Advanced Numerical Algorithms (MU4IN920)

Lecture 2: Iterative methods for solving linear systems

Stef Graillat

Sorbonne Université

Summary of the previous lecture

Monte Carlo Method:

- Basic statistics: random number and generation
- Monte Carlo method and calculation of integrals
- Monte Carlo method and optimization
- Monte Carlo method and counting
- Introduction to Derivatives in Finance
- Calculating the price of an option

Objectives

1. Solving linear systems by iterative methods (Jacobi, Gauss-Seidel, SOR)
2. Conjugate gradient method
3. Krylov subspace methods

References

- An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, J. Shewchuk, 1994
- Scientific Computing, An Introductory Survey, Michael .T. Heath, McGraw-Hill, 2002
- Scientific Computing with Case Studies, Dianne P. O'Leary, SIAM, 2009
- Iterative methods for sparse linear systems, Y. Saad, SIAM, 2007
- Linear and Nonlinear Programming, Luenberger, Ye, 3e édition, Springer, 2010
- Algèbre linéaire numérique : Cours et exercices, Allaire, Kaber Sidi, Ellipses, 2002

Why use iterative methods

We want to solve an equation of the form:

$$Ax = b$$

The direct methods provide the solution x^* in a finite number of operations.

But

- the complexity is in $\mathcal{O}(n^3)$
- do not take into account the sparsity of the matrix (many coefficients are zero).

Iterative methods

- We construct a sequence of vectors (x_k), $k = 0, 1, \ldots$ which tends to x^*.
- The starting point is an approximation x_0 of x^*
- To construct this sequence, we use linearity to decompose the matrix A into an easily invertible part and a remainder part.

General principle

We decompose the matrix A into $A = M - N$, so that M is easily invertible. Then,

$$Ax = b \quad \Leftrightarrow \quad Mx = Nx + b$$

We compute the sequence of vectors (x_i) from a vector x_0 chosen arbitrarily and the relation:

$$M\, x_{k+1} = N\, x_k + b \quad \Leftrightarrow \quad x_{k+1} = M^{-1}N\, x_k + M^{-1}b$$

That is to say

$$\begin{cases} x_0 & \text{given} \\ x_{k+1} & = \quad M^{-1}N\, x_k + M^{-1}b \end{cases}$$

Let's define the problem

Let $C = M^{-1}N$, and $d = M^{-1}b$. We must therefore study the recurrent sequence

$$\begin{cases} x^0 & \text{given} \\ x_{k+1} = & Cx_k + d \end{cases}$$

With:

- x^* is a fixed point of the linear function

$$x \quad \mapsto \quad C\,x + d$$

Question:

- Under what conditions will the sequence converge?

Matrix norms

A matrix norm is a norm defined on $\mathcal{M}_n(\mathbb{C})$ which is compatible with the matrix multiplication, i.e.:

$$\begin{array}{rccc} \|\cdot\| & : & \mathcal{M}_n(\mathbb{C}) & \rightarrow & \mathbb{R}^+ \\ & & A & \mapsto & \|A\| \end{array}$$

such that $\forall A, B \in \mathcal{M}_n(\mathbb{C})$ and $\forall \lambda \in \mathbb{C}$:

- Point-separating: $\|A\| = 0 \Leftrightarrow A = 0$,
- Homogeneity: $\|\lambda A\| = |\lambda| \cdot \|A\|$,
- Triangular inequality: $\|A + B\| \leq \|A\| + \|B\|$,
- $\|A \cdot B\| \leq \|A\| \cdot \|B\|$.

