Practice 2
Resolution of systems of linear equations: iterative methods

Contents

1 Introduction

In computational mathematics, an iterative method attempts to solve a problem (for example, finding the solution of an equation or system of equations) by finding successive approximations to the solution starting from an initial guess. This approach is in contrast to direct methods, which attempt to solve the problem by a finite sequence of operations, and, in the absence of rounding errors, would deliver an exact solution (like solving a linear system of equations $\mathsf{A}\vec{x} = \vec{b}$ by Gaussian elimination). Iterative methods are often useful even for linear problems involving a large number of variables (sometimes of the order of millions), where direct methods would be prohibitively expensive (and in some cases impossible) even with the best available computing power.

We shall study two iterative methods to solve systems of linear equations: the Jacobi and Gauss-Seidel methods. In both cases we will work with a **square system (same number of unknowns and equations) with a unique solution**. Moreover, **all the diagonal elements of the coefficient matrix must be non-zero**.

2 The Jacobi method

Consider a system of linear equations satisfying the above mentioned assumptions. Suppose that its matrix expression is

$$\mathsf{A}\vec{x} = \vec{b}.$$

Consider the following decomposition of the coefficient matrix A:

$$\mathsf{A} = \mathsf{L} + \mathsf{D} + \mathsf{U},$$

where the matrix L is the "lower triangular part" of A, U is the "upper triangular part" of A and D is the "diagonal part" of A. Let us see an example with the aim of clarify this decomposition:

Example 1. Set the matrix

$$\mathsf{A} = \begin{bmatrix} 10 & 3 & 1 \\ 2 & -10 & 3 \\ 1 & 3 & 10 \end{bmatrix}$$

Then $\mathsf{A} = \mathsf{L} + \mathsf{D} + \mathsf{U}$, where

$$\mathsf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 3 & 0 \end{bmatrix}, \quad \mathsf{D} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & -10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \quad \mathsf{U} = \begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}.$$

Observe the following sequence of equivalences:

$$\begin{aligned} &\text{A vector } \vec{x} \text{ is solution of the system } \mathsf{A}\vec{x} = \vec{b} \Leftrightarrow \\ &(\mathsf{L} + \mathsf{D} + \mathsf{U})\vec{x} = \vec{b} \Leftrightarrow \\ &\mathsf{D}\vec{x} = \vec{b} - (\mathsf{L} + \mathsf{U})\vec{x} \Leftrightarrow \\ &\vec{x} = \mathsf{D}^{-1}[\vec{b} - (\mathsf{L} + \mathsf{U})\vec{x}] \end{aligned} \tag{1}$$

Notice that D is invertible because its diagonal elements are non-zero (by assumption). D^{-1} can be computed very easily because it is a diagonal matrix: D^{-1} is the diagonal matrix whose diagonal elements are the inverses of the diagonal elements of D.

The Jacobi method is an iterative technique based on equality (1). It consists of the following steps: take an initial approximation to the solution $\vec{x}_0$, replace it on the left hand side of (1) and compute another approximation $\vec{x}_1$, replace $\vec{x}_1$ on the left hand side of (1) and compute another approximation $\vec{x}_2$, and so on. Doing this, we obtain a sequence of vectors $\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots$ such that

$$\vec{x}_{k+1} = \mathsf{D}^{-1}[\vec{b} - (\mathsf{L} + \mathsf{U})\vec{x}_k], \qquad k = 0, 1, 2, 3, \ldots \tag{2}$$

The important fact here is the following property:

Proposition 1. If this sequence of vectors is convergent then the limit vector is a solution of the system of linear equations.

The proof of this property is very easy: let $\vec{v}$ be the limit vector of the sequence $\{\vec{x}_k\}$. Taking limits at both sides of equality (1) we obtain that $\vec{v} = \mathsf{D}^{-1}[\vec{b} - (\mathsf{L} + \mathsf{U})\vec{v}]$, which is equivalent to say that $\vec{v}$ is a solution of the system of linear equations (taking into account the above sequence of equivalences).

