Developing a harmonic power flow software in distributed generation systems

CÉDRIC BUREAU

Degree project in
Electric Power Systems
Second Level,
Stockholm, Sweden 2012

XR-EE-ES 2012:011
Master thesis report

Developing a harmonic power flow software in distributed generation systems

Student: Cédric BUREAU
Supervisor: Lennart SÖDER

EDF R&D, Département Mesure et Système d'Information des Réseaux Electriques
1, avenue du Général de Gaulle
92141 CLAMART
Abstract

The main topic of this thesis is harmonic power flow and its use in a simulation software that I have developed. The idea of the software is to combine distribution grids’ description, non-linear load models and power flow methods.

Nowadays, power electronics is more and more present in electric devices in distributed generation systems. Those power electronics systems can emit or absorb harmonics that can damage the devices in the grid. Thus, it is important to be able to estimate harmonic behaviour in the grid in order to be able to prevent the possible problems that could occur.

The main contribution of this internship is the precise expression of the needs and goals of the software, and an implementation of its structure. In this thesis, it is explained how the grid’s components and non-linear devices are modelled in the software in order to be able to represent the distribution system. There is also a study the possible input of this software and create a symbolic representation of the grid that is helpful when it comes to load flow calculation. Then, the different load flow and harmonic load flow algorithms that are presented in the literature are analysed and compared them together in order to determine the methods that should be implemented in the future software.

Two of the implemented fundamental load flows with a single-phase system are tested. Thus, it also validates the input reading and the grid representation construction.

The software developed is a first implementation of a more global software that will require further studies. Indeed, the development stage will be done by external contractors or computer science specialists, that will insist on parallelization of algorithms and software optimization, in order to have a software as efficient and fast as possible.
Acknowledgements

First of all, I would like to express my gratitude to my supervisor Professor Lennart Söder, for his advice and support during the project.

I would like to thank the members of the group *Power Quality* of EDF R&D and especially Vincent Diemunsch and Christophe Santander for their support and encouragement and for providing stimulating atmosphere during my internship.
# Table of contents

**ABSTRACT** .................................................................................................................................................. 2

**ACKNOWLEDGEMENT** ............................................................................................................................... 3

**TABLE OF CONTENTS** ................................................................................................................................ 4

1. **INTRODUCTION** ........................................................................................................................................ 7
   1.1. **PRESENTATION OF POWER QUALITY GROUP IN MIRE AT EDF** ................................................... 7
   1.1.1. **Presentation of the company** ........................................................................................................ 7
   1.1.2. **Presentation of Power Quality Group** .......................................................................................... 7
   1.2. **BACKGROUND** .................................................................................................................................... 7
   1.3. **PLAN OF THE THESIS** ..................................................................................................................... 8

2. **SOFTWARE DESCRIPTION** ....................................................................................................................... 8
   2.1. **GLOBAL DESCRIPTION** .................................................................................................................. 8
   2.2. **DEFINITION OF BASIC ELEMENTS** ............................................................................................... 9
   2.2.1. **Node and Bus definition** ............................................................................................................. 9
   2.2.2. **Harmonic definition** .................................................................................................................... 9
   2.3. **CONTRIBUTION OF THIS SOFTWARE** ........................................................................................ 9
   2.3.1. **Existing software** ....................................................................................................................... 9
   2.3.1.1. **HARMONIQUE** ................................................................................................................... 9
   2.3.1.2. **ExperTEC** .......................................................................................................................... 10
   2.3.2. **Advantages of the software** ....................................................................................................... 10

3. **LIBRARY OF COMPONENTS - FREQUENCY MODELS** ........................................................................ 10
   3.1. **LIBRARY PRESENTATION** .............................................................................................................. 10
   3.2. **MATRICES FOR EACH COMPLEX COMPONENT** ........................................................................... 11
   3.2.1. **Model utilization** ...................................................................................................................... 12
   3.2.1.1. **Line and cable** ..................................................................................................................... 12
   3.2.1.2. **Three-phase transformer** ..................................................................................................... 13
   3.2.2. **Example of models calculation** .................................................................................................. 16
   3.2.2.1. **Single-phase transformer** .................................................................................................... 16
   3.2.2.2. **Single-phase autotransformer** .............................................................................................. 18
   3.2.2.3. **Three phase cable** .............................................................................................................. 18
   3.2.2.4. **Three-phase line** ............................................................................................................... 19
   3.2.2.5. **Impedance** ........................................................................................................................... 19
   3.2.3. **Linear Bus** .................................................................................................................................. 20
   3.2.4. **Non linear Bus** .......................................................................................................................... 20

4. **INPUT AND OUTPUT DATA ANALYSIS** ............................................................................................... 21
   4.1. **INPUT DATA ANALYSIS** .............................................................................................................. 21
   4.1.1. **CIM format** ............................................................................................................................... 21
   4.1.2. **ASC format** ................................................................................................................................ 22
   4.2. **OUTPUT DATA ANALYSIS** ........................................................................................................... 22
   4.2.1. **CSV Format** ............................................................................................................................ 22
   4.2.2. **Simple CSV Format** ................................................................................................................. 22

5. **STRUCTURED REPRESENTATION OF THE ELECTRICAL GRID** ....................................................... 23
   5.1. **INTERMEDIARY FILE USED IN THE SOFTWARE** ....................................................................... 23
   5.2. **SYMBOLIC REPRESENTATION** .................................................................................................... 24
   5.3. **NUMERICAL REPRESENTATION WITH ADMITTANCE MATRIX** ................................................ 25
   5.3.1. **Representation with an admittance matrix** .............................................................................. 25
   5.3.1.1. **Association of two component admittance matrix** ................................................................. 25
6. ALGORITHMS REVIEW ........................................................................................................27

6.1. FUNDAMENTAL LOAD FLOW ALGORITHMS .................................................................27

6.1.1. Gauss-Seidel ..................................................................................................................27

6.1.1.1. Method description ......................................................................................................27

6.1.1.2. Advantages and Drawbacks .......................................................................................27

6.1.2. Newton-Raphson ..........................................................................................................27

6.1.2.1. Method description ......................................................................................................27

6.1.2.2. Advantages and Drawbacks .......................................................................................29

6.1.3. Fast Decoupled Load Flow .........................................................................................29

6.1.3.1. Method description ......................................................................................................29

6.1.3.2. Advantages and Drawbacks .......................................................................................30

6.1.4. Backward-Forward Sweep .........................................................................................30

6.1.4.1. Method description ......................................................................................................30

6.1.4.2. Advantages and Drawbacks .......................................................................................31

6.1.5. Compensation based load flow ....................................................................................31

6.1.5.1. Method description ......................................................................................................31

6.1.5.2. Advantages and Drawbacks .......................................................................................32

6.1.6. Algorithms comparison ...............................................................................................32

6.2. HARMONIC LOAD FLOW METHODS ............................................................................33

6.2.1. Harmonic Penetration .................................................................................................33

6.2.1.1. Method description ......................................................................................................33

6.2.1.2. Advantages and Drawbacks .......................................................................................34

6.2.2. Iterative Harmonic Penetration ...................................................................................34

6.2.2.1. Method description ......................................................................................................34

6.2.2.2. Advantages and Drawbacks .......................................................................................35

6.2.3. Simplified Harmonic load flow .....................................................................................35

6.2.3.1. Method description ......................................................................................................35

6.2.3.2. Advantages and Drawbacks .......................................................................................36

6.2.4. Harmonic Newton-Raphson .......................................................................................36

6.2.4.1. Method description ......................................................................................................36

6.2.4.2. Advantages and Drawbacks .......................................................................................39

6.2.5. Harmonic Backward Forward Sweep ..........................................................................39

6.2.5.1. Method description ......................................................................................................39

6.2.5.2. Advantages and Drawbacks .......................................................................................41

6.2.6. Algorithms comparison ...............................................................................................41

6.3. CHOICE OF ALGORITHMS ..........................................................................................41

7. ARCHITECTURE, SOFTWARE ENGINEERING ..............................................................46

7.1. SPECIFICATIONS ..............................................................................................................46

7.2. SOFTWARE ARCHITECTURE .........................................................................................49

8. RESULTS AND COMPARISON .......................................................................................50

9. CONCLUSION ....................................................................................................................50

10. REFERENCE ......................................................................................................................51

ANNEX A : INPUT AND OUTPUT FORMAT EXAMPLES ....................................................52

ANNEX B: NETWORK SEPARATION AND PARALLELIZATION OF CALCULATIONS ..........53

B.1 - CYCLES DETECTION ALGORITHM .............................................................................53

B.2 - RADIAL PART CREATION ALGORITHM .....................................................................55

B.3 - SMALL RADIAL PART SUPPRESSION ALGORITHM ..................................................56

ANNEX C : PACKAGE DIAGRAM ..........................................................................................56
1. Introduction

1.1. Presentation of Power Quality Group in MIRE at EDF

1.1.1. Presentation of the company

EDF (« Electricité De France ») is one of the world’s largest utility company and is present in every parts of the electricity activities, from production to distribution. Since the promulgation of the law in 2004 about the deregulation of the electricity market, two major activities have become subsidiaries of EDF: the electricity power transmission managed by RTE and the distribution managed by ErDF.

The group exists mainly in Europe and South America and tries to develop its activities in Asia and in the USA.

Some relevant statistics:
- around 160000 employees
- 37 million of customers
- 2009 Turnover: €63.34 billion
- 630.4 TWh produced in 2010

1.1.2. Presentation of Power Quality Group

There are more than 2 000 researchers in the Research and Development department in EDF.

The main mission of the department “Measurement and Information Systems of Electrical Networks” is to analyze the behavior of the transmission system, the distribution system, or even private electrical systems. Employees try to increase the global efficiency of those networks. Its annual budget is 13 M€.

The Power Quality Group's mission is to develop methods and tools that will help to control that the electricity sold by EDF respect the quality agreement and the needs of the customers. Some of its activities are:
- representing EDF in the national and international standardization institution
- offering an expertise about electricity quality to other groups in EDF or ErDF
- developing or testing new instrumentation solution
- developing and maintaining the computer tools that helps to realize studies about electricity quality

1.2. Background

Power electronics is more and more present in electric domestic and industrial devices, in distributed generation systems and even in electric cars chargers. Those power electronics systems have a tendency to absorb or emit distorted currents from a 50 Hz sinusoid. Thus, harmonic voltages appear in the grid and they can damage the systems.

