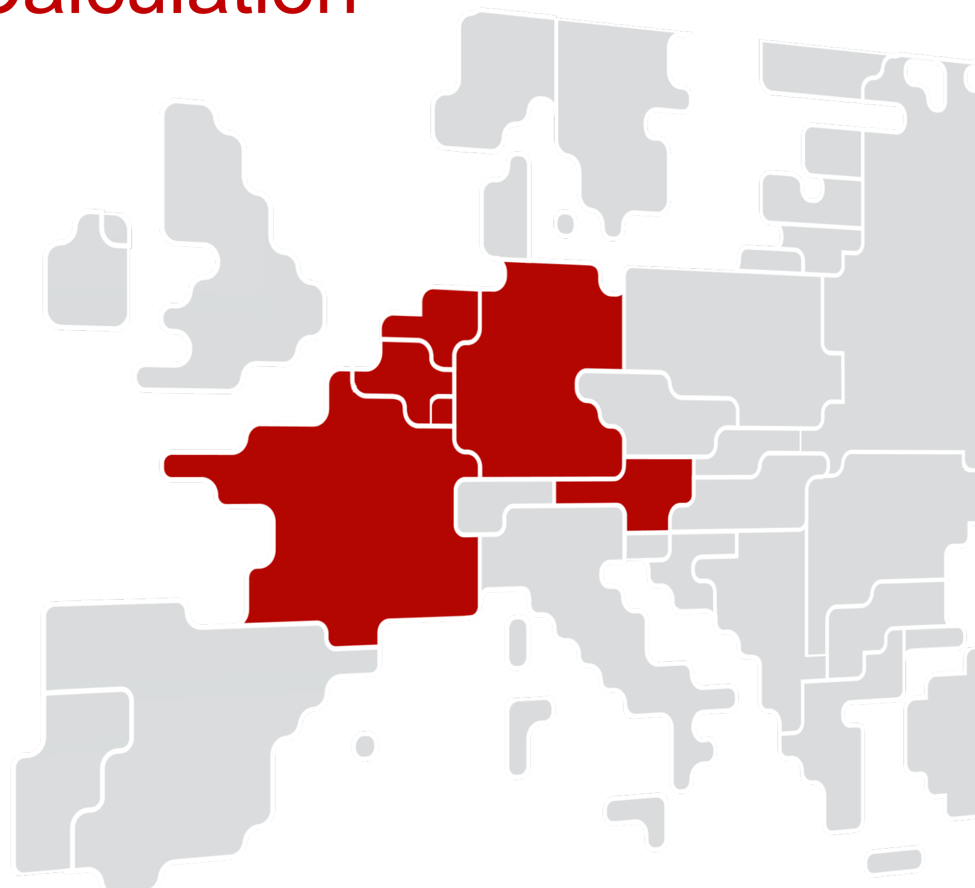
# Background information for CWE Flow Based Capacity Calculation



# DC Load flow

# DC Load flow

The purpose of a DC load flow is to compute the phase angles for every node, before deducing the power flows across the grid.

- Reference flow calculation is calculated at the end of the load flow processing
    - Fref is the active flow on a branch.

## Formulation

- In a DC load flow algorithm, the power flow equations are simplified in such a way that its solution is non-iterative.
- Power flow equations:

$$\mathrm{P_k} = \sum_{j=1}^{n} |V_k||V_j|\left(G_{kj}\cos\left(\theta_k - \theta_j\right) + B_{kj}\sin\left(\theta_k - \theta_j\right)\right)$$

$$\mathrm{Q_k} = \sum_{j=1}^{n} |V_k||V_j|\left(G_{kj}\sin\left(\theta_k - \theta_j\right) - B_{kj}\cos\left(\theta_k - \theta_j\right)\right)$$

These equations are simplified according to the assumptions listed below:

