Numerical Analysis (089180) A.Y. 2022-2023
prof. Simona Perotto Luca Liverotti
simona.perotto@polimi.it luca.liverotti@polimi.it

Lab 7 – Homework Solutions

November 25, 2022

A Homework

Homework 7.1. Consider the Hilbert matrix $\mathbb{H}_n$ of order n

$$\mathbb{H}_n = \begin{bmatrix} 1 & 1/2 & \cdots & 1/n \\ 1/2 & 1/3 & \cdots & 1/(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/(n+1) & \cdots & 1/(2n-1) \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad n = 10, 20, 40.$$

a. Estimate $\mathcal{K}_2(\mathbb{H}_n)$ by choosing the rigth hand side term $\boldsymbol{b}$ of $\mathbb{H}_n\boldsymbol{x} = \boldsymbol{b}$ such that $x = ones(n, 1)$ and a perturbation of the right hand side $\boldsymbol{\delta b}$ randomly generated via MATLAB command `randn` and then multiplied by the scalar number $\alpha = 1e - 6$

b. Compute $\mathcal{K}_2(\mathbb{H}_n)$ using the MATLAB command `cond`.

c. What is the maximum value for n such that the relative error on the solution is equal to 1?

Solution Homework 7.1.

hw_7_1.m

```
clc
clear all
close all

% We can invert the relation
% norm(deltax)/norm(x) <= Kp(A) * norm(deltab)/norm(b)
% to get an estimate for Kp(A):
% Kp(A) >= norm(deltax)/norm(x) * norm(b)/norm(deltab)
% Now we use this estimate (Remenber that if we do not indicate value inside
% the command norm we get the 2-norm)

for n = [10,20 40]
   pert_mag = 1e-6; % Perturbation magnitude
   n_trials = 100;
   H = hilb(n);
   x = ones(n, 1);
   b = H*x;
   KH_est = 0 ;
   for ii = 1:n_trials
     deltab = randn(n, 1) * pert_mag;
     deltax = H \ deltab;
     KH_est_ii = norm(deltax)/norm(x) * norm(b)/norm(deltab);
     KH_est = max(KH_est_ii, KH_est);
   end
   KH_est      % estimation of the conditioning numer
   cond( hilb(n) ) % calculation of the conditioning number
   pause
end
```

```
% The estimates obtained at the point a) are lower. This is expected, since
% the condition number represents the maximum
% possible amplification of the relative error on data. With the first
% approach we might not have considered the worst condition,
% hence the estimate is lower than the result with cond command.

n = [1:20];

for (ii = n)
 H = hilb(ii);
 x_ex = ones(ii, 1);
 b = H*x_ex;
 x = H\b;
 KH(ii) = cond(H);
 err(ii) = norm(x - x_ex) / norm(x);
end

figure
semilogy(n, KH, 'xb-', 'LineWidth',2, 'MarkerSize',8)
hold on
semilogy(n, err, 'xr-', 'LineWidth',2, 'MarkerSize',8)
semilogy(n, ones(n(end),1), 'k--', 'LineWidth',2, 'MarkerSize',8)
set(gca, 'FontSize',16)
set(gca, 'LineWidth', 1.5)
axis([0 n(end)+1 1e-17 1e20])
xlabel('n')
ylabel('K_2(H_n) and err')
legend('K_2(H_n)', 'relative error', 'Location','SouthEast')


% For n > 12 the relative error is 1. This means that the solution is completely meaningless.
```

Homework 7.2. Consider the following matrix

$$\mathbb{A} = \begin{bmatrix} 1.001 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1.001 & 1 & 1 \\ 1 & 1.001 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1.001 \end{bmatrix}.$$

a. Compute the condition number $\mathcal{K}_\infty(\mathbb{A})$ using the MATLAB command `cond`

b. Then compute $\mathcal{K}_\infty(\mathbb{A})$ using the explicit definition of the conditioning number

c. Finally estimate $\mathcal{K}_\infty(\mathbb{A})$ by introducing a perturbation of the right hand side $\boldsymbol{\delta b}$ randomly generated via MATLAB command `randn` and then multiplied by the scalar number $\alpha = 1e - 6$

Solution Homework 7.2.