In order to prevent the problems, it can be interesting to be able to simulate existing network behavior when some non-linear buses are connected and observe where harmonics voltages appear.

The combination of non-linear loads’ models and harmonic power flow software could give interesting information in order to improve systems stability and safety.

In the Power Quality Group, non linear loads frequency models are built. Furthermore, a module that realizes time domain simulation of non-linear loads from those models is developed. Thus, there is a need for a simulation software that could simulate existing grids and realize a harmonic power flow by communicating with that module in order to get an accurate response of the loads.

This internship’s goal is to write the specifications, develop such a software and compare the results
with existing simulation methods.

1.3. Plan of the thesis

In the following part of this thesis, we will globally describe the software we are developing by presenting its purpose and how it operates. Then, we will explain the contribution of this software as regards the existing software used by EDF.

In the third part, we will describe the components’ library by introducing some components models.

In the fourth part, we will describe the possible input format and the output format that would be interesting to develop.

In the fifth part, we will explain how the network’s description will be internally represented and how it will be used by the calculation module.

In the following part, we will describe the existing algorithm that can be used in order to realize fundamental and harmonic power flow. We will try to select some of them to be implemented in the software.

In the seventh part, we will briefly explain the software’s architecture by presenting the uses cases and the main packages that constitute the software.

In the last part, we will discuss the results by comparing our software with the existing ones.

2. Software description

2.1. Global description

The goal of this software is to read a real electrical network description and to make determinist frequency calculations in order to obtain harmonic levels. It must take into account three-phase networks, distributed generation and non-linear loads.

This software shall also be able to communicate with an external module that is developed by the department in order to realize time domain simulation of non-linear loads. For this study, we will consider that the network is linear. Furthermore, I will not implement the connection with the external module. Thus, we will only use known models of non-linear loads.

We shall also be able to be connected with OpenTURNS, a software that models uncertainty of parameters, in order to realize multiple simulations of the same network, modifying at each iteration the value of a few parameters.
Figure 1 - Software description

In this first implementation of the software, I will not implement the connection with OpenTURNS. We will modify ourselves the network if we want to. The user will be able to choose what kind of load flow he wants to use and the voltages and currents he wants to observe.

Even if all the functionalities are not implemented in the first version, we will take care of developing the software in the most modular way in order to easily integrate future modules.

2.2. Definition of basic elements

2.2.1. Node and Bus definition

A node represents a physical point in the system.

A bus represents a group of nodes to which a component is connected.

For instance we have for a three-phase component:

![Figure 2 - Three-phase component example](image)

2.2.2. Harmonic definition

A harmonic is a component frequency of the signal that is a multiple of the fundamental frequency $f_0 = 50Hz$.

Thus, we call the $h^{th}$ harmonic the signal that has a frequency $f = h.f_0$. It is defined by the equation:

$$\forall i \in \{1..n\}, \quad V_i^h = |V_i|^h \exp(j.\theta_i^h)$$

The voltage angle $\theta$ refers to the reference voltage which is taken at the source of the studied system, which is usually the point before the transformer of the substation.

In this study, the last harmonic we will take into account is the $L^{th}$ harmonic.

2.3. Contribution of this software

2.3.1. Existing software

2.3.1.1. HARMONIQUE

The software HARMONIQUE was developed by EDF and is used to calculate frequency simulation. It can calculate voltages and currents on LV and MV electrical networks.

This software has many issues:
Developing a harmonic power flow software in distributed generation systems

- It can only simulate single-phase grids.
- The calculation is independent for every harmonics.
- The current sources used to model non-linear loads are calculated without taking into account the voltage at the considered node.
- The grid size is limited to 300 nodes.
- There is no load flow calculation.

2.3.1.2. ExperTEC

This software was also developed by EDF in the department I am doing this internship. It can realize permanent and transient analysis, and calculate harmonic voltages values.

The main advantages of this software are:
- It can simulate three phase unbalanced systems.
- Components are very detailed and their parameters can be easily changed by a user.
- It can realize simulation on networks that contains up to 5000 nodes.

However, this software has many issues:
- It does not realize a real load flow.
- It has been developed by one person that left the department.

2.3.2. Advantages of the software

The main advantage of our new software is that it will realize both fundamental and harmonic load flow. Different methods will be implemented in order to compare algorithms and determine the most efficient, by taking into account the accuracy of the results and the execution time.

Besides, we will be able to connect this software with an external module that realizes temporal simulation of a non-linear load. Thus we could use this module at each iteration of the harmonic load flow to get an accurate value of the non-linear load behavior, corresponding to a value of harmonic voltages at that load.

Furthermore, we will also be able in the future to use OpenTURNS in order to realize parametric simulations. The software and the intermediary file format will be designed in order to ease the components’ or network’s modifications done by OpenTURNS.

Finally, this software will be able to read files that describes existing distribution network. Thus, we could observe the impact of a non-linear load connection on a specific real grid.

3. Library of components - frequency models

3.1. Library presentation

In order to be able to realize a harmonic load flow, we need to have a frequency model for every network components and ending components called equipments.

We choose to calculate the models from the physical or basic electrical data usually given by the components manufacturers. The input data format shall give those data.
The idea is that we should be able to calculate the \(Y_{bus}\) matrix from the manufacturers’ data.

The components we take into account can be single-phase or three-phase electrical components. Furthermore, we describe the basic elements \(Z\) that can be used to model losses in a component.

**Figure 3 - components library**

In the following section, we will describe the complex calculation of one components \(Y_{bus}\) matrix for each frequency, which take into account impact of the frequency on the component behavior.

The components \(Y_{bus}\) matrix describes the component behaviour. For example, for the component described in Figure 2, the matrix is described by the equation

\[
\begin{bmatrix}
I_1 \\
I_2 \\
I_3 \\
I_4 \\
I_5 \\
I_6
\end{bmatrix}
= \begin{bmatrix}
V_1 \\
V_2 \\
V_3 \\
V_4 \\
V_5 \\
V_6
\end{bmatrix} \times Y_{bus}
\]

In order to write the equations, I used the work made previously in order to develop the software ExperTEC. The development of those models has been done and described in an internal report concerning ExperTEC (cf. [17], [18], [19]).

**3.2. Matrices for each complex component**

In order to model the components, we have to be able to get a component admittance matrix for each
3.2.1. Model utilization

Here we will explain how we can use simple models of single-phase cable, line in order to obtain three phases models of those components. For transformer, we can either use three single phase model to calculate the three phase component admittance matrix or use directly a three phase model based on the electromagnetic equations and then take into account the impact of windings coupling.

3.2.1.1. Line and cable

![Line and cable model](figure4.png)

The \( p \) and \( s \) indexes represent respectively the primary and secondary parts of the component.

In order to analyze this element, we will study separately its serial part and its parallel part. First, we model the serial part of the line.

With this model, the impedance matrix at frequency \( f \) of the serial part is, for the \( abc \) components:

\[
\begin{bmatrix}
\Delta V_a \\
\Delta V_b \\
\Delta V_c
\end{bmatrix} =
\begin{bmatrix}
V_{pa} - V_{sa} \\
V_{pb} - V_{sb} \\
V_{pc} - V_{sc}
\end{bmatrix} =
\begin{bmatrix}
Z_p & Z_m & Z_m \\
Z_m & Z_p & Z_m \\
Z_m & Z_m & Z_p
\end{bmatrix}
\begin{bmatrix}
I_a \\
I_b \\
I_c
\end{bmatrix}.
\]

The entries of the matrix depend on the frequency \( f \).

Then, we use the symmetrical components and the component impedance matrix at frequency \( f \) becomes:

\[
\begin{bmatrix}
\Delta V_0 \\
\Delta V_1 \\
\Delta V_2
\end{bmatrix} =
\begin{bmatrix}
Z_p + 2Z_m & 0 & 0 \\
0 & Z_p - Z_m & 0 \\
0 & 0 & Z_p - Z_m
\end{bmatrix}
\begin{bmatrix}
I_0 \\
I_1 \\
I_2
\end{bmatrix}.
\]

Then, we model the parallel part of the line (the shunt impedance).

For the \( abc \) parameters, the component admittance matrix at frequency \( f \) of the parallel part becomes:
Developing a harmonic power flow software in distributed generation systems

\[
\begin{bmatrix}
I_{pa} \\
I_{pb} \\
I_{pc}
\end{bmatrix} =
\begin{bmatrix}
Y_{qp} + 2Y_{pp} & -Y_{qp} & -Y_{qp} \\
-Y_{qp} & Y_{qp} + 2Y_{pp} & -Y_{qp} \\
-Y_{qp} & -Y_{qp} & Y_{qp} + 2Y_{pp}
\end{bmatrix}
\begin{bmatrix}
V_{pa} \\
V_{pb} \\
V_{pc}
\end{bmatrix}.
\]

Then, we use the symmetrical components and the admittance matrix at frequency \( f \) becomes:

\[
\begin{bmatrix}
I_{p0} \\
I_{p1} \\
I_{p2}
\end{bmatrix} =
\begin{bmatrix}
Y_{qp} & 0 & 0 \\
0 & Y_{qp} + 3Y_{pp} & 0 \\
0 & 0 & Y_{qp} + 3Y_{pp}
\end{bmatrix}
\begin{bmatrix}
V_{p0} \\
V_{p1} \\
V_{p2}
\end{bmatrix}.
\]

Thus, we can consider for the symmetrical components that a three phase line is actually three independent single-phase lines. The parameters of those lines can be calculated from the parameters of the three-phase line.

