CONTENTS

1. ELEMENTS

The program considers the following elements: loads, power lines, transformers, synchronous generators, voltage source converters, and back-to-back converters. The modelling approach and their attributes are particularly detailed below.

1.1. Loads

Loads are shunt devices, that is, they are connected between a given bus and the reference bus. They can be treated as constant impedance loads, which by definition are linear, and power loads. It is common practice to view them as power loads, hence the power flow problem becomes non-linear. Figure 1 depicts the model of the load.

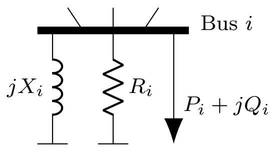

Figure 1. Load model.

In the `load` sheet from the input file, the attributes are as shown in Table 1.

Attribute	Description	Units
`bus`	Bus where the load is connected	-
`R`	Parallel resistance	p.u.
`X`	Parallel reactance	p.u.
`P`	Active power	p.u.
`Q`	Reactive power	p.u.

Table 1. Attributes of the load model.

Note that if `R` and `X` are set to 0, the program understands there is no constant impedance load. The reactance `X` can take both positive and negative values, emulating an inductance or capacitance respectively.

1.2. Power lines and transformers

They are series devices that interconnect buses. Because of their similarity, they are all specified in the `branch` sheet. Nonetheless, their modelling differs slightly.

Power lines are represented by the π equivalent as seen in Figure 2.

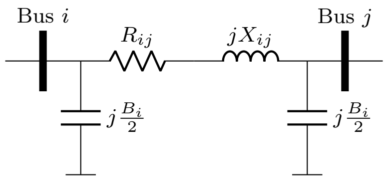

Figure 2. Line model.

Transformers may have an off-nominal tap ratio and may introduce as well a phase shift typically employed to regulate the active power. Its equivalent model is shown in Figure 3.

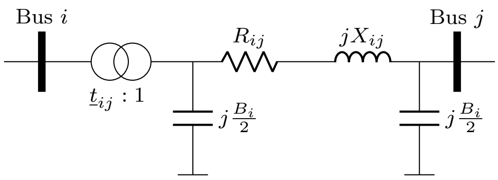

Figure 3. Transformer model.

The full admittance matrix is composed of the following submatrix:

$$\mathbf{Y_{trafo}} = \begin{pmatrix} \underline{Y}_{ii} & \underline{Y}_{ij} \\ \underline{Y}_{ji} & \underline{Y}_{jj} \end{pmatrix}, \tag{1}$$

where each element is given by:

$$\begin{cases} \underline{Y}_{ii} = \dfrac{R_{ij} + jX_{ij} + j\frac{B_i}{2}}{t_{ij}^2}, \\ \underline{Y}_{ij} = -\dfrac{R_{ij} + jX_{ij}}{\underline{t}_{ij}^*}, \\ \underline{Y}_{ji} = -\dfrac{R_{ij} + jX_{ij}}{\underline{t}_{ij}}, \\ \underline{Y}_{jj} = R_{ij} + jX_{ij} + j\frac{B_i}{2}. \end{cases} \tag{2}$$

The parameters to introduce in the `branch` file are gathered in Table 2.

Attribute	**Description**	**Units**
`bus_from`	Initial bus i	-
`bus_to`	Final bus j	-
`R`	Series resistance	p.u.
`X`	Series reactance	p.u.
`B`	Parallel susceptance	p.u.
`tap_module`	Absolute value of $\underline{t}_{ij}$	p.u.
`tap_angle`	Angle of $\underline{t}_{ij}$	degrees

Table 2. Attributes of the branch model.

In the end, series devices such as power lines and transformers are both modelled following the scheme in Figure 3. If one wishes to specify a power line, then the tap changer has to be set at $1\angle 0$. Besides, transformers are commonly modelled with a single series inductive reactance. The user can set a null resistance and susceptance without loss of generality.

