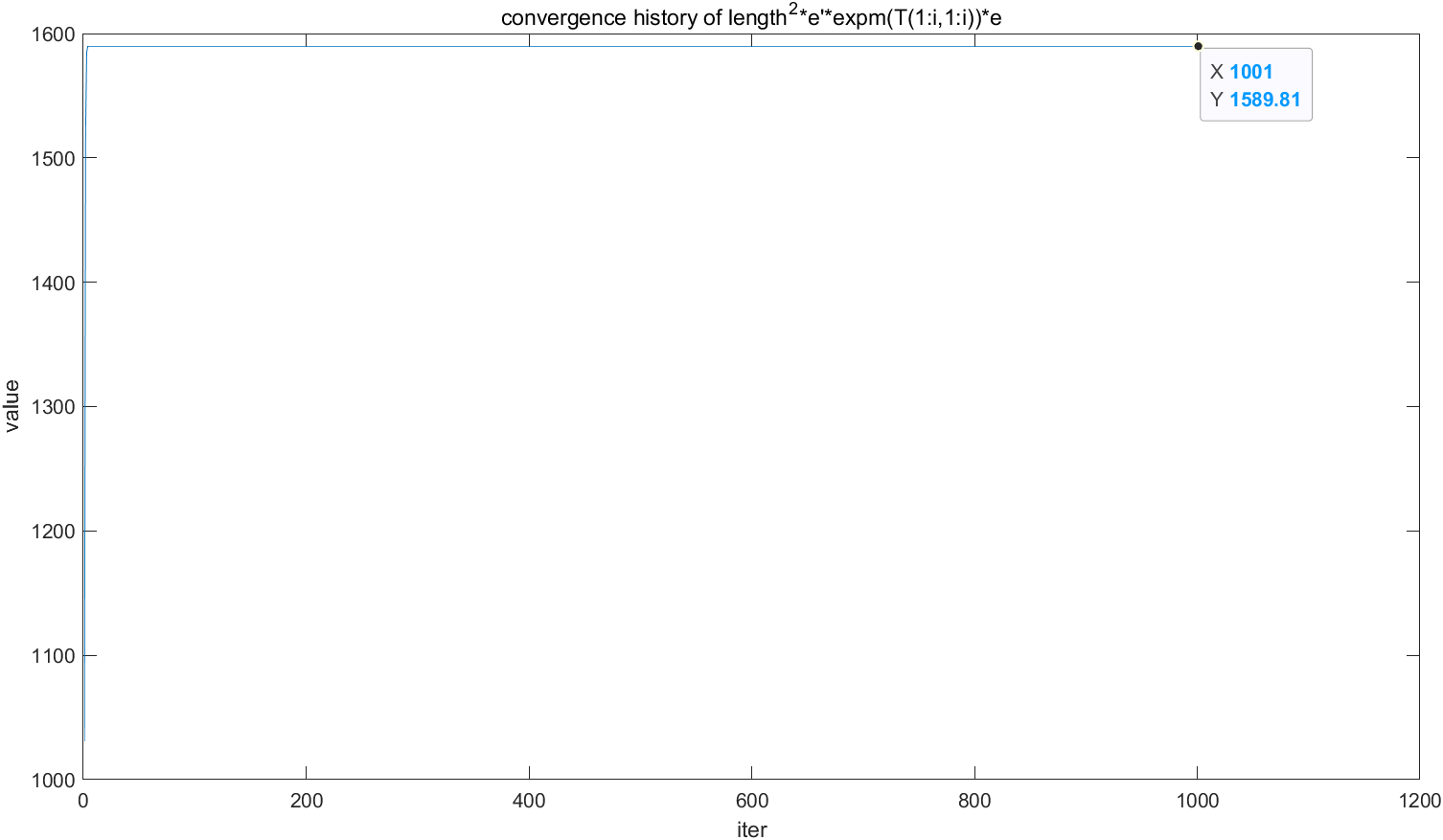




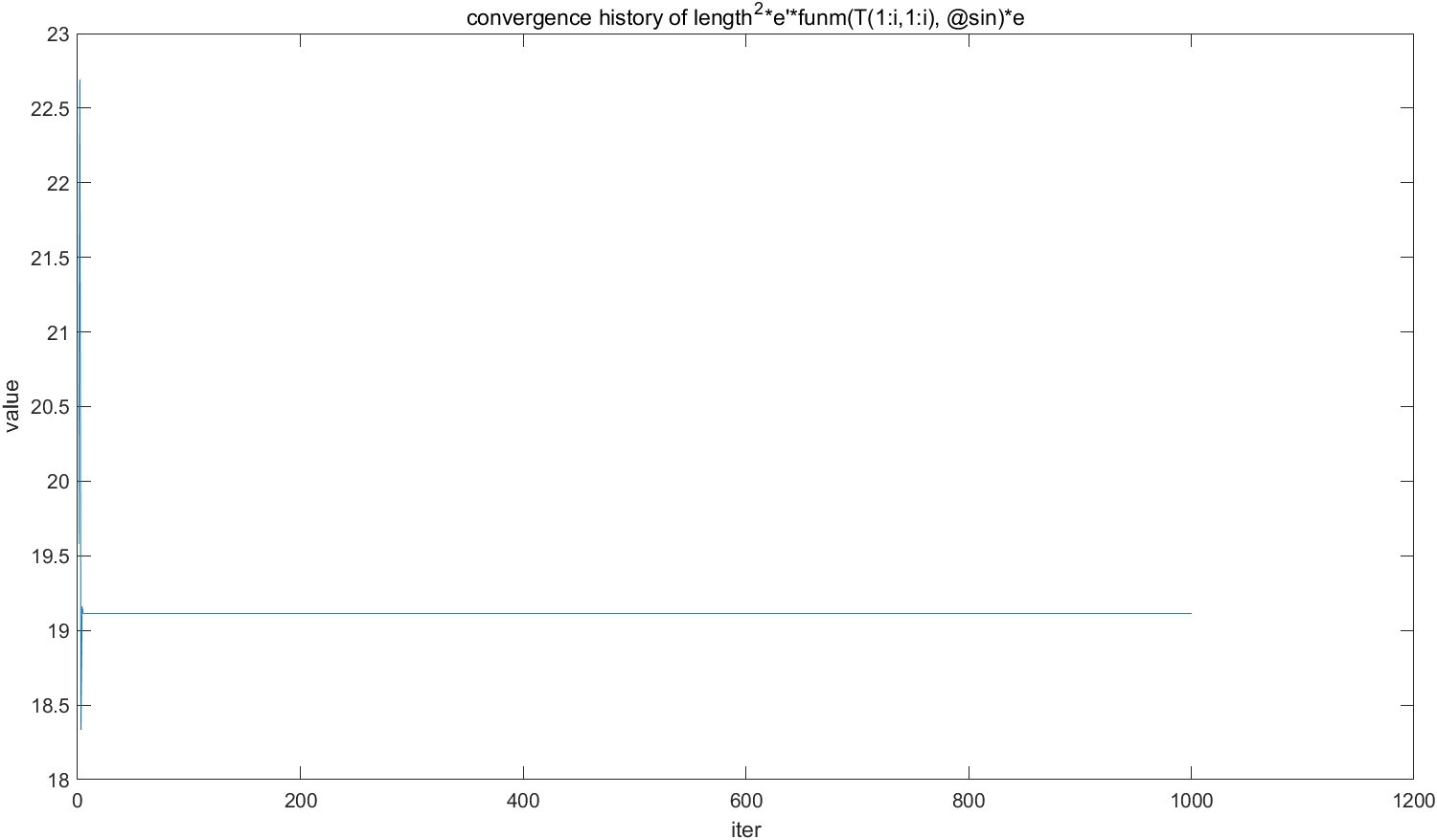
Q：这里特别诡异，按道理来说，如果Rits vector收敛，那么其各个分量也应该收敛，但体现出来的却是，分量大部分情况下收敛，但在特定位置会出现巨大的“震荡”，非常诡异。更离谱都是，最大Rits值和最小Rits值对应的Rits vector的分量出现“震荡”的地方还不太一样！

