Linghao Shi

① (a) $Ax=b$: $\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$

$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\quad r_0 = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$ $\quad d_0 = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$

$x_1(4x_1 + 2x_2) + x_2(2x_1 + 4x_2)$

$= 4x_1^2 + 4x_1x_2 + 4x_2^2 \geq 0$

when $x^TAx = 0$, $x_1 + x_2 = 0$, $x_2 = 0 = x_1$

$x^TAx = 0$ iff $\vec{x} = 0$

$\therefore$ A is SPD

$k=0$: $\alpha = \dfrac{r_0^T r_0}{d_0^T A d_0} = \dfrac{\begin{bmatrix} 8 \\ 10 \end{bmatrix}^T \begin{bmatrix} 8 \\ 10 \end{bmatrix}}{\begin{bmatrix} 8 \\ 10 \end{bmatrix}^T \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} 8 \\ 10 \end{bmatrix}} = \dfrac{41}{244}$

$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{41}{244}\begin{bmatrix} 8 \\ 10 \end{bmatrix} = \begin{bmatrix} 1.34 \\ 1.68 \end{bmatrix}$

$r_1 = \begin{bmatrix} 8 \\ 10 \end{bmatrix} - \frac{41}{244}\begin{bmatrix} 52 \\ 56 \end{bmatrix} = \begin{bmatrix} -0.74 \\ 0.59 \end{bmatrix}$

$\beta_0 \approx 0.0054$, $d_1 = r_1 + \beta_0 d_0 \approx \begin{bmatrix} -0.6942 \\ 0.6446 \end{bmatrix}$

$k=1$: $\alpha_1 = \dfrac{r_1^T r_1}{d_1^T A d_1} \approx 0.4959$

$x_2 = \begin{bmatrix} 1.34 \\ 1.68 \end{bmatrix} + 0.4959\begin{bmatrix} -0.6942 \\ 0.6446 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$r_2 = \begin{bmatrix} -0.74 \\ 0.59 \end{bmatrix} - 0.4959\begin{bmatrix} -1.4875 \\ 1.19 \end{bmatrix} \approx 1.0 \times 10^{-15}\begin{bmatrix} 0.111 \\ -0.111 \end{bmatrix}$

$\therefore x = x_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

(b)

$$Ax=b: \begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix}$$

$$x^TAx = 4x_1^2 + 2x_1x_2 + 2x_1x_2 + 5x_2^2 + 2x_2x_3 + 2x_3^2$$

$$= (2x_1+x_2)^2 + (x_2+x_3)^2 + 3x_2^2 + x_3^2 \geq 0$$

when $x^TAx=0$, $x_1=0, x_2=0, x_3=0$

$x^TAx=0$ iff $\vec{x}=0$ $\quad\therefore$ A is SPD

$k=0$:

$$\alpha_0 = \frac{\begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix}^T \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix}}{\begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 2 & 6 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix}} = \frac{353}{2229}$$

$$x_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \frac{353}{2229} \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix} \approx \begin{bmatrix} 1.2669 \\ 2.3755 \\ 1.2669 \end{bmatrix}$$

$$r_1 = \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix} - \frac{353}{2229} \begin{bmatrix} 62 \\ 99 \\ 31 \end{bmatrix} \approx \begin{bmatrix} -1.8188 \\ -0.6783 \\ 3.0906 \end{bmatrix}$$

$$\beta_0 = \frac{\begin{bmatrix} -1.8188 \\ -0.6783 \\ 3.0906 \end{bmatrix}^T \begin{bmatrix} -1.8188 \\ -0.6783 \\ 3.0906 \end{bmatrix}}{\begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix}^T \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix}} \approx 0.0377$$

$$d_1 = \begin{bmatrix} -1.8188 \\ -0.6783 \\ 3.0906 \end{bmatrix} + 0.0377 \begin{bmatrix} 8 \\ 15 \\ 8 \end{bmatrix} \approx \begin{bmatrix} -1.5169 \\ -0.1123 \\ 3.3925 \end{bmatrix}$$