1.3. Synchronous generators

The synchronous generator is a shunt device, that is, connected to a single bus. It can work as a PQ, PV, or slack bus depending on the preference. Figure 4 presents its simple equivalent model.

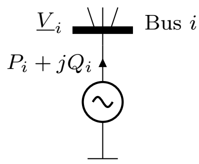

Figure 4. Synchronous generator model.

The attributes that conform it are captured in Table 3.

Attribute	Description	Units
`bus`	Bus of connection	-
`P`	Active power	p.u.
`Q`	Reactive power	p.u.
`V`	Reference voltage	p.u.
`kf`	Frequency droop constant	1/Hz
`kv`	Voltage droop constant	p.u.
`type`	0: slack, 1:PQ, 2:PV	-

Table 3. Attributes of the synchronous generator model.

While all parameters have to be defined, depending on the type (PQ, PV, or slack) some magnitudes are ignored. In other words, in PQ mode `P` and `Q` are relevant, in PV mode `P` and `V`, and if acts as the slack bus, `V` is the only important variable. According to the sign convention, injected powers are positive if they flow from the element towards the bus.

1.4. Voltage source converters

Voltage source converters behave just like synchronous generators from the power flow perspective. Their equivalent model is displayed in Figure 5.

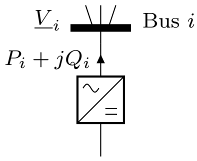

Figure 5. Voltage source converter model.

Again, their parameters are collected in Table 4.

Attribute	Description	Units
`bus`	Bus of connection	-
`P`	Active power	p.u.
`Q`	Reactive power	p.u.
`V`	Reference voltage	p.u.
`kf`	Frequency droop constant	1/Hz
`kv`	Voltage droop constant	p.u.
`type`	0: slack, 1:PQ, 2:PV	-

Table 4. Attributes of the voltage source converter model.

There is no difference in the modelling of voltage source converters with respect to synchronous generators. Nevertheless, the power flow results are divided so as to be associated with its corresponding device.

1.5. Back-to-back converters

The back-to-back converter configuration is assumed to involve two voltage source converters with different controls. It has been considered that both converters are linked by a single resistance. Figure 6 shows the general scheme.

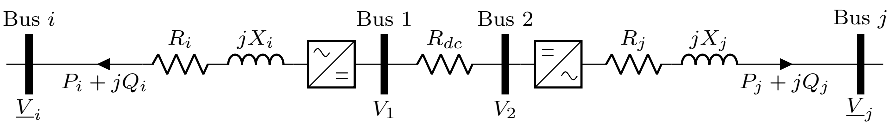

Figure 6. Back-to-back converter model.

The control strategy, on the one hand, consists of setting the powers P_i and Q_i. The latter can obey a voltage droop law, as will be described later on. On the other hand, the second converter regulates the voltage V_2 and the reactive power Q_j. As mentioned before, Q_j can also follow the droop law and hence depend on V_j.

Some of the parameters indicated in Figure 6 act as seeds for the power flow.

In any case, Table 5 contains the attributes by which back-to-back converters are defined.

Attribute	Description	Units
`bus1`	Initial bus i	-
`bus2`	Final bus j	-
`P1`	Active power at bus i	p.u.
`Q1`	Reactive power at bus i	p.u.
`P2`	Active power at bus j	p.u.
`Q2`	Reactive power at bus j	p.u.
`V1`	Reference voltage at bus i	p.u.
`V2`	Reference voltage at bus j	p.u.
`Vdc`	Reference voltage at bus 2	p.u.
`R1`	Filter resistance at bus i	p.u.
`X1`	Filter reactance at bus i	p.u.
`R2`	Filter resistance at bus j	p.u.
`X2`	Filter reactance at bus j	p.u.
`Rdc`	DC resistance	p.u.
`kf1`	Frequency droop constant at bus 1	1/Hz
`kv1`	Voltage droop constant at bus 1	p.u.
`kf2`	Frequency droop constant at bus 2	1/Hz
`kv2`	Voltage droop constant at bus 2	p.u.
`type`	0: slack, 1:PQ, 2:PV	-

Table 5. Attributes of the back-to-back converter model.