# 第二题

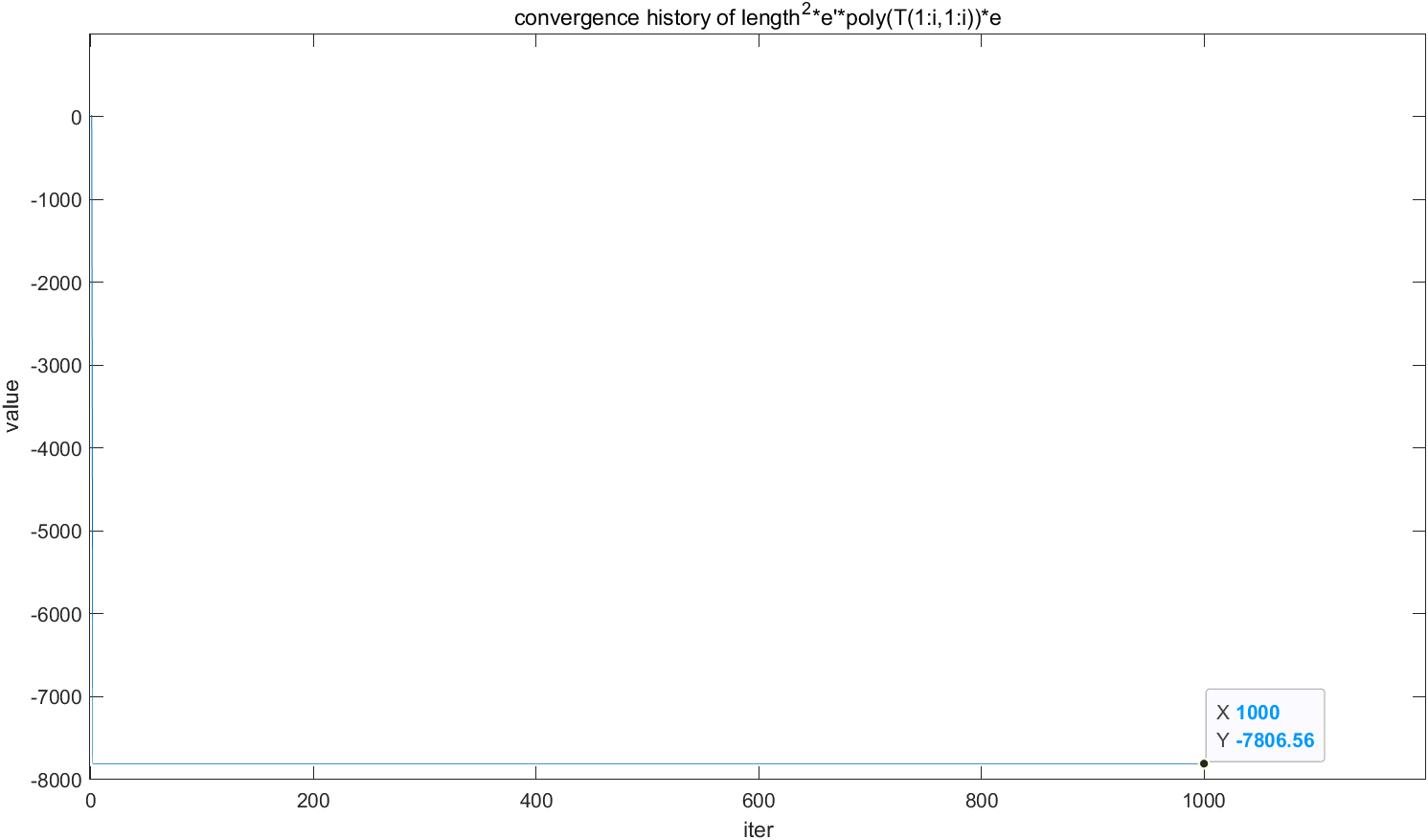
把f取为矩阵指数：



把f取作矩阵正弦：



把f取作矩阵多项式：



可以看出，以上近似效果都非常好，Lanczos过程进行到20次以内就已经基本收敛到正确的值。