Example 2. Consider the following system of linear equations:

$$\left.\begin{array}{rcl} 10x + 3y + z & = & 14 \\ 2x - 10y + 3z & = & -5 \\ x + 3y + 10z & = & 14 \end{array}\right\},$$

The coefficient matrix of this system is the matrix A of Example 1. The recurrence relation (2) is, in this case:

$$\vec{x}_{k+1} = \underbrace{\begin{bmatrix} 1/10 & 0 & 0 \\ 0 & -1/10 & 0 \\ 0 & 0 & 1/10 \end{bmatrix}}_{\mathsf{D}^{-1}} \left(\underbrace{\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}}_{\vec{b}} - \underbrace{\begin{bmatrix} 0 & 3 & 1 \\ 2 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix}}_{\mathsf{L}+\mathsf{U}} \vec{x}_k \right)$$

We apply now the Jacobi method choosing an arbitrary initial vector. Take, for example, $\vec{x}_0 = (0, 0, 0)$. Then

$$\vec{x}_1 = \begin{bmatrix} 1/10 & 0 & 0 \\ 0 & -1/10 & 0 \\ 0 & 0 & 1/10 \end{bmatrix} \left(\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 1 \\ 2 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix} \vec{x}_0 \right)$$

$$= \begin{bmatrix} 1/10 & 0 & 0 \\ 0 & -1/10 & 0 \\ 0 & 0 & 1/10 \end{bmatrix} \left(\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 1 \\ 2 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 7/5 \\ 1/2 \\ 7/5 \end{bmatrix}$$

Now, we compute $\vec{x}_2$:

$$\vec{x}_2 = \begin{bmatrix} 1/10 & 0 & 0 \\ 0 & -1/10 & 0 \\ 0 & 0 & 1/10 \end{bmatrix} \left(\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 1 \\ 2 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix} \vec{x}_1 \right)$$

$$= \begin{bmatrix} 1/10 & 0 & 0 \\ 0 & -1/10 & 0 \\ 0 & 0 & 1/10 \end{bmatrix} \left(\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 1 \\ 2 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix} \begin{bmatrix} 7/5 \\ 1/2 \\ 7/5 \end{bmatrix} \right) = \begin{bmatrix} 111/100 \\ 6/5 \\ 111/100 \end{bmatrix}$$

We want to continue doing iterations in order to "see" if the process is convergent or not. With Scilab is easier:

```
-->A=[10 3 1; 2 -10 3; 1 3 10];
```

```
-->D=diag([diag(A)])
 D  =

    10.     0.      0.
    0.   - 10.     0.
    0.      0.      10.

-->L=tril(A)-D
 L  =

    0.     0.     0.
    2.     0.     0.
    1.     3.     0.

-->U=triu(A)-D
 U  =

    0.     3.     1.
    0.     0.     3.
    0.     0.     0.
```

With the help of the functions **diag**, **tril** and **triu** he have computed easily the matrices L, D and U.

```
-->F=inv(D)
 F  =

    0.1     0.      0.
    0.   - 0.1     0.
    0.      0.      0.1

-->R=L+U
 R  =

    0.     3.     1.
    2.     0.     3.
    1.     3.     0.

-->x0=[0; 0; 0];

-->x1=F*(b-R*x0)
```

```
 x1  =

    1.4
    0.5
    1.4

-->x2=F*(b-R*x1)
 x2  =

    1.11
    1.2
    1.11

-->x3=F*(b-R*x2)
 x3  =

    0.929
    1.055
    0.929

-->x4=F*(b-R*x3)
 x4  =

    0.9906
    0.9645
    0.9906

-->x5=F*(b-R*x4)
 x5  =

    1.01159
    0.9953
    1.01159

-->x6=F*(b-R*x5)
 x6  =

    1.000251
    1.005795
    1.000251
```

We “see” that the process seems to be convergent to the vector $(1, 1, 1)$. You can check that it is, in fact, solution of the system.