Example of norms

Frobenius norm: $\forall A \in \mathcal{M}_n(\mathbb{C})$

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2} = \sqrt{\mathrm{Tr}\left(A^T A\right)}$$

Subordinate matrix norms:
Let $\|\cdot\|_v$ be a vector norm defined on $\mathbb{C}^n$ the function which $\forall A \in \mathcal{M}_n(\mathbb{C})$ associates

$$\|A\| = \max_{x \in \mathbb{C}^n, x \neq 0} \frac{\|Ax\|_v}{\|x\|_v}$$

is a matrix norm called a subordinate matrix norm

Example of subordinate matrix norms

Let the vector norm $\|\cdot\|_\infty$:
$\forall x \in \mathbb{C}^n$

$$\|x\|_\infty = \max_{1\le i\le n} |x_i|$$

The subordinate matrix norm $\|\cdot\|_\infty$ on $\mathcal{M}_n(\mathbb{C})$:
$\forall A \in \mathcal{M}_n(\mathbb{C})$

$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$$

Matrix norms and vector norms

Definition 1

A matrix norm $\|\cdot\|$ is compatible with a vector norm $\|\cdot\|_v$ if $\forall x$

$$\|Ax\|_v \quad \leq \quad \|A\| \, \|x\|_v$$

Properties:

- For any matrix norm $\|\cdot\|$, there exists a vector norm with which it is compatible.
- Any subordinate matrix norms is compatible with its vector norm.

Spectral radius

Definition 2

Let $A \in \mathcal{M}_n(\mathbb{C})$, the spectral radius of A is defined by

$$\rho\,(A) \quad = \quad \max_{1 \le i \le n} |\lambda_i|$$

where λ_i are the eigenvalues of A

Remark: $\rho(\cdot)$ is not a norm.

Properties:

- For any matrix norm $\|\cdot\|$ and for any matrix A:

$$\rho(A) \quad \le \quad \|A\|$$

- $\forall A \in \mathcal{M}_n(\mathbb{C})$, $\forall \varepsilon \in \mathbb{R}^+$, there exists a subordinate matrix norm $\|\cdot\|_*$ such that:

$$\rho(A) \quad \le \quad \|A\|_* \quad \le \quad \rho(A) + \varepsilon$$

Convergence

Let's go back to the convergence of the sequence:

$$\begin{cases} x_0 & \text{given} \\ x_{k+1} = & Cx_k + d \end{cases}$$

Theorem 1

$\forall C \in \mathcal{M}_n(\mathbb{C})$, if there exists a subordinate matrix norm $\|\cdot\|$ such that

$$\|C\| \quad < \quad 1$$

then

1. *The equation $x = Cx + d$ has a unique solution x^*.*
2. *The sequence $x_k \to x^*$ whatever x_0 is.*

Proof

Existence of a solution:

$$\rho(C) \quad \leq \quad \|C\| \quad < \quad 1$$

So the eigenvalues λ of C are such that $|\lambda| < 1$.

It means that the matrix $I - C$ is invertible

So there exits a unique solution for the equation

$$x \quad = \quad Cx + d$$

We call this solution x^*

Proof (cont'd)

Convergence:

Let $e_k = x_k - x^*$ We can deduce a relation between e_k and e_{k-1}. Indeed

$$\begin{aligned} Ce_{k-1} &= C(x_{k-1} - x^*) \\ &= C(x_{k-1}) - C(x^*) \\ &= C(x_{k-1}) + d - x^* \end{aligned}$$

As a consequence $e_k = C\, e_{k-1}$ for $k = 1, 2, \ldots$
So we have

$$e_k = C^k e_0$$

Let the subordinate matrix norm $\|\cdot\|$ and its vector norm $\|\cdot\|_v$ such that $\|C\| < 1$:

$$\|e_k\|_v \leq \|C\|^k \|e_0\|_v$$

So $e_k \to 0$ and $x_k \to x^*$

Other conditions

Given C, how to know if the sequence will converge?

General case:

Theorem 2

There is equivalence between the following propositions:

- *C is a convergent matrix (i.e.) C^k goes to 0)*
- $\rho(C) < 1$
- *There exists a subordinate matrix norm such that $\|C\| < 1$.*

Jacobi method

The Jacobi method corresponds to the splitting

$$A = D - U - L$$

with

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

$$-L = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix} \qquad -U = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{n1} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Jacobi method (cont'd)

The Jacobi method corresponds to the splitting

$$A = D - U - L$$

with

$$M = D \quad \text{et} \quad N = L + U$$

The iteration $Mx^{(k+1)} = Nx^{(k)} + b$ can be written as

$$Dx^{(k+1)} = (L + U)x^{(k)} + b$$

that is to say

$$a_{ii}x_i^{(k+1)} = (b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)})$$

