Introduction to Scientific Computing

(Lecture 5: Linear system of equations / Matrix Splitting)

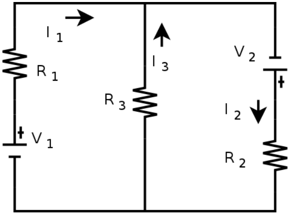
Bojana Rosić, Thilo Moshagen

Institute of Scientific Computing

November 15, 2017

Motivation

Let us resolve the problem scheme by using Kirchhoff's laws:



$$-I_1 + I_2 - I_3 = 0$$
$$-I_1 R_1 + I_3 R_3 = -V_1$$
$$-I_2 R_2 - I_3 R_3 = -V_2$$

- the algebraic sum of all the currents flowing toward a node is equal to zero.

$$\sum I = 0$$

- In any closed circuit, the algebraic sum of all the voltages around the loop is equal to zero

$$\sum V = \sum IR$$

Motivation

Hence, we end up with the system of equations

$$-I_1 + I_2 - I_3 = 0$$

$$-I_1 R_1 + I_3 R_3 = -V_1$$
$$-I_2 R_2 - I_3 R_3 = -V_2$$

which is easy to solve algebraically by elimination

$$I_1 = I_2 + I_3$$

and substitution

$$-(I_2 + I_3)R_1 + I_3 r_3 = -V_1 \Rightarrow I_2 = (V_1 + I_3(R_3 - R_1))R_1^{-1}$$

and so on...

What to do in this case?

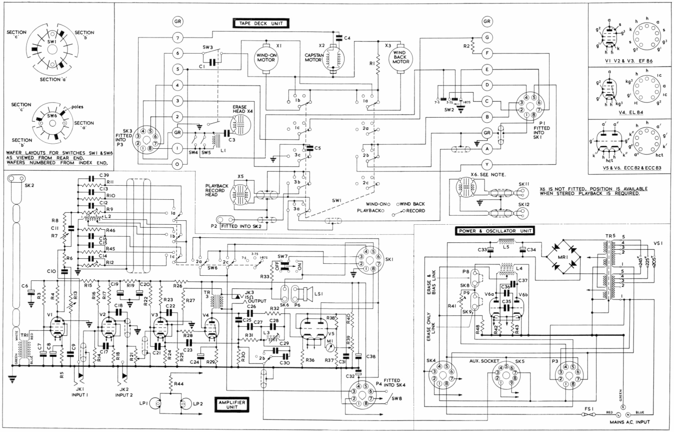
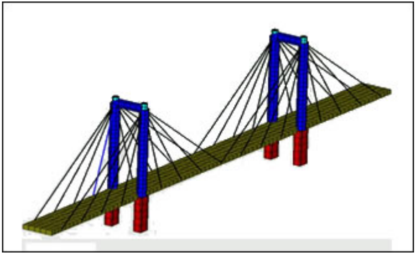


FIG. 20 CIRCUIT DIAGRAM.

Algebraically is difficult to solve.

But, engineering goal

is to solve large-scale (realistic) systems



This usually matches with solving

$$\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x})$$

in which x is the system state.

Special case

Special case are systems in a linear form

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

in which matrix $\boldsymbol{A}$ can be of dimensions larger than 10. Think about

$$\boldsymbol{A} \in \mathbb{R}^{1000 \times 1000}$$

$$\boldsymbol{A} \in \mathbb{R}^{1000000 \times 1000000}$$

Examples are: solving linear partial differential equations (elasticity, heat equation etc.)

In these cases one cannot compute

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

as one would have problem with memory or computation time.

Linear systems

There are many methods one can use to solve the system

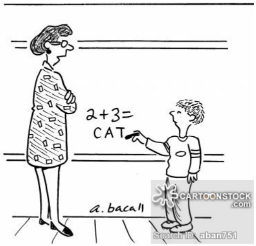
$$\boldsymbol{Ax} = \boldsymbol{b}$$

such as:

- **Stationary iterative methods** (approximate operator)
  - Gauss-Seidel (GS)
  - Jacobi (JM)
  - Successive over-relaxation (SOR)
- **Krylov subspace methods**
  - Conjugate Gradient (CG)
  - Biconjugate gradient method (BiCG)
  - generalized minimal residual method (GMRES)
- **combination**

Solving system

Remembering fixed point iteration, we may try something similar. Let us guess the solution by putting



"There's supposed to be a fine line between a guess and an educated guess."

$$\boldsymbol{x} = \boldsymbol{x}^{(k)}$$

Since we do not know if our guess is correct, we may check its accuracy by evaluating the *residual*

$$\boldsymbol{d}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\mathbf{x}^{(k)}$$

Now we may have educated guess by drawing conclusions from the error value.