$k=1$:

$$\alpha_1 = \frac{r_1^Tr_1}{d_1^TAd_1} = 0.4136$$

$$x_2 = x_1 + \alpha_1 d_1 = \begin{bmatrix} -1.8188 \\ -0.6783 \\ 3.0906 \end{bmatrix} + 0.4136 \begin{bmatrix} -1.5169 \\ -0.1123 \\ 3.3925 \end{bmatrix} \approx \begin{bmatrix} 0.6395 \\ 2.3290 \\ 2.6701 \end{bmatrix}$$

$$r_2 = \begin{bmatrix} -1.8188 \\ -0.6783 \\ 3.0906 \end{bmatrix} - 0.4136 \begin{bmatrix} -6.2922 \\ -0.2029 \\ 6.6727 \end{bmatrix} \approx \begin{bmatrix} 0.7838 \\ -0.5944 \\ 0.3308 \end{bmatrix}$$

$$\beta_1 = \frac{r_2^Tr_2}{r_1^Tr_1} \approx 0.0809$$

$$d_2 = \begin{bmatrix} 0.7838 \\ -0.5944 \\ 0.3308 \end{bmatrix} + 0.0809 \begin{bmatrix} -1.5169 \\ -0.1123 \\ 3.3925 \end{bmatrix} \approx \begin{bmatrix} 0.6611 \\ -0.6035 \\ 0.6051 \end{bmatrix}$$

$k=2$

$$\alpha_2 = \frac{r_2^Tr_2}{d_2^TAd_2} \approx 0.5452$$

$$x_3 = x_2 + \alpha_2 d_2 \approx \begin{bmatrix} 1.000 \\ 2.000 \\ 3.000 \end{bmatrix}$$

$$r_3 = r_2 - \alpha_2(Ad_2) \approx 1.0 \times 10^{-15} \cdot \begin{bmatrix} 0.4441 \\ 0.6661 \\ 0.1110 \end{bmatrix} \approx \vec{0}$$

$$x = x_3 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

③ (a) (i) $f(x) = \begin{bmatrix} x_1^3 + 2x_2^2 - 1 \\ x_1 + 3x_2^4 - 3 \end{bmatrix}$, $\quad x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$f(\vec{x_0}) + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \cdot \Delta x + O(\|\Delta x\|^2)$$

$$= \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 3x_1^2 & 4x_2 \\ x_1 & 12x_2^3 \end{bmatrix} \cdot \Delta\vec{x}$$

(ii) $f(x) = \begin{bmatrix} \cos(x_1^2 + x_2^2) \\ e^{x_2 x_1} + x_1 x_2^2 \end{bmatrix}$, $\quad x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -\sin(x_1^2 + x_2^2) \cdot 2x_1, & -\sin(x_1^2 + x_2^2) \cdot 2x_2 \\ e^{x_2 x_1} \cdot x_2 + x_2^2, & e^{x_2 x_1} \cdot x_1 + 2x_1 x_2 \end{bmatrix} \cdot \Delta\vec{x}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \Delta\vec{x}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

(b) (i) $f(x) = \begin{bmatrix} x_1^4 + x_2^2 - 2 \\ x_1^3 - x_2 + 1 \end{bmatrix} = 0$

$$\vec{x_1} = \vec{x_0} - \begin{bmatrix} 4x_1^3 & 2x_1 \\ 3x_1^2 & -1 \end{bmatrix}^{-1} \cdot f(\vec{x_0})$$

where $\vec{x_0} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

(ii) $f(x) = \begin{bmatrix} e^{x_1 x_2} - x_2^3 x_1 + 2 \\ \sin(x_1 x_2) + \cos(x_3) \\ x_1 x_2 x_3 - x_1^2 \end{bmatrix} = 0$

$$\vec{x_1} = \vec{x_0} - \begin{bmatrix} x_2 e^{x_1 x_2} - x_2^3 & -3x_2^2 x_1 & 0 \\ x_2 \cdot \cos(x_1 x_2) & \cos(x_1 x_2)\, x_1 & -\sin(x_3) \\ x_2 x_3 - 2x_1 & x_1 x_3 & x_1 x_2 \end{bmatrix} f(\vec{x_0})$$

where $\vec{x_0} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

(C) $f(x) = \begin{bmatrix} u^2 + 4v^2 - 4 \\ 4u^2 + v^2 - 4 \end{bmatrix} = 0$

Jacobian $\begin{bmatrix} 2u & 8v \\ 8u & 2v \end{bmatrix}$ initial guess $x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$k=1$: $\begin{bmatrix} 2 & 8 \\ 8 & 2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\vec{y} = [-0.1, -0.1]^T$

