# 迭代方法求解矩阵问题

目录

[迭代方法求解矩阵问题 1](#_Toc498342564)

[全篇上下文中采用的矩阵生成方法 1](#_Toc498342565)

[关于矩阵条件数的控制 1](#_Toc498342566)

[Result & Remark 2](#_Toc498342567)

[迭代算法部分 2](#_Toc498342568)

[CG算法验证与误差分析 3](#_Toc498342569)

[Result & Remark 4](#_Toc498342570)

[SD算法验证与误差分析 4](#_Toc498342571)

[Result & Remark 4](#_Toc498342572)

[预优化Conjugate Gradient Method 5](#_Toc498342573)

[预优化CG算法验证与误差分析 5](#_Toc498342574)

[Result & Remark 5](#_Toc498342575)

## 全篇上下文中采用的矩阵生成方法

function [A,b,x0]=raw\_data(k,cond\_ctr)

% Raw data generating for error test

% Input:

% k - Matrix Rank

% cond\_ctr - condition number control agrument in aprandsym build-in

% function; 'none' indicates using random method generation

% Output:

% A, b and x0 stand for the matrix, the rhs vector and the initial

% solution

if cond\_ctr=='none'

B=rand(k,k);

A=0.5\*(B'+B)+eye(k,k);

% A=B'\*B;

else

A=sprandsym(k,0.3,cond\_ctr);

A=A+eye(k,k);

end

b=rand(k,1);

x0=rand(k,1);

end

### 关于矩阵条件数的控制

clear;

clc;

c\_list=(0.1:0.01:0.7);

m\_list=(35:45);

fitdata=[];

for cond\_ctr=c\_list

for rank\_mat=m\_list

[A,~,~]=raw\_data(rank\_mat,cond\_ctr);

fitdata=[fitdata;cond\_ctr,rank\_mat,cond(A)];

end

end

v\_names={'CP\_parameter','Rank','Condition\_Number'};

relation\_table=table(fitdata(:,1),fitdata(:,2),fitdata(:,3),'VariableNames'...

,v\_names);

fit\_result=fit([relation\_table.CP\_parameter,relation\_table.Rank],...

relation\_table.Condition\_Number,'poly22');

filename = 'relation\_table.xlsx';

xlswrite(filename,fitdata);

### Result & Remark

这一部分是在寻在raw\_data函数中输入参数与形成的随机矩阵之间的关系，利用了matlab的拟合函数fit。输入参数为控制量CP\_parameter(x)、矩阵阶数Rank(y).

运行结果如下：

>> fit\_result

Linear model Poly22:

fit\_result(x,y) = p00 + p10\*x + p01\*y + p20\*x^2 + p11\*x\*y + p02\*y^2

Coefficients (with 95% confidence bounds):

p00 = 58.1 (2.572, 113.6)

p10 = -359.1 (-382.7, -335.5)

p01 = -0.319 (-3.081, 2.443)

p20 = 440 (429, 451)

p11 = 7.066 (6.52, 7.611)

p02 = 2.963e-06 (-0.03439, 0.03439)

使用SAS软件进行指数函数拟合（固定矩阵阶数为40）结果如下：

模拟模型为：

Condition\_Number = a - b \* c \*\* CP\_parameter

参数估计为：

Parameter Estimate

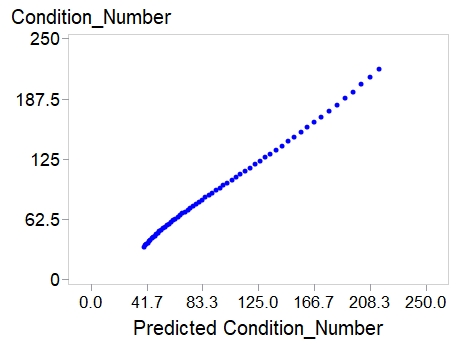
a 13.4587

b -17.9245

c 31.6515

拟合效果图：

（预测值与实际值做比较图）



## 迭代算法部分

function [x,itr] = mat\_gradient(A,b,method,iteration,x0,tol)

% mat\_gradient Conjugate Gradient Method or Steepest Descent Method

% A\*X=B for X with initial X0. The N-by-N coefficient matrix A must be

% symmetric and the right hand side column vector B must have length N.