# 第三题

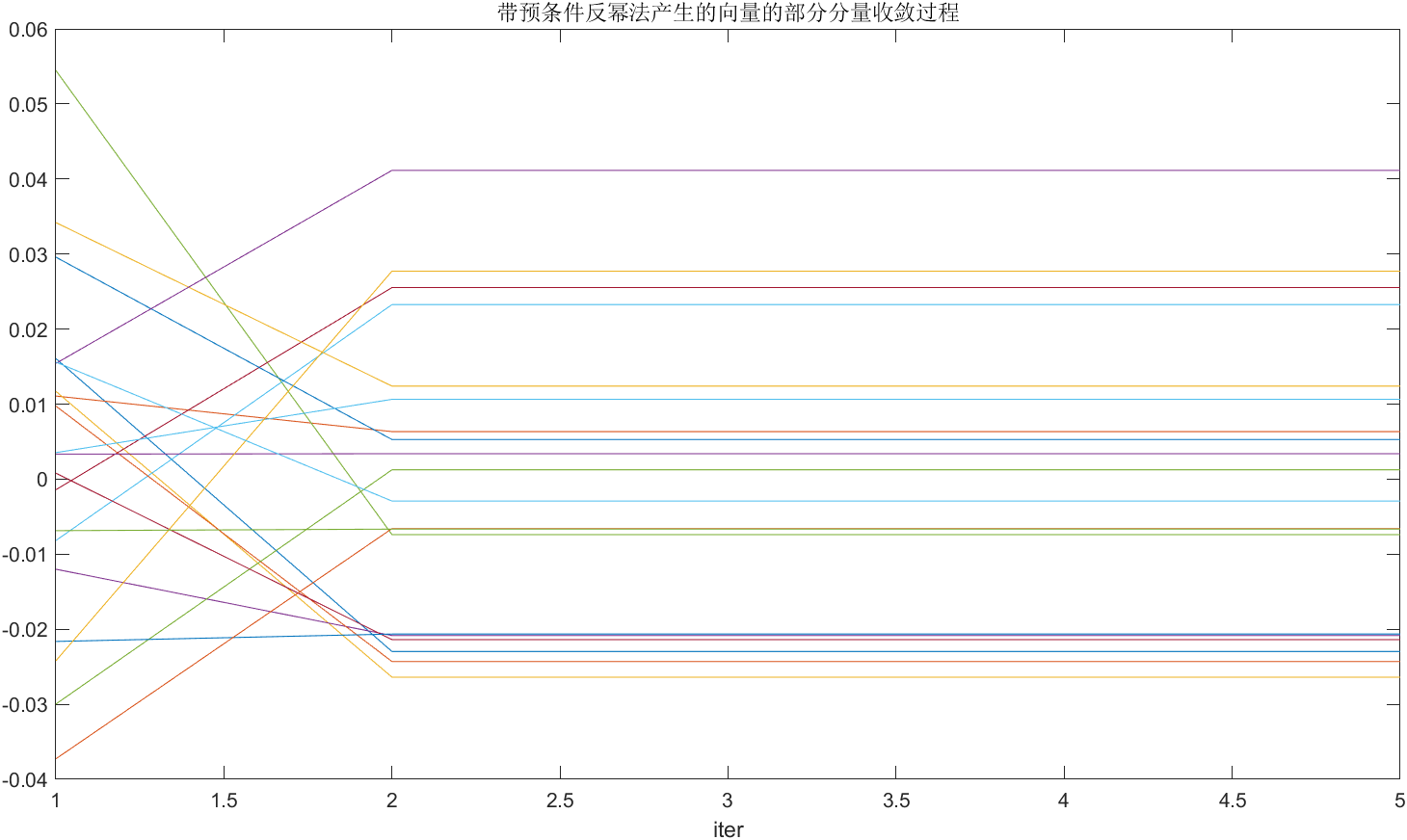
记：Box(v,A) = vT\*f(A)\*v

由于A是对称阵，则

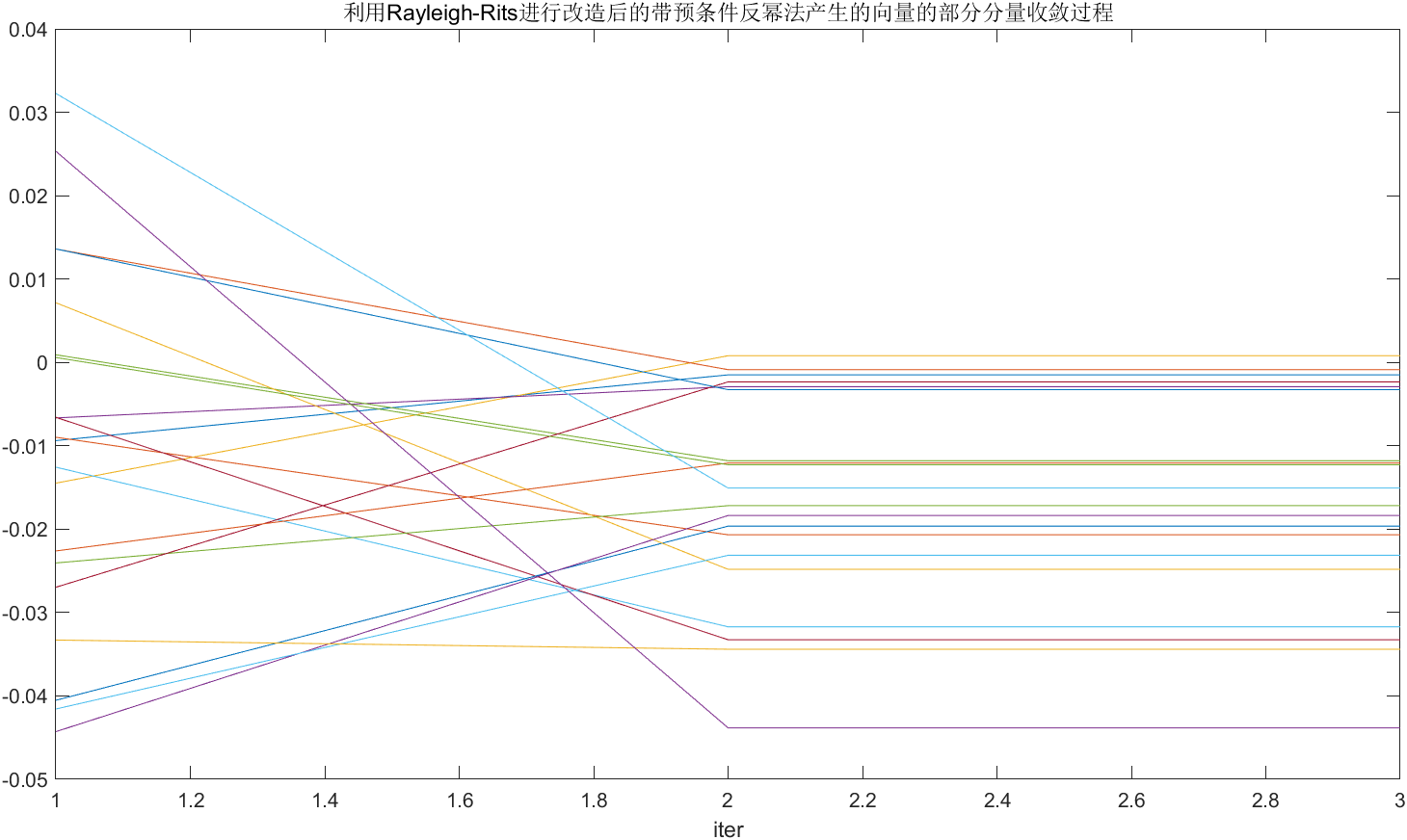
uT\*f(A)\*v = 0.5\*(Box(u+v,A) – Box(v,A) – Box(u,A))

