Numerical Calculus

Pătrulescu Flavius

Technical University of Cluj-Napoca

2024

References

1. R.L. Burden, D.J. Faires, A.M. Burden, *Numerical Analysis*, 10th Ed., Cengage Learning, Boston, 2022.
2. A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Springer-Verlag, New-York, 2000.

Iterative techniques for linear systems (Burden *et al., 2022*)

Iterative techniques are seldom used for solving linear systems of small dimension since the time required for sufficient accuracy exceeds that required for direct techniques, such as Gaussian elimination. For large systems with a high percentage of 0 entries, these techniques are efficient in terms of both computer storage and computation. Systems of this type arise frequently in circuit analysis and in the numerical solution of boundary-value problems.

Iterative techniques for linear systems (Burden *et al.*, *2022*; Quarteroni *et al.* 2000)

In general, *iterative techniques* to solve linear systems involve a process that converts the system

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

into an equivalent system of the form

$$\boldsymbol{x} = R\boldsymbol{x} + \boldsymbol{c}$$

for some fixed matrix (*iteration matrix*) R and vector $\boldsymbol{c}$. After the initial vector $\boldsymbol{x}^{(0)}$ is selected, the sequence of approximate solution vectors is generated by computing

$$\boldsymbol{x}^{(k+1)} = R\boldsymbol{x}^{(k)} + \boldsymbol{c}$$

for each $k = 1, 2, \ldots$

Iterative techniques for linear systems (Burden *et al.*, *2022*; Quarteroni *et al.* 2000)

We consider a decomposition $A = M - K$ with $\det M \neq 0$.
We obtain

$$A\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (M - K)\boldsymbol{x} = b \Leftrightarrow M\boldsymbol{x} = K\boldsymbol{x} + \boldsymbol{b}$$

$$\boldsymbol{x} = \underbrace{M^{-1}K}_{R:=}\,\boldsymbol{x} + \underbrace{M^{-1}\boldsymbol{b}}_{\boldsymbol{c}:=}$$

Find a splitting $A = M - K$ s.t.

- R and $\boldsymbol{c}$ are easy to evaluate
- $\rho(R)$ is *small*.

Iterative techniques for linear systems (Burden *et al.*, 2022)

$$A = D - L - U$$

D is the diagonal matrix of the diagonal entries of A

$-L$ is the strictly lower-triangular part of A

$$L = (l_{ij}) = \begin{cases} -a_{ij}, \ i > j \\ 0, i \leqslant j \end{cases}$$

$-U$ is the strictly upper-triangular part of A

$$U = (u_{ij}) = \begin{cases} -a_{ij}, \ i < j \\ 0, i \geqslant j \end{cases}$$

Jacobi method (Burden *et al.*, *2022*)

$$A\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (D - L - U)\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow D\boldsymbol{x} = (L + U)\boldsymbol{x} + \boldsymbol{b}$$

$$\boldsymbol{x} = \underbrace{D^{-1}(L + U)}_{R_J :=} \boldsymbol{x} + \underbrace{D^{-1}\boldsymbol{b}}_{\boldsymbol{c}_J :=} \Leftrightarrow \boldsymbol{x} = R_J\boldsymbol{x} + \boldsymbol{c}_J$$

$$\boldsymbol{x}^{(k+1)} = R_J\boldsymbol{x}^{(k)} + \boldsymbol{c}_J,\ k \geqslant 0$$

Jacobi method is obtained by solving the ith equation in $A\boldsymbol{x} = \boldsymbol{b}$ for x_i (provided that $a_{ii} \neq 0$). For each $k \geqslant 0$, Jacobi method generates the components $x_i^{(k+1)}$ of $\boldsymbol{x}^{(k+1)}$ from the components of $\boldsymbol{x}^{(k)}$ by

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)}\right), i = 1, 2, \ldots, n.$$

Gauss-Seidel method(Burden *et al., 2022*)

$$A\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (D - L - U)\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (D - L)\boldsymbol{x} = U\boldsymbol{x} + \boldsymbol{b}$$

$$\boldsymbol{x} = \underbrace{(D - L)^{-1}U}_{R_{GS}}\boldsymbol{x} + \underbrace{(D - L)^{-1}\boldsymbol{b}}_{\boldsymbol{c}_{GS}} \Leftrightarrow \boldsymbol{x} = R_{GS}\boldsymbol{x} + \boldsymbol{c}_{GS}$$

$$\boldsymbol{x}^{(k+1)} = R_{GS}\boldsymbol{x}^{(k)} + \boldsymbol{c}_{GS},\ k \geqslant 0$$

For each $k \geqslant 0$, Gauss-Seidel method generates the components $x_i^{(k+1)}$ of $\boldsymbol{x}^{(k+1)}$ from the components of $\boldsymbol{x}^{(k)}$ by

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), i = 1, 2, \ldots, n.$$

Successive Over-Relaxation (S.O.R.) method (Burden *et al.*,2022)

Let $\omega > 0$.

$$\omega A\boldsymbol{x} = \omega\boldsymbol{b} \Leftrightarrow \omega(D - L - U)\boldsymbol{x} = \omega\boldsymbol{b} \Leftrightarrow \omega(D - L)\boldsymbol{x} = \omega U\boldsymbol{x} + \omega\boldsymbol{b}$$

$$\Leftrightarrow (D - \omega L)\boldsymbol{x} = \big((1-\omega)D + \omega U\big)\boldsymbol{x} + \omega\boldsymbol{b} \Leftrightarrow$$

$$\Leftrightarrow \boldsymbol{x} = \underbrace{(D - \omega L)^{-1}\big((1-\omega)D + \omega U\big)}_{R(\omega):=}\boldsymbol{x} + \underbrace{\omega(D - \omega L)^{-1}\boldsymbol{b}}_{\boldsymbol{c}(\omega):=} \Leftrightarrow$$

$$\Leftrightarrow \boldsymbol{x} = R(\omega)\boldsymbol{x} + \boldsymbol{c}(\omega)$$

S.O.R. (Burden *et al.*,2022)

$$\boldsymbol{x}^{(k+1)} = R(\omega)\boldsymbol{x}^{(k)} + \boldsymbol{c}(\omega),\ k \geqslant 0$$

For each $k \geqslant 0$, S.O.R. method generates the components $x_i^{(k+1)}$ of $\boldsymbol{x}^{(k+1)}$ from the components of $\boldsymbol{x}^{(k)}$ by

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega \underbrace{\frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)}_{\text{the component generated by Gauss}-\text{Seidel method}},\ i = \overline{1,n}.$$

For $\omega = 1$ S.O.R. coincides with Gauss-Seidel method. In particular, for $\omega \in (0,1)$, the method is called *under-relaxation*.

Backward Gauss-Seidel (B.G.S.) method (Quarteroni *et al.*, 2000)

An analogue of the Gauss-Seidel method can be constructed by simply exchanging L with U

$$A\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (D - L - U)\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (D - U)\boldsymbol{x} = L\boldsymbol{x} + \boldsymbol{b}$$

$$\boldsymbol{x} = \underbrace{(D - U)^{-1}L}_{R_{BGS}}\boldsymbol{x} + \underbrace{(D - U)^{-1}\boldsymbol{b}}_{\boldsymbol{c}_{BGS}} \Leftrightarrow \boldsymbol{x} = R_{BGS}\boldsymbol{x} + \boldsymbol{c}_{BGS}$$

$$\boldsymbol{x}^{(k+1)} = R_{BGS}\boldsymbol{x}^{(k)} + \boldsymbol{c}_{BGS},\ k \geqslant 0.$$

For each $k \geqslant 0$, backward Gauss-Seidel method generates the components $x_i^{(k+1)}$ of $\boldsymbol{x}^{(k+1)}$ from the components of $\boldsymbol{x}^{(k)}$ by

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

Symmetric Gauss-Seidel method (Quarteroni *et al.*, 2000)

The symmetric Gauss-Seidel method is obtained by combining an iteration of the Gauss-Seidel method with an iteration of the backward Gauss-Seidel method.

$$(D - L)\boldsymbol{x}^{(k+\frac{1}{2})} = U\boldsymbol{x}^{(k)} + \boldsymbol{b},$$

$$(D - U)\boldsymbol{x}^{(k+1)} = L\boldsymbol{x}^{(k+\frac{1}{2})} + \boldsymbol{b}.$$

Eliminating $\boldsymbol{x}^{(k+\frac{1}{2})}$, the following scheme is obtained

$$\boldsymbol{x}^{(k+1)} = R_{SGS}\boldsymbol{x}^{(k)} + \boldsymbol{c}_{SGS},\ k \geqslant 0$$

$$\begin{cases} R_{SGS} = (D - U)^{-1}L(D - L)^{-1}U \\ \boldsymbol{c}_{SGS} = (D - U)^{-1}\big[L(D - L)^{-1} + I_n\big]\boldsymbol{b}. \end{cases}$$

General Iterations Methods (Burden *et al., 2022*)

For any $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ the sequence $(\boldsymbol{x}^{(k)})_k$ defined by

$$\boldsymbol{x}^{(k+1)} = R\boldsymbol{x}^{(k)} + c, \quad k \geqslant 0$$

converges to the unique solution of $\boldsymbol{x} = R\boldsymbol{x} + \boldsymbol{c}$ if and only if

$$\rho(R) < 1 \text{ (spectral radius).}$$

Moreover, if $|||R||| < 1$ for any induced norm then the sequence $(\boldsymbol{x}^{(k)})_k$ converges to $\boldsymbol{x}$ and the following error bounds hold

$$\|\boldsymbol{x} - \boldsymbol{x}^{(k)}\| \leqslant |||R|||^k \|\boldsymbol{x} - \boldsymbol{x}^{(0)}\|$$

and

$$\|\boldsymbol{x} - \boldsymbol{x}^{(k)}\| \leqslant \frac{|||R|||^k}{1 - |||R|||} \|\boldsymbol{x}^{(1)} - \boldsymbol{x}^{0}\|.$$

Convergence Results (Burden *et al.*, 2022)

If A is strictly diagonally dominant, then for any choice of $\boldsymbol{x}^{(0)}$ both the Jacobi and Gauss-Seidel methods give sequences $(\boldsymbol{x}^{(k)})_k$ that converge to the unique solution of $A\boldsymbol{x} = \boldsymbol{b}$.

Stein-Rosenberg: If $a_{ij} < 0$ for each $i \neq j$ and $a_{ii} > 0$ for each $i = \overline{1, n}$ then one and only one of the following statements holds:

1. $0 \leqslant \rho(R_{GS}) < \rho(R_J) < 1$
2. $1 \leqslant \rho(R_J) < \rho(R_{GS})$
3. $\rho(R_J) = \rho(R_{GS}) = 0$
4. $\rho(R_J) = \rho(R_{GS}) = 1$

S.O.R. (Burden *et al.*,2022; Quarteroni *et al.*, 2000)

Kahan: If $a_{ii} \neq 0$, $i = \overline{1,n}$ then $\rho(R(\omega)) > |\omega - 1|$. This implies that the S.O.R. method can converge only if $0 < \omega < 2$ (necessary condition).
Ostrowski-Reich: If A is symmetric and positive definite, then the S.O.R. method is convergent if and only if $0 < \omega < 2$ (condition becomes also sufficient for convergence).

If A is strictly diagonally dominant by rows, the S.O.R. method converges if $0 < \omega \leqslant 1$.

If A is symmetric, positive definite and tridiagonal then $\rho(R_{GS}) = (\rho(R_J))^2 < 1$ and the optimal choice of ω for the S.O.R. method is

$$\omega = \frac{2}{1 + \sqrt{1 - (\rho(R_J))^2}}$$

With this choice of ω, we have $\rho(R(\omega)) = \omega - 1$.