

Linear Systems - Iterative Methods



Numerical Analysis

Profs. Gianluigi Rozza-Luca Heltai

2019-SISSA mathLab Trieste

(Sec. 5.9 of the book)

Solve the linear system $A\mathbf{x} = \mathbf{b}$ using an iterative method consists in building a series of vectors $\mathbf{x}^{(k)}$, $k \geq 0$, in \mathbb{R}^n that converge at the exact solution \mathbf{x} , i.e.:

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}$$

for any initial vector $\mathbf{x}^{(0)} \in \mathbb{R}^n$.

We can consider the following recurrence relation:

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{g}, \quad k \geq 0 \quad (1)$$

where B is a well chosen matrix (depending on A) and \mathbf{g} is a vector (that depends on A and \mathbf{b}), satisfying the relation (of consistence)

$$\mathbf{x} = B\mathbf{x} + \mathbf{g}. \quad (2)$$

Given $\mathbf{x} = A^{-1}\mathbf{b}$, we get $\mathbf{g} = (I - B)A^{-1}\mathbf{b}$; the iterative method is therefore completely defined by the matrix B , known as *iteration matrix*.
By defining the error at step k as

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)},$$

we obtain the following recurrence relation:

$$\mathbf{e}^{(k+1)} = B\mathbf{e}^{(k)} \quad \text{and thus} \quad \mathbf{e}^{(k+1)} = B^{k+1}\mathbf{e}^{(0)}, \quad k = 0, 1, \dots$$

$$B\mathbf{e}^{(k)} = B(\mathbf{x} - \mathbf{x}^{(k)}) = B\mathbf{x} - B\mathbf{x}^{(k)} = B\mathbf{x} - \mathbf{x}^{(k+1)} + \mathbf{g} = \mathbf{x} - \mathbf{x}^{(k+1)} = \mathbf{e}^{(k+1)}$$

We can show that $\lim_{k \rightarrow \infty} \mathbf{e}^{(k)} = \mathbf{0}$ for all $\mathbf{e}^{(0)}$ (and thus for all $\mathbf{x}^{(0)}$) if and only if

$$\rho(B) < 1,$$

or $\rho(B)$ is the *spectral radius* of the matrix B , defined by

$$\rho(B) = \max |\lambda_i(B)|$$

and $\lambda_i(B)$ are the eigenvalues of the matrix B .

The smaller the value of $\rho(B)$, the less iterations are needed to reduce the initial error of a given factor.

Construction of an iterative method

A general way of setting up an iterative method is based on the decomposition of the matrix A :

$$A = P - (P - A)$$

where P is an invertible matrix called *preconditioner* of A .
Hence,

$$A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad P\mathbf{x} = (P - A)\mathbf{x} + \mathbf{b}$$

which is of the form (2) leaving

$$B = P^{-1}(P - A) = I - P^{-1}A \quad \text{and} \quad \mathbf{g} = P^{-1}\mathbf{b}.$$

We can define the corresponding iterative method

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \mathbf{r}^{(k)} \quad k \geq 0$$

where $\mathbf{r}^{(k)}$ represents the *residual* at the iteration k : $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$

We can generalise this method as follows:

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \alpha_k \mathbf{r}^{(k)} \quad k \geq 0 \quad (3)$$

where $\alpha_k \neq 0$ is a parameter that improves the convergence of the series $\mathbf{x}^{(k)}$.
The method (3) is called *Richardson's method*.

The matrix P has to be chosen in such a way that renders the cost of solving (3) small enough. For example a diagonal or triangular P matrix would comply with this criterion.

Jacobi method

If the elements of the diagonal of A are non-zero, we can write

$$P = D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$$

D with the diagonal part of A being:

$$D_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ a_{ij} & \text{if } i = j. \end{cases}$$

The Jacobi method corresponds to this choice with $\alpha_k = 1$ for all k .

We deduce:

$$D\mathbf{x}^{(k+1)} = \mathbf{b} - (A - D)\mathbf{x}^{(k)} \quad k \geq 0.$$

By components:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n. \quad (4)$$

The Jacobi method can be written under the general form

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{g},$$

with

$$B = B_J = D^{-1}(D - A) = I - D^{-1}A, \quad \mathbf{g} = \mathbf{g}_J = D^{-1}\mathbf{b}.$$

Gauss-Seidel method

This method is defined as follows:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, \dots, n.$$

This method corresponds to (1) with $P = D - E$ and $\alpha_k = 1$ ($\forall k \geq 0$) where E is the lower triangular matrix

$$\begin{cases} E_{ij} = -a_{ij} & \text{if } i > j \\ E_{ij} = 0 & \text{if } i \leq j \end{cases}$$

(lower triangular part of A without the diagonal and with its elements' sign inverted).

We can write this method under the form (3), with the iteration matrix $B = B_{GS}$ given by

$$B_{GS} = (D - E)^{-1}(D - E - A)$$

and

$$\mathbf{g}_{GS} = (D - E)^{-1}\mathbf{b}.$$

Example 1. Given the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}.$$

We have then

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 16 \end{pmatrix};$$

Thus, the iteration matrix for the Jacobi method is

$$B_J = D^{-1}(D - A) = I - D^{-1}A = \begin{pmatrix} 0 & -2 & -3 & -4 \\ -5/6 & 0 & -7/6 & -4/3 \\ -9/11 & -10/11 & 0 & -12/11 \\ -13/16 & -14/16 & -15/16 & 0 \end{pmatrix}.$$

For defining the matrix A and extracting its diagonal D and its lower triangular part E (without the diagonal and the sign inverted) with Matlab/Octave, we use the commands

```
>> A = [1,2,3,4;5,6,7,8;9,10,11,12;13,14,15,16];
>> D = diag(diag(A));
>> E = - tril(A,-1);
```

These allow us, for exemple, to compute the iteration matrix B_{GS} for the Gauss-Seidel method in the following way:

```
>> B_GS = (D-E)\(D-E-A);
```

We find:

$$B_{GS} = \begin{pmatrix} 0.0000 & -2.0000 & -3.0000 & -4.0000 \\ 0.0000 & 1.6667 & 1.3333 & 2.0000 \\ 0.0000 & 0.1212 & 1.2424 & 0.3636 \\ 0.0000 & 0.0530 & 0.1061 & 1.1591 \end{pmatrix}.$$

Convergence

We have the following convergence results:

- (Prop 5.3) If the matrix A is strictly diagonally dominant by row, i.e.,

$$|a_{ii}| > \sum_{j=1, \dots, n; j \neq i} |a_{ij}|, \quad i = 1, \dots, n,$$

then the Jacobi and the Gauss-Seidel methods converge.

- If A is **symmetric positive definite**, then the Gauss-Seidel method converges (Jacobi maybe not).
- (Prop 5.4) Let A be a tridiagonal non-singular matrix whose diagonal elements are all non-null. Then the Jacobi and the Gauss-Seidel methods are either **both divergent or both convergent**. In the latter case, $\rho(B_{GS}) = \rho(B_J)^2$.

Richardson method

(Sec. 5.10)

Let consider the following iterative method:

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \alpha_k \mathbf{r}^{(k)}, \quad k \geq 0. \quad (5)$$

If $\alpha_k = \alpha$ (a constant) this method is called **stationary preconditioned Richardson method**; otherwise **dynamic preconditioned Richardson method** when α_k varies during the iterations.

The matrix P is called **preconditioner** of A .

If A and P are **symmetric positive definite**, then there are two optimal criteria to choose α_k :

1. **Stationary case:**

$$\alpha_k = \alpha_{opt} = \frac{2}{\lambda_{min} + \lambda_{max}}, \quad k \geq 0,$$

where λ_{min} and λ_{max} represent the smaller and the larger eigenvalue of the matrix $P^{-1}A$.

2. **Dynamic case:**

$$\alpha_k = \frac{(\mathbf{z}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{z}^{(k)})^T A \mathbf{z}^{(k)}}, \quad k \geq 0,$$

where $\mathbf{z}^{(k)} = P^{-1}\mathbf{r}^{(k)}$ is the preconditioned residual.

This method is also called **preconditioned gradient method**.

If $P = I$ and A is symmetric definite positive, we get the following methods:

- the **Stationary Richardson** if we choose:

$$\alpha_k = \alpha_{opt} = \frac{2}{\lambda_{min}(A) + \lambda_{max}(A)}. \quad (6)$$

- the **Gradient** method if :

$$\alpha_k = \frac{(\mathbf{r}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{r}^{(k)})^T A \mathbf{r}^{(k)}}, \quad k \geq 0. \quad (7)$$

The gradient method can be written as:

Let $\mathbf{x}^{(0)}$ be given, set $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$, then for $k \geq 0$,

$$\begin{aligned} P\mathbf{z}^{(k)} &= \mathbf{r}^{(k)} \\ \alpha_k &= \frac{(\mathbf{z}^{(k)})^T \mathbf{r}^{(k)}}{(\mathbf{z}^{(k)})^T A\mathbf{z}^{(k)}} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)} \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k A\mathbf{z}^{(k)}. \end{aligned}$$

We have to apply once A and inverse P at each iteration. P should then be such that the resolution of the associated system results easy (i.e. it requires a reasonable amount of computing cost). For example, we can choose a diagonal P (Like in the gradient or the stationary Richardson cases) or triangular.

Convergence of Richardson method

When A and P are s.p.d. and with the two optimal choices for α , we can show that the preconditioned Richardson Method converges to x when $k \rightarrow \infty$, and that

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_A \leq \left(\frac{K(P^{-1}A) - 1}{K(P^{-1}A) + 1} \right)^k \|\mathbf{x}^{(0)} - \mathbf{x}\|_A, \quad k \geq 0, \quad (8)$$

where $\|\mathbf{v}\|_A = \sqrt{\mathbf{v}^T A \mathbf{v}}$ and $K(P^{-1}A)$ is the condition number of $P^{-1}A$.

Remark If A et P are s.p.d., we have that

$$K(P^{-1}A) = \frac{\lambda_{max}}{\lambda_{min}}.$$

Demonstration The iteration matrix of the method is given by $R_\alpha = I - \alpha P^{-1}A$, where the eigenvalues of R_α are of the form $1 - \alpha\lambda_i$. The method is convergent if and only if $|1 - \alpha\lambda_i| < 1$ for $i = 1, \dots, n$, therefore $-1 < 1 - \alpha\lambda_i < 1$ for $i = 1, \dots, n$. As $\alpha > 0$, this is the equivalent to $-1 < 1 - \alpha\lambda_{max}$, from where the necessary and sufficient condition for convergence remains $\alpha < 2/\lambda_{max}$. Consequently, $\rho(R_\alpha)$ is minimal if $1 - \alpha\lambda_{min} = \alpha\lambda_{max} - 1$, i.e., for $\alpha_{opt} = 2/(\lambda_{min} + \lambda_{max})$. By substitution, we obtain

$$\rho_{opt} = \rho(R_{opt}) = 1 - \alpha_{opt}\lambda_{min} = 1 - \frac{2\lambda_{min}}{\lambda_{min} + \lambda_{max}} = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{min} + \lambda_{max}}$$

what allows us to complete the proof. □

$\|x^k - x\| = \|B^k e(0)\|$ because of slide 3
 $\leq \|B\|^k \|x(0) - x\|$ and $\|B\| = \rho(B)$ which we just computed

In the dynamic case, we get a result that allows us to optimally choose the iteration parameters at each step, if the matrix A is symmetric definite positive:

Theorem 1 (Dynamic case). *If A is symmetric definite positive, the optimal choice for α_k is given by*

$$\alpha_k = \frac{(\mathbf{r}^{(k)}, \mathbf{z}^{(k)})}{(A\mathbf{z}^{(k)}, \mathbf{z}^{(k)})}, \quad k \geq 0 \quad (9)$$

like slide 15 but with scalar product

where

$$\mathbf{z}^{(k)} = P^{-1} \mathbf{r}^{(k)}. \quad (10)$$

Demonstration On the one hand we have

$$\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)} = A(\mathbf{x} - \mathbf{x}^{(k)}) = -A\mathbf{e}^{(k)}, \quad (11)$$

and thus, using (10),

$$P^{-1}A\mathbf{e}^{(k)} = -\mathbf{z}^{(k)}, \quad (12)$$

where $\mathbf{e}^{(k)}$ represents the error at the step k . On the other hand

$$\mathbf{e}^{(k+1)} = \mathbf{e}^{(k+1)}(\alpha) = \underbrace{(I - \alpha P^{-1}A)}_{R_\alpha} \mathbf{e}^{(k)}.$$

We notice that, in order to update the residual, we have the relation

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha A \mathbf{z}^{(k)} = \mathbf{r}^{(k)} - \alpha A P^{-1} \mathbf{r}^{(k)}.$$

Thus, expressing as $\|\cdot\|_A$ the vector norm associated to the scalar product $(\mathbf{x}, \mathbf{y})_A = (A\mathbf{x}, \mathbf{y})$, what means, $\|\mathbf{x}\|_A = (A\mathbf{x}, \mathbf{x})^{1/2}$ we can write

$$\begin{aligned} \|\mathbf{e}^{(k+1)}\|_A^2 &= (A\mathbf{e}^{(k+1)}, \mathbf{e}^{(k+1)}) = -(\mathbf{r}^{(k+1)}, \mathbf{e}^{(k+1)}) \\ &= -(\mathbf{r}^{(k)} - \alpha A P^{-1} \mathbf{r}^{(k)}, \mathbf{e}^{(k)} - \alpha P^{-1} A \mathbf{e}^{(k)}) \\ &= -(\mathbf{r}^{(k)}, \mathbf{e}^{(k)}) + \alpha[(\mathbf{r}^{(k)}, P^{-1} A \mathbf{e}^{(k)}) + (A \mathbf{z}^{(k)}, \mathbf{e}^{(k)})] \\ &\quad - \alpha^2 (A \mathbf{z}^{(k)}, P^{-1} A \mathbf{e}^{(k)}) \end{aligned}$$

Now we choose α as the α_k that minimises $\|\mathbf{e}^{(k+1)}(\alpha)\|_A$:

$$\left. \frac{d}{d\alpha} \|\mathbf{e}^{(k+1)}(\alpha)\|_A \right|_{\alpha=\alpha_k} = 0$$

We then obtain

$$\alpha_k = \frac{1}{2} \frac{(\mathbf{r}^{(k)}, P^{-1} A \mathbf{e}^{(k)}) + (A \mathbf{z}^{(k)}, \mathbf{e}^{(k)})}{(A \mathbf{z}^{(k)}, P^{-1} A \mathbf{e}^{(k)})} = \frac{1}{2} \frac{-(\mathbf{r}^{(k)}, \mathbf{z}^{(k)}) + (A \mathbf{z}^{(k)}, \mathbf{e}^{(k)})}{-(A \mathbf{z}^{(k)}, \mathbf{z}^{(k)})}$$

and using the equality $(A \mathbf{z}^{(k)}, \mathbf{e}^{(k)}) = (\mathbf{z}^{(k)}, A \mathbf{e}^{(k)})$ knowing that A is symmetric definite positive, and noting that $A \mathbf{e}^{(k)} = -\mathbf{r}^{(k)}$, we find

$$\alpha_k = \frac{(\mathbf{r}^{(k)}, \mathbf{z}^{(k)})}{(A \mathbf{z}^{(k)}, \mathbf{z}^{(k)})}$$



For the stationary case and for the dynamic one we can prove that, if A and P are symmetric definite positive, the series $\{\mathbf{x}^{(k)}\}$ given by the Richardson method (stationary and dynamic) converges towards \mathbf{x} when $k \rightarrow \infty$, and

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_A \leq \left(\frac{K(P^{-1}A) - 1}{K(P^{-1}A) + 1} \right)^k \|\mathbf{x}^{(0)} - \mathbf{x}\|_A, \quad k \geq 0, \quad (13)$$

where $\|\mathbf{v}\|_A = \sqrt{\mathbf{v}^T A \mathbf{v}}$ and $K(P^{-1}A)$ is the conditioning of the matrix $P^{-1}A$.

Remark. In the case of the gradient method or the Richardson stationary method the error estimation becomes

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_A \leq \left(\frac{K(A) - 1}{K(A) + 1} \right)^k \|\mathbf{x}^{(0)} - \mathbf{x}\|_A, \quad k \geq 0. \quad (14)$$

Remark. If A and P are symmetric definite positive, we have

$$K(P^{-1}A) = \frac{\lambda_{\max}(P^{-1}A)}{\lambda_{\min}(P^{-1}A)}.$$

The conjugate gradient method

(Sec. 5.11)

When A and P are s.p.d, there exists a very efficient and effective method to iteratively solve the system: the **conjugate gradient method**

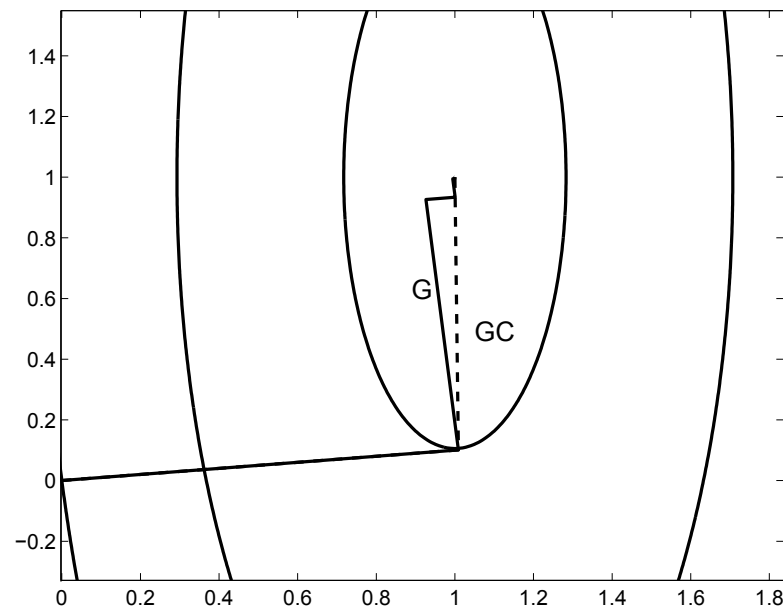
Let $\mathbf{x}^{(0)}$ be given; we compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)} = P^{-1}\mathbf{r}^{(0)}$, $\mathbf{p}^{(0)} = \mathbf{z}^{(0)}$, then for $k \geq 0$,

$$\begin{aligned}\alpha_k &= \frac{\mathbf{p}^{(k)T} \mathbf{r}^{(k)}}{\mathbf{p}^{(k)T} A \mathbf{p}^{(k)}} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)} \\ \mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k A \mathbf{p}^{(k)} \\ P \mathbf{z}^{(k+1)} &= \mathbf{r}^{(k+1)} \\ \beta_k &= \frac{(A \mathbf{p}^{(k)})^T \mathbf{z}^{(k+1)}}{(A \mathbf{p}^{(k)})^T \mathbf{p}^{(k)}} \\ \mathbf{p}^{(k+1)} &= \mathbf{z}^{(k+1)} - \beta_k \mathbf{p}^{(k)} .\end{aligned}$$

The error estimate is given by

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_A \leq \frac{2c^k}{1 + c^{2k}} \|\mathbf{x}^{(0)} - \mathbf{x}\|_A, \quad k \geq 0 \quad \text{où} \quad c = \frac{\sqrt{K_2(P^{-1}A)} - 1}{\sqrt{K_2(P^{-1}A)} + 1}.$$

(15)



Example 2. Let consider the following linear system:

$$\begin{cases} 2x_1 + x_2 &= 1 \\ x_1 + 3x_2 &= 0 \end{cases} \quad (16)$$

whose matrix is $A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$ is s.p.d. The solution to this system is $x_1 = 3/5 = 0.6$ et $x_2 = -1/5 = -0.2$.