What would be our next guess?

Thus, our next guess $\mathbf{x}^{(k+1)}$ will be smaller or larger than $\mathbf{x}^{(k)}$ depending on the residual $\mathbf{d}^{(k)}$. Hence,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{v}^{(k)}$$

in which $\mathbf{v}^{(k)}$ denotes correction.

Our goal is now to find the best correction!

Naturally, the best correction $\mathbf{v}^{(k)}$ would be the correction which satisfies the equation

$$\mathbf{x}^{(k)} + \mathbf{v}^{(k)} = \mathbf{x}.$$

so the error. The only problem is that we do not know $\mathbf{x}$!!

But,

we do know that the best correction $\mathbf{v}^{(k)}$ has to satisfy

$$\boldsymbol{A}\mathbf{x} = \boldsymbol{A}(\mathbf{x}^{(k)} + \mathbf{v}^{(k)}) = \mathbf{b}$$

This in turn gives us equation for the correction

$$\boldsymbol{A}\mathbf{v}^{(k)} = \mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)} = \mathbf{d}^{(k)}$$

which further yields

$$\mathbf{v}^{(k)} = \boldsymbol{A}^{-1}(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}) = \boldsymbol{A}^{-1}\mathbf{d}^{(k)}$$

Thsi is explicit solving again.

Solution: pose the problem differently

Do not invert matrix $\boldsymbol{A}$, but find some simpler matrix $\boldsymbol{C} \approx \boldsymbol{A}$ which is easy to invert!

Then the problem

$$\mathbf{v}^{(k)} = \boldsymbol{A}^{-1}\mathbf{d}^{(k)}$$

reduces to

$$\mathbf{v}^{(k)} = \boldsymbol{C}^{-1}\mathbf{d}^{(k)}$$

i.e.

$$\boldsymbol{C}\mathbf{v}^{(k)} = \mathbf{d}^{(k)}$$

which takes only **small computational time**.

Linear iteration scheme

Then we can formulate the general linear iteration scheme

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \underbrace{\boldsymbol{C}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right)}_{\mathbf{v}^{(k)}}.$$

in which $\boldsymbol{C}$ can be chosen in different ways. With respect to the type of approximation one may distinguish:

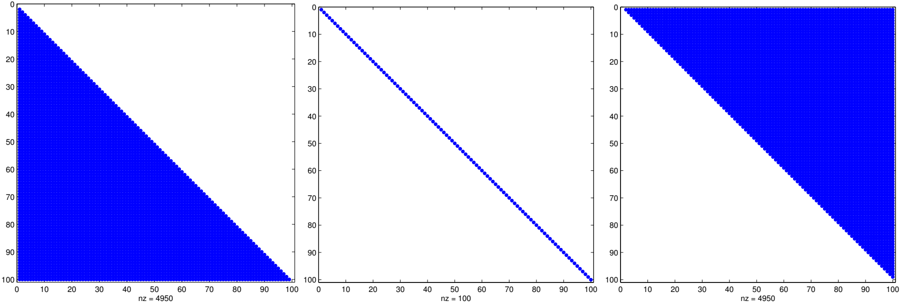
- Jacobi method
- Gauss-Seidel method
- The successive over relaxation method, etc.

These methods choose $\boldsymbol{C}$ from LDU decomposition of a matrix $\boldsymbol{A}$.

LDU decomposition

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

where $\boldsymbol{L}$ is a lower triangular matrix with main diagonal equals 0, D is a diagonal matrix und $\boldsymbol{U}$ is an upper triangular matrix with main diagonal equals zero. Decomposition of $\boldsymbol{A}$:



$$L_{ij} = 0, j \geq i \quad \text{diag}(a_{ii}) \quad U_{ij} = 0, j \leq i$$

Jacobi method

Let us take that $\boldsymbol{C}$ is approximated only by diagonal

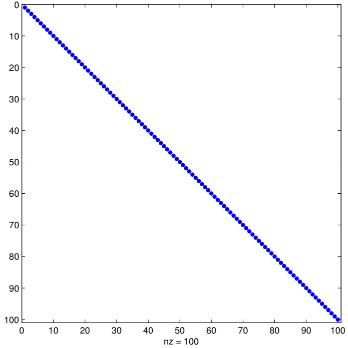
$$\boldsymbol{C} := \boldsymbol{D} = \mathrm{diag}(a_{ii}).$$

Then it follows

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \underbrace{\boldsymbol{D}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right)}_{\mathbf{v}^{(k)}}.$$

With $\boldsymbol{D}^{-1} = \mathrm{diag}\left(a_{ii}^{-1}\right)$ and

$$\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right)_i = b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k)}$$



Jacobi method

The iterative procedure in element-wise form reads:

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

Hence, the values a_{ii} must be different than zero.