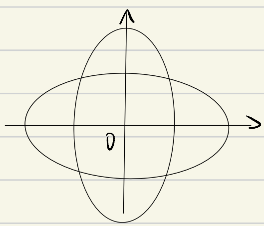
$\vec{x_1} = \vec{x_0} + \vec{y} = [0.9, 0.9]^T$

$k=2$: $\begin{bmatrix} 1.8 & 7.2 \\ 7.2 & 1.8 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = -\begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix}$

$\vec{y} = [-0.006, -0.006]^T$

$\vec{x_2} = \vec{x_1} + \vec{y} = [0.894, 0.894]^T$

Consider the symmetry, then we will left with four solution



$\begin{bmatrix} -0.894 \\ 0.894 \end{bmatrix}$ $\begin{bmatrix} 0.894 \\ -0.894 \end{bmatrix}$

$\begin{bmatrix} 0.894 \\ 0.894 \end{bmatrix}$ $\begin{bmatrix} 0.894 \\ -0.894 \end{bmatrix}$

Question 2

Here is the code for input:

```
A = zeros(100,100);
n = 100;
xe = zeros(100,1);
for i = 1:n
    for j = 1:n
        if i == j
            A(i,j) = 1 + j;
        elseif i == j - 1
            A(i,j) = -1;
        elseif i == j + 1
            A(i,j) = -1;
        end
    end
    xe(i,1)=n-i+1;
end
b = A *xe;
x0=zeros(100,1);
[x,nit] = conjugateGradient(A,b,x0,100,10^-5);

RFE = norm(xe-x, "inf")/norm(x,"inf");
fprintf('CG: n=%d, rtol=%8.2e, nit=%d max-RFE = %8.2e \n',n,rtol,nit,RFE);
```

Here is the code for my solve function:

```
function [x,nit] = conjugateGradient( A,b, x0, maxIterations, rtol )
    delta = norm(b - A * x0, 2)/norm(b, 2);
    res = b - A * x0;
    direc= res;
    k = 1;
    while (k < maxIterations) && (delta > rtol)
        w = A * direc;
        r = res' * res;
        prevdelta = delta;
        steplen = r/(direc' * w);
        x1 = steplen * direc + x0;
        res = res - steplen * w;
        newRes = (res' * res) / r;
        direc = newRes * direc + res;
        delta = norm(b - A * x1, 2) / norm(b, 2);
        ratio = delta / prevdelta;
        fprintf('CG: k=%3d, delta = %8.2e, ratio=%8.2e\n',k,delta,ratio);
        x0 = x1;
        k = k + 1;
    end
    x = x1;
    nit = k;
end
```