Jacobi method (cont'd)

From a vector $x^{(0)}$, we construct the sequence $(x^{(k)})_{k\geq 0}$ in the following way

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

This method is only defined if all a_{ii} are nonzero

```
function x=Jacobi(A,y,xo,itmax)
n = length(y);
x = xo;
xold = x;
for it=1:itmax
   for i=1:n
      x(i) = (y(i)-A(i,[1:i-1,i+1:n])*xold([1:i-1,i+1:n]))/A(i,i);
   end
   xold=x;
end
```

Jacobi method (cont'd)

Carl Gustav Jakob Jacobi
German mathematician
10 December 1804 – 18 February 1851

Gauss-Seidel method

The Gauss-Seidel method corresponds to the splitting

$$A = D - U - L$$

with

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

$$-L = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix} \qquad -U = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{n1} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Gauss-Seidel method (cont'd)

The Gauss-Seidel method corresponds to the splitting

$$A = D - U - L$$

with

$$M = D - L \quad \text{et} \quad N = U$$

The iteration $Mx^{(k+1)} = Nx^{(k)} + b$ can be written as

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b$$

that is to say

$$a_{ii}x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})$$

Gauss-Seidel method (cont'd)

From a vector $x^{(0)}$, we construct the sequence $(x^{(k)})_{k\geq 0}$ in the following way

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})$$

This method is only defined if all a_{ii} are nonzero

```
function x=GS(A,y,xo,itmax)
n = length(y);
x = xo;
for it=1:itmax
  for i=1:n
     x(i)=(y(i)-A(i,1:i-1)*x(1:i-1)-A(i,i+1:n)*x(i+1:n))/A(i,i);
  end
end
```

Gauss-Seidel method (cont'd)

Johann Carl Friedrich Gauss
German mathematician
30 April 1777 - 23 February 1855

Philipp Ludwig von Seidel
German mathematician
24 October 1821 - 13 August 1896

Comparison between Jacobi and Gauss-Seidel methods

Jacobi: only the elements of $x^{(k)}$ are used to calculate the elements of $x^{(k+1)}$.

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})$$

Gauss-Seidel: we use the new components of $x^{(k+1)}$ as soon as possible

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)})$$

Method of successive over-relaxation (SOR)

Given $\omega \in]0, 2[$, the SOR method corresponds to the splitting

$$A = D - U - L$$

avec

$$M = \frac{D}{\omega} - L \quad \text{et} \quad N = \frac{1-\omega}{\omega} D + U$$

From a vector $x^{(0)}$, we construct the sequence $(x^{(k)})_{k \geq 0}$ in the following way

$$x_i^{(k+1)} = \omega \frac{1}{a_{ii}} (b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}) + (1-\omega) x_i^{(k)}$$

Gradient descent (also often called steepest descent) method

We assume that the matrix A is symmetric positive definite ($A = A^T$ and for all $x \neq 0, x^T Ax > 0$)

- We can show that a critical point of the function

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

  is a solution of the equation $Ax = b$. Indeed $\nabla f(x) = Ax - b$
- We can solve the linear system $Ax = b$ by solving the following minimization problem

$$\min_x f(x)$$

In the sequel, we denote by $r(x) = b - Ax$ the residual. We can notice that $r(x) = -\nabla f(x)$

Steepest descent method

$x_0 =$ `intial approximation`
`for` $k = 0, 1, 2, \ldots$
    `compute` $p_k = -\nabla f(x_k) = r_k$
    $x_{k+1} = x_k + \alpha_k p_k$ `where` α_k `is the solution ot the`
        `problem` $\min_\alpha f(x_k + \alpha p_k)$
`end for`

We can find an explicit expression for α_k. Indeed

$$\begin{aligned} f(x_k + \alpha p_k) &= \frac{1}{2}(x_k + \alpha p_k)^T A (x_k + \alpha p_k) - b^T (x_k + \alpha p_k) \\ &= \frac{1}{2}\alpha^2 p_k^T A p_k + \alpha p_k^T A x_k - \alpha b^T p_k + \text{constant} \end{aligned}$$