Exercise

Let us use Jacobi method to solve

$$\begin{aligned} 2x + \quad y \quad &= 11 \\ 5x + 7y &= 13 \end{aligned}$$

by starting at $\mathbf{x}^{(0)} = (1; 1)^T$. Hence,

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 \\ 5 & 7 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 11 \\ 13 \end{pmatrix}$$

Exercise

The first iteration reads

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \boldsymbol{D}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(0)}\right)$$

where $\boldsymbol{D}$ is the diagonal part of the matrix $\boldsymbol{A}$:

$$\boldsymbol{D} = \begin{pmatrix} 2 & 0 \\ 0 & 7 \end{pmatrix}$$

with the inverse

$$\boldsymbol{D}^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/7 \end{pmatrix}.$$

Exercise

Hence,

$$\mathbf{x}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1/2 & 0 \\ 0 & 1/7 \end{pmatrix} \left(\begin{pmatrix} 11 \\ 13 \end{pmatrix} - \begin{pmatrix} 2 & 1 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right)$$

$$\mathbf{x}^{(1)} = \begin{pmatrix} 5.0000 \\ 1.1429 \end{pmatrix}$$

After 22 iterations one obtains:

$$\mathbf{x}^{(22)} = \begin{pmatrix} 7.1110 \\ -3.2222 \end{pmatrix}$$

Exercise

Let us check solution

$$\boldsymbol{Ax}^{(22)} = \begin{pmatrix} 2 & 1 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} 7.1110 \\ -3.2222 \end{pmatrix} = \begin{pmatrix} 11 \\ 13 \end{pmatrix}$$

Exercise

Let us use Jacobi method to solve

$$\begin{aligned} 3x + 2y - z &= 1 \\ 2x - 2y + 4z &= -2 \\ -x + \tfrac{1}{2}y - z &= 0 \end{aligned}$$

Hence,

$$\boldsymbol{A} = \begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$$

Exercise

The diagonal part of the matrix is

$$\boldsymbol{D} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \Rightarrow$$

$$\boldsymbol{D}^{-1} = \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & -1/2 & 0 \\ 0 & 0 & -1/1 \end{pmatrix}$$

and the Jacobi method reads:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \underbrace{\boldsymbol{D}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right)}_{\mathbf{v}^{(k)}}.$$

Exercise

Start with

$$\mathbf{x}^{(0)} = \mathbf{0} = (0,0,0)^T$$

such that

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \boldsymbol{D}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(0)}\right)$$

$$\mathbf{x}^{(1)} = \mathbf{0} + \boldsymbol{D}^{-1}\left(\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix} - \begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{pmatrix}\mathbf{0}\right)$$

holds.

Exercise

Hence,

$$\mathbf{x}^{(1)} = \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & -1/2 & 0 \\ 0 & 0 & -1/1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1 \\ 0 \end{pmatrix}$$

Similarly, one may write

$$\mathbf{x}^{(2)} = \begin{pmatrix} 1/3 \\ 1 \\ 0 \end{pmatrix} + \boldsymbol{D}^{-1} \left(\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix} - \begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{pmatrix} \begin{pmatrix} 1/3 \\ 1 \\ 0 \end{pmatrix} \right)$$

$$\mathbf{x}^{(2)} = \begin{pmatrix} -0.3333 \\ 1.3333 \\ 0.1667 \end{pmatrix}$$

Exercise

Iter	x_1	x_2	x_3
1	0.33	1.00	0.00
2	-0.33	1.33	0.17
3	-0.50	1.00	1.00
100	-0.43e6	1.94e6	1.29e6
1000	-0.29e59	1.3e59	0.87e59

However, by iterating one may notice that the solution does not converge. This happens even by changing the initial condition. The question is why this happens?

