C07 - The Gauss-Seidel load flow method

The Gauss-Seidel (GS) load flow method is one of the first developed for computer-aided analysis of EPS steady state, back in the 1950 years. Its simple, yet precise algorithm makes it usable on low performance computer configurations. This is why it is still included in today's EPS analysis software such as EDSA-Paladin or Power World Simulator, where it can be used standalone or as starting method for the Newton-Raphson load flow algorithm.

Its advantages are:

- has a short and simple algorithm, easy to implement using any programming language
- ensures convergence in some special cases where the Newton-Raphson algorithm fails, such as operating states close to instability, or when the initial approximation for the solution is chosen too far from the exact solution.

Its disadvantages are:

- requires a high number of iterations, which grows with the EPS size
- it is considered that the precision of the results is somewhat lower compared with the Newton-Raphson method.

The general mathematical model

The Gauss-Seidel method for load flow studies is based on the method with the same name written for the mathematical problem of solving linear equations systems. In the following, the general mathematical model, and then the adaptation for load flow calculations will be presented.

A n-sized (n equations, n unknown variables) linear equations system can be written as:

$$\begin{aligned} &a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1n} \cdot x_n = b_1 \\ &a_{21} \cdot x_1 + a_{22} \cdot x_2 + \ldots + a_{2n} \cdot x_n = b_2 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{n1} \cdot x_1 + a_{n2} \cdot x_2 + \ldots + a_{nn} \cdot x_n = b_n \end{aligned} \tag{7.1}$$

or, in matrix form:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \tag{7.2}$$

or, in compact writing:

$$[A]\cdot[x]=[b] \tag{7.3}$$

Matrix [A] can be separated into two matrices, as follows:

$$[A]=[N]-[P], \tag{7.4}$$

which allows (7.3) to be rewritten as

$$[N]\cdot[x]=[P]\cdot[x]+[b] \tag{7.5}$$

Equation (7.5) allows the definition of the generic recurrence equation for any interation *t* as:

$$[N]\cdot[x]^{t+1}=[P]\cdot[x]^{t}+[b] \tag{7.6}$$

Thus, the approximation from iteration *(t+1)* can be computed using the approximation from iteration *(t)* with:

$$[x]^{t+1}=[N]^{-1}\cdot[P]\cdot[x]^{t}+[N]^{-1}\cdot[b] \tag{7.7}$$

Starting from an initial approximation $[x]^0$, and using (7.7) in an iterative process, a chain of successive approximations $[x]^1$, $[x]^2$, $[x]^3$, … can be computed, which, in certain scenarios, converges to the exact solution.

Usually, matrices $[N]$ and $[P]$ from (7.4) are defined based on the standard decomposition

$$[A]=[L]+[D]+[R] \tag{7.8}$$

so that matrix $[N]$, which in (7.7) needs the computation of its inverse, to be as simple as possible.

In (7.8), *[L]* *(left)* is an lower triangular matrix (with the elements below the main diagonal taken from matrix *[A]*), *[D]* is a diagonal matrix, with the elements from the main diagonal equal to the main diagonal elements of matrix *[A]*, while *[R]* (right) is an upper triangular matrix, with elements above the main diagonal taken from matrix *[A]*.

For the Gauss-Seidel method, $[N]$ and $[P]$ are written as:

$$[N]=[D]+[L] \quad \text{and} \quad [P]=-[R] \tag{7.9}$$

Thus, (7.6) can be rewritten as:

$$([D]+[L])\cdot[x]^{t+1}=[b]-[R]\cdot[x]^{t} \tag{7.10}$$

and, for a single unknown variable:

$$x_i^{t+1}=\frac{b_i}{a_{ii}}-\sum_{j=1}^{i-1}\frac{a_{ij}}{a_{ii}}\cdot x_j^{t+1}-\sum_{j=i+1}^{n}\frac{a_{ij}}{a_{ii}}\cdot x_j^{t} \qquad i = 1,\ldots,n \tag{7.11}$$

Example:

Equation (7.11) is used to compute new approximations for the unknown variables in the following manner. Let's assume a 4 equations, 4 unknown variables equation system written as:

$$\begin{aligned}
a_{11}\cdot x_1 + a_{12}\cdot x_2 + a_{13}\cdot x_3 + a_{14}\cdot x_4 &= b_1\\
a_{21}\cdot x_1 + a_{22}\cdot x_2 + a_{23}\cdot x_3 + a_{24}\cdot x_4 &= b_2\\
a_{31}\cdot x_1 + a_{32}\cdot x_2 + a_{33}\cdot x_3 + a_{34}\cdot x_4 &= b_3\\
a_{41}\cdot x_1 + a_{42}\cdot x_2 + a_{43}\cdot x_3 + a_{44}\cdot x_4 &= b_4
\end{aligned}$$

or, in matrix form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14}\\ a_{21} & a_{22} & a_{23} & a_{24}\\ a_{31} & a_{32} & a_{33} & a_{34}\\ a_{41} & a_{42} & a_{43} & a_{44}\end{bmatrix}\cdot\begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}=\begin{bmatrix} b_1\\ b_2\\ b_3\\ b_4\end{bmatrix}$$

For solving this equations system, an initial approximation for the solution must be given: $[x]^{(0)}=[x_1^{(0)} \quad x_2^{(0)} \quad x_3^{(0)} \quad x_4^{(0)}]$, and using (7.11) and elements from matrix *[A]* and vector *[b]*, all known,

in iteration 1:

$x_1^{(1)}$ is computed based on $x_1^{(0)} \quad x_2^{(0)} \quad x_3^{(0)} \quad x_4^{(0)}$

$x_2^{(1)}$ is computed based on $x_1^{(1)} \quad x_2^{(0)} \quad x_3^{(0)} \quad x_4^{(0)}$

$x_3^{(1)}$ is computed based on $x_1^{(1)} \quad x_2^{(1)} \quad x_3^{(0)} \quad x_4^{(0)}$

$x_4^{(1)}$ is computed based on $x_1^{(1)} \quad x_2^{(1)} \quad x_3^{(1)} \quad x_4^{(0)}$

in iteration 2:

$x_1^{(2)}$ is computed based on $x_1^{(1)} \quad x_2^{(1)} \quad x_3^{(1)} \quad x_4^{(1)}$

$x_2^{(2)}$ is computed based on $x_1^{(2)} \quad x_2^{(1)} \quad x_3^{(1)} \quad x_4^{(1)}$

$x_3^{(2)}$ is computed based on $x_1^{(2)} \quad x_2^{(2)} \quad x_3^{(1)} \quad x_4^{(1)}$

$x_4^{(2)}$ is computed based on $x_1^{(2)} \quad x_2^{(2)} \quad x_3^{(2)} \quad x_4^{(1)}$

and so on, for a number of iterations, until a stopping criterion is met.

The modified Gauss-Seidel method uses and acceleration procedure which computes first the x_i^{t+1} approximation with the standard (7.11) formula, to which then applies a correction written as:

$$\begin{aligned} x_i^{t+1} &\leftarrow x_i^{t+1} + \omega_1 \cdot (x_i^{t+1} - x_i^t) \\ x_i^{t+1} &\leftarrow x_i^t + \omega_2 \cdot (x_i^{t+1} - x_i^t) \end{aligned} \tag{7.12}$$

where ω_1 and ω_2 are called acceleration factors.

In order to ensure convergence, matrix *[A]* should be diagonally dominant, i.e.

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{i-1} |a_{ij}| \qquad |a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{i-1} |a_{ji}| \tag{7. 13}$$

The stopping criterion is chosen based on a qualitative principle (for instance, the deviation between two successive approximations of the solution falls under a specified threshold) or a quantitative principle (a given number of iterations are computed).

The Gauss-Seidel method for load flow analysis

In order to adapt the general GS method to the specific case of electrical networks, the general nodal equation is used:

$$[\underline{Y}_n]\cdot[\underline{U}_n] = [\underline{J}_n] \tag{7.14}$$

in which $[Y_n]$ is the bus admittance matrix, $[\underline{U}_n]$ is the bus voltages vector, the unknown variables that need to be computed, and $[\underline{J}_n]$ is the bus current injection vector.