2. FORMULATION

2.1. General scheme

Since the system is formed by non-linear equations, the solver is based on the Newton-Raphson method. The system is formally defined as:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{f}$ denotes the non-linear equations and $\boldsymbol{x}$ is the set of unknowns.

The Newton-Rapshon linearizes the system and solves the following:

$$\boldsymbol{\Delta f} = -\boldsymbol{J}\boldsymbol{\Delta x}, \tag{4}$$

where $\boldsymbol{\Delta f}$ represents the residual vector, $\boldsymbol{J}$ stands for the Jacobian, and $\boldsymbol{\Delta x}$ is the vector containing the sought variation of unknowns.

The residuals are formed by the active and reactive power equations. In compact form:

$$\boldsymbol{\Delta f} = [\boldsymbol{\Delta f_p}; \boldsymbol{\Delta f_q}]^T, \tag{5}$$

where $\boldsymbol{\Delta f_p}$ is the active power residual and $\boldsymbol{\Delta f_q}$ symbolizes the reactive power residual.

Similarly, the unknowns are structured as follows:

$$\boldsymbol{\Delta x} = [\boldsymbol{\delta}; \boldsymbol{V}; f], \tag{6}$$

where $\boldsymbol{\delta}$ is the vector of the voltages angles, $\boldsymbol{V}$ is the vector of voltages magnitudes, and f corresponds to the frequency of the system to be found.

Thus, the full Jacobian $\boldsymbol{J}$ can be constituted by six submatrices as indicated below:

$$\begin{pmatrix} \boldsymbol{\Delta f_p} \\ \boldsymbol{\Delta f_q} \end{pmatrix} = -\begin{pmatrix} \frac{d\boldsymbol{f_p}}{d\boldsymbol{\delta}} & \frac{d\boldsymbol{f_p}}{d\boldsymbol{V}} & \frac{d\boldsymbol{f_p}}{df} \\ \frac{d\boldsymbol{f_q}}{d\boldsymbol{\delta}} & \frac{d\boldsymbol{f_q}}{d\boldsymbol{V}} & \frac{d\boldsymbol{f_q}}{df} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{V} \\ f \end{pmatrix}, \tag{7}$$

which for simplicity are represented as:

$$\begin{pmatrix} \boldsymbol{\Delta f_p} \\ \boldsymbol{\Delta f_q} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{J1} & \boldsymbol{J2} & \boldsymbol{J3} \\ \boldsymbol{J4} & \boldsymbol{J5} & \boldsymbol{J6} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{V} \\ f \end{pmatrix}, \tag{8}$$

Regarding the power flow expressions, recall that the complex power in matrix form is:

$$\boldsymbol{S} = [\boldsymbol{V}]\boldsymbol{Y}^*\boldsymbol{V}^*, \tag{9}$$

where $[\boldsymbol{V}]$ indicates the diagonal matrix of the $\boldsymbol{V}$ vector, and $\boldsymbol{Y}^*$ is the conjugated admittance matrix. With this, the residual of active power becomes:

$$\boldsymbol{\Delta f_p} = \Re(\boldsymbol{S}) - \boldsymbol{P_{ref}} - [\boldsymbol{k_f}](f_{ref} - f), \tag{10}$$

where $\boldsymbol{P_{ref}}$ refers to the vector of active power setpoints, $[\boldsymbol{k_f}]$ is the diagonal matrix filled with frequency droop constants, f_{ref} is the frequency reference typically of 50 Hz, and f is the unknown frequency. Note that in the case of a load, its k_f would take a null value, hence its active power demand will be constant independently of the frequency.