We can indeed calculate for each symmetrical component the admittance matrix of the corresponding line, at frequency \( f \):

\[
\begin{bmatrix}
I_p \\
I_s
\end{bmatrix} =
\begin{bmatrix}
Y^p + Y^s & -Y^s \\
-Y^s & Y^p + Y^s
\end{bmatrix}
\begin{bmatrix}
V_p \\
V_s
\end{bmatrix}.
\]

Thus we can determine the component admittance matrix for all the symmetrical components, at frequency \( f \), by combining the three admittance matrices of the three symmetrical components:

\[
Y_{012} =
\begin{bmatrix}
Y^p_0 + Y^s_0 & 0 & 0 & -Y^s_0 & 0 & 0 \\
0 & Y^p_1 + Y^s_1 & 0 & 0 & -Y^s_1 & 0 \\
0 & 0 & Y^p_2 + Y^s_2 & 0 & 0 & -Y^s_2 \\
-Y^s_0 & 0 & 0 & Y^p_0 + Y^s_0 & 0 & 0 \\
0 & -Y^s_1 & 0 & 0 & Y^p_1 + Y^s_1 & 0 \\
0 & 0 & -Y^s_2 & 0 & 0 & Y^p_2 + Y^s_2
\end{bmatrix}
\]

By taking into account the length of the line, we have: \( Y_{T_{012}} \).

The component admittance matrix for the abc components for the differential values of the voltages at frequency \( f \) is then obtained:

\[
Y_{abc} = T_S^{-1} Y_{T_{012}} T_S^{-1}
\]

Where \( T_S \) is Fortescue’s Matrix:

\[
T_S =
\begin{bmatrix}
1 & 1 & 1 \\
1 & \alpha^2 & \alpha \\
1 & \alpha & \alpha^2
\end{bmatrix}
\]

### 3.2.1.2. Three-phase transformer

**First Method**

In this part, we will explain how we can use a model of single-phase transformer that is different for zero sequence and positive, negative sequence, in order to calculate the component admittance matrix of a three-phase transformer.

The phases of the transformer are called a, b and c and we have:
The component admittance matrix of the transformer at frequency \( f \) is defined by:

\[
\begin{bmatrix}
[y^a_p] \\
[y^b_p] \\
[y^c_p]
\end{bmatrix} = Y_{abc} \begin{bmatrix}
[i^a_p] \\
[i^b_p] \\
[i^c_p]
\end{bmatrix}
\]

Then, we use the symmetrical component: \( y^a_p \) and \( y_{abc} \) with

\[
\begin{bmatrix}
[y^0_p] \\
[y^1_p] \\
[y^2_p]
\end{bmatrix} = Y_{abc} \begin{bmatrix}
[i^0_p] \\
[i^1_p] \\
[i^2_p]
\end{bmatrix}
\]

The component admittance matrix at frequency \( f \) becomes:

\[
Y_{012} = \begin{bmatrix}
T^{-1}_{S} & T_{S}^{-1}
Y_{pp} . T_{S} & T_{S}^{-1} Y_{ps} \cdot T_{S}
Y_{sp} . T_{S} & T_{S}^{-1} Y_{ss} \cdot T_{S}
\end{bmatrix}
\]

We obtain three independent modes. We can determine for each of them a value of the admittance matrix by using a model of single-phase transformer. Entries of matrices depend on the frequency \( f \). Thus, by calculating the value of the zero, positive and negative sequence, we can obtain the values of \( Y_{pp}, Y_{sp}, Y_{ps} \) and \( Y_{ss} \) and then the matrix \( Y \) for the abc components, at frequency \( f \).

**Second Method**

From the transformer electromagnetic equations, we use a model described in an internal report within EDF R&D (cf. [17]) for each kind of flux: free flux, independent flux and forced flux.

We simplify those models with the following hypothesis: equality of reluctances. It means that the transformer has a symmetric structure. Thus, as it is described in [17], we find a 6x6 matrix \( Z_{abc} \) that is the differential impedance matrix in the abc components system, that describes the relation between currents and potential differences, at frequency \( f \):

\[
\begin{bmatrix}
V^a_{pa} \\
V^b_{pb} \\
V^c_{pc} \\
V^a_{sa} \\
V^b_{sb} \\
V^c_{sc}
\end{bmatrix} = Z_{abc} \begin{bmatrix}
I^a_p \\
I^b_p \\
I^c_p \\
I^a_s \\
I^b_s \\
I^c_s
\end{bmatrix}
\]

The entries of \( Z_{abc} \) depend on the frequency \( f \).

By inverting it, we find the corresponding admittance matrix \( Y_{abc} \).
In order to take into account the impact of windings coupling on the component admittance matrix, we have to calculate the nodal admittance matrix $Y_{Nodal\_abc}$ (12x12) at frequency $f$ defined by

$$
\begin{bmatrix}
I_0 \\
I_1 \\
\vdots \\
I_{10} \\
I_{11}
\end{bmatrix}
= Y_{Nodal\_abc}
\begin{bmatrix}
V_0 \\
V_1 \\
\vdots \\
V_{10} \\
V_{11}
\end{bmatrix},
$$

by doing the following calculation:

$$
Y_{Nodal\_abc} = \begin{bmatrix}
1 & -1 & 0 & \cdots & 0 \\
-1 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-1 & -1 & -1 & \cdots & 1
\end{bmatrix}
$$

Then, we calculate a matrix $C$ that represents the impact of windings coupling on this admittance matrix. For instance, for a Wye-Delta-1 transformer, this coupling matrix will be:
Developing a harmonic power flow software in distributed generation systems

Finally, we get an 8x8 component admittance matrix $Y'$ that fully describes the transformer behavior at a given frequency $f$:

$$C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

The calculation is made using the following formula:

$$Y = C \cdot Y_{\text{Nodal, abc}} \cdot C'^T.$$

It is this matrix that will be returned when one asks its admittance at a given frequency to a component.

### 3.2.2. Example of models calculation

#### 3.2.2.1. Single-phase transformer

This study is done at a given frequency $f = \frac{\omega}{2\pi}$.

Manufacturer’s data:

- $V_p$: primary rms voltage in V
- $V_s$: secondary rms voltage in V
- $S_a$: apparent power in VA
- $v_{sc}$: short circuit voltage in V
- $I_0$: no-load current in A
- $R_p$, $R_s$: winding resistance in $\Omega / m$

**Model:**
Figure 6 – Single-phase transformer model

Model's elements calculation:
First we have to calculate:

The turns ratio: \( n = \frac{V_p}{V_s} \)

primary leakage inductance: \( l_{11} = \frac{V_m}{100} \frac{Z_{np}}{2\omega} \) with \( Z_{np} = \frac{V_p^2}{S_n} \)

secondary leakage inductance: \( l_{22} = \frac{V_m}{100} \frac{Z_{ns}}{2\omega} \) with \( Z_{ns} = \frac{V_i^2}{S_n} \)

Thus, we can know the model elements:

\[
L_p = \frac{V_p}{\omega I_0}
\]

\[
k = 1 - \frac{l_{11}}{L_p}
\]

\[
L_s = \frac{l_{22}}{1 - k}
\]

\[
M = k \sqrt{L_p L_s}
\]

\[
Y_{bus} \text{ matrix, at frequency } f = \frac{\omega}{2\pi}
\]

\[
Y = \begin{bmatrix} y_s & y_m \\ y_m & y_p \end{bmatrix} \text{ with: } \\
\begin{align*}
y_s &= \frac{R_s + jL_s \omega}{R_p + jL_p \omega + \left(R_p L_s + R_s L_p\right) \omega + \left(M^2 - L_p L_s\right) \omega^2} \\
y_p &= \frac{R_p R_s + j\left(R_p L_s + R_s L_p\right) \omega + \left(M^2 - L_p L_s\right) \omega^2}{R_p R_s + j\left(R_p L_s + R_s L_p\right) \omega + \left(M^2 - L_p L_s\right) \omega^2 - jM \omega} \\
y_m &= \frac{R_p R_s + j\left(R_p L_s + R_s L_p\right) \omega + \left(M^2 - L_p L_s\right) \omega^2}{R_p R_s + j\left(R_p L_s + R_s L_p\right) \omega + \left(M^2 - L_p L_s\right) \omega^2}
\end{align*}
\]
3.2.2.2. Single-phase autotransformer

This study is done at a given frequency \( f = \frac{\omega}{2\pi} \).

Data: the given data are the same as transformer's one.

Model:

\[ R_{po} = R_p - R_s \]

\( L_{po} \) is calculated by solving the following equation:

\[
L_{po}^2 - 2L_{po}(L_p + (2k^2 - 1)L_s) + (L_p - L_s)^2 = 0
\]

\( Y_{bus} \) matrix:

Increasing voltage autotransformer:

\[
Y = \begin{bmatrix}
R_p + R_{so} + j\omega(L_p + L_{so} + 2M) & -R_p - j\omega(L_p + M) \\
-R_p - j\omega(L_p + M) & R_p + j\omega L_p
\end{bmatrix} \frac{1}{R_p R_{so} + 2j\omega R_s R_{so} + \omega^2(M^2 - L_p L_{so})}
\]

Decreasing voltage autotransformer:

\[
Z = \begin{bmatrix}
R_p + j\omega L_s & -R_s - j\omega (L_s + M) \\
-R_s - j\omega (L_s + M) & R_s + R_{po} + j\omega (L_{po} + L_s + 2M)
\end{bmatrix} \frac{1}{R_s R_{po} + 2j\omega R_p R_{po} + \omega^2(M^2 - L_s L_{po})}
\]

3.2.2.3. Three phase cable

In order to model, the three phase cable, I use model that have been developed in EDF R&D. Those models are confidential and I cannot detail them in the report.
Developing a harmonic power flow software in distributed generation systems

With this model, we can find positive, negative and zero sequence a numerical value of admittance matrix, at frequency f.

\[
\begin{bmatrix}
I_p \\
I_s
\end{bmatrix} = \begin{bmatrix}
Y^p + Y^s & -Y^s \\
-Y^s & Y^p + Y^s
\end{bmatrix} \begin{bmatrix}
V_p \\
V_s
\end{bmatrix}
\]

Thus, we can have the component admittance matrix for the three-phase component at frequency f as it is explained in 3.2.1.1:

\[
Y_{abc} = \begin{bmatrix}
T_s & 0 \\
0 & T_s
\end{bmatrix} \begin{bmatrix}
Y_0^p + Y_0^s & 0 & 0 & -Y_0^s & 0 & 0 \\
0 & Y_1^p + Y_1^s & 0 & 0 & -Y_1^s & 0 \\
0 & 0 & Y_2^p + Y_2^s & 0 & 0 & -Y_2^s \\
- Y_0^s & 0 & 0 & Y_0^p + Y_0^s & 0 & 0 \\
0 & - Y_1^s & 0 & 0 & Y_1^p + Y_1^s & 0 \\
0 & 0 & - Y_2^s & 0 & 0 & Y_2^p + Y_2^s
\end{bmatrix} \begin{bmatrix}
T_s & 0 \\
0 & T_s
\end{bmatrix}^{-1}
\]

3.2.2.4. Three-phase line

In order to model, the three-phase line, I use model that have been developed in EDF R&D. Those models are confidential and I cannot detail them in the report.