We can do the same but using a more efficient (shorter) Scilab code (with the help of the command **for**):

```
-->x=[0; 0; 0];

-->for i=1:6
-->x=F*(b-R*x);
-->end;

-->x
 x  =

    1.000251
    1.005795
    1.000251
```

The vector returned after the loop is, directly, $\vec{x}_6$. If we want to be "more sure" of the convergence, we may apply more iterations:

```
-->x=[0; 0; 0];

-->for i=1:50
-->x=F*(b-R*x);
-->end;

-->x
 x  =

    1.
    1.
    1.
```

3 Gauss-Seidel method

This method is a slight modification of the Jacobi method. In most cases the number of necessary iterations to obtain an approximate solution is smaller than for Jacobi method.

Given a system of linear equations $\mathsf{A}\vec{x} = \vec{b}$ (accomplishing the assumptions stablished at the beginning of the Section), it is used the same decomposition of the matrix A explained in the Jacobi method:

$$\mathsf{A} = \mathsf{L} + \mathsf{D} + \mathsf{U}.$$

However, here we rewrite the equality $\mathsf{A}\vec{x} = \vec{b}$ as

$$(L + D)\vec{x} = \vec{b} - U\vec{x}. \tag{3}$$

This is the crucial equality in the Gauss-Seidel method. We begin with an arbitrary initial vector $\vec{x}_0$ and we compute $\vec{x_1}$ in such a way that

$$(\mathsf{L} + \mathsf{D})\vec{x}_1 = \vec{b} - U\vec{x}_0.$$

Then, we compute $\vec{x}_2$ in such a way that

$$(\mathsf{L} + \mathsf{D})\vec{x}_2 = \vec{b} - U\vec{x}_1,$$

and so on: for each $k = 0, 1, 2, \ldots$ we compute $\vec{x}_{k+1}$ such that

$$(\mathsf{L} + \mathsf{D})\vec{x}_{k+1} = \vec{b} - U\vec{x}_k \tag{4}$$

Since the matrix $\mathsf{L} + \mathsf{D}$ is lower triangular, the components of the vector $\vec{x}_{k+1}$ are computed from the components of $\vec{x}_k$ by **forward substitution**. As in the case of the Jacobi method, if the sequence of vectors $\vec{x}_0, \vec{x}_1, \ldots$ is convergent then the limit vector is the solution of the system. Let us clarify these facts with an example.

Example 3. We shall apply Gauss-Seidel method to the system of Example 2.

$$\mathsf{L} + \mathsf{D} = \begin{bmatrix} 10 & 0 & 0 \\ 2 & -10 & 0 \\ 1 & 3 & 10 \end{bmatrix}, \quad \mathsf{U} = \begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}$$

We start also with the initial vector $\vec{x}_0 = (0, 0, 0)$. The first iteration is:

$$\underbrace{\begin{bmatrix} 10 & 0 & 0 \\ 2 & -10 & 0 \\ 1 & 3 & 10 \end{bmatrix}}_{\mathsf{L+D}} \vec{x}_1 = \underbrace{\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}}_{\vec{b}} - \underbrace{\begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathsf{U}} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\vec{x}_0} \tag{5}$$

If we call

$$\vec{x}_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The equality (5) can be written as

$$\underbrace{\begin{bmatrix} 10 & 0 & 0 \\ 2 & -10 & 0 \\ 1 & 3 & 10 \end{bmatrix}}_{\mathsf{L+D}} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}.$$

And this is a system of linear equations whose coefficient matrix is lower triangular:

$$\left.\begin{array}{rcl} 10x & = & 14 \\ 2x - 10y & = & -5 \\ x + 3y + 10z & = & 14 \end{array}\right\}.$$