Convergence

From

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{D}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right) = F(\mathbf{x}^{(k)})$$

we may conclude that this is one kind of fixed point iteration scheme with the Lipschitz constant

$$F'(\mathbf{x}) = \mathbf{1} - \boldsymbol{D}^{-1}\boldsymbol{A}$$

Since

$$\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U}$$

with $\boldsymbol{R} = \boldsymbol{L} + \boldsymbol{U}$ is the rest of the matrix

$$F'(\mathbf{x}) = \mathbf{1} - \boldsymbol{D}^{-1}(\boldsymbol{D} + \boldsymbol{R}) = -\boldsymbol{D}^{-1}\boldsymbol{R}$$

Convergence

To check if the method is convergent, one has to check contractivity

$$q = \sup \|F'\| = \sup \|\boldsymbol{D}^{-1}\boldsymbol{R}\|_2 < 1.$$

The last relation is equivalent to the condition for *the spectral radius*

$$\rho \stackrel{\text{def}}{=} \max_i(|\lambda_i|) < 1$$

where λ_i are the eigenvalues of $\boldsymbol{D}^{-1}\boldsymbol{R}$.

Exercise

In the first example one has

$$\boldsymbol{D}^{-1}\boldsymbol{R} = \begin{pmatrix} 0 & 0.5000 \\ 0.7143 & 0 \end{pmatrix}$$

Hence,

$$\rho = 0.5976 < 1$$

Exercise

In the second example one has

$$D^{-1}R = \begin{pmatrix} 0 & 0.6667 & -0.3333 \\ -1.0000 & 0 & -2.0000 \\ 1.0000 & -0.5000 & 0 \end{pmatrix}$$

Hence,

$$\rho = 1.14 > 1$$

Can we make this conclusion apriori?

If matrix $\boldsymbol{A}$ is **strictly or irreducibly diagonally dominant**, Jacobi method will converge. Strict row diagonal dominance means that for each row, the absolute value of the diagonal term is greater than the sum of absolute values of other terms:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|.$$

This stems from (see Golub/van Loan Chap. 10.1)

$$\rho(\boldsymbol{D}^{-1}(\boldsymbol{D}+\boldsymbol{R})) \leq ||\boldsymbol{D}^{-1}(\boldsymbol{D}+\boldsymbol{R})||_{\infty} = \max_{i} \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right| < 1$$

Gauss-Seidel method

Going back to the general scheme

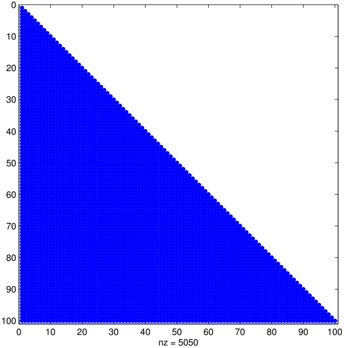
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{C}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right).$$

we may take the approximation for $\boldsymbol{C}$ to be equal

$$\boldsymbol{C} := \boldsymbol{L} + \boldsymbol{D}$$

Then it follows

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + (\boldsymbol{L} + D)^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right).$$



Gauss-Seidel method

However, **we said before that inversion is avoided in numerical schemes**. Due to this reason, let us multiply both sides of equation by $(\boldsymbol{L}+\boldsymbol{D})$:

$$(\boldsymbol{L}+\boldsymbol{D})\mathbf{x}^{(k+1)} = (\boldsymbol{L}+\boldsymbol{D})\mathbf{x}^{(k)} + \mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}.$$

Having that $\boldsymbol{A} = \boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U}$ we obtain

$$(\boldsymbol{L}+\boldsymbol{D})\mathbf{x}^{(k+1)} = (\boldsymbol{L}+\boldsymbol{D})\mathbf{x}^{(k)} + \mathbf{b} - (\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U})\mathbf{x}^{(k)} = \mathbf{b} - \boldsymbol{U}\mathbf{x}^{(k)}$$

Splitting the left side of equation, one obtains

$$\boldsymbol{D}\mathbf{x}^{(k+1)} = \mathbf{b} - \boldsymbol{L}\mathbf{x}^{(k+1)} - \boldsymbol{U}\mathbf{x}^{(k)}.$$

Gauss-Seidel method

In component form one obtains

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

and

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

Compare Jacobi and GS method

The Jacobi method in element-wise form reads:

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

and Gauss-Seidel:

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

More motivating view on GS: It is Jacobi method while using the latest infos.