hw_7_2.m

```
clc
clear all
close all

% Definition of the matrix A
A = ones(5,5);
A(1,1) = 1.001;
A(3,2) = 1.001;
A(2,3) = 1.001;
A(5,5) = 1.001;
% visualization of the matrix A
A
% Calculation of the conditioning number in inf norm with Matlab
KA = cond(A, inf)
%calculation of the conditioning number wih the definition
KA_def = norm(A, inf) * norm(inv(A), inf)
```

```
% Estimation of the conditioning number by inverting the relation
% norm(deltax)/norm(x) <= Kp(A) * norm(deltab)/norm(b)
% to get an estimate for Kp(A):
% Kp(A) >= norm(deltax)/norm(x) * norm(b)/norm(deltab)
n_trials = 100;
KA_store = zeros(n_trials,1);
pert_mag = 1e-6; % Perturbation magnitude
x = ones(5,1);
b = A*x;

for (i=1:n_trials)
  deltab = pert_mag * randn(5,1);
  deltax = A\deltab;
  KA = norm(deltax,inf)/norm(x,inf) * norm(b,inf)/norm(deltab,inf);
  KA_store(i) = KA;
end
KA_est = max(KA_store)

figure
set(gcf, 'Position', [100 100 1000 500])
plot([1:n_trials], KA_store, 'bx-','LineWidth',2, 'MarkerSize',8)
hold on, box on
plot([1:n_trials], cond(A,inf)*ones(n_trials,1), 'r--', 'LineWidth',2)
axis([0 n_trials+1 0 1.1*cond(A,inf)])
set(gca,'LineWidth',1.5)
set(gca,'FontSize',16)
xlabel('trials')
ylabel('K_{est}(A)')
```

Homework 7.3. Consider a linear system whose coefficient matrix is obtained with the commands » `B = rand(5) + diag(10*ones(5,1));` » `A = B*B';`

a. Analyze the convergence of the Jacobi and of the Gauss-Seidel methods.

b. Check the results using assuming the exact solution $xex = [1, 1, 1, 1, 1]^T$ and setting $x_0 = [0, 0, 0, 0, 0]^T$, tol=1.e-12, maxit=100

Solution Homework 7.3. Not provided.

Homework 7.4. Consider the system $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$ with

$$\mathbb{A} = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

a. Study the convergence of the Jacobi and of the Gauss-Seidel methods when applied to the system $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$.

b. Compute the spectral radius of the iteration matrix for both the methods.

c. How many iterations are required to obtain the solution $\boldsymbol{x}$ with a tolerance equal to 10^{-8} and choosing as initial guess $\boldsymbol{x}^{(0)} = \boldsymbol{0}$?

Solution Homework 7.4.

hw_7_4.m

```
clc
clear all
close all
```

```
% Definition of the matrix A
A = diag(3*ones(6,1)) - diag(ones(5,1), 1) - diag(ones(5,1), -1);
A(1, 6) = -1;
A(2, 5) = -1;
A(5, 2) = -1;
A(6, 1) = -1;
% Visualization of the matrix A
A
% Definition of the vector b
b = [1 0 1 1 0 1]';

A == A'
eig(A) > 0
% The matrix is clearly symmetric, and the computation of the eigenvalues
% shows that is positive definite, so the Gauss-Seidel method converges.
% We cannot conclude anything on the convergence of the Jacobi method because
% the matrix is only weakly diagonally dominant.

% Calculation of the spectral radius of the iteration matrices
D = diag( diag(A) );
E = -tril(A, -1);
% Bj= inv(D)*(D-A)
Bj = D \ (D-A)
% Bgs = inv(D-E)*(D-E-A)
Bgs = (D-E) \ (D-E-A)
rhoBj = max( abs( eig(Bj) ) )
rhoBgs = max( abs( eig(Bgs) ) )

% Both Jacobi and Gauss-Seidel method will converge because
% their spectral radius are less than 1

x0 = zeros(6,1);
tol = 1e-8;

% Jacobi
normBj = norm(Bj, 2) % Choose 2-norm becuase the implementation of stationary_method provides
stopping criterion in 2-norm. This is actually the spectral radius itself, since matrix A is
symmetric.
gj = D \ b;
x1j = Bj*x0 + gj;

% norm(e^{k+1}) <= norm(Bj)*norm(e^{k}); or with tringular inequality
% norm(e^{k+1}) <= norm(Bj)*(norm((x-x^{k+1})+(x^{k+1}-x^{k})))<=norm(Bj)*(norm(x-x^{k+1})+
norm(x^{k+1}-x^{k}))
% = norm(Bj)*(norm(e^{k+1})+norm(x^{k+1}-x^{k})) from which
% norm(x-x^{k+1})<= norm(Bj)*norm(x^{k+1}-x^{k})) / (1-norm(Bj)) because norm(Bj)<1
% now starting from k=0 and applying recursively the inequality we end up
% with the following estimation:
% norm(x-x^{k+1}) <= norm(Bj)^{k+1}*norm(x^{1}-x^{0})) / (1-norm(Bj)
% from which taking the log having set norm(x-x^{k+1}) = tol:
% k ~= log(tol*(1-norm(Bj)/norm(x^{1}-x^{0}))/ log (norm(Bj)

kjmin_2 = log(tol*(1 - normBj)/norm(x1j-x0)) / log(normBj)

% Rounding up to the first integer tbhe estimate says that tolerance
% will be satisfied in no more than 91 iterations.

% Gauss-Seidel
%normBgs = norm(Bgs)
ggs = (D-E) \ b;
x1gs = Bgs*x0 + ggs;
normBgs = norm(Bgs, 2) ;

% following the same procedure used for the jacobi method
kgsmin_2 = log(tol*(1 - normBgs)/norm(x1gs-x0)) / log(normBgs)

% The estimate says that tolerance will be satisfied in no more than 89 iterations

[xj,niterj] = stationary_method(Bj, gj, x0, tol, 1000)
[xgs,nitergs] = stationary_method(Bgs, ggs, x0, tol, 1000)
```

```
% We get convergence in respectively 83 and 45 iteration, in accordance
% with what we found from the estimates.
```