Using (7.14), the current $\underline{J}_i$ from a generic bus i can be written as:

$$\underline{J}_i = \underline{Y}_{ii} \cdot \underline{U}_i - \sum_{\substack{k=1 \\ k\neq i}}^{n} \underline{Y}_{ik} \cdot \underline{U}_k \qquad i = 1,\ldots,N;\ i\neq e \tag{7.15}$$

In practice, bus loads are usually known as active and reactive power, not as currents. The bus current can be written as:

$$\underline{J}_i = \frac{\underline{S}_i^*}{\underline{U}_i^*} = \frac{P_i - j\cdot Q_i}{\underline{U}_i^*} \qquad i = 1,\ldots,N;\ i\neq e \tag{7.16}$$

Combining (7.15) and (7.16), the bus voltage $\underline{U}i$ can be written as:

$$\underline{U}_i = \frac{1}{\underline{Y}_{ii}} \cdot \left(\frac{P_i - j\cdot Q_i}{\underline{U}_i^*} + \sum_{\substack{k=1 \\ k\neq i}}^{N} \underline{Y}_{ik} \cdot \underline{U}_k \right) \qquad i = 1,\ldots,N;\ i\neq e \tag{7.17}$$

Taking into account (7.3), (7.11) and (7.14), (7.17) gives the iteration formula for the Gauss-Seidel method for load flow calculations:

$$\underline{U}_i^{t+1} = \frac{1}{\underline{Y}_{ii}} \cdot \left(\frac{P_i - j \cdot Q_i^{t+1}}{(\underline{U}_i^t)^*} + \sum_{k=1}^{i-1} \underline{Y}_{ik} \cdot \underline{U}_k^{t+1} + \sum_{k=i+1}^{N} \underline{Y}_{ik} \cdot \underline{U}_k^t \right) \qquad \begin{matrix} i = 1,\ldots,N \\ i \neq e \end{matrix} \tag{7.18}$$

To speed up convergence, acceleration factors can be used, as in (7.12). The modified Gauss-Seidel method applies corrections to voltages computed in the current iteration based on their values from the previous iteration and the deviation for two successive iterations:

$$\underline{U}_i^{t+1} \leftarrow \underline{U}_i^t + \alpha \cdot (\underline{U}_i^{t+1} - \underline{U}_i^t) \tag{7.19}$$

The acceleration factor alpha is usually chosen in the [0, 2] range, and the recommended values are between 1.4 and 1.6. A sub-unit value of alpha leads usually to convergence slow down.

As stopping criteria, one of the following can be used:

- the deviation of the slack bus power injection computed in two successive iterations falls below a given threshold:

$$\Delta S_e = \left| \underline{S}_e^{t+1} - \underline{S}_e^t \right| \leq \varepsilon \tag{7.20}$$

- the maximum deviation of bus voltages computed in two successive iterations falls below a given threshold:

$$\Delta U_{\max} = \max_{\substack{i=1,\ldots,N \\ i \neq e}} \left| \underline{U}_i^{t+1} - \underline{U}_i^t \right| \leq \varepsilon \tag{7.21}$$

- all deviations between bus voltages computed in two successive iterations fall below a given threshold:

$$\left| \underline{U}_i^{t+1} - \underline{U}_i^t \right| \leq \varepsilon, \qquad \forall i = 1,\ldots,N \tag{7.22}$$

The complete algorithm for the Gauss-Seidel method for EPS load flow analysis follows below.

1. Provide the input (topology, electrical and load) parameters for the analyzed EPS
2. Provide the initial approximation for the bus voltages $\underline{U}_i^0$, $i=1..N,\ i \neq e$ and initialize the iterations count($t=0$);
3. Compute the apparent power injection at the slack bus at the beginning of the iteration:

$$\underline{S}_e^{init} = U_e \cdot \underline{J}_e^* = U_e^2 \cdot \underline{Y}_{ee}^* - U_e \cdot \sum_{\substack{k=1 \\ k \neq e}}^{N} \underline{Y}_{ek}^* \cdot (\underline{U}_k^0)^*$$

4. For the PV buses PU (i=1,…, N and i is a PV bus).
   4.1. Compute a corrected value for the bus voltage $\underline{U}_i^0$:

$$\underline{U}_i^{cor} = \frac{U_i^{setpoint} \cdot \underline{U}_i^0}{U_i^0}$$

4.2. Compute the bus reactive power:

$$Q_i^0 = \mathrm{Im}\left(\underline{U}_i^{cor} \cdot \underline{J}_i^*\right) = \mathrm{Im}\left(\left(U_i^{cor}\right)^2 \cdot \underline{Y}_{ii}^* - \underline{U}_i^{cor} \sum_{\substack{k=1 \\ k \neq i}}^{N} \underline{Y}_{ik}^* \cdot (\underline{U}_k^0)^*\right)$$

4.3. Check if the computed reactive power falls in the min-max range defined for the bus:

- if $Q_i^{min} \leq Q_i^0 \leq Q_i^{max}$, bus i remains PV, and the corrected bus voltage is kept: $\underline{U}_i^0 = \underline{U}_i^{cor}$.
- if $Q_i^0 \leq Q_i^{min}$, bus i becomes only for this iteration a PQ bus, for which $Q_i^0 = Q_i^{min}$, and the voltage correction is discarded, keeping the previous value $\underline{U}_i^0$.
- if $Q_i^0 > Q_i^{max}$, bus i becomes only for this iteration a PQ bus, for which $Q_i^0 = Q_i^{max}$, , and the voltage correction is discarded, keeping the previous value $\underline{U}_i^0$.