% Input:

% A - input matrix

% b - vector b

% x0 - initial vector

% method - sd,cg or sm;

% Steepest Descent Method, Conjugate Gradient Method or small w setting

% tol - tolerance The default is 1e-10.

% itaration - setting the max iteration; 'auto' for default setting

if nargin==4

x0=zeros(b);

tol=1e-10;

elseif nargin==5

tol=1e-10;

elseif nargin~=6

error('mat\_gradient: invalid input parameters');

end

r = b - A\*x0;

e=[];

if norm(r) < tol

itr=0;

return

end

x=x0;

if method=='cg'

if iteration=='auto'

iteration=length(b);

end

p=r;

w=r'\*r;

for itr=1:iteration

y=A\*p;

temp=w;

alpha=w/(p'\*y);

x=x+alpha\*p;

r=r-alpha\*y;

w=r'\*r;

if sqrt(w)<tol

break

end

beta=w/temp;

p=r+beta\*p;

end

elseif method=='sm'

if iteration=='auto'

iteration=1e5;

end

for itr=1:iteration

p=r;

y=A\*p;

w=r'\*r;

e=[e,w];

if sqrt(w)<tol

break

end

alpha=0.001;

x=x+alpha\*p;

r=r-alpha\*y;

end

else

if iteration=='auto'

iteration=1e5;

end

for itr=1:iteration

y=A\*r;

w=r'\*r;

if sqrt(w)<tol

break;

end

alpha=(r'\*r)/(r'\*y);

x=x+alpha\*r;

r=r-alpha\*y;

end

end

### CG算法验证与误差分析

in\_k=(1:1:300);

% in\_k array indicate Way\_A test, whose entry is used as test matrix rank.

k\_s=min(in\_k);

% for making data table.

tol=1e-9;

% c.f. comment in function file conjgrad.m

rank0=30;n0=300;

% rank0 and n0 indicate Way\_B test, which stand the fixed matrix rank and

% test times.

cond\_ctr=0.1;

% c.f. comment in fuction file raw\_data.m

for k=in\_k

[A,b,x0]=raw\_data(k,cond\_ctr);

[x,itr] = mat\_gradient(A,b,'cg',1e2,x0,tol);

accurate\_x=A\b;

t1(k-k\_s+1,:)=[k,itr,norm(accurate\_x-x),cond(A)];

end

% generating error data for Way\_A

vnames={'Rank','Iteration','Error','Condition\_Number'};

op\_table1=table(t1(:,1),t1(:,2),t1(:,3),t1(:,4),'VariableNames',vnames);

% generating data table for Way\_A

figure

subplot(3,1,1);

plot(t1(:,1),t1(:,3),'Color','blue','LineStyle','-','LineWidth',1.2,...

'Marker','d','MarkerFaceColor','yellow','MarkerSize',3);

title({'Caculation error in matlab V.s. rank of matix',...

'in Conjugate Gradient Method'});

xlabel('Rank of matrix');

ylabel('Error');

subplot(3,1,2);

plot(t1(:,4),t1(:,3),'LineStyle','none','LineWidth',1.2,...

'Marker','o','MarkerFaceColor','yellow','MarkerSize',9);

title({'Caculation error in matlab V.s. condition number of matix',...

'in Conjugate Gradient Method'});

xlabel('Condition Number of matrix');

ylabel('Error');

% two plots are about visulizing data in Way\_A

subplot(3,1,3);

for k=1:n0

[A,b,x0]=raw\_data(k,cond\_ctr);

[x,itr] = mat\_gradient(A,b,'cg',x0,tol);

accurate\_x=A\b;

t2(k,:)=[rank0,itr,norm(accurate\_x-x),cond(A)];

end

% generating error data for Way\_B

op\_table2=table(t2(:,1),t2(:,2),t2(:,3),t2(:,4),'VariableNames',vnames);

% generating data table for Way\_B

plot(t2(:,4),t2(:,3),'LineStyle','none','LineWidth',1.2,...

'Marker','o','MarkerFaceColor','yellow','MarkerSize',9);

str=sprintf(join(['Caculation error in matlab V.s. cond\_n of matix',...