Analogously, for the reactive power:

$$\boldsymbol{\Delta f_q} = \Im(\boldsymbol{S}) - \boldsymbol{Q_{ref}} - [\boldsymbol{k_v}][\boldsymbol{\nu}](\boldsymbol{V_{ref}} - \boldsymbol{\nu}), \tag{11}$$

where $\boldsymbol{Q_{ref}}$ represents the specified reactive power injections, $[\boldsymbol{k_v}]$ groups the voltage droop constants in the diagonal, $\boldsymbol{V_{ref}}$ is the vector of voltage magnitudes references, and $\boldsymbol{\nu}$ contains the absolute value of the actual voltages.

The six matrices that compose the Jacobian are obtained by deriving Equations 10 and 11 with respect to the unknowns. In compact form:

$$\begin{cases} \boldsymbol{J1} = \Re\left(j[\boldsymbol{V}]([\boldsymbol{Y}\boldsymbol{V}]^* - \boldsymbol{Y}^*[\boldsymbol{V}]^*)\right), \\ \boldsymbol{J2} = \Re\left([\boldsymbol{V}]([\boldsymbol{Y}\boldsymbol{V}]^* + \boldsymbol{Y}^*[\boldsymbol{V}]^*)[\boldsymbol{\nu}]^{-1}\right), \\ \boldsymbol{J3} = [\boldsymbol{k_f}], \\ \boldsymbol{J4} = \Im\left(j[\boldsymbol{V}]([\boldsymbol{Y}\boldsymbol{V}]^* - \boldsymbol{Y}^*[\boldsymbol{V}]^*)\right), \\ \boldsymbol{J5} = \Im\left([\boldsymbol{V}]([\boldsymbol{Y}\boldsymbol{V}]^* + \boldsymbol{Y}^*[\boldsymbol{V}]^*)[\boldsymbol{\nu}]^{-1}\right) + \boldsymbol{2}[\boldsymbol{k_v}][\boldsymbol{\nu}], \\ \boldsymbol{J6} = \boldsymbol{0}. \end{cases} \tag{12}$$

Note that the Jacobians have to be sliced. This means that not all entries have to be selected. For instance, slicing $\boldsymbol{J1}$ would mean selecting the rows where the active power is a specified magnitude. Overall, defining the Jacobians and the residuals in matrix form allows us to easily handle these expressions and achieve shorter computational times.

2.2. AC/DC power flow integration

Reconsider the back-to-back scheme shown in Figure 7. The first converter controls the power at bus i whereas the second converter regulates the voltage

V_2 and the reactive power Q_j.

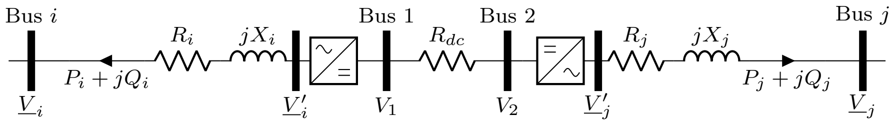

Figure 7. Back-to-back converter model.

If bus i acts as a PQ bus, then the voltage $\underline{V}'_i$ is found with:

$$\underline{V}'_i = \underline{V}_i + \frac{P_i - jQ_i}{\underline{V}^*_i}(R_i + jX_i), \tag{13}$$

With this, the power transmitted by the first converter is:

$$P'_i = \Re\left(\underline{V}'_i \frac{P_i + jQ_i}{\underline{V}_i}\right), \tag{14}$$

and assuming an efficiency of 100%, this is the same active power that enters bus number 1. Let this power be $P_{dc,1}$. As voltage V_2 is set, V_1 has to be found by solving:

$$P_{dc,1} = \frac{V_2 - V_1}{R_{dc}} V_1, \tag{15}$$

which results in the closed-form solution:

$$V_1 = \frac{V_2 + \sqrt{V_2^2 - 4R_{dc}P_{dc,1}}}{2}. \tag{16}$$

Then the power entering bus 2 is simply:

$$P_{dc,2} = \frac{V_1 - V_2}{R_{dc}} V_2, \tag{17}$$

and this corresponds to the active power P'_j.