With this model, we can find for the positive, negative and zero sequence a numerical value of admittance matrix.

Thus, we can have the component admittance matrix for the three-phase component at the frequency f as it explained in 3.2.1.1.

3.2.2.5. Impedance

This component is a fixed impedance and the frequency does not modify its value.

Data:
- \(R\): resistance in \(\Omega\)
- \(X\): reactance in \(\Omega\)

Ybus matrix:
- If the impedance is connected between two different nodes of the network, the component admittance matrix is:
  \[
  Y = \begin{bmatrix}
  \frac{1}{R + jX} & \frac{1}{R + jX} \\
  \frac{1}{R + jX} & \frac{1}{R + jX}
  \end{bmatrix}
  \]

- If the impedance is connected between a node of the network and the ground, the component admittance matrix that is used is:
  \[
  Y = \frac{1}{R + jX}
  \]
3.2.3. Linear Bus

In this study, we only consider single-phase linear loads.

Those components are not included in the system admittance matrices, for every harmonics.

At fundamental frequency, linear buses are defined by the two parameters that are known. Thus, a linear bus can be a PQ bus, a PV bus or a $\theta V$ bus which is called the slack bus.

Even if those buses do not produce harmonic currents, they have a passive behavior as regards harmonics that go through the network. This behavior can be modeled by the following impedance (cf. [3]), where the upper part is an impedance and the lower part is a reactance.

$$\frac{(V^1)^2}{P^1}$$

$$\frac{h(V^1)^2}{Q^1}$$

**Figure 8 – Linear Bus Harmonic Model**

With $P^1$: fundamental active power

$Q^1$: fundamental reactive power

$V^1$: fundamental voltage

$h$: harmonic rank

This model is equivalent to a current injection, which is the sum of the current injection in the reactance and the current injection in the impedance:

$$I^h = \frac{V^h \cdot P^1}{(V^1)^2} - \frac{jV^h \cdot Q^1}{h(V^1)^2}$$

3.2.4. Non linear Bus

Those components are not included in the system admittance matrix.

The fundamental behavior of those buses is the same behavior as PQ bus. It means that during a fundamental load flow calculation, we will consider that the NLD has a known active and reactive fundamental power.

For harmonic calculations, we will consider that the non-linear loads are equivalent harmonic current injectors (cf. [2], [23]). Models that describe those loads are obtained by different ways:

- Some models that describe very specific loads have been developed by a partner university for EDF (cf. [21]). The format of injected current according to those models is given by:

$$I^k_i = g^k_i \left(V^1_i, V^2_i, ..., V^k_i, Z_{eq}\right)$$

where $L$: last harmonic

$k$: harmonic number

$i$: node number
Developing a harmonic power flow software in distributed generation systems

\[ Z_{eq} \]: equivalent impedance of the network seen from the load.

- Some behaviors will be obtained by communicating with an external module that realizes time
domain analysis of non-linear loads. The input will be \((V_i^1)^\delta, (V_i^2)^\delta, \ldots, (V_i^{L_i})^\delta\) and \(Z_{eq}\). The output will be \((I_i^k)^\delta\).

- Some models have been deduced from measurements. The format of injected current
according to those models is given by: \((I_i^k)^\delta = g_k((V_i^1)^\delta, (V_i^2)^\delta, \ldots, (V_i^{L_i})^\delta)\). \hspace{1cm} \text{Eq. A}

In the first version of this software, only some models obtained from measurements will be
available.

For instance, harmonic measurements have been done with a specific load (cf. [20]). After a
treatment, the model obtained has the format:

\[
\begin{bmatrix}
I^1_1 & I^1_2 & \cdots & \cdots & \cdots & I^1_{17} \\
I^3_1 & I^3_2 & \cdots & \cdots & I^3_{17} \\
I^5_1 & I^5_2 & \cdots & \cdots & \cdots \\
I^7_1 & I^7_2 & \cdots & \cdots & \cdots \\
I^9_1 & I^9_2 & \cdots & \cdots & \cdots \\
I^{11}_1 & I^{11}_2 & \cdots & \cdots & \cdots \\
I^{13}_1 & I^{13}_2 & \cdots & \cdots & \cdots \\
\end{bmatrix}
\begin{bmatrix}
y_{11} \\
y_{12} \\
y_{21} \\
y_{22} \\
\vdots \\
\vdots \\
y_{71} \\
y_{72} \\
\end{bmatrix}
= \begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
y_5 \\
\vdots \\
y_7 \\
y_8 \\
\end{bmatrix}
\begin{bmatrix}
V^1 \\
V^2 \\
V^3 \\
V^4 \\
V^5 \\
\vdots \\
V^7 \\
V^8 \\
\end{bmatrix}
\hspace{1cm} \text{Eq. B}

4. Input and Output data analysis

In this part, we will explain what input format we have to use in order to transfer data from a database
to the calculation software and what output format we choose to use in order to be able to visualize
and treat the results.

4.1. Input data analysis

We want to input a standard format used by EDF and the distribution system manager ErDF in order
to describe the distribution system. Two different formats are currently used: ASC and CIM. The ASC
format is used by ErDF for a long time and the CIM is a more recent standard format.

4.1.1. CIM format

It is the most detailed format between the two. It is actually a flat XML representation of the network in
which every component is separated in many different subcomponents. The description of the components is very detailed and many information are irrelevant to our study (for example, installation date of a cable).

### 4.1.2. ASC format

This format is actually a flat text file in which every component is described. Those files can be created from CIM file thanks to existing software. The ASC format does not contain all the needed parameters (for example, transformer winding’s configuration) but its reading is easier than CIM’s one because one component is described by only one compact group of data.

We decide to use firstly ASC file as input for our software. Thus, we will have to add the needed information that is not described in the format by using default values or user defined values.

### 4.2. Output data analysis

With this software, the user will be able to choose the voltages and currents he wants to observe. When he asks the system to observe some harmonic voltages at a specific component or some harmonic currents at a specific bus, the information will be stored in an export setting. The user can add as many export setting as he wants.

#### 4.2.1. CSV Format

The CSV format is designed to be easily read by Microsoft Excel. Thus, results analysis and treatment would be efficient.

This format contains a main header that contains the input path, the simulation description and the software version.

The idea of this file is to create a table where the line corresponds to the components and the buses the user wants to observe and the columns correspond to the value the user wants to observe at least one time.

Thus, this format also contains a second header that corresponds to the columns title and shows all the values that will be observed for at least one component. For instance, in a single phase network, if the user wants to observe fundamental and third harmonic voltages for a component C1 and fundamental and fifth harmonic voltages for a component C2, the second header would be:

```
SimulationNumber ; ComponentName ; V^fund ; θ^fund ; V^3 ; θ^3 ; V^5 ; θ^5
```

Then, for each export setting, which corresponds to one component or bus, a line is written in the file with its identifier and all the value the user wants to observe at their proper places.

One output file corresponds to one initial input network, one choice of algorithms and one list of export settings. If the network is modified, with OpenTURNS for instance, the results are written in the same file.

A complete example is given in ANNEX A.

#### 4.2.2. Simple CSV Format

The simple CSV format is designed to be easily read by a human being. The data are more difficult to analyze with another software.

In this format, there is no second header and the results are written in the order described by the export settings, giving the name of the value and the value for each parameter.

A complete example is given in ANNEX A.
In the first implementation of this software, we will implement the simple CSV format module.

5. **Structured representation of the electrical grid**

5.1. **Intermediary file used in the software**

We need a file format in order to communicate with OpenTURNS, which requires a text file in order to be able to read the parameters value. Thus, we will use an intermediary file that describes the system and contains all the relevant information that OpenTURNS may change. When we will use OpenTURNS to modify the value of a data, it will only change the value in the intermediary file and the input file will remain the same.

![Figure 10 - Intermediary File and OpenTURNS action](image)

In the intermediary file, every system’s component is described by a single line that contains all its parameters:

```plaintext
Id_compo # type # param_1 = x_1 ; … ; param_p = x_p # N1 = Id_N1 ; N1 = Id_N2 # topology_info
```

The component’s type may be:

- Terminal load
- Transformer
- Line, Cable
- Switch
- Auto transformer
- Current source, voltage source

The topological information, when they are known, contains:

- Geographic coordinates
- Segment it belongs to
- The substation
- MV/LV substation
- MV feeder
- LV feeder
The idea of this format is to have a simple and efficient way to describe a network that is particularly adapted to our needs.

This format will also be used as an input format in the first implementation of the software.

### 5.2. Symbolic representation

We will store the network information in a symbolic representation. This representation is made of a symbolic matrix which indicates the components that are connected between two nodes. An element of the matrix is actually a list of the components that are connected between the two nodes corresponding to the line and the column indexes. Each of these components is then described with its type and parameters in the list of components.

Besides, the diagonal elements of the matrix indicate the loads that are connected at this node. Each of these loads is then described with its type and parameters in the list of components.

We also have to store the position of sources in order to be able to determine the radial and nodal parts of the grid.

![Figure 11 - Example network](image)

**Symbolic table of the grid:**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{i1}$</td>
<td>$C_{21}$</td>
<td>...</td>
<td>$C_{ij}$</td>
<td>...</td>
<td>$C_{in}$</td>
</tr>
<tr>
<td>$C_{i2}$</td>
<td>$C_{22}$</td>
<td>...</td>
<td>$C_{2j}$</td>
<td>...</td>
<td>$C_{2n}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$C_{j1}$</td>
<td>$C_{j2}$</td>
<td>...</td>
<td>$C_{ij}$</td>
<td>...</td>
<td>$C_{jn}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$C_{n1}$</td>
<td>$C_{n2}$</td>
<td>...</td>
<td>$C_{nj}$</td>
<td>...</td>
<td>$C_{nn}$</td>
</tr>
</tbody>
</table>

where the $C_{ij}$ represent the connected components between node $i$ and node $j$.