Preliminary convergence studies

- A is strictly diagonal dominant by row. Hence Jacobi and Gauss-Seidel methods converge.
- A is regular, tridiagonal with non-zero diagonal elements. Then $\rho(B_{GS}) = \rho(B_J)^2$. Therefore we expect a quicker convergence of Gauss-Seidel w.r.t. Jacobi.
- A is s.p.d., hence the gradient and the conjugate gradient methods converge. Moreover (see error estimates), the CG shall converge faster.

We want to approximate the solution with an iterative method starting with

$$\mathbf{x}^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

We can see that

$$\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)} = \begin{pmatrix} -\frac{3}{2} \\ -\frac{5}{2} \end{pmatrix}$$

and

$$\|\mathbf{r}^{(0)}\|_2 = \sqrt{(\mathbf{r}^{(0)})^T \mathbf{r}^{(0)}} = \frac{\sqrt{34}}{2} \approx 2.9155.$$

Jacobi method

$$\mathbf{x}^{(k+1)} = B_J \mathbf{x}^{(k)} + \mathbf{g}_J, \quad k \geq 0, \quad \text{where } B_J = I - D^{-1}A \text{ and } \mathbf{g}_J = D^{-1}\mathbf{b}.$$

We have

$$B_J = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{3} & 0 \end{pmatrix}$$

$$\mathbf{g}_J = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$$

$$\text{and } \rho(B_J) = \max |\lambda_i(B_J)| = \max(\text{abs}(\text{eig}(B_J))) = 0.4082.$$

For $k = 0$ (first iteration) we find:

$$\mathbf{x}^{(1)} = B_J \mathbf{x}^{(0)} + \mathbf{g}_J = \begin{pmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{3} & 0 \end{pmatrix} \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ -\frac{1}{3} \end{pmatrix} \approx \begin{pmatrix} 0.25 \\ -0.3333 \end{pmatrix}.$$

Notice that

$$\mathbf{r}^{(1)} = \mathbf{b} - A\mathbf{x}^{(1)} = \begin{pmatrix} 0.8333 \\ 0.75 \end{pmatrix} \quad \text{and} \quad \|\mathbf{r}^{(1)}\|_2 = 1.1211.$$

Gauss-Seidel method

$$\mathbf{x}^{(k+1)} = B_{GS}\mathbf{x}^{(k)} + \mathbf{g}_{GS}, \quad k \geq 0, \quad \text{where } B_{GS} = (D - E)^{-1}(D - E - A)$$

$$\text{and } \mathbf{g}_{GS} = (D - E)^{-1}\mathbf{b}.$$

We have

$$B_{GS} = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ -\frac{1}{6} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{6} \end{pmatrix}$$

$$\mathbf{g}_{GS} = \begin{pmatrix} \frac{1}{2} & 0 \\ -\frac{1}{6} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{6} \end{pmatrix}$$

In this case $\rho(B_{GS}) = \max|\lambda_i(B_{GS})| = \max(\text{abs}(\text{eig}(B_{GS}))) = 0.1667$.

We can verify that $\rho(B_{GS}) = \rho(B_J)^2$.

For $k = 0$ (first iteration) we find:

$$\mathbf{x}^{(1)} = B_{GS}\mathbf{x}^{(0)} + \mathbf{g}_{GS} = \begin{pmatrix} 0 & -\frac{1}{2} \\ 0 & \frac{1}{6} \end{pmatrix} \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{6} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ -\frac{1}{12} \end{pmatrix} \approx \begin{pmatrix} 0.25 \\ -0.0833 \end{pmatrix}.$$

We have

$$\mathbf{r}^{(1)} = \mathbf{b} - A\mathbf{x}^{(1)} = \begin{pmatrix} 0.5833 \\ 0 \end{pmatrix} \quad \text{and} \quad \|\mathbf{r}^{(1)}\|_2 = 0.5833.$$

Preconditioned gradient method with $P = D$

We set $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} -\frac{3}{2} \\ -\frac{5}{2} \end{pmatrix}$.

For $k = 0$, we have:

$$P\mathbf{z}^{(0)} = \mathbf{r}^{(0)} \quad \Leftrightarrow \quad \mathbf{z}^{(0)} = P^{-1}\mathbf{r}^{(0)} = \begin{pmatrix} -\frac{3}{4} \\ -\frac{5}{6} \end{pmatrix}$$

$$\alpha_0 = \frac{(\mathbf{z}^{(0)})^T \mathbf{r}^{(0)}}{(\mathbf{z}^{(0)})^T A\mathbf{z}^{(0)}} = \frac{77}{107}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{z}^{(0)} = \begin{pmatrix} 0.4603 \\ -0.0997 \end{pmatrix}$$

$$\mathbf{r}^{(1)} = \mathbf{r}^{(0)} - \alpha_0 A\mathbf{z}^{(0)} = \begin{pmatrix} 0.1791 \\ -0.1612 \end{pmatrix} \quad \text{and} \quad \|\mathbf{r}^{(1)}\|_2 = 0.2410.$$

Conjugated preconditioned gradient method with $P = D$

We set $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)} = P^{-1}\mathbf{r}^{(0)}$ and $\mathbf{p}^{(0)} = \mathbf{z}^{(0)}$. For $k = 0$, we have:

$$\alpha_0 = \frac{(\mathbf{p}^{(0)})^T \mathbf{r}^{(0)}}{(\mathbf{p}^{(0)})^T A \mathbf{p}^{(0)}} = \frac{(\mathbf{z}^{(0)})^T \mathbf{r}^{(0)}}{(\mathbf{z}^{(0)})^T A \mathbf{z}^{(0)}}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{p}^{(0)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{z}^{(0)}$$

$$\mathbf{r}^{(1)} = \mathbf{r}^{(0)} - \alpha_0 A \mathbf{p}^{(0)} = \mathbf{r}^{(0)} - \alpha_0 A \mathbf{z}^{(0)}.$$

We see that the first iteration $\mathbf{x}^{(1)}$ matches with the one obtained by the preconditioned gradient method.

We then complete the first iteration of the preconditioned conjugate gradient method:

$$P\mathbf{z}^{(1)} = \mathbf{r}^{(1)} \quad \Leftrightarrow \quad \mathbf{z}^{(1)} = P^{-1}\mathbf{r}^{(1)} = \begin{pmatrix} 0.0896 \\ -0.0537 \end{pmatrix}$$

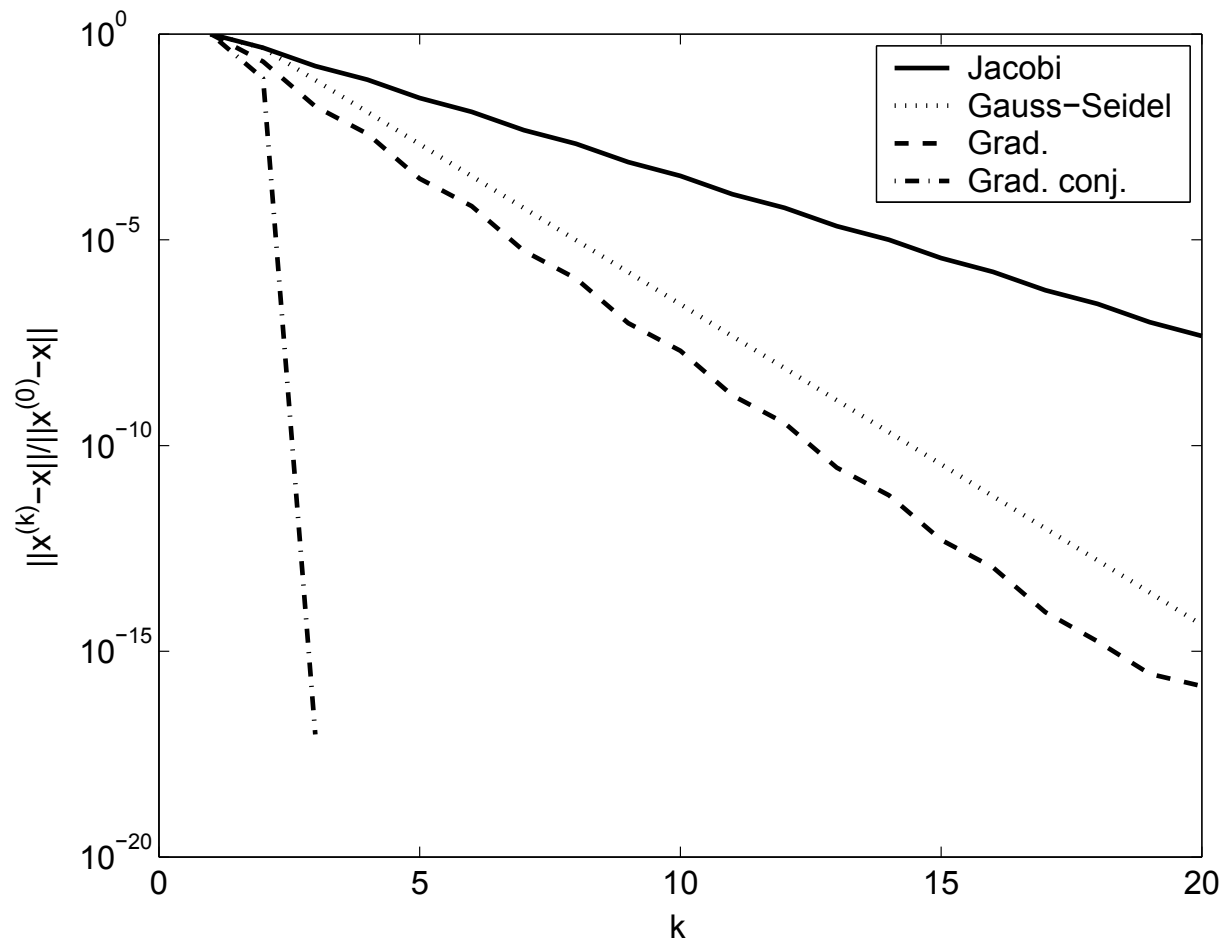
$$\beta_0 = \frac{(A\mathbf{p}^{(0)})^T \mathbf{z}^{(1)}}{(A\mathbf{p}^{(0)})^T A\mathbf{p}^{(0)}} = \frac{(A\mathbf{z}^{(0)})^T \mathbf{z}^{(1)}}{(A\mathbf{z}^{(0)})^T \mathbf{z}^{(0)}} = -0.0077$$

$$\mathbf{p}^{(1)} = \mathbf{z}^{(1)} - \beta_0 \mathbf{p}^{(0)} = \mathbf{z}^{(1)} - \beta_0 \mathbf{z}^{(0)} = \begin{pmatrix} 0.0838 \\ -0.0602 \end{pmatrix}.$$

At the second iteration, with the four different methods, we have:

Method	$\mathbf{x}^{(2)}$	$\mathbf{r}^{(2)}$	$\ \mathbf{r}^{(2)}\ _2$
Jacobi	$\begin{pmatrix} 0.6667 \\ -0.0833 \end{pmatrix}$	$\begin{pmatrix} -0.2500 \\ -0.4167 \end{pmatrix}$	0.4859
Gauss-Seidel	$\begin{pmatrix} 0.5417 \\ -0.1806 \end{pmatrix}$	$\begin{pmatrix} 0.0972 \\ 0 \end{pmatrix}$	0.0972
PG	$\begin{pmatrix} 0.6070 \\ -0.1877 \end{pmatrix}$	$\begin{pmatrix} -0.0263 \\ -0.0438 \end{pmatrix}$	0.0511
PCG	$\begin{pmatrix} 0.60000 \\ -0.2000 \end{pmatrix}$	$\begin{pmatrix} -0.2220 \\ -0.3886 \end{pmatrix} \cdot 10^{-15}$	$4.4755 \cdot 10^{-16}$

Behavior of the relative error applied to the system (16) :



Example 3. Let now consider another example:

$$\begin{cases} 2x_1 + x_2 &= 1 \\ -x_1 + 3x_2 &= 0 \end{cases} \quad (17)$$

whose solution is $x_1 = 3/7$, $x_2 = 1/7$.

Preliminary convergence studies

The associated matrix is $A = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix}$.

- A is strictly diagonal dominant by row. Hence Jacobi and Gauss-Seidel methods converge.
- A is regular, tridiagonal with non-zero diagonal elements. Then $\rho(B_{GS}) = \rho(B_J)^2$. Therefore we expect a quicker convergence of Gauss-Seidel w.r.t. Jacobi.
- A is **not s.p.d.**, therefore we have no idea if the gradient or the conjugate gradient converge.

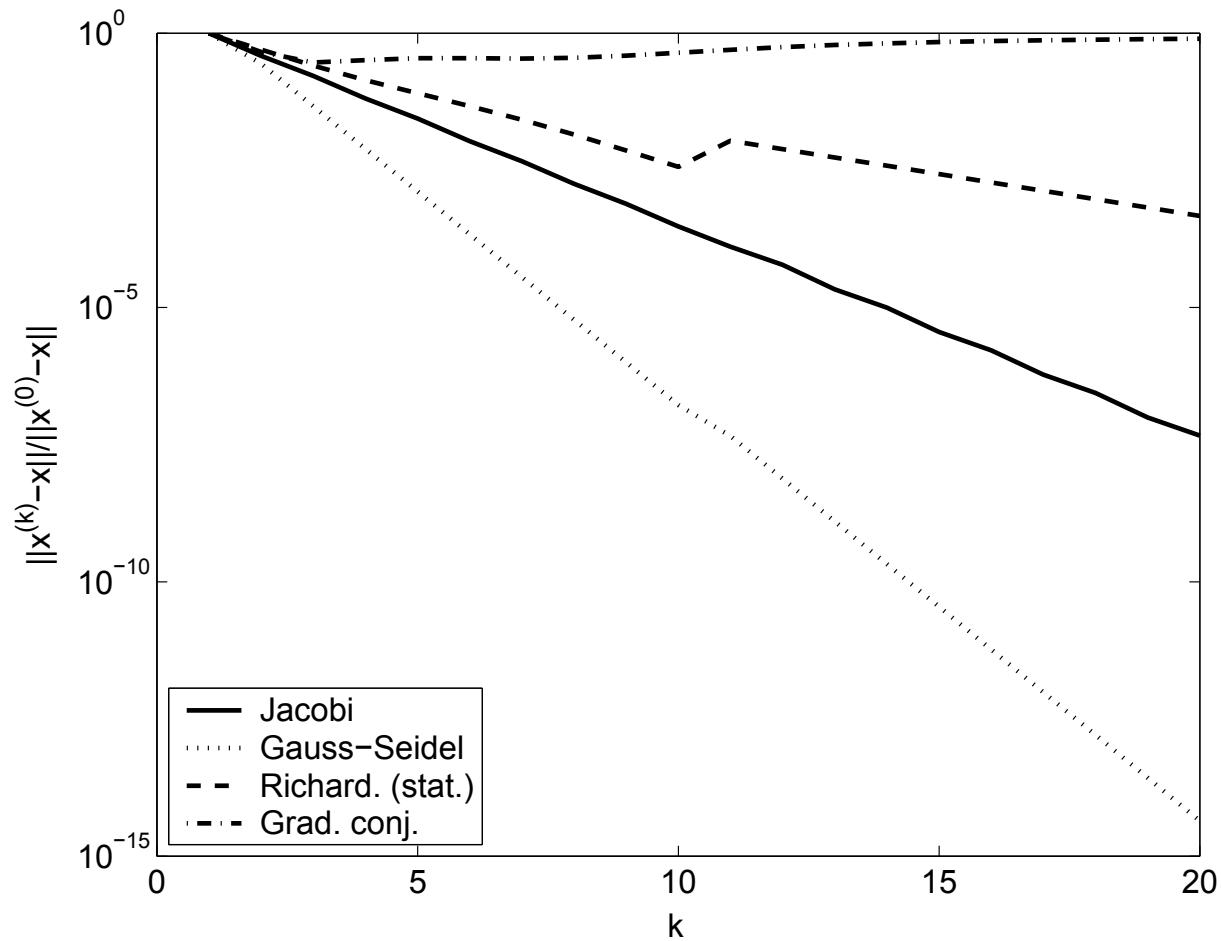
We approximate the solution with an iterative method starting from

$$\mathbf{x}^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

The following figure shows the value of $\frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}^{(0)} - \mathbf{x}\|}$ for the Jacobi, Gauss-Seidel, Richardson stationary (preconditioned with $\alpha = 0.5$ and $P = D = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$), and the preconditioned (with $P = D$) conjugate gradient methods.

Remark that this time the preconditioned conjugate gradient method doesn't converge.

Behavior of the relative error applied to the system (17) :



Convergence Criteria

(Sec. 5.12)

We have the following error bound:

If A is *s.p.d*, then

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}\|} \leq K(A) \frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|}. \quad (18)$$

The relative error at the iteration k is bounded by the condition number of A times the residual scaled with the right hand side.

We can also use another relation in case of a preconditioned system:

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}\|} \leq K(P^{-1}A) \frac{\|P^{-1}\mathbf{r}^{(k)}\|}{\|P^{-1}\mathbf{b}\|}.$$

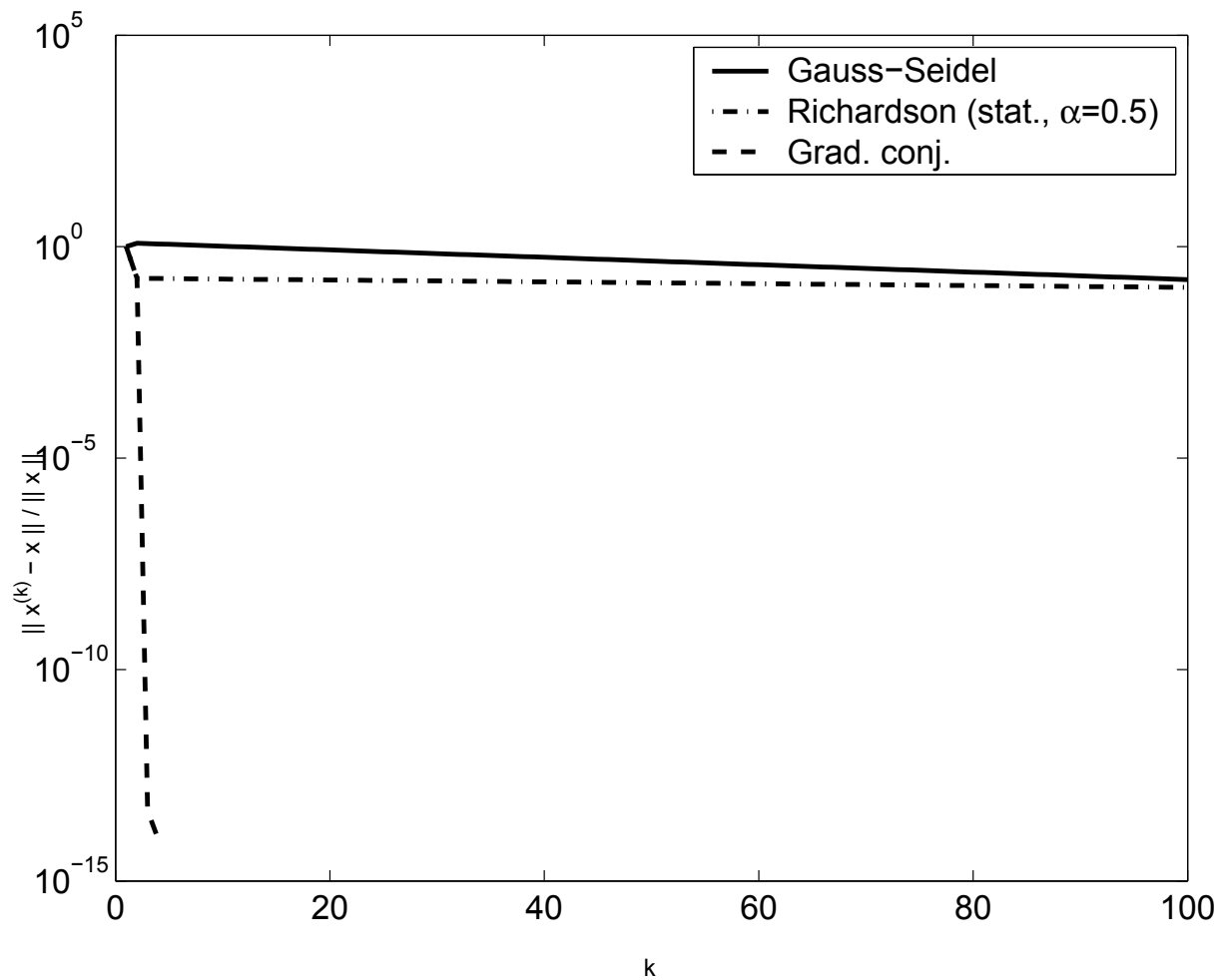
Some examples of convergence of iterative methods applied to linear systems of the kind $A\mathbf{x} = \mathbf{b}$.

Example 4. Lets start with the matrix

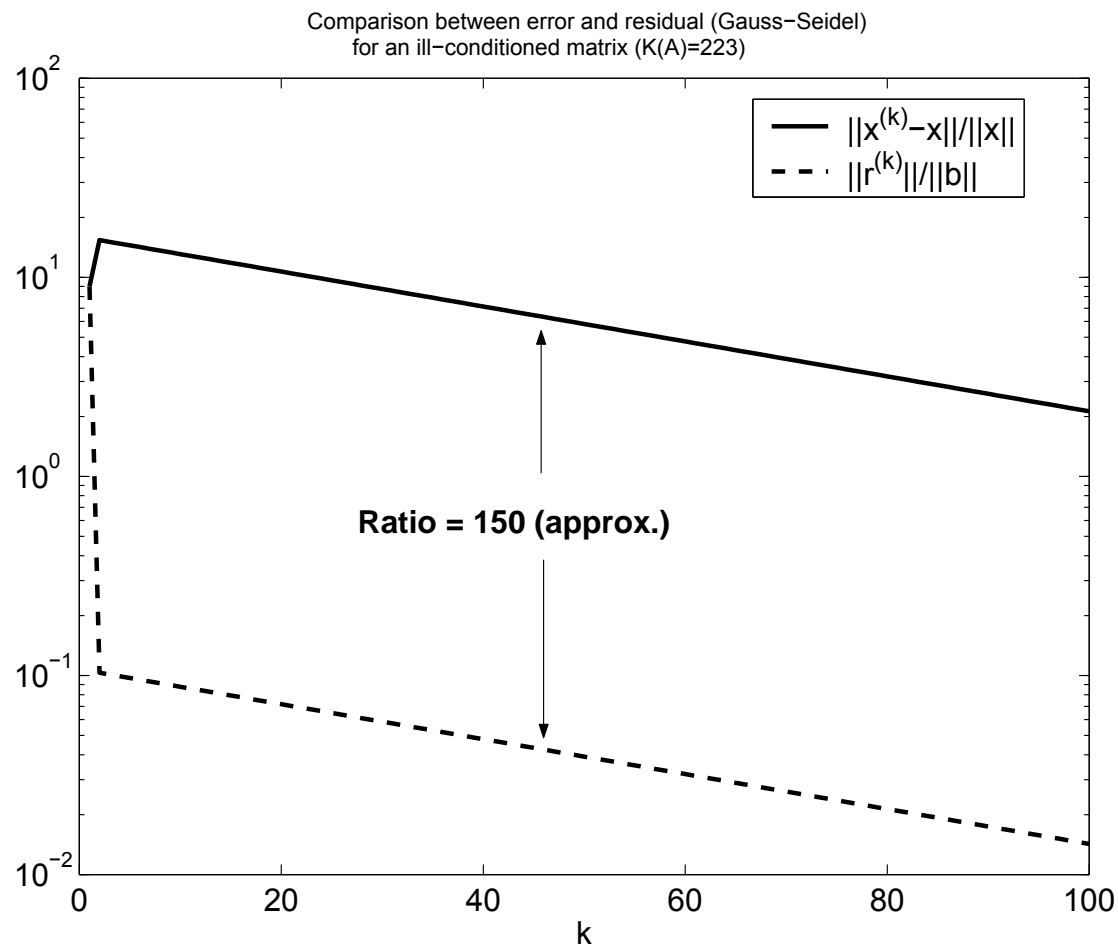
$$A = \begin{pmatrix} 5 & 7 \\ 7 & 10 \end{pmatrix} \quad (19)$$

The condition number of this matrix is $K(A) \approx 223$. We consider the Gauss-Seidel method and stationary preconditioned Richardson method with $\alpha = 0.5$ and $P = \text{diag}(A)$. We have that $\rho(B_{GS}) = 0.98$ and $\rho(B_{Rich}) = 0.98$, where $B_{Rich} = I - \alpha P^{-1}A$ is the iterative matrix for the stationary Richardson method.

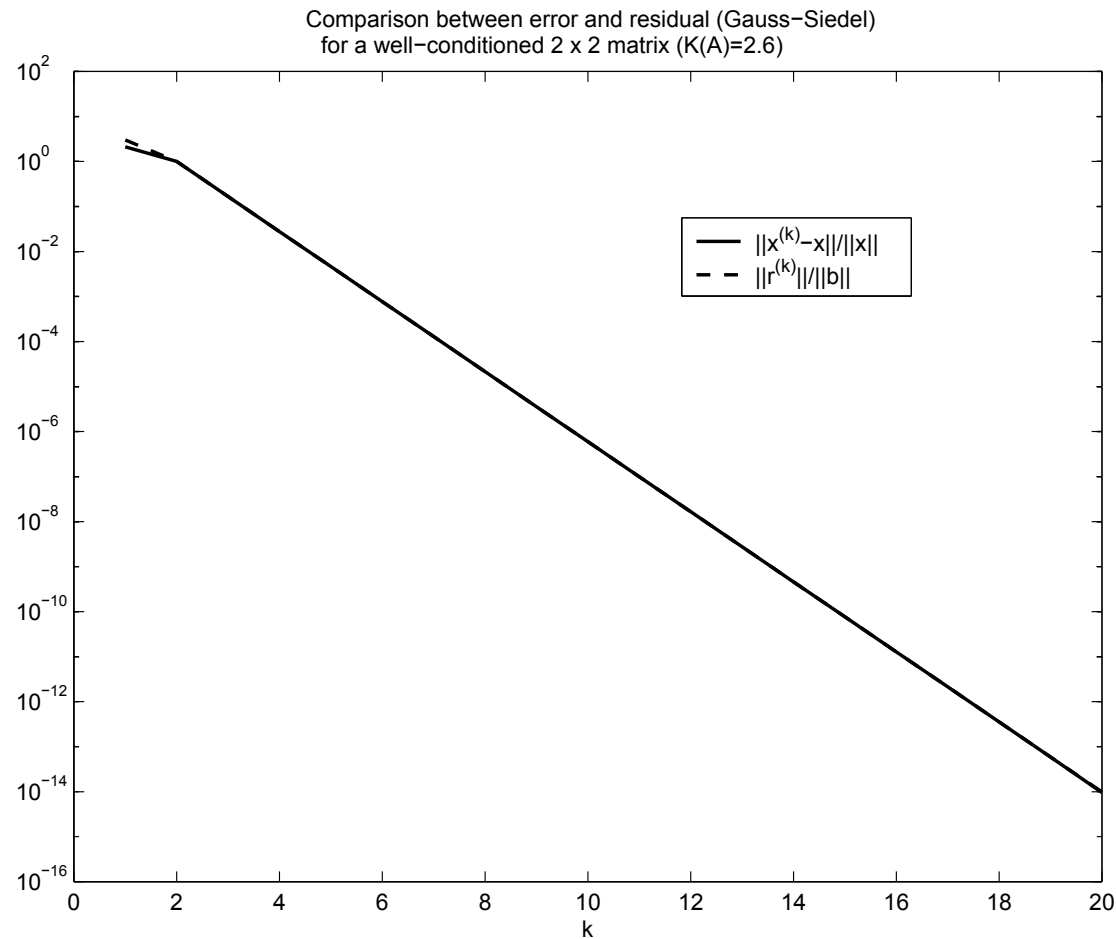
This figure shows the relative error behavior for Gauss-Seidel method and stationary preconditioned Richardson method and the conjugate gradient preconditioned with the same matrix



Comparison between error and residual (Gauss-Seidel) (19). Recall that $K(A) \approx 223$



We can compare the previous figure with the same curves for well conditioned matrix 2×2 ($K(A) \approx 2.6$):



Comparison between the relative error and the residual, for a well conditioned matrix.

Example 5. We take the Hilbert matrix and we solve a linear system for different size n . We set $P = \text{diag}(A)$ and use the preconditioned gradient method. The stopping tolerance is set to 10^{-6} on the relative residual. We take $\mathbf{x}^{(0)} = \mathbf{0}$.

We evaluate the relative error

```
for j=2:7;
    n=2*j; i=j-1; nn(i)=n;
    A=hilb(n);
    x_ex=ones(n,1); b=A*x_ex;
    Acond(i)=cond(A);
    tol=1.e-6; maxit=10000;
    R=diag(diag(A));
    x0=zeros(n,1);
    [x,iter_gr(i)]=gradient(A,b,x0,maxit,tol,R);
    error_gr(i)=norm(x-x_ex)/norm(x_ex);
end
```

For the iterative method, we set a tolerance of 10^{-6} on the relative residual. Because the matrix is ill conditioned, it is to be expected to get a relative error in the solution greater than 10^{-6} (see inequality (18)).

		Gradient method		
n	$K(A)$	Error	Iterations	Residual
4	1.55e+04	8.72e-03	995	1.00e-06
6	1.50e+07	3.60e-03	1813	9.99e-07
8	1.53e+10	6.30e-03	1089	9.96e-07
10	1.60e+13	7.99e-03	875	9.99e-07
12	1.67e+16	5.09e-03	1355	9.99e-07
14	2.04e+17	3.91e-03	1379	9.98e-07

Some observations

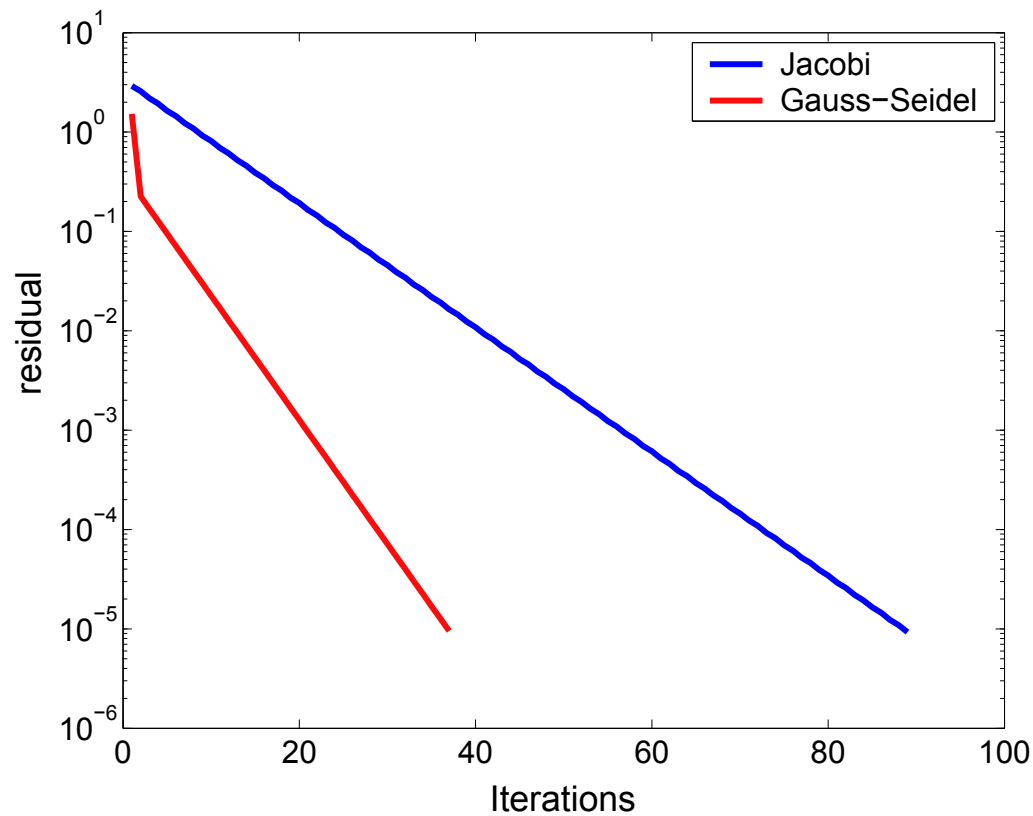
Memory and computational costs

Computational cost (*flops*) and used memory (*bytes*): we consider the Cholesky and the conjugate gradient methods for sparse matrices of size n (originated in the approximation of solutions to the Poisson equation in the finite elements method) with m non zero elements.

		Cholesky		Conjugate gradient		flops(Chol.)/ flops(GC)	Mem(Chol.)/ Mem(GC)
n	m/n^2	flops	Memory	flops	Memory		
47	0.12	8.05e+03	464	1.26e+04	228	0.64	2.04
83	0.07	3.96e+04	1406	3.03e+04	533	1.31	2.64
150	0.04	2.01e+05	4235	8.86e+04	1245	2.26	3.4
225	0.03	6.39e+05	9260	1.95e+05	2073	3.27	4.47
329	0.02	1.74e+06	17974	3.39e+05	3330	5.15	5.39
424	0.02	3.78e+06	30815	5.49e+05	4513	6.88	6.83
530	0.01	8.31e+06	50785	8.61e+05	5981	9.65	8.49
661	0.01	1.19e+07	68468	1.11e+06	7421	10.66	9.23

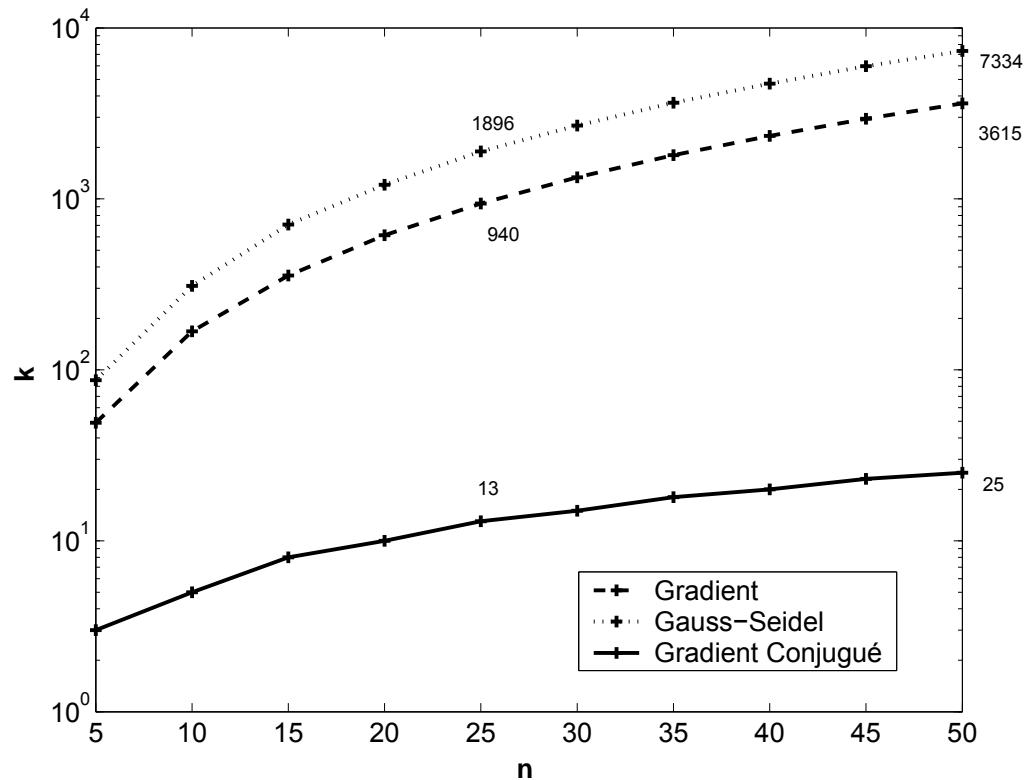
Iterative methods: Jacobi, Gauss-Seidel

Behaviour of the error for a well conditioned matrix ($K = 20$)



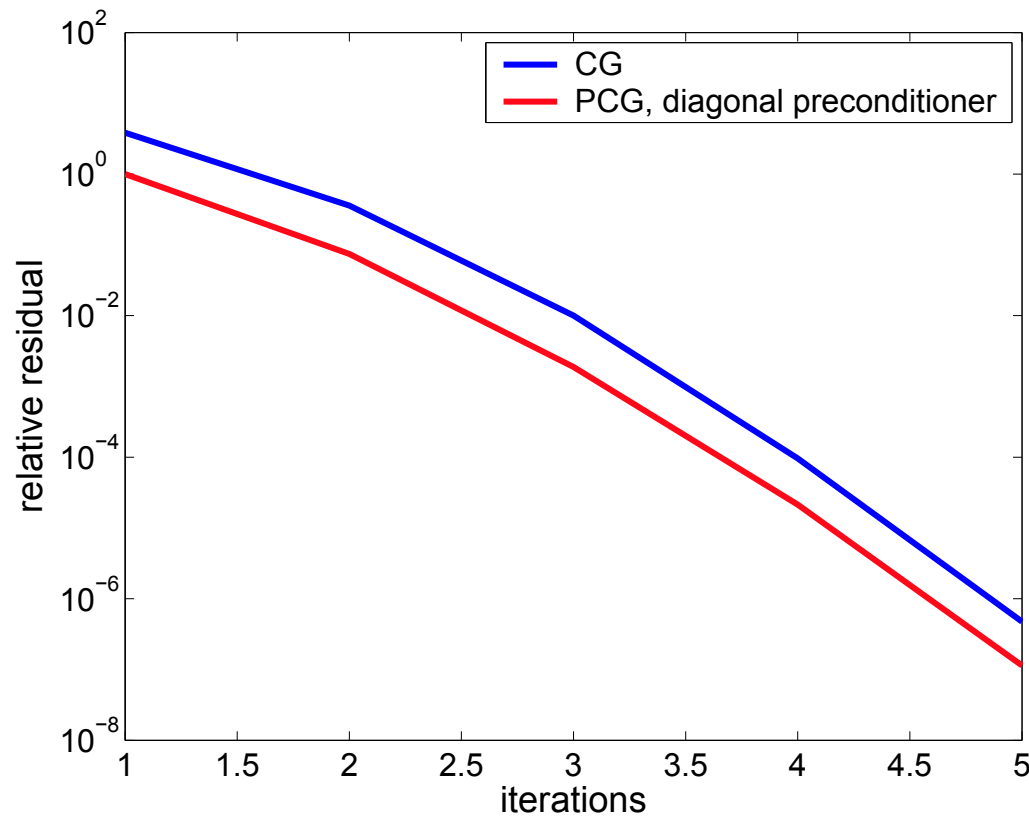
Iterative methods: G-S, Gradient, Gradient Conjugate

Number of iterations as function of the size n of a stiffness matrix, tolerance 10^{-6}



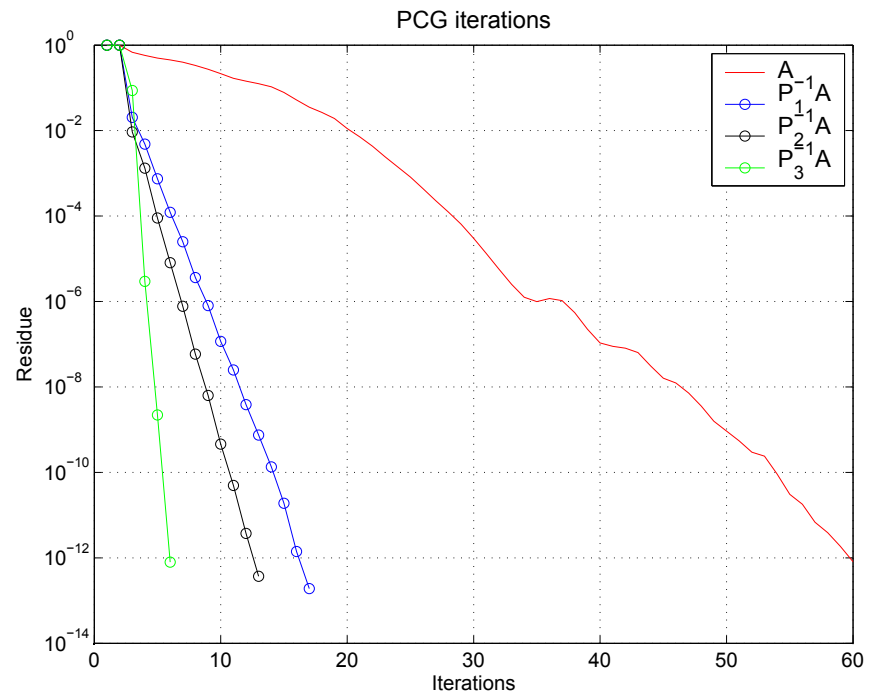
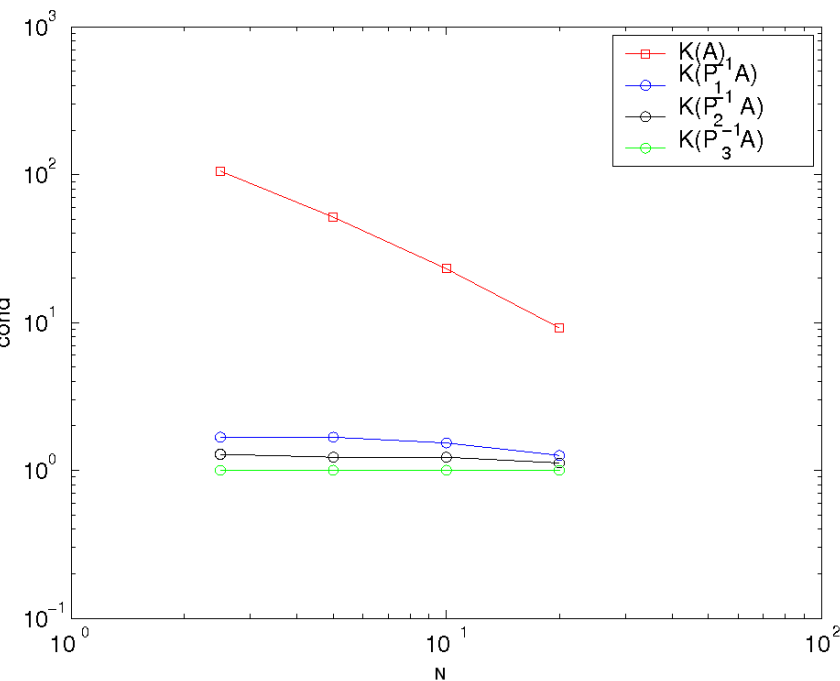
Preconditioning

The convergence conjugate gradient and the preconditioned conjugate gradient methods with a diagonal preconditioner ($K = 4 \cdot 10^8$)



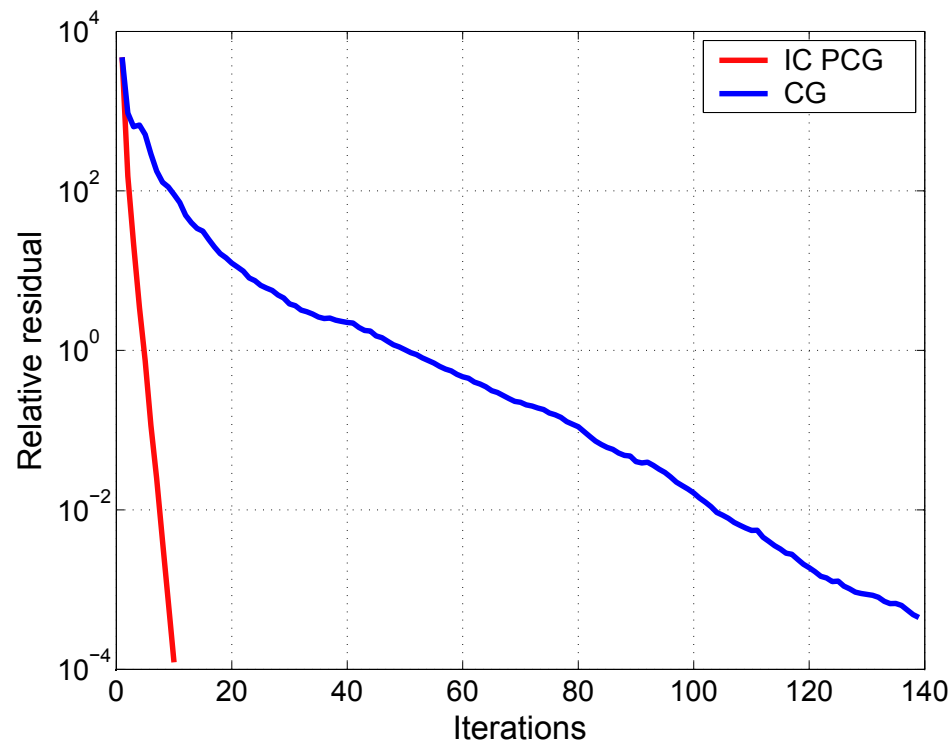
Preconditioning

Role of the preconditioning over the condition number of a matrix
(approximation by the finite element method of the Laplacian operator)