Homework 7.5. Consider the system $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$ with

$$\mathbb{A} = \begin{bmatrix} 50 & 1 & 2 \\ 1 & 5 & 2 \\ 2 & 2 & 7 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

a. Compute $\|\mathbb{A}\|_\infty$ and $\|\mathbb{A}\|_1$.

b. Is $\mathbb{A}$ a symmetric and positive definite matrix?

c. Compute the $\mathbb{LU}$ factorization of $\mathbb{A}$ using the Matlab command `lu`.

d. Compute $\mathcal{K}_2(\mathbb{A})$. Is $\mathbb{A}$ a well-conditioned matrix $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$?

e. What methods (direct or iterative) can be employed to solve the system? Motivate your choice. If the Jacobi method belongs to this list, estimate both the convergence rate and the error after 4 iterations starting from the initial guess $\boldsymbol{x}^{(0)} = \mathbf{0}$. Validate numerically the results.

Solution Homework 7.5.

hw_7_5.m

```
clc
clear all
close all

% Definition of the matrix A
A = [50 1 2;
     1 5 2;
     2 2 7];
% Definition of the B vector
b = [1 0 0]';
% calculation of the 1-norm and inf-norm of the matrix A
norm(A,1)
norm(A,inf)

% Cheking the symmetry of matrix A
A == A'
% Checking the positive definite (i.e eigenvalues all positive)
eig(A) > 0
% yes the matrix is symmetry and psotive definite

% Let calculate the LU decomposition
[L1, U1] = lu(A)

% Let calculate the conditioning number
cond(A,2)
% The condition number is quite small, so the matrix is well conditioned.

% All the methods we saw are suitable for the solution of the linear system,
% since the matrix is symmetric and positive definite.

% let consider the Jacobi method
D = diag( diag(A) );
E = -tril(A, -1);
% Bj= inv(D)*(D-A)
Bj = D \ (D-A)
% calculation of the spectral radoius of the matrix
rhoBj = max(eig(Bj))

% calculation of the ssymptotic convergence rate
rate_jacobi = -log10(rhoBj)
```

```
% We now compute the estimate of the error after 4 iterations
x0 = zeros(3,1);
x1 = Bj*x0 + D\b;
% We estimate the error after four iteration with
% norm(x-x^{k+1}) <= norm(Bj)^{k+1}*norm(x^{1}-x^{0})) / (1-norm(Bj)
esterr4 = norm(Bj)^4/(1-norm(Bj))*norm(x1-x0)

% We now compute the approximated solution after four steps.
xj = x0;
gj = D \ b;
for i=1:4
   xj = Bj*xj + gj;
end

% Let rewrite the solution at the forrth step in the variable x4
x4 = xj
% Let calculate the xact solution
xtrue = A\b;
% The real error at the fourth iteration is
norm(xtrue-x4)

% The error is lower than the estimate as we expected it to be.
```