# 第四题

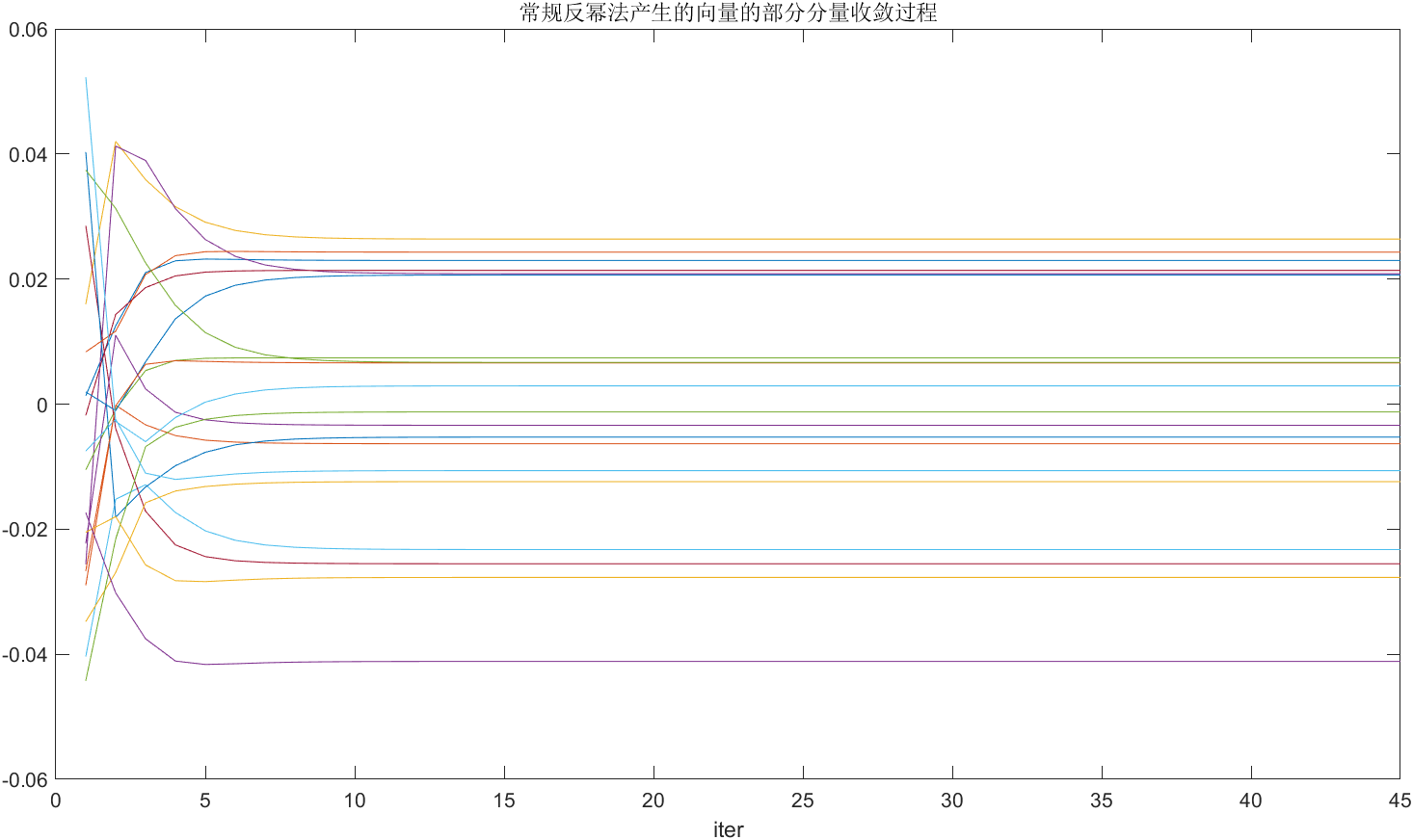
带预条件的反幂法计算出的特征向量的部分分量收敛过程如下（5次就收敛）：



带预条件且用Rayleigh-Rits改进过的反幂法计算出的特征向量的部分分量收敛过程如下（3次就收敛）：



原始反幂法计算出的特征向量的部分分量收敛过程如下（收敛次数最多，45次）：



# 第五题

**我在Paige-Saunders双对角化那里卡住了，按照Matrix Computation那本书写的，但不知道为何求出来的B一直是错的。**

**只能暂时把源代码放一份，如下：**

%% LSQR

m = 20;

n = 5;