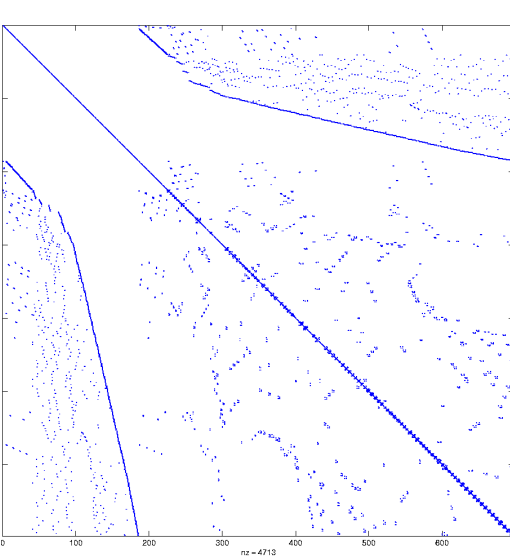
Preconditioning

The convergence of the conjugate gradient method and the preconditioned conjugate gradient method with a preconditioner based on an Cholesky incomplete decomposition for a sparse matrix originated from the finite element method ($K = 1.5 \cdot 10^3$)

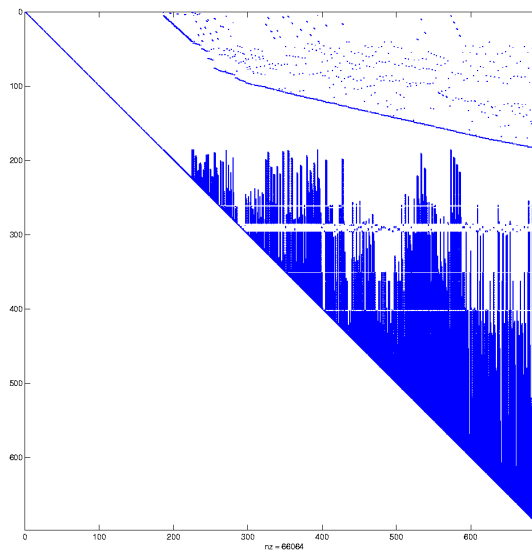


Pre-conditioning

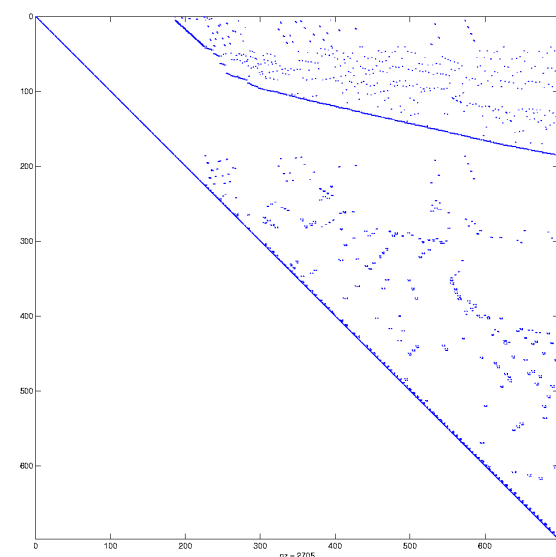
Comparison between the non zero elements of the sparse matrix A from the previous example, its Cholesky factor R and the matrix \tilde{R} obtained by the Cholesky incomplete decomposition:



A



R



\tilde{R}

Non-linear systems

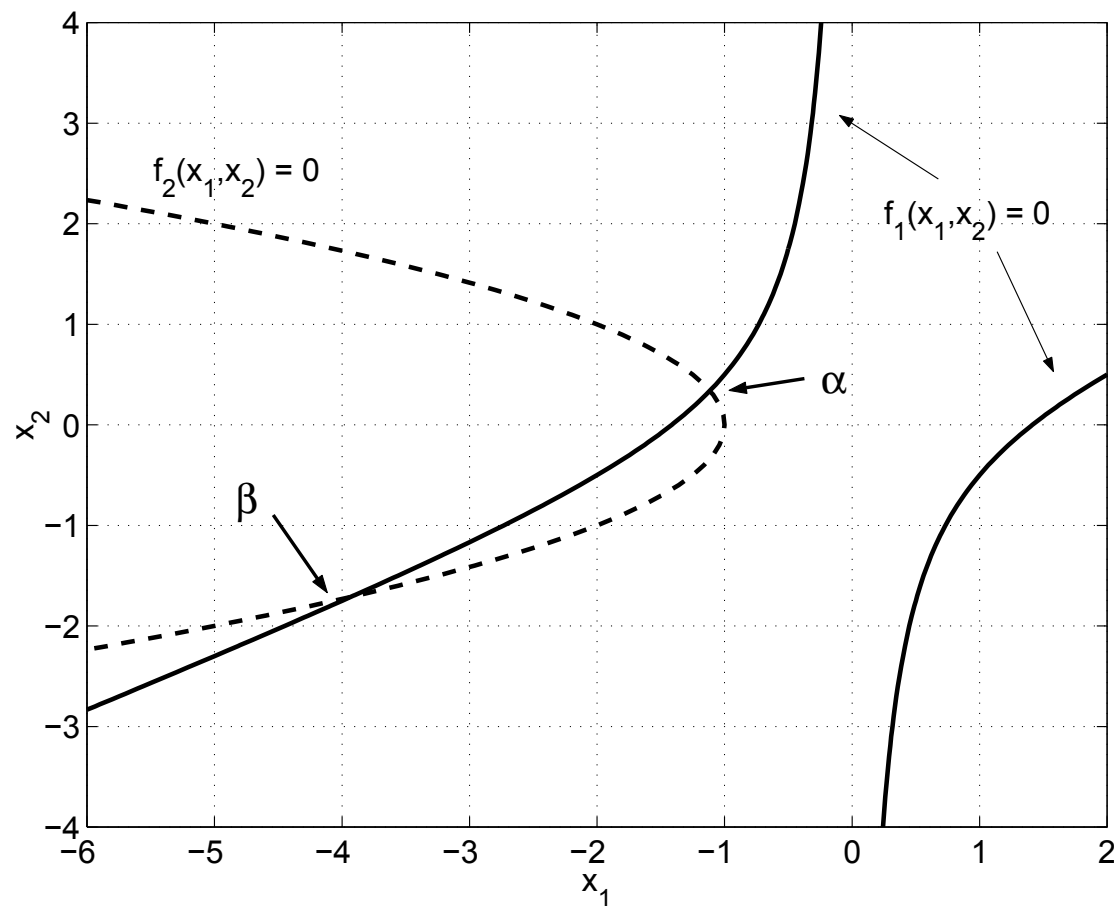
Example 6. Lets consider the following system of non-linear equations:

$$\begin{cases} x_1^2 - 2x_1x_2 = 2 \\ x_1 + x_2^2 = -1. \end{cases} \quad (20)$$

This system can be written in the form:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \quad \text{i.e.} \quad \begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases}$$

where $\mathbf{f} = (f_1, f_2)$, $f_1(x_1, x_2) = x_1^2 - 2x_1x_2 - 2$ and $f_2(x_1, x_2) = x_1 + x_2^2 + 1$.



Curves $f_1 = 0$ and $f_2 = 0$ in the square $-6 \leq x_1 \leq 2, -4 \leq x_2 \leq 4$

We want to generalise the **Newton method** for the case of non-linear systems. To do this, we define the **Jacobian** matrix of the vector \mathbf{f} :

$$J_{\mathbf{f}}(\mathbf{x} = (x_1, x_2)) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 2x_2 & -2x_1 \\ 1 & 2x_2 \end{bmatrix}.$$

If $J_{\mathbf{f}}(\mathbf{x}^{(k)})$ is invertible, the Newton method for non linear systems is written : *Let $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)})$, we compute for $k = 0, 1, 2, \dots$*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [J_{\mathbf{f}}(\mathbf{x}^{(k)})]^{-1} \mathbf{f}(\mathbf{x}^{(k)}), \quad k = 0, 1, 2, \dots \quad (21)$$

We can write (21) as

$$[J_{\mathbf{f}}(\mathbf{x}^{(k)})](\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = -\mathbf{f}(\mathbf{x}^{(k)}), \quad k = 0, 1, 2, \dots \quad (22)$$

Description of the first step of the algorithm: given the vector $\mathbf{x}^{(0)} = [1, 1]^T$.
We calculate:

$$J_{\mathbf{f}}(\mathbf{x}^{(0)}) = \begin{bmatrix} 0 & -2 \\ 1 & 2 \end{bmatrix}.$$

We determine $\mathbf{x}^{(1)}$ as solution of the equation :

$$[J_{\mathbf{f}}(\mathbf{x}^{(0)})](\mathbf{x}^{(1)} - \mathbf{x}^{(0)}) = -\mathbf{f}(\mathbf{x}^{(0)}).$$

And we continue with (22).