List of connected components (list of Identifiers):
### Components’ list:

<table>
<thead>
<tr>
<th>Component Id</th>
<th>Type of component</th>
<th>Parameter 1</th>
<th>Parameter 2</th>
<th>...</th>
<th>Parameter p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line_1</td>
<td>line</td>
<td>Length = 1km</td>
<td>Section = 150mm²</td>
<td></td>
<td>Resistivity = ...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load_j</td>
<td>PQ Load</td>
<td>P=100W</td>
<td>Q=500W</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Sources’ list:

| Bus_2        | ... | ... |

### 5.3. Numerical representation with admittance matrix

#### 5.3.1. Representation with an admittance matrix

##### 5.3.1.1. Association of two component admittance matrix

5.3.1.1. In series

If two components are in series, without any load between them, the network can be simplified by putting them together into one unique component.

Thus, if two junctions A and B have respectively for component admittance matrix

\[
Y_A = \begin{bmatrix}
Y_{A,11} & Y_{A,21} \\
Y_{A,12} & Y_{A,22}
\end{bmatrix}
\]

and

\[
Y_B = \begin{bmatrix}
Y_{B,11} & Y_{B,21} \\
Y_{B,12} & Y_{B,22}
\end{bmatrix}
\]

and are in series, the component admittance matrix of the equivalent component is:

\[
Y = \begin{bmatrix}
Y_{A,11} - Y_{A,12}Y_kY_{A,21} & -Y_{A,12}Y_kY_{B,21} \\
-Y_{B,21}Y_kY_{A,21} & Y_{B,22} - Y_{B,21}Y_kY_{B,21}
\end{bmatrix}
\]

with

\[
Y_k = \left( Y_{A,22} + Y_{B,11} \right)^{-1}
\]

### 5.3.1.2. Creation of the system admittance matrix

In 3.2, we explain how we can calculate the component admittance matrix of every component of the
Developing a harmonic power flow software in distributed generation systems

network, for each frequency. All those matrices are in the abc component system.

In this part, we will explain how we can put together those component admittance matrices in order to create a system admittance matrix that will be used in the algorithms. The idea is to create, for each frequency a unique system admittance matrix that describe how the nodes, as they are introduced in 2.2.1, are connected together. This matrix takes into account the three-phase components.

For instance, for a single component $C_1$ that is between the node $k$ and the node $l$, the component admittance matrices corresponding to the component are:

$$Y^h_{c_1} = \begin{bmatrix} y^h_{kk,c_1} & y^h_{kl,c_1} \\ y^h_{lk,c_1} & y^h_{ll,c_1} \end{bmatrix}$$

with $h \in \{1...h_{max}\}$: harmonic rank

For a three-phase component $C_2$ that is between the nodes $p,q,r$ and the nodes $s,t,u$, the component admittance matrices are:

$$Y^h_{c_2} = \begin{bmatrix} y^h_{pp,c_1} & y^h_{pq,c_1} & y^h_{pr,c_1} & y^h_{ps,c_1} & y^h_{pt,c_1} & y^h_{pu,c_1} \\ y^h_{qp,c_1} & y^h_{qq,c_1} & y^h_{qr,c_1} & y^h_{qs,c_1} & y^h_{qt,c_1} & y^h_{qu,c_1} \\ y^h_{qp,c_1} & y^h_{qp,c_1} & y^h_{rr,c_1} & y^h_{rs,c_1} & y^h_{rt,c_1} & y^h_{ru,c_1} \\ y^h_{sp,c_1} & y^h_{sp,c_1} & y^h_{sr,c_1} & y^h_{ss,c_1} & y^h_{ss,c_1} & y^h_{su,c_1} \\ y^h_{sp,c_1} & y^h_{sp,c_1} & y^h_{sr,c_1} & y^h_{ss,c_1} & y^h_{ss,c_1} & y^h_{su,c_1} \\ y^h_{sp,c_1} & y^h_{sp,c_1} & y^h_{sr,c_1} & y^h_{ss,c_1} & y^h_{ss,c_1} & y^h_{su,c_1} \end{bmatrix}$$

with $h \in \{1...h_{max}\}$: harmonic rank

Thus, for a network with $n$ nodes, we are able to create the system admittance matrix, $Y^h_{bus}$, at each harmonic $h$, in abc component system:

$$Y^h_{bus} = \begin{bmatrix} Y^h_{11} & ... & Y^h_{1i} & ... & Y^h_{1j} & ... & Y^h_{1n} \\ ... & ... & ... & ... & ... & ... & ... \\ Y^h_{i1} & ... & Y^h_{ii} & ... & Y^h_{ij} & ... & Y^h_{in} \\ ... & ... & ... & ... & ... & ... & ... \\ Y^h_{j1} & ... & Y^h_{ji} & ... & Y^h_{jj} & ... & Y^h_{jn} \\ ... & ... & ... & ... & ... & ... & ... \\ Y^h_{n1} & ... & Y^h_{ni} & ... & Y^h_{nj} & ... & Y^h_{nn} \end{bmatrix}$$

where: $- Y^h_{ij} = \sum_{C \in E_{ij}} Y^h_{ij,c}$

$- E_{ij}$ : Set of component connected between node $i$ and $j$. 

Accessibilité : Page 26 sur 58
6. Algorithms review

6.1. Fundamental Load Flow algorithms

6.1.1. Gauss-Seidel

6.1.1.1. Method description

\[ n: \text{number of nodes in the system} \]
\[ Y = B + jG: \text{admittance matrix of the system} \]
\[ I_i^\delta: \text{injected current at node } i \text{ calculated at the } \delta^{th} \text{ iteration} \]
\[ V_i^\delta = |V_i^\delta|e^{j\theta_i^\delta}: \text{voltage at node } i \text{ compared with simulation reference potential calculated at the } \delta^{th} \text{ iteration} \]
\[ \theta_i: \text{Angle between the voltage at node } i \text{ and the angular simulation reference} \]

In this method (cf. [13], [14],[24]), we iteratively calculate the value of voltage at each node.

Method:
First we set values \( V^\delta=0 \) of voltages. We can know the solution of \( I = Y.V \) by calculating iteratively

\[ V_i^{\delta+1} = \frac{1}{Y_{ii}} \left( I_i^{\delta} - \sum_{j=1}^{n-1} Y_{ij} V_{j}^{\delta+1} - \sum_{j=i+1}^{n} Y_{ij} V_{j}^{\delta} \right), i \in \{1..n\} \]

Where currents are calculated with the following formula: 
\[ I_i^\delta = \frac{P_i - jQ_i}{|V_i^\delta|^2} \]

When \( Q_i \) is unknown (for a PV load), we use the formula:
\[ Q_i = |V_i| \sum_{j=1}^{n} |V_j| \left( G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j) \right) \]

6.1.1.2. Advantages and Drawbacks

The main advantage of this method is that the calculations are simple. This method requires lots of iterations in order to obtain a result. Thus, the execution time is rather high. Furthermore, the convergence of the method is quite slow.

6.1.2. Newton-Raphson

6.1.2.1. Method description

\( P_i: \) Calculated injected active power at node \( i \), from the voltages
\( Q_i: \) Calculated injected reactive power at node \( i \), from the voltages
Developing a harmonic power flow software in distributed generation systems

\( P_{GDi} \): Net generation of active power at node i

\( Q_{GDi} \): Net generation of reactive power at node i

In this method (cf. [2], [24]), we try to obtain the solution \( x_0 \) of a system: 
\[
g(x) = f(x) - b = 0
\]

We iteratively calculate the approximated value \( x^\delta \) of \( x_0 \) by calculating
\[
\Delta x^\delta = \left[ \frac{\partial g(x)}{\partial x} \right]_{x=x^\delta}^{-1} g(x^\delta).
\]

The difference between calculated powers and the setpoint powers represent the system and are the followings:
\[
\begin{align*}
P_i &= V_i \left[ \sum_{j=1}^{n} |V_j| \left( G_{ij} \cos(\theta_i - \theta_j) - B_{ij} \sin(\theta_i - \theta_j) \right) \right] \\
Q_i &= V_i \left[ \sum_{j=1}^{n} |V_j| \left( G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j) \right) \right]
\end{align*}
\]

**Method:**

First we set values \( \theta \) of voltages. We set \( n \): number of loads. The unique \( V \) bus (main source) is conventionally fixed as the first node. Then we execute the following loop:

1) For each PQ load, we calculate \( \Delta P_i = P_i - P_{GDi} \) and \( \Delta Q_i = Q_i - Q_{GDi} \) with \( i \in \{2..m+1\} \), \( m \): number of PQ loads.

For each PV load, we calculate \( \Delta P_i = P_i - P_{GDi} \) with \( i \in \{m+2..m+p+1\} \), \( p \): number of PV loads.

We have: \( n = m + p + 1 \)

2) We calculate the jacobian:

\[
JAC = \begin{bmatrix}
\frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial Q} \\
\frac{\partial P}{\partial V} & \frac{\partial P}{\partial \theta}
\end{bmatrix}\begin{bmatrix}
\frac{\partial P}{\partial V} & \frac{\partial P}{\partial \theta} \\
\frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial \theta}
\end{bmatrix} \ldots \begin{bmatrix}
\frac{\partial P}{\partial V} & \frac{\partial P}{\partial \theta} \\
\frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial \theta}
\end{bmatrix} \ldots \begin{bmatrix}
\frac{\partial P}{\partial V} & \frac{\partial P}{\partial \theta} \\
\frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial \theta}
\end{bmatrix} \ldots \begin{bmatrix}
\frac{\partial P}{\partial V} & \frac{\partial P}{\partial \theta} \\
\frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial \theta}
\end{bmatrix}
\]

The size is \((n+m-1)\times(n+m-1)\).
3) We update

\[ X^{\delta+1} = X^{\delta} + [JAC]^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \theta_2 \\ \vdots \\ \theta_n \\ V_2 \\ \vdots \\ V_{m+1} \end{bmatrix} + [JAC]^{-1} \begin{bmatrix} \Delta P_n \\ \Delta Q_n \end{bmatrix} \]

We execute the loop until \( \Delta P_i \) and \( \Delta Q_i \) are small enough.