A = randn(m,n); % 最小二乘问题中，m>n

B = zeros(m,n);

main\_diag = zeros(1,n);

sub\_diag = zeros(1,n); % 第一个元素不在此对角线上，但为了编程方便加上它

x0 = zeros(n,1);

b = randn(m,1);

residual = b - A\*x0;

uc = residual./norm(residual); %uc是U的第一列

U = zeros(m);

U(1:m,1) = uc;

p = uc;

sub\_diag(1,1) = 1;

V = zeros(n);

X=zeros(n,2); %用于储存产生的向量

X(1:n,1) = x0;

error = zeros(1,n); % 用于储存残差

error(1,1) = norm(residual);

k = 1;

while(sub\_diag(1,k)>1e-16 && k<=n)

% 首先进行二对角化

U(1:m,k) = p./sub\_diag(1,k);

if k==1

r = A'\*U(1:m,k);

else

r = A'\*U(1:m,k)- sub\_diag(1,k)\*V(1:n,k-1);

end

main\_diag(1,k) = norm(r);

V(1:n,k) = r./main\_diag(1,k);

p = A\*V(1:n,k) - main\_diag(1,k)\*U(1:m,k);

sub\_diag(1,k+1) = norm(p);

B(k,k) = main\_diag(1,k);

B(k+1,k) = sub\_diag(1,k+1);

% 然后进行LSQR过程

Bk = B(1:k+1,1:k);

v = norm(residual)\*[1,zeros(1,k)]';

% 对Bk和v同时做Givens旋转

for j = 1:k

alpha = Bk(j,j);

beta = Bk(j+1,j);

c = alpha/sqrt(alpha^2+beta^2);

s = beta/sqrt(alpha^2+beta^2);

G = eye(k+1);

G(j,j) = c;

G(j+1,j+1) = c;

G(j,j+1) = s;

G(j+1,j) = -s;

Bk = G\*Bk;

v = G\*v;

end

Rk = Bk(1:k,1:k);

y = Rk\v(1:k,1);

xk = x0 + V(1:n,1:k)\*y;

X(1:n,k+1) = xk;

error(1,k+1) = norm(Bk\*y-v);

k = k+1;