The power P_j, which has to be updated, can be estimated as:

$$P_j^{(k+1)} = P'_j - \left|\frac{P_j^{(k)} - jQ_j}{\underline{V}^*_j}\right|^2 R_j, \tag{18}$$

where $P_j^{(k+1)}$ denotes the updated power and $P_j^{(k)}$ is the considered power in the previous iteration.

It has to be mentioned that this approach works well as long as the back-to-back converter does not form an island. In case two AC systems are isolated, the general Jacobian would become singular. Hence, the proposed solution would no longer be valid.

3. PROGRAM DESCRIPTION

From the perspective of a high level of abstraction, Figure 8 shows the required steps to go from the input file to the solution of the power flow.

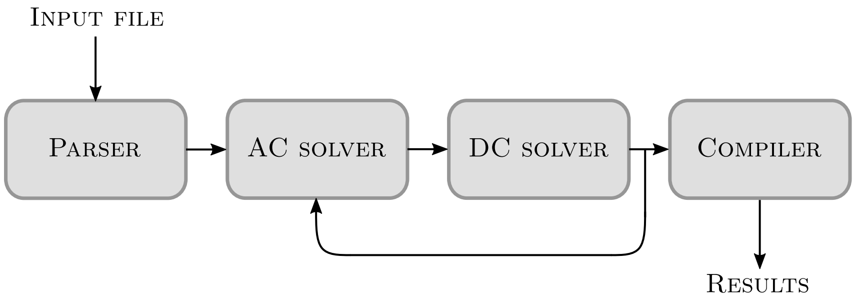

Figure 8. Overview of the program.

The parser goes from the raw `.xlsx` input file to structured data. A general grid class is employed to store the vectors and matrices. Besides, the parser also processes this information by building the admittance matrix, for instance.

The AC solver is responsible for solving the typical AC power flow, as described in Chapter 2. The DC solver is subsequently called to update the magnitudes of the back-to-back converter. This in turn influences the AC power flow, so there is an outer loop to iterate in this regard.

Finally, once the error has reached a threshold, the compiler is used to calculate the final results. They are organized inside a final results class.

3.1. Algorithms

The engine as such operates as illustrated in Algorithm 1.

Algorithm 1: Main stages of the full solver.

Data: `grid`, ϵ
Result: Solved system in `results`
Initialize objects with `grid=parse(grid)`;
while $\max(\boldsymbol{\Delta f}) > \epsilon$ **do**
  `grid` = `solver(grid)`;
  `grid` = `solver_dc(grid)`;
end
`results` = `Results(grid)`;

The AC solver, encapsulated inside the `solver()` function, is the main part of the engine. In short, it calculates the residuals, computes the Jacobian and its inverse, and obtains the variations of the unknowns. Algorithm 2 presents the steps to follow.

Algorithm 2: Main instructions of the AC solver.

Data: `grid`, ϵ, `n`
Result: Updated unkowns in `grid`
[`vec_d`, `vec_v`] = `vec_dv`(`grid`);
[`vec_p`, `vec_q`] = `vec_pq`(`grid`);
while $\max(\boldsymbol{\Delta f}) > \epsilon$ *and* $\boldsymbol{k} < \boldsymbol{n}$ **do**
  $\boldsymbol{\Delta f}$ = `residuals(grid, vec_p, vec_q)`;
  $\boldsymbol{J}$ = `jacobian(grid, vec_p, vec_q, vec_d, vec_v)`;
  $\boldsymbol{\Delta x} = \boldsymbol{J}^{-1}\boldsymbol{\Delta f}$;
  `k` += 1;
end

The first two lines of code select the indices where the angles and magnitudes of the voltages are unknown, and the indices where the active and reactive power are specified. These lists are then used to form the residuals and the Jacobian. This is in essence what slicing is about. The iterations inside the loop finish if the error is sufficiently small or if the number of iterations exceeds a certain maximum. There are no iterations in the DC solver as it is limited to updating the unspecified power.