6.1.2.2. Advantages and Drawbacks

This method has a good convergence and the execution time is rather low for many different systems. Thus, it is particularly adapted for a system that has a topology which has no specific characteristic (cf. [2]).

This method can be used with almost every networks but it may be more efficient to use specific methods that are more adapted to the systems we want to study.

6.1.3. Fast Decoupled Load Flow

6.1.3.1. Method description

This method is a modification of the Newton-Raphson’s method (cf. [2], [4], [15], [16]).

1) First, we neglect \( \frac{\partial P}{\partial |V|} \) and \( \frac{\partial Q}{\partial \theta} \). Thus, the system we have to solve becomes:

\[
\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \theta} & 0 \\ 0 & \frac{\partial Q}{\partial |V|} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta |V| \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta |V| \end{bmatrix}
\]

The system can be separated into two distinct systems:

\[
\begin{cases}
\Delta P = H \Delta \theta \\
\Delta Q = N \frac{\Delta |V|}{|V|}
\end{cases}
\]

In order to do this simplification, three conditions have to be valid:

- The value of voltages must be close to their nominal value.
- \( \frac{R}{X} \) on the lines must be weak
- Phases shift between two nodes with a line in between must be close to 0.

Thus, by calculating the values of elements of \( N \) and \( H \), we obtain the following system:
Developing a harmonic power flow software in distributed generation systems

\[
\begin{align*}
\left[ \Delta P \right] &= \left[ V.B'.V \right] \left[ \Delta \theta \right] \\
\left[ \Delta Q \right] &= \left[ V.B''.V \right] \left[ \frac{\Delta |V|}{|V|} \right]
\end{align*}
\]
where at this stage \( B' \) and \( B'' \) are elements of \( \text{Im}(Y_{bus}) \).

2) Then, we make some simplifications:

- We omit in \( B' \) network elements representation that mainly modified MVAR flows.
- We omit in \( B'' \) network elements representation that mainly modified angle.
- In the first equation, influence of MVAR flows on calculation of \( \Delta \theta \) is suppressed by setting right-hand \( V \) to 1 p.u.

Thus, the final system is:

\[
\begin{align*}
\left[ \Delta P \right] &= \left[ B' \right] \left[ \Delta \theta \right] \\
\left[ \Delta Q \right] &= \left[ B'' \right] \left[ \frac{\Delta |V|}{|V|} \right]
\end{align*}
\]

6.1.3.2. Advantages and Drawbacks

With this method, the Newton-Raphson’s method calculations are simplified. Indeed, equations about \( \theta \) and equations about \( V \) can be solved successively. Furthermore, \( B' \) and \( B'' \) are constant the calculations are consequently really fast.

The hypothesis that states that \( \frac{R}{X} \) on the lines must be weak is not always true, especially in the distribution network. Thus, this method requires modifications in order to be used with the systems we want to study.

6.1.4. Backward-Forward Sweep

6.1.4.1. Method description

\( B_i^\delta \): Branch current in branch \( i \) (upstream current) calculated at the \( \delta \)-th iteration

Method:

In this method, the network needs to be radial (cf. [6], [7], [25]). The idea is to calculate the currents by going backward into the tree and calculate the voltages by going forward into the tree.
Figure 12 - BFS illustration

First we set values $V_{\delta=0}$ of voltages. Then we execute the following loop:

1) Backward sweep: Calculation of injected current in each node: $I_i^\delta = \left( \frac{P_i - jQ_i}{V_i^\delta} \right)$.

Then we can calculate the branch currents $B_i^\delta$ thanks to the description of the system.

2) Forward sweep: Calculation of voltages thanks to the $Y_{bus}$ matrix, starting from the first node and deducting the voltage drops caused by the branch currents.

3) We update the voltages

We execute the loop until the differences between voltages calculated during two successive iterations are small enough.

There are some variants of this BFS where powers are calculated during the Backward sweep and the voltages are calculated during the Forward sweep.

6.1.4.2. Advantages and Drawbacks

This method is especially suitable for distribution network because they often have a radial structure. Indeed, the calculations are quite simple and an accurate solution can be obtained fast. The convergence is also good.

Furthermore, it is possible to realize parallel processing by analyzing the different feeders separately.

This method requires the network to be radial.

6.1.5. Compensation based load flow

6.1.5.1. Method description

This method is similar to the Backward Forward Sweep, but can also be used if there are cycles on the network (cf. [5]).

First, we have to choose the breakpoints that make the network not radial. The current at breakpoints is $J_j$ where $j \in 1..m$ and $m$ is the number of breakpoints. In order to be able to execute BFS, those breakpoints are modeled by 2 separate points at which is injected $J_j$ and $-J_j$.

![Diagram of Breakpoint Representation]

Figure 13 – Breakpoint representation

Now, we want to calculate $Z$ defined by $V_b = Z.J$ where $V_b$ is the vector of voltage differences between the 2 separate points that represent the breakpoint and $J$ the vector of current at
breakpoints.

**Calculation of \( Z \):**
For a specific \( j \), we set \( J_j = 1 \) and \( J_i = 0 \), \( \forall i \neq j \). Thus, by running a BFS, we find a value of \( V_b \).

\[
egin{bmatrix}
0 \\
\vdots \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix} = \begin{bmatrix} Z_{ij} \\ \vdots \\ Z_{nj} \end{bmatrix}
\]

We know that \( V_b = Z_{J_j} \). Thus, we have: \( V_b = \begin{bmatrix} Z_{ij} \\ \vdots \\ Z_{nj} \end{bmatrix} \). By executing this process for every \( j \), we find the value of \( Z \).

**Method:**
First we set \( J^d = 0 \) and calculate \( Z \). Then we execute the following loop:

1) BFS
2) Calculation of \( V_{b..j} \), for each breakpoint \( j \).
3) Calculation of currents that would be necessary to inject at each breakpoint in order to have \( V_b \) equal to \( 0 \): \( \Delta J = Z^{-1} V_b \)
4) We update \( J = J + \Delta J \)

We execute the loop until \( V_b \) is small enough.

### 6.1.5.2. Advantages and Drawbacks

With this method, it is possible to solve the load flow even if the network is not purely radial.

The execution time become very high if the network is too densely meshed.

### 6.1.6. Algorithms comparison

The main characteristics of load flow methods are presented in the following table:

<table>
<thead>
<tr>
<th>Method</th>
<th>Constraints, Hypothesis</th>
<th>Advantages</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gauss-Seidel</td>
<td>-</td>
<td>Simple calculation</td>
<td>Bad convergence</td>
</tr>
<tr>
<td>Newton-Raphson</td>
<td>-</td>
<td>No hypothesis on the network’s structure</td>
<td>Heavy calculation</td>
</tr>
<tr>
<td>Fast Decoupled Load Flow</td>
<td>Weak ( \frac{R}{X} ) on the lines</td>
<td>Simplification of Newton-Raphson’s calculation</td>
<td>Hypothesis are often invalid for distribution network</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>V</td>
<td>\approx V_{nominal} )</td>
</tr>
<tr>
<td></td>
<td>( \theta \approx 0 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backward-Forward Sweep</td>
<td>Radial network</td>
<td>Suitable for distribution network</td>
<td>Not for meshed system</td>
</tr>
</tbody>
</table>
6.2. Harmonic Load Flow methods

In this part, we will present some general harmonic load flow methods that can be used. The choice between those methods has to be mainly made by looking at the nonlinear loads models we want to use. We also have to take into account the complexity and the accuracy of the method.

We will not use the positive, negative and zero sequence modeling. The nodes are the physical nodes, described in the abc component system. Furthermore, the calculation of injected current by non linear devices is explained in each method’s description. This calculation is also different if the device is a single phase device or a three phase device.

Some methods that we will present require NLD models that are simpler than the one described in eq A.

### Notation

- $i = 1$: Slack bus: reference bus
- $i = 2..g$: PV bus
- $i = g + 1..c$: PQ bus
- $i = c + 1..n$: Non-Linear Device: NLD
- L: last harmonic we take into account.
- $I_i^k$: Current of harmonic $k$ injected at node $i$, $k = 1,3..L$
- $V^k_i = U^k_i.e^{i\theta}$ voltage at harmonic $k$ at node $i$, $k = 1,3..L$
- $Y^k$: System admittance matrix at harmonic $k$

#### 6.2.1. Harmonic Penetration

##### 6.2.1.1. Method description

In this method, we consider that harmonic voltages have no influence on NLD behavior (cf. [8]). Thus NLD behaviors are defined by, which is a simplification of equation A:

$$I^k_i = g^k(V^1_i), \quad \forall k = 3..L$$

Where the function $g$ depends on the nature of the NLD.

If the NLD is a three-phase load, then $I^k_i$ is a vector: $I^k_i = \begin{bmatrix} I_{i,a}^k \\ I_{i,b}^k \\ I_{i,c}^k \end{bmatrix}$ and $V^1_i$ is also a vector:

$$V^1_i = \begin{bmatrix} V_{i,a}^1 \\ V_{i,b}^1 \\ V_{i,c}^1 \end{bmatrix}$$

<table>
<thead>
<tr>
<th>Compensation-based Load Flow</th>
<th>Weakly meshed system</th>
<th>Suitable for weakly meshed distribution network</th>
<th>Calculation time is very long when the network is highly meshed</th>
</tr>
</thead>
</table>

Table 1 – Fundamental Algorithm comparison
If the NLD is a single-phase load, then $I^k_i$ and $V^1_i$ are complex numbers.