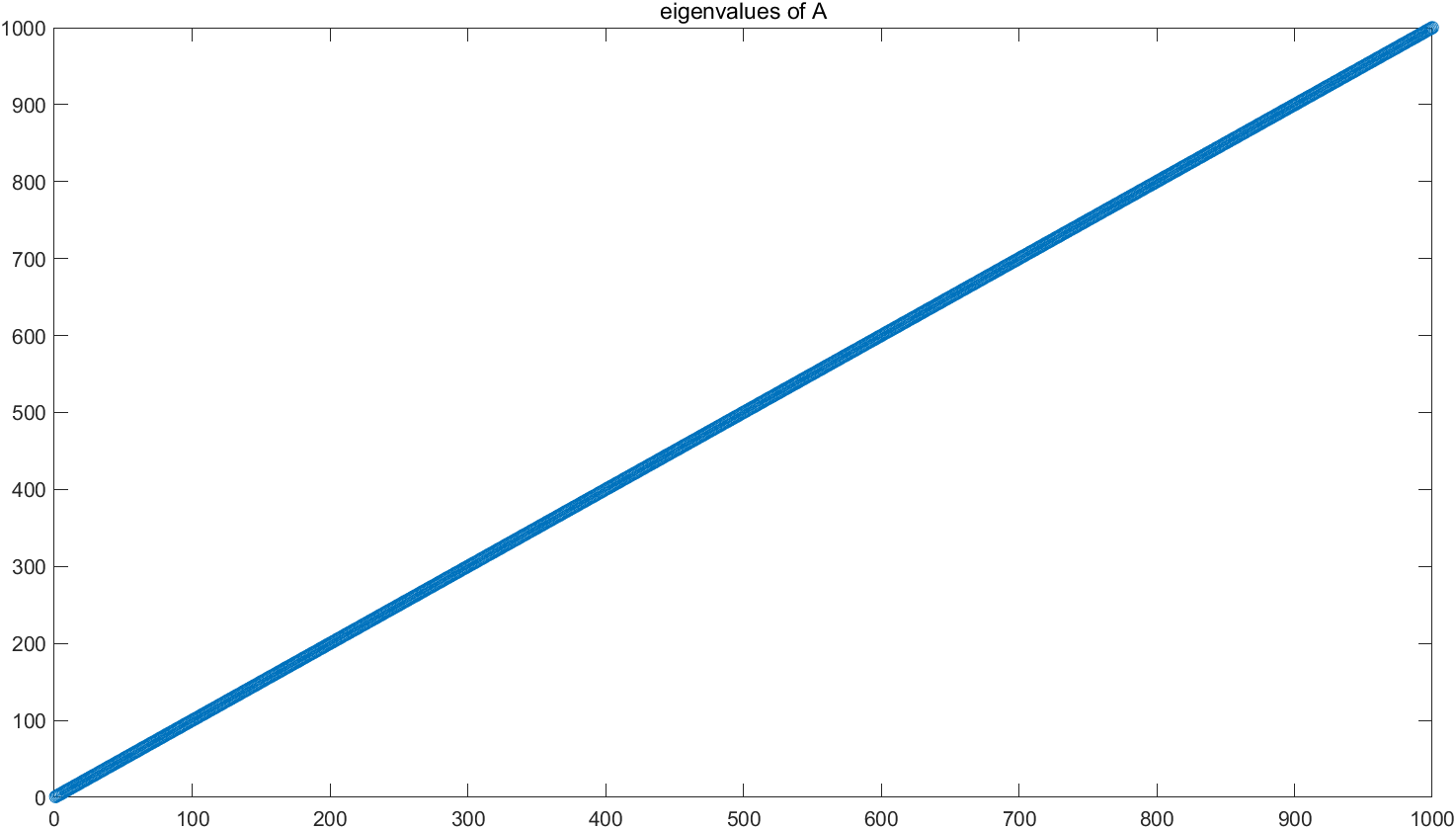
end

figure()

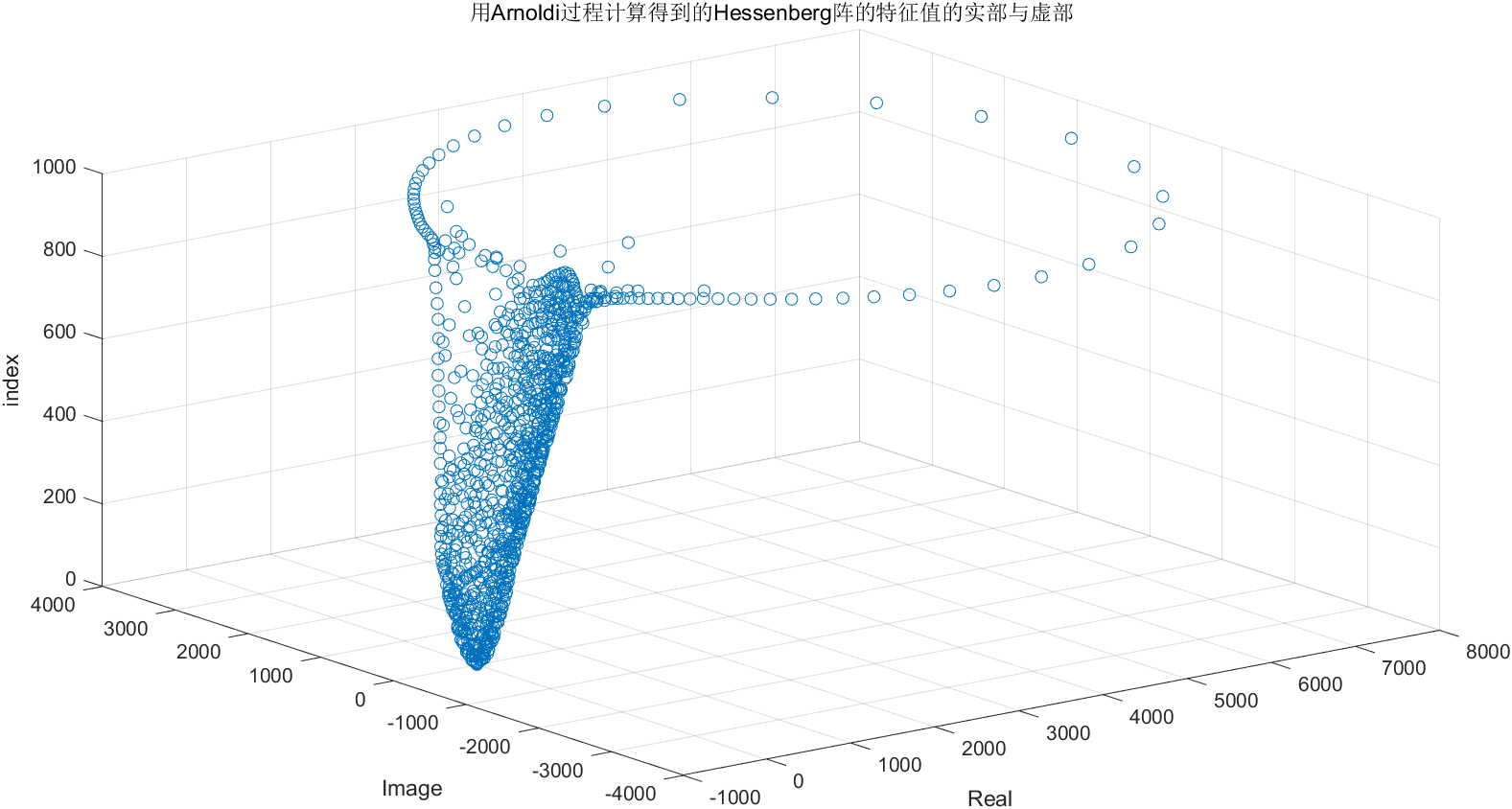
plot(error)