'\n for fixed rank of matrix','\n in Conjugate Gradient Method']));

title(str);

xlabel('Condition Number of matrix(fixed rank)');

ylabel('Error');

% plot about visulizing data in Way\_B

### Result & Remark

结果图见尾注[[1]](#endnote-1)

从图中可以看出，当矩阵条件数被控制住的时候，误差被控制在比较小的范围内，基本不受矩阵随机性的影响；但是随着矩阵阶数有上升的趋势，从中间图猜测是由于条件数的增加导致的。

### SD算法验证与误差分析

clc;clear;

in\_k=(1:1:10);

% in\_k array indicate Way\_A test, whose entry is used as test matrix rank.

k\_s=min(in\_k);

% for making data table.

tol=1e-9;

% c.f. comment in function file conjgrad.m

rank0=8;n0=100;

% rank0 and n0 indicate Way\_B test, which stand the fixed matrix rank and

% test times.

cond\_ctr=0.01;

% c.f. comment in fuction file raw\_data.m

for k=in\_k

[A,b,x0]=raw\_data(k,cond\_ctr);

[x,itr] = mat\_gradient(A,b,'sd',1e2,x0,tol);

accurate\_x=A\b;

cond\_temp=cond(A);

err=norm(accurate\_x-x);

t1(k-k\_s+1,:)=[k,itr,err,cond\_temp];

end

% generating error data for Way\_A

vnames={'Rank','Iteration','Error','Condition\_Number'};

op\_table1=table(t1(:,1),t1(:,2),t1(:,3),t1(:,4),'VariableNames',vnames);

% generating data table for Way\_A

figure

subplot(3,1,1);

plot(t1(:,1),t1(:,3),'Color','blue','LineStyle','-','LineWidth',1.2,...

'Marker','d','MarkerFaceColor','yellow','MarkerSize',3);

title({'Caculation error in matlab V.s. rank of matix',...

'in Steepest Descent Method'});

xlabel('Rank of matrix');

ylabel('Error');

axis([k\_s k\_s+length(in\_k)-1 0 inf]);

subplot(3,1,2);

plot(t1(:,4),t1(:,3),'LineStyle','none','LineWidth',1.2,...

'Marker','o','MarkerFaceColor','yellow','MarkerSize',9);

title({'Caculation error in matlab V.s. condition number of matix',...

'in Steepest Descent Method'});

xlabel('Condition Number of matrix');

ylabel('Error');

% two plots are about visulizing data in Way\_A

subplot(3,1,3);

for k=1:n0

[A,b,x0]=raw\_data(k,cond\_ctr);

[x,itr] = mat\_gradient(A,b,'sd',1e2,x0,tol);

accurate\_x=A\b;

cond\_temp=cond(A);

err=norm(accurate\_x-x);

if err>1e2

si\_count=si\_count+1;

coll{si\_count}={A,b,x0,cond\_temp};

end

t2(k-k\_s+1,:)=[k,itr,err,cond\_temp];

end

% generating error data for Way\_B

op\_table2=table(t2(:,1),t2(:,2),t2(:,3),t2(:,4),'VariableNames',vnames);

% generating data table for Way\_B

plot(t2(:,4),t2(:,3),'LineStyle','none','LineWidth',1.2,...

'Marker','o','MarkerFaceColor','yellow','MarkerSize',9);

str=sprintf(join(['Caculation error in matlab V.s. cond\_n of matix',...

'\n for fixed rank of matrix','\n in Steepest Descent Method']));

title(str);

xlabel(['Condition Number of matrix(fixed rank ',num2str(rank0),')']);

ylabel('Error');

% plot about visulizing data in Way\_B

### Result & Remark

运行结果见尾注[[2]](#endnote-2)

这次演示只给出矩阵阶数较小的情况，当矩阵阶数增加，耗时增加，稳定性也得到损失（误差加大）

## 预优化Conjugate Gradient Method

function [x, k] = cgp(x0,A,b,C,mit,tol)

% Synopsis:

% x0: initial point

% A: Matrix A of the system Ax=b

% C: Preconditioning Matrix can be left or right

% mit: Maximum number of iterations

% tol: residue norm tolerance

% x: Estimated solution point

% k: Number of iterations done

%

r=b-A\*x0;

tir=C\r;

p=tir;

x=x0;

for k=(1:mit)

d=A\*p;

alpha=(tir'\*r)/(d'\*p);

x=x+alpha\*p;

temp\_r=r;

temp\_tir=tir;

if norm(r)<tol

break;

end

r=r-alpha\*d;

tir=C\r;

beta=(r'\*tir)/(temp\_r'\*temp\_tir);

p=tir+beta\*p;

end

end

### 预优化CG算法验证与误差分析

clear;

clc;

cond\_ctr='none';

mit=1e4;

tol=1e-7;

c\_list=(0.1:0.2:0.8);

count=1;

for cond\_ctr=c\_list

for j=1:400

k=40;