**Method:**
1) First we run a fundamental Load Flow in order to know the fundamental voltages at every nodes of the system.
2) With the equations that describe NLD behaviors, we can calculate harmonic currents for every NLD. Indeed we calculate for each harmonic, for each NLD:
   \[ I^k_i = g^k \left( V^1_i \right). \]
   For the linear loads we calculate injected harmonic currents with the formula in 3.2.3:
   \[ I^k_i = \left( \frac{P^1_i - j Q^1_i}{h \left(V^1_i \right)} \right) \text{ and } Q^1_i = \left| V^1_i \right| \left| \sum_{j=1}^{\infty} \left| V^j_i \right| \left( G^1_{ij} \cdot \sin \left( \theta^1_i - \theta^1_j \right) - B^1_{ij} \cdot \cos \left( \theta^1_i - \theta^1_j \right) \right) \right| \text{ if necessary.} \]
3) Then we can calculate $V^k = \left( V^k \right)^{-1} I^k$, $k = 3..L$.

**6.2.1.2. Advantages and Drawbacks**
In this method, the calculations are simple and the execution time is low.
However, this method does not give accurate results because there is only one iteration. We can notice that it has a global tendency to give higher value of harmonic voltages than the reality (cf. [8], [22]).

**6.2.2. Iterative Harmonic Penetration**

**6.2.2.1. Method description**
In this method, we take into account the influence of harmonics voltages on harmonic injected currents by NLD (cf. [8]). Thus, NLD behavior is represented by, which is a simplification of equation A:
   \[ I^k_i = g^k \left( V^1_i, ..., V^L_i \right), \forall k = 3..L \]
However, we consider that harmonic voltages have no influence on the fundamental voltages at NLD.

If the NLD is a three-phase load, then $I^k_i$ is a vector: $I^k_i = \begin{bmatrix} I^k_{i,a} \\ I^k_{i,b} \\ I^k_{i,c} \end{bmatrix}$ and $V^1_i$ is also a vector:
   \[
   V^k_i = \begin{bmatrix} V^k_{i,a} \\ V^k_{i,b} \\ V^k_{i,c} \end{bmatrix}
   \]

If the NLD is a single-phase load, then $I^k_i$ and $V^k_i$ are complex numbers.

**Method:**
First we run a fundamental Load Flow in order to know the fundamental voltages at every node of the system.
Then, we run the process:
1) We set values $V^{1=0}$ of voltages, for each harmonic.
2) We can calculate the harmonic currents:

\[
(I^k_i)^\delta = g^k \left( (V^1_i)^\delta, ..., (V^k_i)^\delta, ..., (V^L_i)^\delta \right)
\]

3) We solve the system \(Y^h V^h = I^h\) for every harmonic to determine \(V^h\) with different method, for example: - by inverting the matrix \(Y^h\)

- by using a Gauss Seidel method.

This process is run iteratively until \((V^h_i)^\delta\) is close enough to \((V^h_i)^\delta-1\) for every bus, for every harmonic.

6.2.2.2. Advantages and Drawbacks

By taking into account interactions between harmonic voltages, we find more accurate results.
This method uses the Gauss-Seidel method or a direct resolution, which may have convergence issues.

6.2.3. Simplified Harmonic load flow

6.2.3.1. Method description

In this method, we take into account the influence of harmonic voltages on harmonic injected currents by NLD and fundamental voltage (cf. [8], [9]).

\[
(I^k_i)^\delta = g^k \left( (V^1_i)^\delta, ..., (V^k_i)^\delta, ..., (V^L_i)^\delta \right)
\]

If the NLD is a three-phase load, then \(I^k_i\) is a vector: \(I^k_i = \begin{bmatrix} I^k_{i,a} \\ I^k_{i,b} \\ I^k_{i,c} \end{bmatrix}\) and \(V^1_i\) is also a vector:

\[
V^k_i = \begin{bmatrix} V^k_{i,a} \\ V^k_{i,b} \\ V^k_{i,c} \end{bmatrix}
\]

If the NLD is a single-phase load, then \(I^k_i\) and \(V^k_i\) are complex numbers.

Method:

First we set values \(V^{\delta=0}\) of voltages, for the fundamental and each harmonic.

1) We run a fundamental Load Flow in order to know the fundamental voltages at every node of the system.

2) Then we can calculate the harmonic current for each bus \((I^k_i)^\delta = g^k \left( (V^1_i)^\delta, ..., (V^k_i)^\delta, ..., (V^L_i)^\delta \right)\), for each harmonic and for the fundamental.

3) Then we can solve the system \(Y^h V^h = I^h\) for every harmonic and the fundamental with different
method, for example: - by inverting the matrix $Y^h$
- by using a Gauss Seidel method
- by using a Newton-Raphson method

The goal is to get a value of the voltage at every node, for each harmonic and the fundamental.

This process is run iteratively until $(V_i^h)^k$ is close enough to $(V_i^h)^{k-1}$ for every load, for the fundamental and every harmonic.

6.2.3.2. Advantages and Drawbacks

This method gives more accurate results than the previous one.

However, because we solve multiple times the fundamental load flow, the execution time can be high.

6.2.4. Harmonic Newton-Raphson

6.2.4.1. Method description

This method is an extended version of the fundamental Newton-Raphson method.

In this method, the NLD are represented by current injections: $I_i^k = g^k(V_i^L, V_i^1, \beta_i^1, \beta_i^2)$, $i = c+1..n$ where $c$ is the index of the first NLD, where $\beta^1$ and $\beta^2$ are the nonlinear bus control parameters (cf. [2], [10], [11]). For example, for a full-wave bridge rectifier, $\beta^1$ is the firing angle of the semiconductor-controlled rectifiers and $\beta^2$ is the commutating impedance.

If the NLD is a three-phase load, then $I_i^k$ is a vector: $I_i^k = \begin{bmatrix} I_{i,a}^k \\ I_{i,b}^k \\ I_{i,c}^k \end{bmatrix}$ and $V_i^k$ is also a vector:

$$
V_i^k = \begin{bmatrix} V_{i,a}^k \\ V_{i,b}^k \\ V_{i,c}^k \end{bmatrix}
$$

If the NLD is a single-phase load, then $I_i^k$ and $V_i^k$ are complex numbers.

The main difference with previous method is that all harmonics are here treated at the same time.

Here, we want to solve the equations that express the difference between the lines currents and the currents injected by the NLD and the difference between the injected power and the net power production at every node.

We want to calculate successively the values of $\Delta V$ and $\Delta M$ with the following equation:
Developing a harmonic power flow software in distributed generation systems

\[ \Delta V = + [J]^{-1} \Delta M \] where \( V = \begin{bmatrix} V^1 \\ V^3 \\ V^L \end{bmatrix} \) and \( \Phi = \begin{bmatrix} \phi^1_c \\ \phi^2_c \\ \phi^n_c \end{bmatrix} \). The definition of \( J \) is given in the method description below. This equation includes all harmonics.

**Method:**

First we set values \( V^{0,0} \) of voltages. Then we execute the following loop:

1) Calculation of \( I^k_{i, c+1} \) with equations: \( I^k_{i, c+1} = g^k \left( V^1_{i, c+1}, V^L_{i, c+1}, \phi^1_{i, c+1}, \phi^2_{i, c+1} \right) \), for each harmonic.

Calculation of lines currents with the following formulas, for each harmonic:

\[
\begin{align*}
I^{h}_{\text{Line}, i} &= \left\{ \begin{array}{l}
\text{Re}(I^{h}_{\text{Line}, i}) = \sum_{j=1}^{n} V^j_{i} \left( |B^h_{i, j} \cos(\theta^h_{j}) - G_{i, j} \sin(\theta^h_{j})| \right) \\
\text{Im}(I^{h}_{\text{Line}, i}) = \sum_{j=1}^{n} V^j_{i} \left( |B^h_{i, j} \sin(\theta^h_{j}) + G_{i, j} \cos(\theta^h_{j})| \right)
\end{array} \right. \quad i = 1..n - 1
\end{align*}
\]

2) Calculation of power mismatch: \( \Delta W = \begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \\ \vdots \\ \Delta P_{c} \\ \Delta Q_{c} \\ \Delta P_{c+1} \\ \Delta Q_{c+1} \\ \vdots \\ \Delta P_{n} \\ \Delta Q_{n} \end{bmatrix} \). This calculation is similar to the one we used in the fundamental Newton-Raphson method when summing all the harmonics.

Calculation of \( \Delta I^1 = \begin{bmatrix} \text{Re}(I^{1}_{\text{Line}, c+1} - I^{1}_{c+1}) \\ \text{Im}(I^{1}_{\text{Line}, c+1} - I^{1}_{c+1}) \\ \vdots \\ \text{Re}(I^{1}_{\text{Line}, n} - I^{1}_{n}) \\ \text{Im}(I^{1}_{\text{Line}, n} - I^{1}_{n}) \end{bmatrix} \) and \( \Delta I^h = \begin{bmatrix} \text{Re}(I^{h}_{\text{Line}, c+1} - I^{h}_{c+1}) \\ \text{Im}(I^{h}_{\text{Line}, c+1} - I^{h}_{c+1}) \\ \vdots \\ \text{Re}(I^{h}_{\text{Line}, n} - I^{h}_{n}) \\ \text{Im}(I^{h}_{\text{Line}, n} - I^{h}_{n}) \end{bmatrix} \).
Developing a harmonic power flow software in distributed generation systems

Calculation of \( \Delta M \):

\[
\begin{bmatrix}
\Delta W \\
\Delta I^3 \\
\Delta I^5 \\
\vdots \\
\Delta I^L \\
\Delta I^1
\end{bmatrix}
\]

3) We set:

- \( J^h = \begin{bmatrix} 0_{2(m-2)\times 2n} & D \end{bmatrix} \): matrix of partial derivatives of the \( h \)-th harmonic mismatch \( P \) and \( Q \) at NLDs.