The minimum of f with respect to α is obtained for

$$p_k^T A x_k + \alpha p_k^T A p_k - b^T p_k = 0$$

Steepest descent method (cont'd)

We can find an explicit expression for α_k. Indeed

$$\begin{aligned}
f(x_k + \alpha p_k) &= \frac{1}{2}(x_k + \alpha p_k)^T A (x_k + \alpha p_k) - b^T (x_k + \alpha p_k) \\
&= \frac{1}{2}\alpha^2 p_k^T A p_k + \alpha p_k^T A x_k - \alpha b^T p_k + \text{constant}
\end{aligned}$$

The minimum of f with respect to α is obtained for

$$p_k^T A x_k + \alpha p_k^T A p_k - b^T p_k = 0$$

that is to say

$$\alpha = -\frac{p_k^T (A x_k - b)}{p_k^T A p_k} = \frac{p_k^T r_k}{p_k^T A p_k}$$

Rate of convergence

Let x^* be the solution of the linear system $Ax = b$. We denote

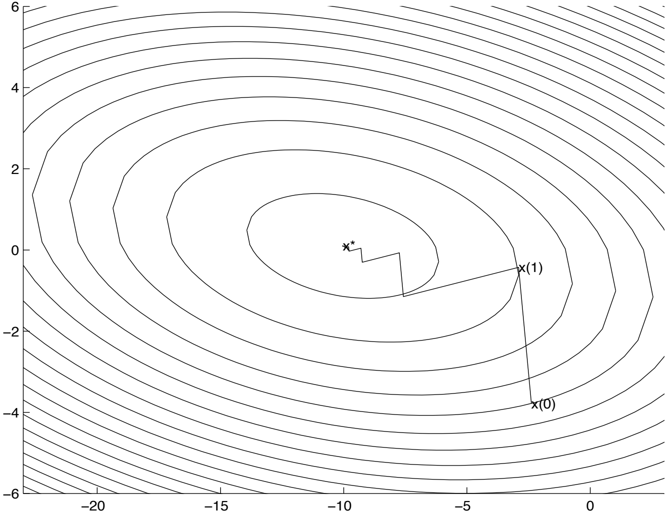
$$E(x) = \frac{1}{2}(x - x^*)^T A(x - x^*)$$

This function is minimal for $x = x^*$ and is a way to measure the convergence.

We can show that the steepest descent method has a rate of convergence

$$E(x_k) \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)^{2k} E(x_0)$$

Example



After 20 iterations, the error has been reduced by a factor 10^{-5}

A-conjugate directions

- As we have seen, the steepest descent algorithm can be very low
- Here we want to modify the steepest descent algorithm so that it converges in at most n steps (n being the size of the matrix)
- The idea is to find n linearly independent vectors p_k, $k = 0, \ldots, n-1$ that are A-conjugate,

$$p_k^T A p_j = 0, \quad k \neq j$$

- As they are linearly independent, they form a basis and

$$x^* - x_0 = \sum_{j=0}^{n-1} \alpha_j p_j$$

A-conjugate directions (cont'd)

- We have

$$x^* - x_0 = \sum_{j=0}^{n-1} \alpha_j p_j$$

- By multiplying the left hand side by $p_k^T A$, we obtain

$$p_k^T A(x^* - x_0) = p_k^T (b - Ax_0) = p_k^T r_0$$

- By multiplying the right hand side by $p_k^T A$, we obtain

$$p_k^T A \sum_{j=0}^{n-1} \alpha_j p_j = \alpha_k p_k^T A p_k$$

- As a consequence

$$\alpha_k = \frac{p_k^T r_0}{p_k^T A p_k}$$

The conjugate gradient method

We can solve the linear system $Ax^* = b$ with the following algorithm:

```
Pick $x_0$ and $A$-conjugate directions $p_k$, $k = 0, \dots, n-1$

for $k = 0, 1, \dots, n-1$
    Set $\alpha_k = \frac{p_k^T r_0}{p_k^T A p_k}$
    Let $x_{k+1} = x_k + \alpha_k p_k$
end for
```