# ****第六题****

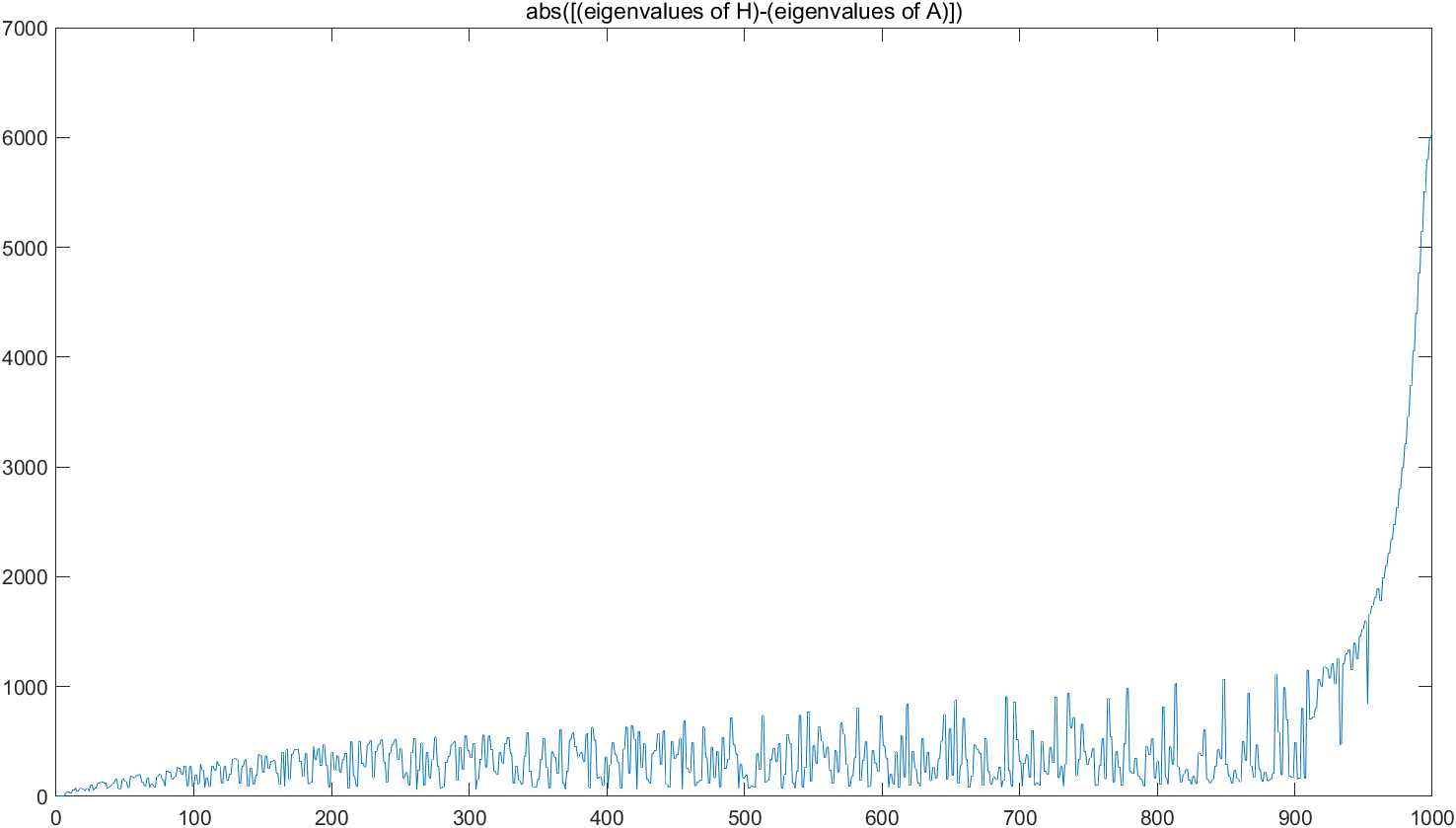
原非对称矩阵A的特征值如下（1到1000）：



计算得到的上Hessenberg阵的特征值为（较大特征值处不知为何计算得极其不准确）：



H的特征值与A的特征值相减再取模（可以说基本没有特征值算得准！特别离谱）：



Q：用了非常多例子测试，均是如此。按照邵老师上课说的，Arnoldi过程应该是两端的特征值算的准中间的算不准。我跑出来的结果，不仅中间的算不准，两端的更算不准，实在无语！ 不知道究竟是为什么？