Now, by forward substitution we obtain $x = 7/5$, $y = 39/50$ and $z = 513/500$, that is:

$$\vec{x}_1 = \begin{bmatrix} 7/5 \\ 39/50 \\ 513/500 \end{bmatrix}.$$

The second iteration is:

$$\underbrace{\begin{bmatrix} 10 & 0 & 0 \\ 2 & -10 & 0 \\ 1 & 3 & 10 \end{bmatrix}}_{\text{L+D}} \vec{x}_2 = \underbrace{\begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}}_{\vec{b}} - \underbrace{\begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{U}} \underbrace{\begin{bmatrix} 7/5 \\ 39/50 \\ 513/500 \end{bmatrix}}_{\vec{x}_1} \tag{6}$$

If we call now

$$\vec{x}_2 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The above equality can be written as

$$\underbrace{\begin{bmatrix} 10 & 0 & 0 \\ 2 & -10 & 0 \\ 1 & 3 & 10 \end{bmatrix}}_{\text{L+D}} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5317/500 \\ -4039/500 \\ 14 \end{bmatrix}.$$

And this is a system of linear equations whose coefficient matrix is lower triangular:

$$\left.\begin{array}{rcl} 10x & = & 5317/500 \\ 2x - 10y & = & -4039/500 \\ x + 3y + 10z & = & 14 \end{array}\right\}.$$

Now, by forward substitution we obtain $x = 5317/5000$, $y = 3189/3125$ and $z = 246879/250000$, that is:

$$\vec{x}_2 = \begin{bmatrix} 5317/5000 \\ 3189/3125 \\ 246879/250000 \end{bmatrix}.$$

And so on...

Let's do it with Scilab. First, we must **execute** the file *SustitucionProgresiva.sci*, where we have defined a function, **SustitucionProgresiva**, that computes the solution **by forward substitution** of a system $\mathsf{T}\vec{x} = \vec{b}$, where T is an invertible lower triangular matrix. The syntax is: $SustitucionProgresiva(T, b)$, where T is the coefficient matrix and b is the vector of independent terms.

```
-->M=L+D; x0=[0; 0; 0];

-->x1=SustitucionProgresiva(M,b-U*x0)
 x1  =

    1.4
    0.78
    1.026

-->x2=SustitucionProgresiva(M,b-U*x1)
 x2  =

    1.0634
    1.02048
    0.987516

-->x3=SustitucionProgresiva(M,b-U*x2)
 x3  =

    0.9951044
    0.9952757
    1.0019069

-->x4=SustitucionProgresiva(M,b-U*x3)
 x4  =

    1.0012266
    1.0008174
    0.9996321

-->x5=SustitucionProgresiva(M,b-U*x4)
 x5  =

    0.9997916
    0.9998480
    1.0000665

-->x6=SustitucionProgresiva(M,b-U*x5)
 x6  =

    1.000039
    1.0000277
```

```
    0.9999878
```

Using the **for** function we can perform more iterations, if we want:

```
-->x=x0;

-->for(i=1:50) x=SustitucionProgresiva(M,b-U*x); end;

-->x
 x  =

    1.
    1.
    1.
```

4 Convergence criterion

It is possible that, when the Jacobi or Gauss-Seidel method is applied, the obtained sequence of vectors be divergent. However, when the coefficient matrix is of a special type, we can guarantee the convergence of both methods.

Definition 1. A square matrix A is **strictly diagonally dominant** if for all rows the absolute value of the diagonal element in a row is strictly greater than than the sum of absolute values of the rest of the elements in that row.

Sometimes the coefficient matrix of a system of linear equations is not strictly diagonally dominant, but if we change the order of the equations and/or change the order of the unknowns the new system has a coefficient matrix that is strictly diagonally dominant.

This is the condition that assures the convergence of Jacobi and Gauss-Seidel methods:

Theorem 1. If a square matrix A is strictly diagonally dominant, then the Jacobi and Gauss-Seidel methods are convergent.