At the end, $x_n = x^*$. Moreover as $p_k^T r_0 = p_k^T r_k$ (due to A-conjugacy), we obtain α_k with the same formula as for the classic steepest descent algorithm

It remains to find n A-conjugate directions

Gram-Schmidt algorithm

Given n linearly independent vectors v_k, $k = 0, \ldots, n-1$, we can compute n A-conjugate vectors spanning the same space

Let $p_0 = v_0$

for $k = 0, 1, \ldots, n-2$

$$\texttt{compute} \quad p_{k+1} = v_{k+1} - \sum_{j=0}^{k} \frac{p_j^T A v_{k+1}}{p_j^T A p_j} p_j$$

end for

Conjugate gradient method

The conjugate gradient algorithm is a special case of the conjugate direction algorithm. In this case, we intertwine the calculation of the new x vector and the new p vector. In fact, the set of linearly independent vectors v_k we use in the Gram-Schmidt process is just the set of residuals r_k.

Let x_0 an initial guess, $r_0 = b - Ax_0$ and $p_0 = r_0$
for $k = 0, 1, \ldots, n-1$
    Compute $\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$
    Compute the new search direction p_{k+1} by
        Gram-Schmidt on r_{k+1} and the previous p
        vectors to make p_{k+1} A-conjugate to the
        previous directions.
end for

Conjugate gradient method (cont'd)

It turns out that $p_j^T A r_{k+1} = 0$ pour $j < k$. As a consequence Gram-Schmidt formula reduces to

$$p_{k+1} = r_{k+1} - \frac{p_k^T A r_{k+1}}{p_k^T A p_k} p_k$$

Let x_0 an initial guess, $r_0 = b - Ax_0$ and $p_0 = r_0$
for $k = 0, 1, \ldots, n-1$
    Compute $\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$ (equivalent to $\frac{r_k^T r_k}{p_k^T A p_k}$)
        $x_{k+1} = x_k + \alpha_k p_k$
        $r_{k+1} = r_k - \alpha_k A p_k$
    Compute the new search direction
        $\beta_k = -\frac{p_k^T A r_{k+1}}{p_k^T A p_k}$ (equivalent a $\frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$)
        $p_{k+1} = r_{k+1} + \beta_k p_k$
end for

Conjugate gradient method (cont'd)

- After K, $K \leq n$, the algorithm terminates with $r_K = 0$ and $x_K = x^*$.
- We can show that

$$E(x_k) \leq \left(\frac{1-\sqrt{\kappa^{-1}}}{1+\sqrt{\kappa^{-1}}}\right)^{2k} E(x_0)$$

where κ is the condition number of A, $\kappa = \lambda_{\max}/\lambda_{\min}$

Krylov subspaces and Krylov methods

Definition 3

Given a matrix $A \in \mathbb{R}^{n\times n}$, a vector $r \in \mathbb{R}^n$, the Krylov subspace of order k generated by A and r, denoted $\mathcal{K}_k(A, r)$, is the linear subspace spanned by the vectors

$$r, Ar, A^2r, \dots, A^{k-1}r.$$

Krylov methods consist in searching the iterated x_k in the space $x_0 + \mathcal{K}_k(A, r_0)$ where $r_0 = b - Ax_0$.

Krylov methods and conjugate gradient method

- It can be shown that the conjugate gradient algorithm consists in searching $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ satisfying $r_k = b - Ax_k \perp \mathcal{K}_k(A, r_0)$

- We can also show that the conjugate gradient algorithm consists in finding x_k that minimizes the function $f(x) = \frac{1}{2}x^T Ax - b^T x$ on the subspace $x_0 + \mathcal{K}_k(A, r_0)$

Krylov methods

Alexei Nikolaevich Krylov
Russian naval engineer
15 August 1863 - 26 October 1945

Conclusion

- Efficient algorithms in practice, especially for the conjugate gradient algorithm
- Algorithms mostly used for sparse matrices