[A,b,x0]=raw\_data(k,cond\_ctr);

C(:,:,2)=diag(diag(A));

C(:,:,1)=eye(k,k);

for i=1:2

[~,itr\_pcg(j,i)]=cgp(x0,A,b,C(:,:,i),mit,tol);

end

itr\_pcg;

end

%subplot(length(c\_list),1,count)

figure

count=count+1;

p=plot(itr\_pcg);

p(1).LineWidth = 2;

p(2).Marker = '\*';

legend('CG','Jacobi PCG')

title('iterations for ''CG'' and ''PCG''')

ylabel('iterations')

xlabel('Fixed matrix rank 40')

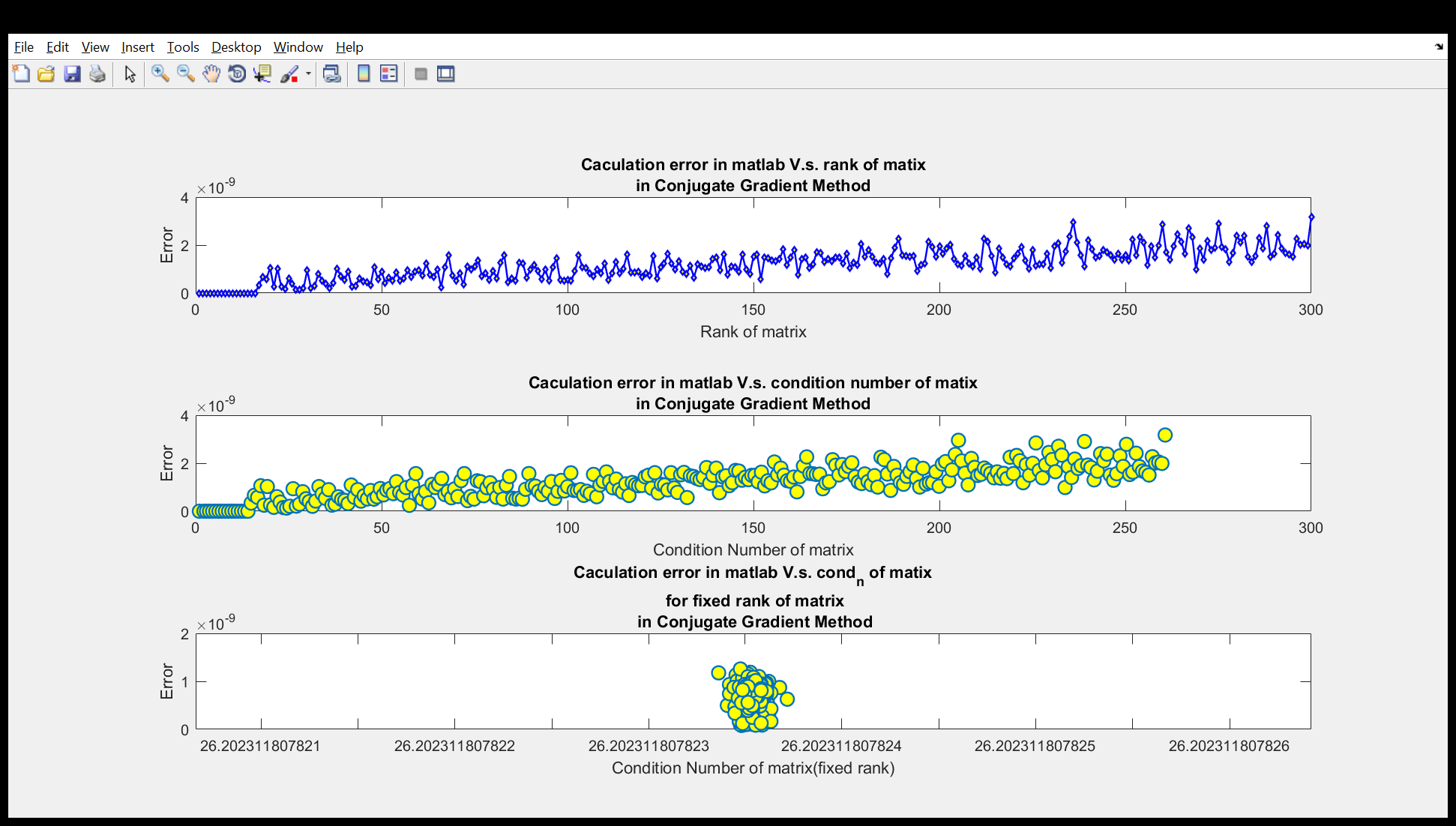
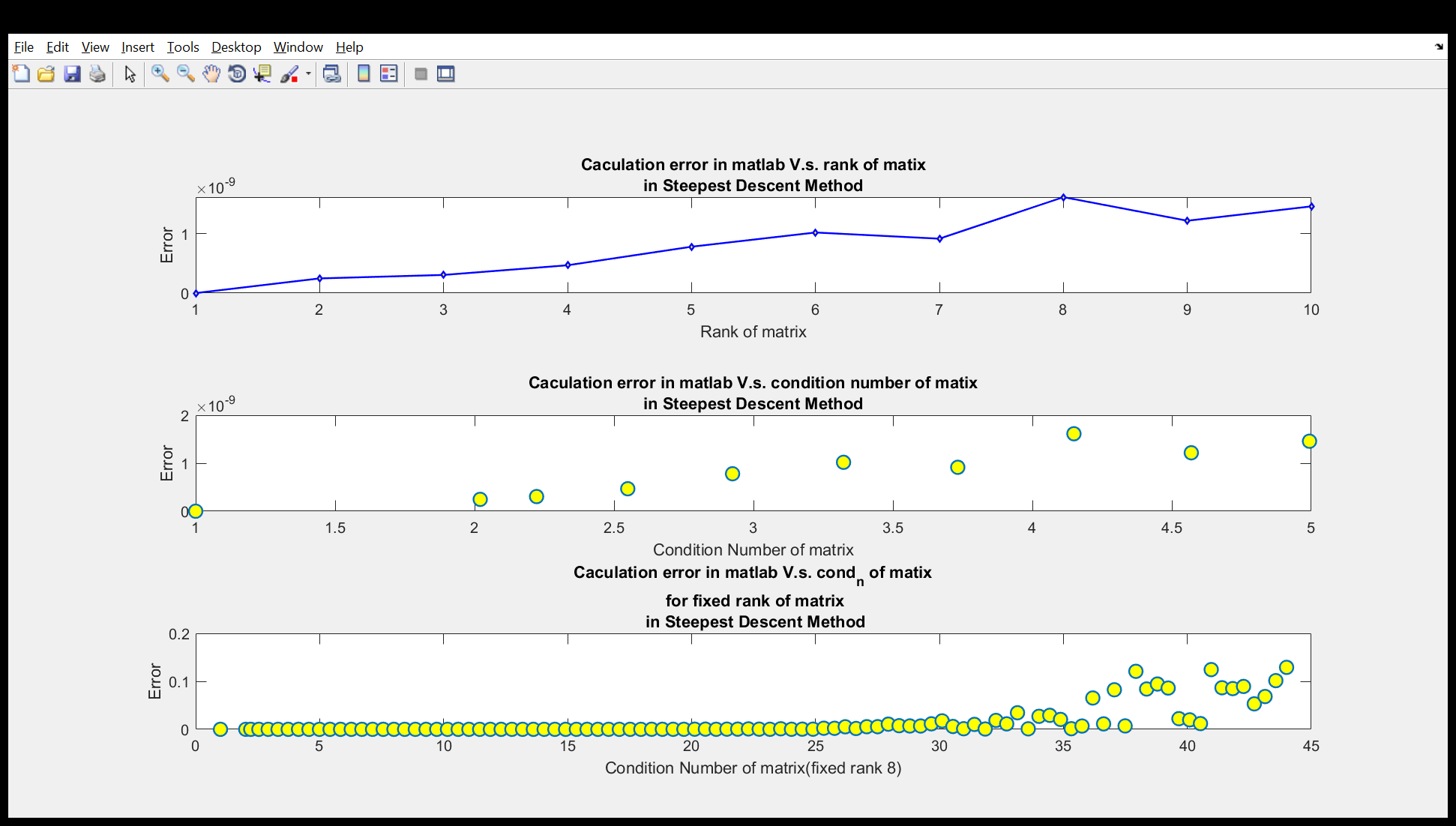
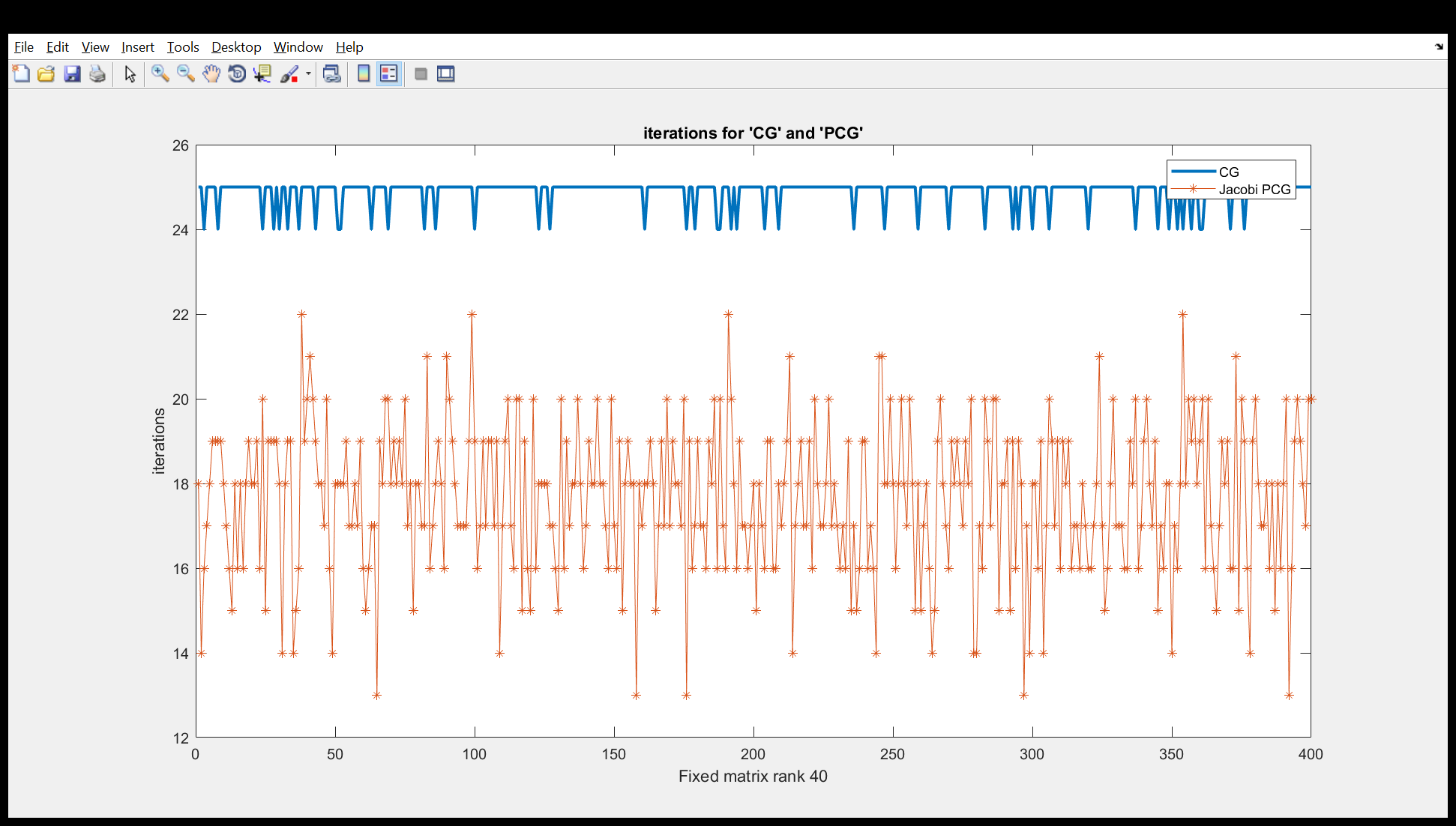
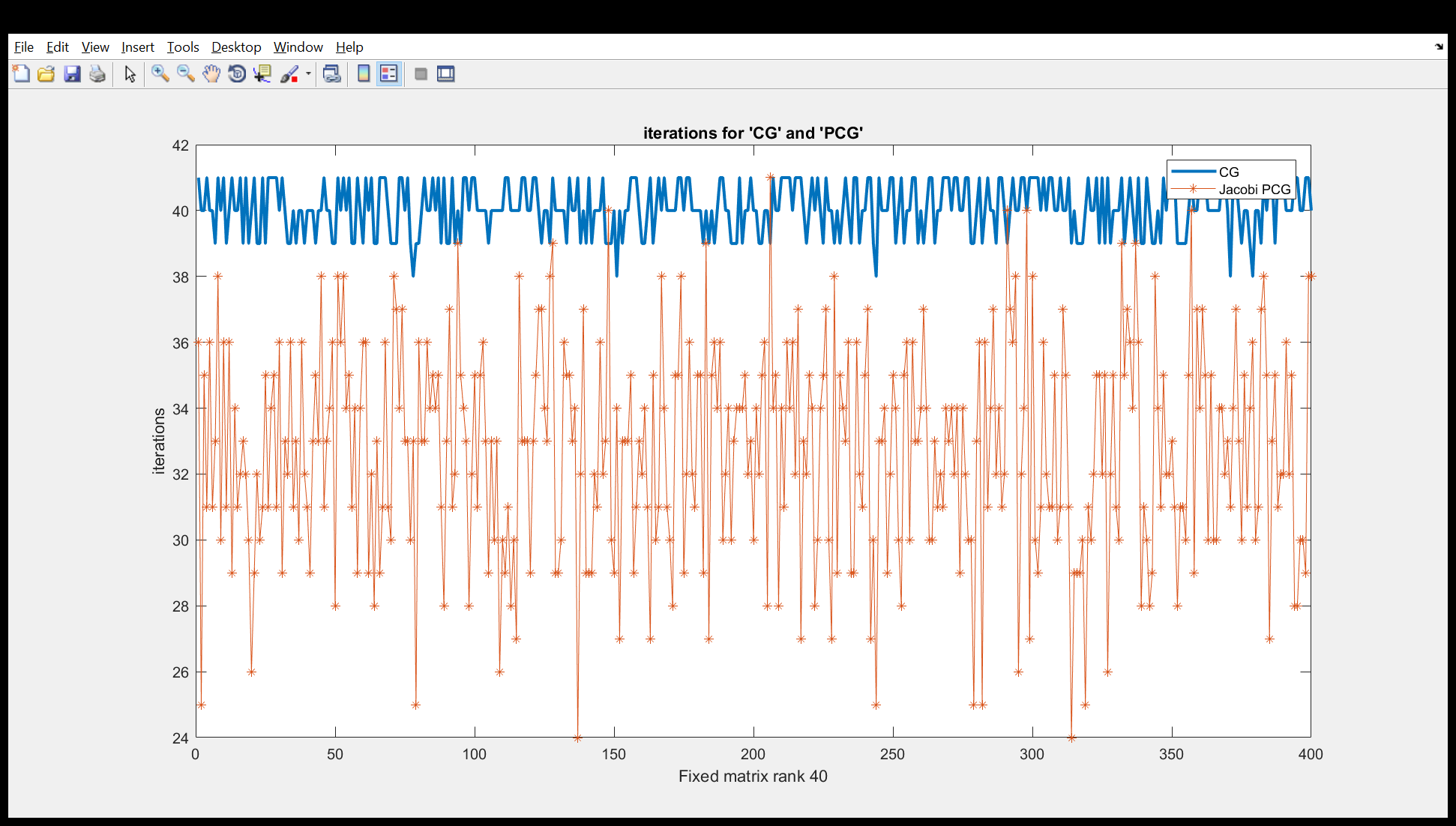
end

### Result & Remark

本次演示共有四张图（从程序设计中可以看出），选取其中三幅进行展示每张图由cond\_ctr变量控制；下面以尾注形式给出：

cond\_ctr=0.1[[3]](#endnote-3)

cond\_ctr=0.7[[4]](#endnote-4)

1.  [↑](#endnote-ref-1)
2.  [↑](#endnote-ref-2)
3.  [↑](#endnote-ref-3)
4.  [↑](#endnote-ref-4)