1. Line resistance is neglected since it is significantly less than the line reactance.

$$\mathrm{P_k} = \sum_{j=1}^{n} |V_k||V_j|\left(B_{kj}\sin\left(\theta_k - \theta_j\right)\right)$$

$$\mathrm{Q_k} = \sum_{j=1}^{n} |V_k||V_j|\left(-B_{kj}\cos\left(\theta_k - \theta_j\right)\right)$$

$V_k$: *voltage magnitude bus k*
$Y_{ik} = G_{ik} + B_{ik}$: *Admittance between bus i and k*
$\theta_k$: *Voltage phase angle bus* $V_k$
$P_k$: *Active power injected at bus k*
$Q_k$: *Reactive power injected at bus k*

# DC Load flow

2. The angular separation across any transmission circuit is small, and the sine of small angle is the angle itself; the cosine approaches 1.

$$\mathrm{P_k} = \sum_{j=1}^{n} |V_k||V_j| \left(\mathrm{B_{kj}}(\theta_\mathrm{k} - \theta_\mathrm{j})\right)$$

$$\mathrm{Q_k} = \sum_{j=1}^{n} |V_k||V_j| \left(-\mathrm{B_{kj}}\right)$$

3. The shunt admittances are not considered.
4. The real power flow is significantly larger than the reactive power flow, so that the reactive power flow is discarded.

The bus voltage magnitudes are determined from the voltage values given in the configuration file. They are converted in p.u. using the nominal voltages provided in CGM file.
The equation to solve is then:

$$\mathrm{P_k} = \sum_{j=1\ j\neq k}^{n} |V_k||V_j| \left(\mathrm{B_{kj}}(\theta_\mathrm{k} - \theta_\mathrm{j})\right)$$

More generally, by taking into account the transformer ratios and the phase shifts:

$$\mathbf{P_k} = \sum_{j=1\ j\neq k}^{n} \rho_k \rho_j |V_k||V_j| \left(\mathbf{B_{kj}}(\theta_\mathbf{k} - \theta_\mathbf{j} + \alpha_\mathbf{kj})\right)$$

Where

$\rho_k$ is the transformer ratio at node k if the branch kj is a transformer (1 if no transformer)
$\alpha_{kj}$ is the phase shift if the branch kj is a phase-shifter (0 if no phase shifter)

## Algorithm

The DC power flow equations are given in matrix form as $\mathbf{P} = \mathbf{B'\theta}$

Where P is the vector of nodal injections
Θ is the vector of nodal phase angles
B’ is the nodal admittance matrix with the following elements:

$$\mathrm{b'_{ii}} = \sum_{j=1\ j\neq i}^{n} \mathrm{b_{ij}}$$

$$\mathrm{b'_{ij}} = -b_{ij} = -\frac{1}{X_{ij}}$$

The matrix B’ is singular because all angles are not independent. To make the system solvable, one of the equations in the system is removed, and the bus associated with that row is selected as the angle reference, with a fixed value of 0 degrees.

In forming the B’ matrix, phase-shifter transformers are treated like transmission lines
Once the angles calculated, the power flow across branch i-j may be deduced according to:

$$\mathbf{P_{ij} = \rho_i \rho_j V_i V_j B_{ij}(\theta_i - \theta_j + \alpha_{ij})}$$