Here is the output:

```
CG: k=  1, delta = 3.78e-01, ratio=3.78e-01
CG: k=  2, delta = 1.93e-01, ratio=5.09e-01
CG: k=  3, delta = 1.14e-01, ratio=5.93e-01
CG: k=  4, delta = 7.45e-02, ratio=6.52e-01
CG: k=  5, delta = 5.20e-02, ratio=6.98e-01
CG: k=  6, delta = 3.82e-02, ratio=7.35e-01
CG: k=  7, delta = 2.93e-02, ratio=7.68e-01
CG: k=  8, delta = 2.34e-02, ratio=7.98e-01
CG: k=  9, delta = 1.93e-02, ratio=8.25e-01
CG: k= 10, delta = 1.64e-02, ratio=8.48e-01
CG: k= 11, delta = 1.42e-02, ratio=8.65e-01
CG: k= 12, delta = 1.24e-02, ratio=8.73e-01
CG: k= 13, delta = 1.08e-02, ratio=8.72e-01
CG: k= 14, delta = 9.27e-03, ratio=8.61e-01
CG: k= 15, delta = 7.83e-03, ratio=8.44e-01
CG: k= 16, delta = 6.45e-03, ratio=8.24e-01
CG: k= 17, delta = 5.20e-03, ratio=8.06e-01
CG: k= 18, delta = 4.10e-03, ratio=7.89e-01
CG: k= 19, delta = 3.19e-03, ratio=7.77e-01
CG: k= 20, delta = 2.45e-03, ratio=7.68e-01
CG: k= 21, delta = 1.86e-03, ratio=7.61e-01
CG: k= 22, delta = 1.41e-03, ratio=7.56e-01
CG: k= 23, delta = 1.06e-03, ratio=7.50e-01
CG: k= 24, delta = 7.85e-04, ratio=7.43e-01
CG: k= 25, delta = 5.77e-04, ratio=7.35e-01
CG: k= 26, delta = 4.19e-04, ratio=7.26e-01
CG: k= 27, delta = 3.00e-04, ratio=7.16e-01
CG: k= 28, delta = 2.12e-04, ratio=7.06e-01
CG: k= 29, delta = 1.48e-04, ratio=6.97e-01
CG: k= 30, delta = 1.02e-04, ratio=6.89e-01
CG: k= 31, delta = 6.93e-05, ratio=6.81e-01
CG: k= 32, delta = 4.68e-05, ratio=6.75e-01
CG: k= 33, delta = 3.13e-05, ratio=6.69e-01
CG: k= 34, delta = 2.07e-05, ratio=6.63e-01
CG: k= 35, delta = 1.36e-05, ratio=6.58e-01
CG: k= 36, delta = 8.89e-06, ratio=6.52e-01
CG: n=100, rtol=1.00e-05, nit=37 max-RFE = 6.29e-05
```

4.

Here is the function:

```
function xc = solveSystemByNewton( f,fx,x0,tol )
    xc =x0;
    fc = f(xc);
    k = 1;
    while tol <= norm(fc,inf)
        xc = xc - fx(xc) ^ (-1) * fc;
        fc = f(xc);
        n = size(xc);
        fprintf('solveSystemByNewton: it=%d: x=[%12.6e', k, xc(1));
        for j=2:n
            fprintf(',%12.6e',xc(j));
        end
        fprintf('] ||f(x)||_inf=%8.2e\n', norm(fc,inf));
        k = k+1;
    end
return
```

Here is the input code and output result:

```
>> f = @(x)[(x(1)^2+x(2)^2-1);((x(1)-1)^2+x(2)^2-1)];
fx = @(x)[2*x(1), 2*x(2); (2*x(1)-2), 2*x(2)];
tol = 10^(-10);

x0 = [1, 1]';
xc = solveSystemByNewton(f,fx,x0,tol)

x0 = [-1, -1]';
xc = solveSystemByNewton(f,fx,x0,tol)
solveSystemByNewton: it=1: x=[5.000000e-01,1.000000e+00] ||f(x)||_inf=2.50e-01
solveSystemByNewton: it=2: x=[5.000000e-01,8.750000e-01] ||f(x)||_inf=1.56e-02
solveSystemByNewton: it=3: x=[5.000000e-01,8.660714e-01] ||f(x)||_inf=7.97e-05
solveSystemByNewton: it=4: x=[5.000000e-01,8.660254e-01] ||f(x)||_inf=2.12e-09
solveSystemByNewton: it=5: x=[5.000000e-01,8.660254e-01] ||f(x)||_inf=1.11e-16

xc =

          0.5
      0.86603

solveSystemByNewton: it=1: x=[5.000000e-01,-2.000000e+00] ||f(x)||_inf=3.25e+00
solveSystemByNewton: it=2: x=[5.000000e-01,-1.187500e+00] ||f(x)||_inf=6.60e-01
solveSystemByNewton: it=3: x=[5.000000e-01,-9.095395e-01] ||f(x)||_inf=7.73e-02
solveSystemByNewton: it=4: x=[5.000000e-01,-8.670663e-01] ||f(x)||_inf=1.80e-03
solveSystemByNewton: it=5: x=[5.000000e-01,-8.660260e-01] ||f(x)||_inf=1.08e-06
solveSystemByNewton: it=6: x=[5.000000e-01,-8.660254e-01] ||f(x)||_inf=3.90e-13

xc =

          0.5
     -0.86603
```

5.