5. Compute the bus voltage in this iteration:

$$\underline{U}_i^1 = \frac{1}{\underline{Y}_{ii}} \cdot \left(\frac{P_i - j \cdot Q_i^0}{(\underline{U}_i^0)^*} + \sum_{k=1}^{i-1} \underline{Y}_{ik} \cdot \underline{U}_k^1 + \sum_{k=i+1}^{N} \underline{Y}_{ik} \cdot \underline{U}_k^0\right) \quad i = 1,\ldots,N;\ i \neq e$$

6. Compute the apparent power injection at the slack bus at the end of the iteration:

$$\underline{S}_e^{final} = U_e \cdot \underline{J}_e^* = U_e^2 \cdot \underline{Y}_{ee}^* - U_e \cdot \sum_{\substack{k=1 \\ k \neq e}}^{N} \underline{Y}_{ek}^* \cdot (\underline{U}_k^1)^*$$

7. Check the stopping criterion:
   - if $|\underline{S}_e^{final} - \underline{S}_e^{init}| \leq \varepsilon$, the stopping threshold is reached and the iterative process stops (go to step 8);
   - otherwise, go back to step 4, using as voltage approximations the voltages computed in this iteration ($\underline{U}_i^0 \leftarrow \underline{U}_i^1$) and the newly computed slack bus apparent power injection ($\underline{S}_e^{init} \leftarrow \underline{S}_e^{final}$);
8. The final bus voltages are the ones computed in the last iteration $\underline{U}_i^1$, $i = 1,\ldots,N$ şi $i \neq e$. The same for the slack bus apparent power injection, $\underline{S}_e^{final}$.
9. Compute the load flow auxiliary values (branch power flows, branch losses, branch voltage drops, branch loadings etc)

The branch power flows are computed with (7.23), using the annotations from Fig. 7.1

$$\begin{aligned} \underline{S}_{ik} &= U_i^2 \cdot \underline{Y}_{i0}^* + \underline{U}_i \cdot (\underline{U}_k - \underline{U}_i)^* \cdot \underline{Y}_{ik}^* \\ \underline{S}_{ki} &= U_k^2 \cdot \underline{Y}_{k0}^* + \underline{U}_k \cdot (\underline{U}_i - \underline{U}_k)^* \cdot \underline{Y}_{ki}^* \end{aligned} \qquad (7.23)$$

(usually, $\underline{Y}_{ik} = \underline{Y}_{ki}$, except transformers with complex ratio, for which $\underline{Y}_{ik} \neq \underline{Y}_{ki}$,).

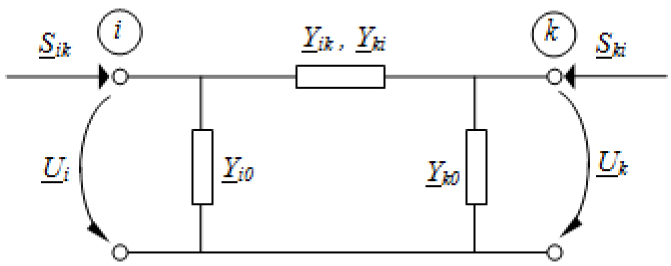

Fig. 7.1 - The PI quadrupole for computing power flows on a generic branch branch $i - k$

The branch losses and voltage drops can be immediately computed with:

$$\Delta \underline{S}_{ik} = \Delta \underline{S}_{ki} = \underline{S}_{ik} + \underline{S}_{ki} \tag{7.24}$$

$$\Delta \underline{U}_{ik} = \underline{U}_{k} - \underline{U}_{i} \tag{7.25}$$

When voltages in magnitude and angle at both ends and the bus admittance is known, the branch current is computed with:

$$\underline{I}_{ik} = \frac{\underline{U}_i - \underline{U}_k}{\underline{Z}_{ik}} \tag{7.26}$$