# Nodal PTDF computation

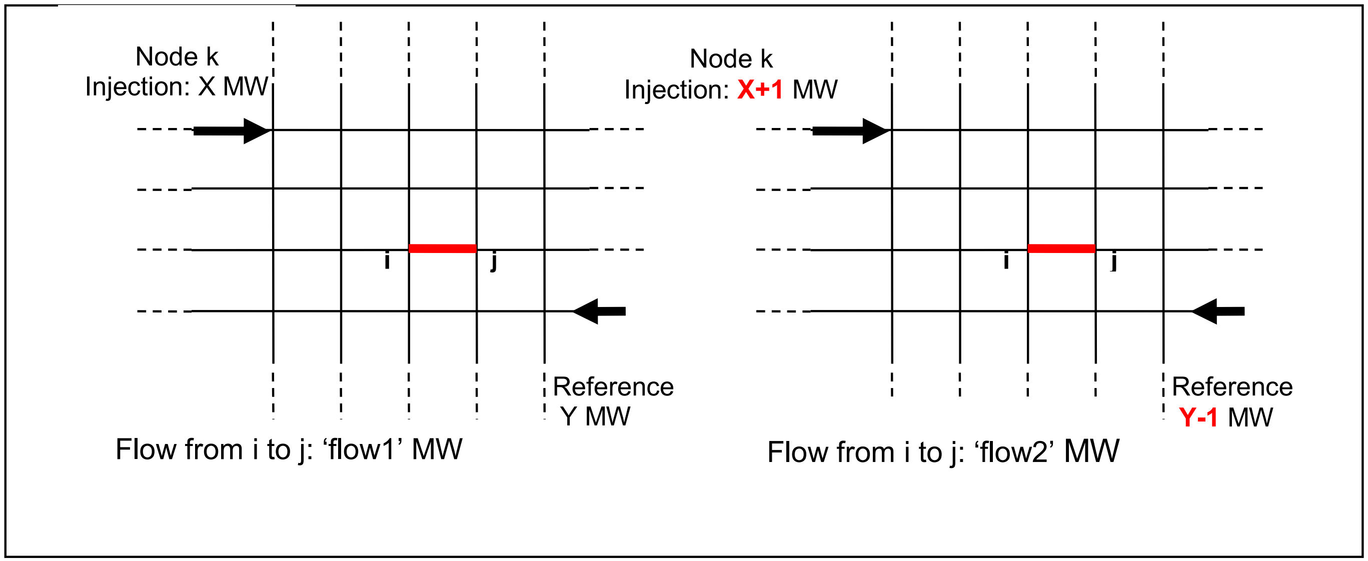
# Nodal PTDF computation

## The nodal PTDF factors are calculated for each constraint.

- They represent the linearized sensitivity factors between the constrained quantity (MW flow), and a pair of source/sink controls.
- Source controls include nodal real power injections/withdrawals. The sink control is the slack bus.

## The sensitivity of a constraint to an injection/withdrawal identifies the change in the MW flow on the constrained branch if the control injection (or withdrawal) is increased (or decreased) by 1 MW.

- The sensitivities depend on the network topology and on the reference chosen to pick up the additional MW.
- This reference corresponds by default to the reference bus.




In the example above, the sensitivity of the MW flow from node i to node j with respect to the injection at node k is a MW value equal to ‘$flow_2$’ minus ‘$flow_1$’.

# Nodal PTDF computation

## Formulation

The incremental power transfer distribution factor (PTDF) is the derivative of the line flow with respect to a change in injection (and corresponding change in withdrawal).

With the DC assumptions, the flow between nodes i and j is given by the following formula:

$$P_{ij} = \rho_i \rho_j V_i V_j B_{ij} (\theta_i - \theta_j + \alpha_{ij})$$

By taking derivatives with respect to real power injection at a node k, we obtain the following formulation for shift factors:

$$\frac{\partial P_{ij}}{\partial P_k} = \rho_i \rho_j V_i V_j B_{ij} \left( \frac{\partial \theta_i}{\partial P_k} - \frac{\partial \theta_j}{\partial P_k} \right)$$

If the derivatives of bus angles with respect to injections $\frac{\partial \theta_i}{\partial P_k}$ are known, shift factors can be determined.

Those derivatives can be computed using the equation: $\Delta P = B' \Delta\theta$. This gives $\frac{\partial \theta_i}{\partial P_k} = [B'_{ik}]^{-1}$

Shift factors with regard to unit injections can then be expressed as follows:

$$\frac{\partial P_{ij}}{\partial P_k} = B_{ij} \left( [B'_{ik}]^{-1} - [B'_{jk}]^{-1} \right)$$

## Algorithm

To numerically compute the sensitivity of the MW flow of a line linking node i and node j to MW injection at each node, the following equation is solved: $\boldsymbol{B'\theta = \psi}$

Where:

$\boldsymbol{\psi}$ is a sparse N vector whose ith element is $B_{ij}$ and whose jth element is $-B_{ij}$.