Here is the input:

```
n = 500;
A = diag(sqrt(1:n)) + diag(cos(1:(n-10)),10) + diag(cos(1:(n-10)),-10);
xe = ones(n,1);
b = A*xe;
x0 = zeros(n,1);
maxIterations = 40;
rtol = 10^(-15);

% condition 1
M = eye(n,n);
fprintf('\nNo preconditioner:\n');
[x,nit1,rbe1] = preconditionedConjugateGradient( M, A,b, x0, maxIterations, rtol);

% condition 2
M = diag(diag(A));
fprintf('\nThe diagonal (Jacobi) preconditioner:\n');
[x,nit2,rbe2] = preconditionedConjugateGradient( M, A,b, x0, maxIterations, rtol);

% condition 3
D = diag(diag(A));
M = (tril(A,-1) + D)*D^(-1)*(D + triu(A, 1));
fprintf('\nThe Gauss-Seidel preconditioner:\n');
[x,nit3,rbe3] = preconditionedConjugateGradient( M, A,b, x0, maxIterations, rtol);

xlabel("iterations");
ylabel("rbe");
legend("no preconditioner","Jacobi","Gauss-Seidel")
```

Here is the function:

```
function [x,nit,rbe] = preconditionedConjugateGradient( M, A,b, x0, ...
    maxIterations, rtol)
    res = b - A * x0;
    direc = M ^ (-1) * res;
    cond = direc;
    delta = norm(b - A * x0, 2)/norm(b, 2);
    k = 1;
    while k <= maxIterations && delta >= rtol
        w = A * direc;
        rz = res' * cond;
        prevdelta = delta;
        steplen = rz /(direc' * w);
        res = res - steplen * w;
        cond = M ^ (-1) * res;
        b0 = (res' * cond) / rz;
        x1 = steplen * direc + x0;
        direc = b0 * direc + cond;
        delta = norm(b - A * x1, 2)/norm(b, 2);
        fprintf('PCG: k=%3d, delta = %8.2e, ratio=%8.2e\n',k,delta,delta/prevdelta);
        x0 = x1;
        rbe(k) = delta;
        k = k + 1;
    end
x = x1;
nit = k;
return
```