Exercise

Let us use Gauss-Seidel method to solve

$$\begin{aligned} 2x + \ \ y \ \ &= 11 \\ 5x + 7y &= 13 \end{aligned}$$

by starting at $\mathbf{x}^{(0)} = (1;1)^T$. Hence,

$$\boldsymbol{A} = \begin{pmatrix} 2 & 1 \\ 5 & 7 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 11 \\ 13 \end{pmatrix}$$

Exercise

The first iteration reads

$$x_i^{(1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(0)} \right] \qquad i = 1, \dots, n.$$

$$x_1^{(1)} = \frac{1}{a_{11}} [b_1 - \sum_{j=2}^{2} a_{1j} x_j^{(0)}] = 5$$

$$x_2^{(1)} = \frac{1}{a_{22}} [b_2 - \sum_{j=1}^{1} a_{2j} x_j^{(1)}] = -1.7143$$

It takes approximately the same number of iterations as Jacobi to get the solution.

Exercise

Let us use Gauss-Seidel method to solve

$$\begin{aligned} 3x + 2y - \ \ z &= \ \ 1 \\ 2x - 2y + 4z &= -2 \\ -x + \tfrac{1}{2}y - \ \ z &= \ \ 0 \end{aligned}$$

Hence,

$$\boldsymbol{A} = \begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$$

Exercise

The first iteration reads

$$x_i^{(1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(0)} \right]$$

$$x_1^{(1)} = \frac{1}{a_{11}} [b_1 - \sum_{j=2}^{3} a_{1j} x_j^{(0)}] = 0.333$$

$$x_2^{(1)} = \frac{1}{a_{22}} [b_2 - \sum_{j=1}^{1} a_{2j} x_j^{(1)} - \sum_{j=3}^{3} a_{2j} x_j^{(0)}] = -1.333$$

$$x_3^{(1)} = \frac{1}{a_{33}} [b_3 - \sum_{j=1}^{2} a_{3j} x_j^{(1)}] = 0.333$$

Exercise

Similarly to the Jacobi method, the Gauss-Seidel does not converge for this system of equations.

Convergence

From

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + (\boldsymbol{L} + \boldsymbol{D})^{-1} \left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)} \right).$$

we may conclude that this is one kind of fixed point iteration scheme with the Lipschitz constant

$$F'(\mathbf{x}) = \mathbf{1} - (\boldsymbol{L} + \boldsymbol{D})^{-1} \boldsymbol{A}$$

Since

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U}$$

one has

$$F'(\mathbf{x}) = \mathbf{1} - (L + \boldsymbol{D})^{-1}(L + \boldsymbol{D} + \boldsymbol{U}) = -(L + \boldsymbol{D})^{-1} \boldsymbol{U}$$

Convergence

Thus, the method is convergent if the spectral radius of

$$\rho((\boldsymbol{L} + \boldsymbol{D})^{-1}\boldsymbol{U}) < 1$$

The convergence properties of the Gauss–Seidel method are dependent on the matrix A. Namely, the procedure is known to converge if either:

- A is symmetric positive-definite (symmetric $A = A^T$, positive definite matrix is a symmetric matrix A for which all eigenvalues are positive).
- A is strictly or irreducibly diagonally dominant.

Convergence

In the first example one has that

$$\rho((\boldsymbol{L} + \boldsymbol{D})^{-1}\boldsymbol{U}) = 0.3571 < 1$$

and in the second example

$$\rho((\boldsymbol{L} + \boldsymbol{D})^{-1}\boldsymbol{U}) = 1.24 > 1$$

Successive over relaxation method (SOR)

The SOR-scheme tries to improve the convergence properties of the Gauß-Seidel method. The idea is to introduce a relaxation parameter $\omega > 0$ such that

$$\boldsymbol{C} := \frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}$$

Inserting yields

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right).$$

Multiplying by $\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}$, and using $A = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U}$ one obtains

$$\left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)\mathbf{x}^{(k+1)} \quad = \quad \left(\frac{1}{\omega} - 1\right)\boldsymbol{D}\mathbf{x}^{(k)} + \mathbf{b} - \boldsymbol{U}\mathbf{x}^{(k)}$$

SOR method

Then

$$\frac{1}{\omega}\boldsymbol{D}\mathbf{x}^{(k+1)} = \frac{1-\omega}{\omega}\boldsymbol{D}\mathbf{x}^{(k)} - \boldsymbol{L}\mathbf{x}^{(k+1)} + \mathbf{b} - \boldsymbol{U}\mathbf{x}^{(k)}$$

and

$$\boldsymbol{D}\mathbf{x}^{(k+1)} = (1-\omega)D\mathbf{x}^{(k)} + \omega\left(\mathbf{b} - \boldsymbol{L}\mathbf{x}^{(k+1)} - \boldsymbol{U}\mathbf{x}^{(k)}\right).$$

The representation with components is now given by

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

and

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

SOR method

Note that when $\omega = 1$ the method

$$\boldsymbol{D}\mathbf{x}^{(k+1)} = (1-\omega)\boldsymbol{D}\mathbf{x}^{(k)} + \omega\left(\mathbf{b} - \boldsymbol{L}\mathbf{x}^{(k+1)} - \boldsymbol{U}\mathbf{x}^{(k)}\right).$$

becomes

$$\boldsymbol{D}\mathbf{x}^{(k+1)} = \left(\mathbf{b} - \boldsymbol{L}\mathbf{x}^{(k+1)} - \boldsymbol{U}\mathbf{x}^{(k)}\right).$$

which is Gauß-Seidel method.

SOR: how to choose ω?

With an optimal choice of ω we can improve the convergence of the SOR-method in comparison with the Gauß-Seidel method. But the optimal value for ω depends on the problem, i.e. in general one has to try several values. One should start with larger values, i.e. $\omega = 1.7$, then $\omega = 1.3$ ... and take a look on the convergence behaviour.

Convergence

From

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right)$$

we may conclude that this is one kind of fixed point iteration scheme with the Lipschitz constant

$$F'(\mathbf{x}) = \mathbf{1} - \left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)^{-1}\boldsymbol{A}$$

and hence the spectral radius must satisfy

$$\rho(\mathbf{1} - \left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)^{-1}\boldsymbol{A}) < 1$$

to get convergence.

Convergence

For $\omega = 1.7$ the spectral radius in the previous two examples becomes

$$\rho(\mathbf{1} - \left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)^{-1} A) = 0.7 < 1$$

and in the second example

$$\rho(\mathbf{1} - \left(\frac{1}{\omega}\boldsymbol{D} + \boldsymbol{L}\right)^{-1} \boldsymbol{A}) = 3.008 > 1$$

General convergence results

The equation

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{C}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right).$$

for the general iteration method can be transformed into

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{C}^{-1}\left(\mathbf{b} - \boldsymbol{A}\mathbf{x}^{(k)}\right) = \boldsymbol{C}^{-1}\mathbf{b} + (\boldsymbol{I} - \boldsymbol{C}^{-1}\boldsymbol{A})\mathbf{x}^{(k)}$$

i.e.

$$\mathbf{x}^{(k+1)} = \boldsymbol{M}\mathbf{x}^{(k)} + \mathbf{g}$$

in which $\boldsymbol{M} = \boldsymbol{I} - \boldsymbol{C}^{-1}\boldsymbol{A}$ and $\mathbf{g} = \boldsymbol{C}^{-1}\mathbf{b}$. This equation is often called normal form. Every linear iteration method can be written in normal form. The matrix $\boldsymbol{M}$ is called iteration matrix of the method.

General convergence results

For the derivation of convergence criteria we need the spectral radius of a matrix which can be defined in the following way:

Definition

For a given matrix $\boldsymbol{M} \in \mathbb{R}^{n,n}$ let λ_i, $i = 1, \ldots, n$ be the eigenvalues of $\boldsymbol{M}$. Then

$$\rho(\boldsymbol{M}) := \max_{i=1,\ldots,n} |\lambda_i|$$

is called the spectral radius of $\boldsymbol{M}$.

General convergence results

Theorem

The linear iteration method with the iteration matrix $\boldsymbol{M}$ converges for an arbitrary start vector $\mathbf{x}^{(0)}$ if and only if $\rho(\boldsymbol{M}) < 1$ and if in a dedicated matrix norm $\|\cdot\|$, the condition $\|\boldsymbol{M}\| \leq 1$ is satisfied. In this case the following error estimates are true:

$$\left\|\mathbf{x}^{(k)} - \mathbf{x}_*\right\| \leq \frac{\|\boldsymbol{M}\|^k}{1 - \|\boldsymbol{M}\|} \left\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right\|, \tag{1}$$

$$\left\|\mathbf{x}^{(k)} - \mathbf{x}_*\right\| \leq \frac{\|\boldsymbol{M}\|}{1 - \|\boldsymbol{M}\|} \left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right\|. \tag{2}$$

Inequality (1) is called a priori estimate and (2) is called a posteriori estimate.

Literature

- C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations
- Yousef Saad, Iterative methods for sparse linear systems
- Martin Stoll, Solving Linear Systems using the Adjoint