$\boldsymbol{\theta}$ is obtained thanks to standard forward and backward elimination techniques and holds the line PTDFs with respect to a change in power injection at each node.

# Net Position Forecast

## Background

- The objective of CWE TSOs was to develop a robust and extensible methodology to forecast CWE NPs, firstly for D-1 market in order to enhance the flow-based qualification.

- Net position forecast uses a statistical approach, based on linear regression
  - Two years of historical data are used for calibration taking into account only past Net Positions of the CWE countries (No assumptions on the market based on « market data » such as prices).

- NP methodology is used operationally on the CWE region since November 2017 and forecasts are consistently better than the heuristic.
- CWE TSOs are following the quality of NP forecast along the year.
- The forecasts of NP can be improved using other data (renewable production and consumption forecasts, outages of production units, to name a few)

# Qhull algorithm

**The Qhull algorithm implemented by GE is similar to the reference one explained on the following website: http://www.qhull.org/**

For the sake of clarity, please find the process which uses the Qhull algorithm.

**Process (Vertices Algorithm)**

The Vertices module provides the vertices of a domain described by linear equalities and inequalities. The Qhull is used for that purpose.

- As Qhull takes only inequalities as input, an elimination of the equalities present in the set of constraints is necessary. Moreover there may be an issue if the equalities are not linearly independent. If it is the case, a first step should be added to remove linearly dependent equalities.
- Qhull requires also that the user provides an initial point clearly inside the domain:
- If the initial point is not clearly inside the domain, a blocking error is raised. If no initial point is provided, the point (0, 0….0) is used.

**Elimination of equalities**

It is possible to determine each vertex of the initial set of constraints with n linearly independent equalities where n is the number of variables and p>n inequalities by:

1. eliminating n variables from the p inequalities;
2. determining the vertices of the resulting subset of inequalities;
3. and building back the vertices in the original variables space.

Let us consider:

$$\left.\begin{array}{l} L \cdot x = a \\ M \cdot x \le b \end{array}\right\} \quad \text{[initial]}$$

where :

- $x$ is the n-vector of variables
- $L$ is a $(p,n)-$matrix of rank $p$
- $a$ is the p-vector
- $M$ is a $(q,n)-$matrix with $q > n$
- $b$ is the q-vector

As $\text{rank}(L) = p$, $L$ can be decomposed in two parts:

$$L \cdot x = L_B \cdot x_B + L_S \cdot x_S$$

where :

- $L_B$ is a inversible $(p,p)-$matrix
- $L_S$ is a $(p,n-p)-$matrix

Hence:

$$L \cdot x = a$$

Can be rewritten as:

$$x_B = L_B^{-1} \cdot (a - L_S \cdot x_S)$$

The same partition of variables $x_B$ and $x_S$ can be used to decompose M:

$$M \cdot x = M_B \cdot x_B + M_S \cdot x_S$$

where :

- $M_B$ is a inversible $(q,p)-$matrix
- $M_S$ is a $(q,n-p)-$matrix

Hence:

$$M \cdot x \le b$$

Can be rewritten as:

$$M_B \cdot L_B^{-1} \cdot (a - L_S \cdot x_S) + M_S \cdot x_S \le b$$

Finally we get:

$$(M_S - M_B \cdot L_B^{-1} \cdot L_S) \cdot x_S \le b - M_B \cdot L_B^{-1} \cdot a \qquad \text{[reduce]}$$

And:

$$x_B = L_B^{-1} \cdot (a - L_S \cdot x_S)$$

As a conclusion, it is possible to determine each vertex of the initial set by the following process:

- determine each vertex of reduce set of inequalities (variables xs [reduce])
- and then build the vertex of the full set [initial] by complements with the variable xb. based on the relation between $x_B$ and $x_S$.