Here is the output for three different preconditioners:

```
No preconditioner:
PCG: k=  1, delta = 2.05e-01, ratio=2.05e-01
PCG: k=  2, delta = 7.17e-02, ratio=3.49e-01
PCG: k=  3, delta = 3.23e-02, ratio=4.51e-01
PCG: k=  4, delta = 1.71e-02, ratio=5.30e-01
PCG: k=  5, delta = 1.02e-02, ratio=5.98e-01
PCG: k=  6, delta = 6.71e-03, ratio=6.55e-01
PCG: k=  7, delta = 4.61e-03, ratio=6.87e-01
PCG: k=  8, delta = 3.18e-03, ratio=6.90e-01
PCG: k=  9, delta = 2.17e-03, ratio=6.81e-01
PCG: k= 10, delta = 1.47e-03, ratio=6.76e-01
PCG: k= 11, delta = 9.84e-04, ratio=6.72e-01
PCG: k= 12, delta = 6.55e-04, ratio=6.66e-01
PCG: k= 13, delta = 4.39e-04, ratio=6.69e-01
PCG: k= 14, delta = 3.05e-04, ratio=6.96e-01
PCG: k= 15, delta = 2.25e-04, ratio=7.36e-01
PCG: k= 16, delta = 1.66e-04, ratio=7.39e-01
PCG: k= 17, delta = 1.16e-04, ratio=7.01e-01
PCG: k= 18, delta = 7.95e-05, ratio=6.84e-01
PCG: k= 19, delta = 5.45e-05, ratio=6.85e-01
PCG: k= 20, delta = 3.70e-05, ratio=6.80e-01
PCG: k= 21, delta = 2.49e-05, ratio=6.71e-01
PCG: k= 22, delta = 1.56e-05, ratio=6.26e-01
PCG: k= 23, delta = 9.04e-06, ratio=5.81e-01
PCG: k= 24, delta = 5.15e-06, ratio=5.69e-01
PCG: k= 25, delta = 2.82e-06, ratio=5.49e-01
PCG: k= 26, delta = 1.68e-06, ratio=5.96e-01
PCG: k= 27, delta = 1.07e-06, ratio=6.36e-01
PCG: k= 28, delta = 6.41e-07, ratio=5.98e-01
PCG: k= 29, delta = 3.97e-07, ratio=6.19e-01
PCG: k= 30, delta = 2.46e-07, ratio=6.21e-01
PCG: k= 31, delta = 1.84e-07, ratio=7.45e-01
PCG: k= 32, delta = 1.31e-07, ratio=7.14e-01
PCG: k= 33, delta = 8.26e-08, ratio=6.29e-01
PCG: k= 34, delta = 4.66e-08, ratio=5.64e-01
PCG: k= 35, delta = 2.51e-08, ratio=5.39e-01
PCG: k= 36, delta = 1.49e-08, ratio=5.91e-01
PCG: k= 37, delta = 8.15e-09, ratio=5.49e-01
PCG: k= 38, delta = 5.01e-09, ratio=6.15e-01
PCG: k= 39, delta = 2.74e-09, ratio=5.46e-01
PCG: k= 40, delta = 1.57e-09, ratio=5.75e-01
```

```
The diagonal (Jacobi) preconditioner:
PCG: k=  1, delta = 2.66e-02, ratio=2.66e-02
PCG: k=  2, delta = 3.36e-03, ratio=1.26e-01
PCG: k=  3, delta = 8.10e-04, ratio=2.41e-01
PCG: k=  4, delta = 2.40e-04, ratio=2.97e-01
PCG: k=  5, delta = 6.04e-05, ratio=2.52e-01
PCG: k=  6, delta = 1.61e-05, ratio=2.66e-01
PCG: k=  7, delta = 3.21e-06, ratio=1.99e-01
PCG: k=  8, delta = 8.40e-07, ratio=2.62e-01
PCG: k=  9, delta = 1.77e-07, ratio=2.11e-01
PCG: k= 10, delta = 3.13e-08, ratio=1.77e-01
PCG: k= 11, delta = 5.70e-09, ratio=1.82e-01
PCG: k= 12, delta = 1.31e-09, ratio=2.29e-01
PCG: k= 13, delta = 2.22e-10, ratio=1.69e-01
PCG: k= 14, delta = 3.63e-11, ratio=1.64e-01
PCG: k= 15, delta = 5.70e-12, ratio=1.57e-01
PCG: k= 16, delta = 1.03e-12, ratio=1.80e-01
PCG: k= 17, delta = 1.51e-13, ratio=1.46e-01
PCG: k= 18, delta = 2.19e-14, ratio=1.45e-01
PCG: k= 19, delta = 3.49e-15, ratio=1.60e-01
PCG: k= 20, delta = 6.20e-16, ratio=1.78e-01
```

```
The Gauss-Seidel preconditioner:
PCG: k=  1, delta = 3.52e-03, ratio=3.52e-03
PCG: k=  2, delta = 2.05e-04, ratio=5.83e-02
PCG: k=  3, delta = 1.13e-05, ratio=5.52e-02
PCG: k=  4, delta = 4.69e-07, ratio=4.14e-02
PCG: k=  5, delta = 1.13e-08, ratio=2.40e-02
PCG: k=  6, delta = 3.11e-10, ratio=2.76e-02
PCG: k=  7, delta = 8.47e-12, ratio=2.72e-02
PCG: k=  8, delta = 1.80e-13, ratio=2.13e-02
PCG: k=  9, delta = 3.91e-15, ratio=2.18e-02
PCG: k= 10, delta = 2.11e-16, ratio=5.38e-02
```

Here is the graph: