EPFL

Numerical Analysis and Computational Mathematics

Fall Semester 2025 – CSE Section

Prof. Laura Grigori

Assistant: Israa Fakih

Session 10 – November 19, 2025

Solutions – Linear systems: iterative methods

Exercise I (MATLAB)

a) We consider the following implementation of the MATLAB function:

```
function [ x, k, res ] = preconditioned_gradient( A, b, P, x0, tol, kmax )
% PRECONDITIONED_GRADIENT solve the linear system A x = b by means
% of the Preconditioned Gadrient method; the preconditioning matrix must be
% non singular. Stopping criterion based on the residual.
%  [ x, k, res ] = preconditioned_gradient( A, b, P, x0, tol, kmax )
%  Inputs:  A    = matrix (square matrix)
%           b    = vector (right hand side of the linear system)
%           P    = preconditioning matrix (non singular, same size of A)
%           x0   = initial solution (colum vector)
%           tol  = tolerence for the stopping criterion based on residual
%           kmax = maximum number of iterations
%  Outputs: x    = solution vector (column vector)
%           k    = number of iterations at convergence
%           res  = value of the norm of the residual at convergence
%

k = 0;
x = x0;
r = b - A * x;
res = norm( r );

while( k < kmax && res > tol )
    z = P \ r;
    alpha = ( z' * r ) / ( z' * A * z );
    x = x + alpha * z;
    r = r - alpha * A * z;
    res = norm( r );
    k = k + 1;
end

return
```

b) Since the matrix A is symmetric and positive definite, we know that the gradient method is convergent for all choices of the initial solution $\mathbf{x}^{(0)}$. Moreover, since P_2 is symmetric and positive definite, we know that the preconditioned gradient method is also convergent.

We consider the following MATLAB commands.

```
n = 4;
A = diag( 5 * ones( n, 1 ), 0 ) + diag( 1 * ones( n - 1, 1 ), 1 ) + ...
    diag( 1 * ones( n - 1, 1 ), -1 ) + diag( 1 * ones( n - 2, 1 ), 2 ) + ...
    diag( 1 * ones( n - 2, 1 ), -2 );
x_ex = ones( n, 1 );
b = A * x_ex;
tol = 1.0e-6;   kmax = 100;
x0 = zeros( n, 1 );
% gradient method (P=I)
P1 = eye( n );
[ x1, k1, res1 ] = preconditioned_gradient( A, b, P1, x0, tol, kmax );
err1 = norm( x_ex - x1 )
%   err1 =
%      1.6781e-08
k1, res1
%   k1 =
%         6
%   res1 =
%      1.2614e-07
```

The gradient method ensures convergence to the approximate solution $\mathbf{x}^{(k_c)}$ satisfying the prescribed tolerance in $k_c = 6$ iterations.

For the preconditioned gradient method with $P = P_2$, we obtain:

```
% preconditioned gradient method (P=P_2)
P2 = diag( diag( A ) ) + diag( diag( A, 1 ), 1 ) + diag( diag( A, - 1 ), - 1 );
[ x2, k2, res2 ] = preconditioned_gradient( A, b, P2, x0, tol, kmax );
err2 = norm( x_ex - x2 )
%   err2 =
%      2.1285e-08
k2, res2
%   k2 =
%      4
%   res2 =
%      1.5999e-07
```

The convergence to the approximate solution for the prescribed tolerance occurs in $k_c = 4$ iterations.

c) We start by recalling the following Proposition.

Proposition 1 *If the matrix $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant by row, then the Jacobi and Gauss-Seidel iterative methods converge to the solution $\mathbf{x}$ of the linear system associated to A, say $A\mathbf{x} = \mathbf{b}$, for any choice of the initial solution $\mathbf{x}^{(0)}$.*

Since A is strictly diagonally dominant by row, convergence follows for both methods.

d) We consider the following MATLAB commands.

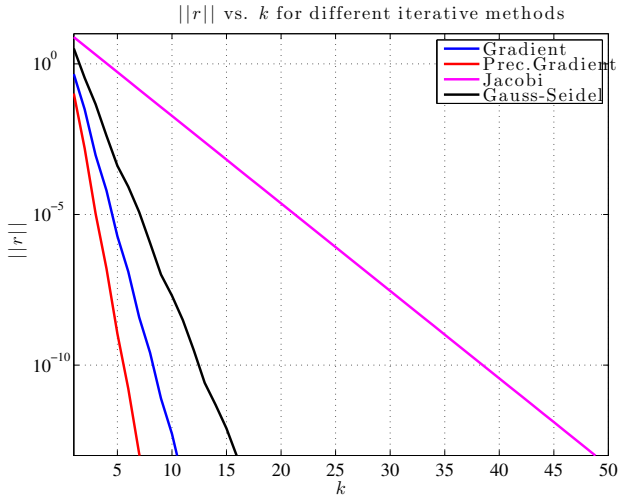

Figure 1: Norm of the residual $r^{(k)}$ vs. number of iterations k for the gradient, preconditioned gradient, Jacobi, and Gauss-Seidel methods.

```
res_PG1_v = [];     res_PG2_v = [];
res_J_v = [];       res_GS_v = [];
klimit = 50;  tol = 1e-14;
k_vect = 1 : klimit;
for kmax = k_vect
    % gradient
    [ xPG1, kPG1, resPG1 ] = preconditioned_gradient( A, b, P1, x0, tol, kmax );
    res_PG1_v = [ res_PG1_v, resPG1 ];
    % preconditioned gardient
    [ xPG2, kPG2, resPG2 ] = preconditioned_gradient( A, b, P2, x0, tol, kmax );
    res_PG2_v = [ res_PG2_v, resPG2 ];
    % Jacobi
    [ xJ, kJ, resJ ] = jacobi( A, b, x0, tol, kmax );
    res_J_v = [ res_J_v, resJ ];
    % Gauss-Seidel
    [ xGS, kGS, resGS ] = gauss_seidel( A, b, x0, tol, kmax );
    res_GS_v = [ res_GS_v, resGS ];
end
semilogy( k_vect, res_PG1_v, '-b', k_vect, res_PG2_v, '-r', ...
          k_vect, res_J_v, '-m', k_vect, res_GS_v, '-k'  );
axis( [ 1 klimit 1e-13 10 ])
legend('Gradient', 'Prec.Gradient', 'Jacobi', 'Gauss-Seidel' );
```

We obtain the result reported in Figure 1. We deduce that, in this case, the preconditioned gradient method with $P = P_2$ ensures a faster convergence than with the gradient, Jacobi, and Gauss-Seidel methods.

Exercise II (MATLAB)

a) The stopping criterion based on the relative residual is satisfactory if the conditioning number of the matrix A is not "too large". Indeed, the following estimate for the relative error on the

solution $e_{rel}^{(k)} = \frac{\|\mathbf{x}-\mathbf{x}^{(k)}\|}{\|\mathbf{x}\|}$ holds:

$$e_{rel}^{(k)} \leq K_2(A)\, r_{rel}^{(k)}, \qquad \text{for all } k = 0, 1, \ldots,$$

where $r_{rel}^{(k)} = \frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|}$ is the relative residual and $K_2(A)$ is the condition number of the matrix A.

We verify that the condition number of the matrix A is very "large" by using the following MATLAB commands:

```
n1 = 15;
A1 = hilb( n1 );
k2_1 = cond( A1 )
%   k2_1 =
%       4.4333e+17
```

b) We verify the answer given at point a) with the following MATLAB commands:

```
x1_ex = ones( n1, 1 );
b1 = A1 * x1_ex;
x1_0 = zeros( n1, 1 );
% criterion based on RELATIVE residual
tol_rel = 1.0e-5;       kmax = 1000;
tol =  tol_rel * norm( b1 );
[ x1_gs, k1_gs, res1_gs ] = gauss_seidel( A1, b1, x1_0, tol, kmax );
k1_gs
%   k1_gs =
%       599
res_rel1_gs = res1_gs / norm( b1 )
%   res_rel1_gs =
%       9.9853e-06
err_rel1_gs = norm( x1_ex - x1_gs ) / norm( x1_ex )
%   err_rel1_gs =
%       0.0412
```

Convergence to the approximate solution occurs in $k_c = 599$ iterations. We observe that the relative residual at convergence is $r_{rel}^{(k_c)} = 9.9853 \cdot 10^{-6}$, while the relative error is $e_{rel}^{(k_c)} = 4.1220 \cdot 10^{-2}$, which is significantly larger than $r^{(k_c)}$.

c) The stopping criterion based on the difference of successive iterates is satisfactory if the spectral radius of the iteration matrix B, denoted by $\rho(B)$, is significantly smaller than 1 ($\rho(B) \ll 1$), while it is unsatisfactory when $\rho(B) \simeq 1$. Indeed, when B is symmetric and positive definite, we have the explicit bound: $e^{(k)} \leq \frac{1}{1-\rho(B)}\, \delta^{(k)}$, for $k = 1, \ldots$, where $e^{(k)} = \left\|\mathbf{e}^{(k)}\right\| = \left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|$ and $\delta^{(k)} = \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|$.[1]

We compute the spectral radius of the iteration matrix $B_{2,GS}$ associated to the Gauss-Seidel method for the matrix A_2.

```
n2 = 100;
```

[1] Note that the iteration matrix $B_{GS} = I - (D - E)^{-1}A$ is not in general symmetric positive definite in the usual inner product induced by the matrix norm $\|\cdot\|_2$.

```
A2 = diag( ( 4 + 5e-5 ) * ones( n2, 1 ), 0 ) + ...
     diag( - 2 * ones( n2 - 1, 1 ), 1 ) + diag( - 2 * ones( n2 - 1, 1 ), -1 );
D2 = diag( diag( A2 ) );
E2 = - tril( A2, -1 );
B2_GS = eye( n2 ) - inv( D2 - E2 ) * A2;
format short e
rho2_GS = max( abs( eig( B2_GS ) ) )
%   rho2_GS =
%      9.9901e-01
format
```

We observe that the Gauss-Seidel method is convergent, since $\rho(B_{2,GS}) = 9.9901 \cdot 10^{-1} < 1$. Still, the convergence will be slow as $\rho(B_{2,GS}) \simeq 1$. Moreover, we deduce that the stopping criterion based on the difference of successive iterates is unsatisfactory since $\rho(B_{2,GS}) \simeq 1$.

d) We consider the following implementation in MATLAB of the function.

```
function [ x, k, diff ] = gauss_seidel_difference_iterates( A, b, x0, tol, kmax )
% GAUSS_SEIDEL solve the linear system A x = b by means
% of the Gauss-Seidel iterative method; diagonal elements of A
% must be nonzero. Stopping criterion based on the difference of successive
% iterates
%  [ x, k, diff ] = gauss_seidel( A, b, x0, tol, kmax )
%  Inputs:  A    = matrix (square matrix)
%           b    = vector (right hand side of the linear system)
%           x0   = initial solution (colum vector)
%           tol  = tolerance for the stopping criterion based on difference
%                  of successive iterates
%           kmax = maximum number of iterations
%  Outputs: x    = solution vector (column vector)
%           k    = number of iterations at convergence
%           diff = difference (in norm) between successive iterates
%

n = size( A, 1 );

k = 0;
x = x0;
diff = tol + 1;

x_old = x0;

while( k < kmax && diff > tol )
    for i = 1 : n
        j_v = 1 : i - 1;
        j_v_old = i + 1 : n;
        x( i ) = 1 / A( i, i ) * ( b( i ) ...
                                   - A( i, j_v ) * x( j_v ) ...
                                   - A( i, j_v_old ) * x_old( j_v_old ) );
    end
    diff = norm( x - x_old );
    k = k + 1;
    x_old = x;
end

return
```

We use the previous function to verify the result of point c) by means of the following MATLAB commands.

```
x2_ex = ones( n2, 1 );
b2 = A2 * x2_ex;
tol = 1.0e-5;    kmax = 10000;
x2_0 = zeros( n2, 1 );
[ x2_gs, k2_gs, diff2_gs ] = gauss_seidel_difference_iterates( A2, b2, x2_0, ...
                                                               tol, kmax );
k2_gs, diff2_gs,
%   k2_gs =
%           6852
%   diff2_gs =
%      9.9964e-06
format short e
err2_gs = norm( x2_ex - x2_gs )
%   err2_gs =
%      1.0065e-02
format
```

We obtain that the convergence to the approximate solution $\mathbf{x}^{(k_c)}$ requires $k_c = 6852$ iterations. The final error is $e^{(k_c)} = 1.0065 \cdot 10^{-2}$, whereas the norm of the difference of the last two approximate solutions is $\delta^{(k_c-1)} = 9.9964 \cdot 10^{-6}$. We verify that the stopping criterion based on the norm of successive iterates is unsatisfactory for the Gauss-Seidel method, since $\rho(B_{2,GS}) \simeq 1$, and the error is underestimated.

Exercise III (Theoretical)

a) The choice $P = P_1 = P_1(\beta) = \beta D$ for $\beta = 1$ corresponds to the Jacobi iterative method. We observe that the elements of A on the diagonal are nonzero, so that the preconditioning matrix P_1 is invertible for all $\beta > 0$.

In order to calculate the values of $\beta > 0$ ensuring that the iterative method defined by the preconditioning matrix $P_1(\beta) = \beta D$ is convergent for all the initial solutions $\mathbf{x}^{(0)}$, first we need to calculate the associated iteration matrix $B_1 = I - P_1^{-1}A$. Then, we calculate the values of β for which the spectral radius of the iteration matrix $B_1(\beta)$, denoted $\rho_1 = \rho_1(\beta)$, is < 1. We recall that $\rho_1 = \rho_1(\beta) = \max_{i=1,2} |\lambda_{1,i}(\beta)|$, where $\lambda_{1,i}(\beta)$ for $i = 1, 2$ are the eigenvalues of the iteration matrix $B_1(\beta)$.

We obtain

$$B_1(\beta) = I - \frac{1}{\beta} D^{-1} A = \begin{bmatrix} \left(1 - \frac{1}{\beta}\right) & \frac{1}{3\beta} \\ \frac{1}{2\beta} & \left(1 - \frac{1}{\beta}\right) \end{bmatrix},$$

and we compute the eigenvalues:

$$\lambda_{1,1}(\beta) = 1 - \frac{1}{\beta}\left(1 + \frac{1}{\sqrt{6}}\right) \qquad \text{and} \qquad \lambda_{1,2}(\beta) = 1 - \frac{1}{\beta}\left(1 - \frac{1}{\sqrt{6}}\right).$$

We plot the eigenvalues $\lambda_{1,1}(\beta)$ and $\lambda_{1,2}(\beta)$ vs. β in Figure 2 (left) for $\beta \in (0, 7)$. We plot in Figure 2 (right) the magnitude of the eigenvalues $|\lambda_{1,1}(\beta)|$ and $|\lambda_{1,2}(\beta)|$. In Figure 3 we plot the spectral radius of the iteration matrix $B_1(\beta)$. From Figure 3 we deduce that $\rho_1(\beta) = 1$

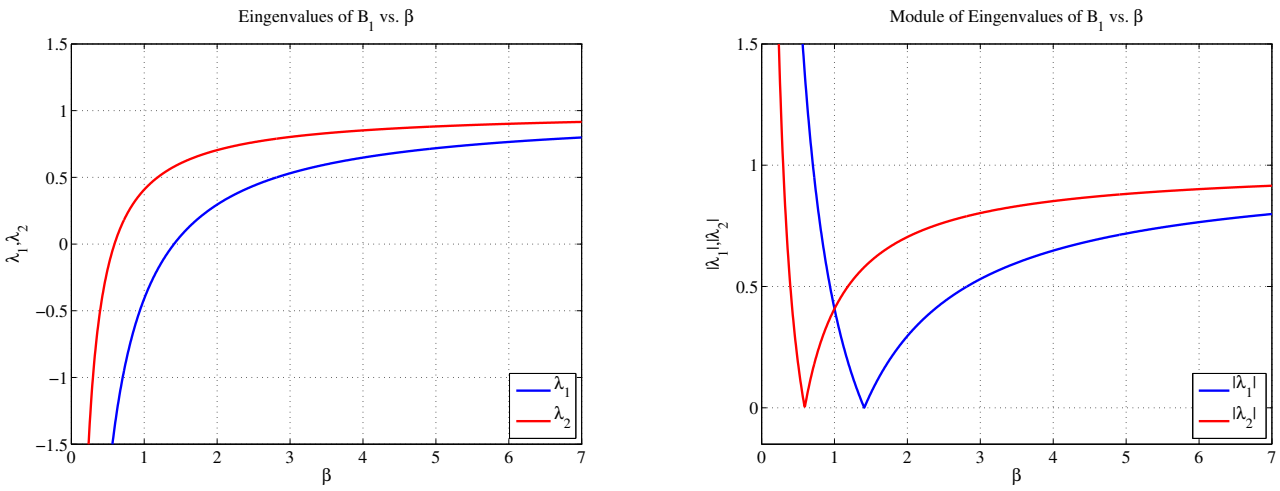

Figure 2: Eigenvalues $\lambda_{1,1}(\beta)$, $\lambda_{1,2}(\beta)$ (left) and their magnitudes (right) vs. β for the iteration matrix $B_1(\beta)$.

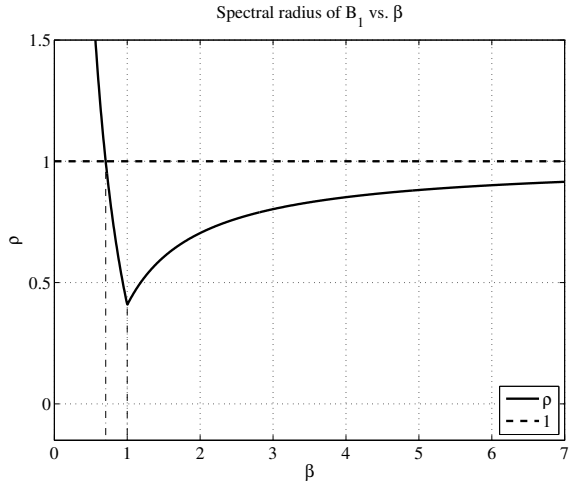

Figure 3: Spectral radius $\rho_1(\beta)$ vs. β for the iteration matrix $B_1(\beta)$.

when $|\lambda_{1,1}(\beta)| = 1$, which yields $\beta = \beta_{1,0} = \frac{1}{2}\left(1 + \frac{1}{\sqrt{6}}\right)$. We also observe that $\rho_1(\beta) < 1$ for large β.

We conclude that the iterative method corresponding to the choice of $P = P_1(\beta) = \beta D$ is convergent for $\beta > \beta_{1,0} = \frac{1}{2}\left(1 + \frac{1}{\sqrt{6}}\right)$. However, we observe that for β "large" the convergence of the iterative method would be extremely slow.

b) The fastest convergence to the solution of the iterative method is obtained for $\beta = \beta_{1,min}$, where the smallest value of the spectral radius $\rho_1(\beta)$ is attained. From Figure 3 we deduce that $\beta_{1,min}$ can be found at the intersection of the curves $|\lambda_{1,1}(\beta)|$ and $|\lambda_{1,2}(\beta)|$. After simple algebraic operations, we find the two intersection points $\beta_{1,min} = 1$ and $\rho_{1,min} = \frac{1}{\sqrt{6}}$. We conclude that the fastest convergence is achieved for $\beta_{1,min} = 1$, i.e. by the Jacobi method.

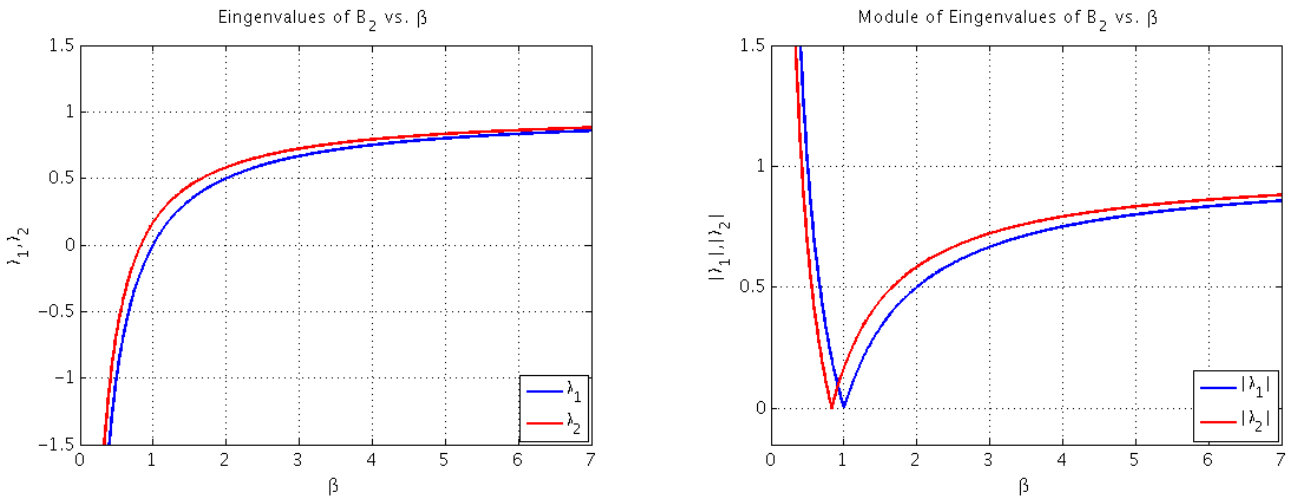

Figure 4: Eigenvalues $\lambda_{2,1}(\beta)$, $\lambda_{2,2}(\beta)$ (left) and their magnitudes (right) vs. β for the iteration matrix $B_2(\beta)$.

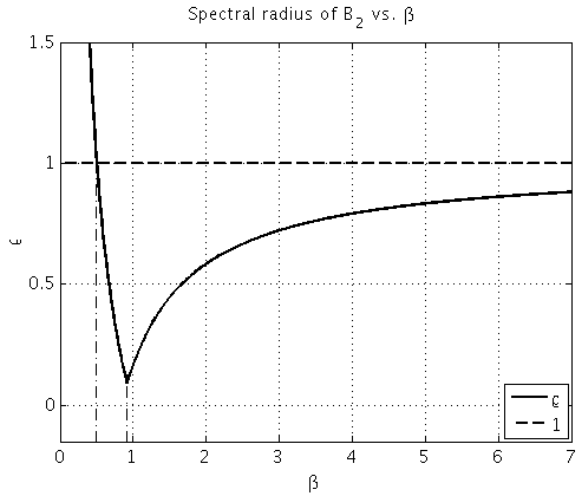

Figure 5: Spectral radius $\rho_2(\beta)$ vs. β for the iteration matrix $B_2(\beta)$.

c) We consider the preconditioner $P = P_2(\beta) = \beta(D - E)$. We observe that the case $\beta = 1$ corresponds to the Gauss-Seidel method. The elements of A on the diagonal are nonzero, so that P_2 is invertible for all $\beta > 0$.

Our goal consists in determining the values of β for which the convergence of the iterative method to the solution is ensured for all $\mathbf{x}^{(0)}$. To this aim, we repeat the procedure of step a), setting $P = P_2(\beta) = \beta(D - E)$ and computing the spectral radius of the corresponding iterative matrix $B_2(\beta)$. We obtain

$$\lambda_{2,1}(\beta) = 1 - \frac{1}{\beta} \qquad \text{and} \qquad \lambda_{2,2}(\beta) = 1 - \frac{5}{6\beta}.$$

We plot these eigenvalues in Figure 4 for $\beta \in (0, 7)$. In Figure 5 we plot the spectral radius

$\rho_2(\beta) = \max_{i=1,2} |\lambda_{2,i}(\beta)|$ of the iteration matrix $B_2(\beta)$. We deduce that $\rho_2(\beta) = 1$ when $|\lambda_{2,1}(\beta)| = 1$, i.e. for $\beta = \beta_{2,0} = \frac{1}{2}$.

We conclude that the iterative method corresponding to the choice of $P = P_2(\beta) = \beta(D - E)$ is convergent for any initial solution $\mathbf{x}^{(0)}$ as long as $\beta > \beta_{2,0} = \frac{1}{2}$. Once again, we observe that, for large β, the convergence of the iterative method is extremely slow.

d) As in point b), from Figure 5 we deduce that $\rho_{2,min}$ corresponds to the value $\beta_{2,min}$ for which the curves $|\lambda_{2,1}(\beta)|$ and $|\lambda_{2,2}(\beta)|$ intersect. After algebraic manipulations, we obtain $\beta_{2,min} = \frac{11}{12}$, corresponding to $\rho_{2,min} = \frac{1}{11}$. We conclude that the fastest convergence is achieved, in this case, for $\beta = \beta_{2,min}$. We observe that the iterative method defined for $\beta = \beta_{2,min} = \frac{11}{12}$ does not correspond to the Gauss-Seidel method.

e) By comparing the preconditioning matrices P_1 and P_2 for $\beta = \beta_{1,min}$ and $\beta_{2,min}$, we obtain $\rho_{1,min} = \frac{1}{\sqrt{6}}$ and $\rho_{2,min} = \frac{1}{11}$, respectively. Since $\rho_{2,min} < \rho_{1,min}$, we select the iterative method corresponding to the choice of $P = P_2(\beta) = \beta(D - E)$, with $\beta = \beta_{2,min} = \frac{11}{12}$.