with respect to \( \left| V^h \right| \) and \( \theta^h : D^h = \begin{bmatrix} \frac{\partial P}{\partial \theta^h} & \frac{\partial P}{\partial \left| V^h \right|} \\ \frac{\partial Q}{\partial \theta^h} & \frac{\partial Q}{\partial \left| V^h \right|} \end{bmatrix} \)

\[
-H^h = \begin{bmatrix}
\frac{\partial \text{Re}(I_{c+1}^h)}{\partial \beta_{c+1}^1} & \frac{\partial \text{Re}(I_{c+1}^h)}{\partial \beta_{c+1}^2} & \cdots & 0 & 0 \\
\frac{\partial \text{Im}(I_{c+1}^h)}{\partial \beta_{c+1}^1} & \frac{\partial \text{Im}(I_{c+1}^h)}{\partial \beta_{c+1}^2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & \frac{\partial \text{Re}(I_n^h)}{\partial \beta_n^1} & \frac{\partial \text{Re}(I_n^h)}{\partial \beta_n^2} \\
0 & 0 & \cdots & \frac{\partial \text{Im}(I_n^h)}{\partial \beta_n^1} & \frac{\partial \text{Im}(I_n^h)}{\partial \beta_n^2}
\end{bmatrix}
\]

- \( YG^{h,j} = \begin{bmatrix} Y^{h,j} & G^{h,j} \end{bmatrix} \) with \( G^{h,j} = \begin{bmatrix} \begin{bmatrix} \partial \text{Re}(I_{c+1}^h) & \partial \text{Re}(I_{c+1}^h) & \cdots & 0 & 0 \\ \partial \theta_{c+1}^j & \partial \left| V_{c+1}^j \right| & \cdots & 0 & 0 \\ \partial \text{Im}(I_{c+1}^h) & \partial \text{Im}(I_{c+1}^h) & \cdots & 0 & 0 \\ \partial \theta_{c+1}^j & \partial \left| V_{c+1}^j \right| & \cdots & 0 & 0 \\
0 & 0 & \cdots & \frac{\partial \text{Re}(I_n^h)}{\partial \theta_n^j} & \frac{\partial \text{Re}(I_n^h)}{\partial \left| V_n^j \right|} \\
0 & 0 & \cdots & \frac{\partial \text{Im}(I_n^h)}{\partial \theta_n^j} & \frac{\partial \text{Im}(I_n^h)}{\partial \left| V_n^j \right|}
\end{bmatrix} \end{bmatrix} \)

and
Developing a harmonic power flow software in distributed generation systems

\[
Y^{h,h} = \begin{bmatrix}
\frac{\partial \text{Re}(I_{\text{Line}_1}^h)}{\partial I^h_1} & \frac{\partial \text{Re}(I_{\text{Line}_1}^h)}{\partial I^h_2} & \cdots & \frac{\partial \text{Re}(I_{\text{Line}_1}^h)}{\partial I^h_n} \\
\frac{\partial \text{Re}(I_{\text{Line}_2}^h)}{\partial I^h_1} & \frac{\partial \text{Re}(I_{\text{Line}_2}^h)}{\partial I^h_2} & \cdots & \frac{\partial \text{Re}(I_{\text{Line}_2}^h)}{\partial I^h_n} \\
\frac{\partial \text{Im}(I_{\text{Line}_1}^h)}{\partial I^h_1} & \frac{\partial \text{Im}(I_{\text{Line}_1}^h)}{\partial I^h_2} & \cdots & \frac{\partial \text{Im}(I_{\text{Line}_1}^h)}{\partial I^h_n} \\
\frac{\partial \text{Im}(I_{\text{Line}_2}^h)}{\partial I^h_1} & \frac{\partial \text{Im}(I_{\text{Line}_2}^h)}{\partial I^h_2} & \cdots & \frac{\partial \text{Im}(I_{\text{Line}_2}^h)}{\partial I^h_n}
\end{bmatrix}
\]

Thus, we can calculate \( J = \begin{bmatrix}
J^1 & J^3 & \cdots & J^L & 0 \\
YG^{3,1} & YG^{3,3} & \cdots & YG^{3,L} & H^3 \\
YG^{5,1} & YG^{5,3} & \cdots & YG^{5,L} & H^5 \\
YG^{L,1} & YG^{L,3} & \cdots & YG^{L,L} & H^L \\
YG^{1,1} & YG^{1,3} & \cdots & YG^{1,L} & H^1
\end{bmatrix}\)

4) We update: \( V^{s+1} = V^s + [J]^{-1} \Delta M \) where \( V = \begin{bmatrix}
V^1 \\
V^3 \\
\vdots \\
V^L \\
\Phi
\end{bmatrix} \) and \( \Phi = \begin{bmatrix}
\beta^1_{s+1} \\
\beta^2_{s+1} \\
\vdots \\
\beta^1_n \\
\beta^2_n
\end{bmatrix} \)

We execute this loop until \( \Delta M \) is small enough.

6.2.4.2. Advantages and Drawbacks

This method takes into account every possible influence of harmonic voltages on the non-linear loads’ behaviour. Thus, the results are more accurate.

However we have to inverse and realize calculations on large scale matrix. Thus, the calculation time is very high.

6.2.5. Harmonic Backward Forward Sweep

6.2.5.1. Method description

This method is similar to the BFS method. A BFS is used for every harmonic and the harmonic injected currents are calculated for every non-linear load (cf. [2]).

\([B_i^h]^{y} \) : Branch current in branch \( i \) (upstream current) at harmonic \( h \) calculated at the \( \delta^y \) iteration.
Developing a harmonic power flow software in distributed generation systems

\[(V_i)^{\delta} = \begin{bmatrix} \ldots \\ (V_i^k)^{\delta} \\ \ldots \end{bmatrix} \]

If there are three phases in the branch, then \( B_i^h \) is a vector: 
\[ B_i^h = \begin{bmatrix} B_{i,a}^h \\ B_{i,b}^h \\ B_{i,c}^h \end{bmatrix}. \]

If the NLD is a single-phase load, then \( B_i^h \) is a complex number.

**Method:**

In this method, the network needs to be radial (cf. [6], [7]). The idea is to calculate the currents by going backward into the tree and calculate the voltages by going forward into the tree.

First we set values \( V_i^{\theta=0} \) of voltages at each node \( i \). Then we execute the following loop:

1. For each \( h \):
   1.1. Backward sweep: from \((V_i)^{\delta}\), we calculate the injected current in each node.
   - For the linear loads, we calculate harmonic currents with the formula in 3.2.3:
     \[ I_i^h = \frac{(P_i^h - jQ_i^h)}{h(V_i^1)} \]
     and 
     \[ Q_i^h = V_i^1 \sum_{j=1}^{n} V_j^1 \left( G_{ij}^1 \sin(\theta_i^1 - \theta_j^1) - B_{ij}^1 \cos(\theta_i^1 - \theta_j^1) \right) \] if necessary.
   - For the non-linear loads, we calculate the currents at each bus with the following formula:
     \[ (I_i^k)^{\delta} = g^k \left((V_i^1)^{\delta}, \ldots, (V_i^k)^{\delta}, \ldots, (V_i^l)^{\delta}\right) \]

     If the non-linear load is a three-phase load, then \( I_i^k \) is a vector: 
     \[ I_i^k = \begin{bmatrix} I_{i,a}^k \\ I_{i,b}^k \\ I_{i,c}^k \end{bmatrix} \]
     and \( V_i^k \) is also a vector:
     \[ V_i^k = \begin{bmatrix} V_{i,a}^k \\ V_{i,b}^k \\ V_{i,c}^k \end{bmatrix} \]

     If the non-linear load is a single-phase load, then \( I_i^k \) and \( V_i^k \) are complex numbers.

   1.2. Then we can calculate the branch currents \((B_i^h)^{\delta}\) thanks to the description of the system, for each node.

   1.3. Forward sweep: Calculation of voltages \((V_i^1)^{\delta}\) thanks to the \( Y_{bus}^h \) matrix, starting from the first node and deducting the voltage drops caused by the branch currents.

2. We update the voltages \((V_i)^{\delta}\).
This process is run iteratively until \( (V_{i}^{h})^{\delta} \) is close enough to \( (V_{i}^{h})^{\delta-1} \) for every load, for the fundamental and every harmonic.

### 6.2.5.2. Advantages and Drawbacks

This method is particularly adapted to distribution network because they often have a radial structure. Indeed, the calculations are quite simple and an accurate solution can be obtained fast. The convergence is also good.

Furthermore, it is possible to realize parallel processing by analyzing the different feeder separately. However, the network must be radial.

### 6.2.6. Algorithms comparison

The main characteristics of harmonic load flow methods are presented in the following table:

<table>
<thead>
<tr>
<th>Method</th>
<th>Constraints, Hypothesis</th>
<th>Advantages</th>
<th>Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harmonic Penetration</td>
<td>Harmonic voltages have no influence on NLD’s behavior</td>
<td>Simple calculation</td>
<td>Non exact method</td>
</tr>
<tr>
<td>Iterative Harmonic</td>
<td>Harmonic voltages have influence on NLD’s current injection</td>
<td>Exact results</td>
<td>Do not take into account all the influences</td>
</tr>
<tr>
<td>Penetration</td>
<td>Harmonic voltages have influence on NLD’s current injection and fundamental voltage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simplified Harmonic</td>
<td>Harmonic voltages have influence on NLD’s current injection and fundamental voltage</td>
<td>More accurate results</td>
<td>Heavy calculation. Need a network reduction, which may be complex to find</td>
</tr>
<tr>
<td>Load Flow</td>
<td>Need knowledge of derivative of the function that associates I and V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harmonic Newton-Raphson</td>
<td>All the interactions are taken into account</td>
<td>More accurate results</td>
<td>Very heavy and complex calculation</td>
</tr>
<tr>
<td>Harmonic BFS</td>
<td>Radial network</td>
<td>Suitable for distribution network</td>
<td>Not for meshed system</td>
</tr>
</tbody>
</table>

Table 2 - Harmonic algorithm comparison

### 6.3. Choice of algorithms

**Choice of Fundamental Algorithm:**

**Gauss-Seidel:**

The first fundamental algorithm that I implemented is the Gauss-Seidel algorithm. This algorithm is
easy to implement and help us to validate the software. Furthermore, the structure of this algorithm is useful in the implementation of harmonic load flow.

Newton-Raphson:

Then, we implemented the Newton-Raphson algorithm. This algorithm is global and should converge with many networks.

Backward-Forward Sweep:

I will not implement this algorithm. Indeed, its parallel implementation will require high skill in programming. However, this method will be implemented by the next person that will work on this software.

In order to use this method, we need to be able to identify the radial part of the system, which are the part where there is no cycle in it. The description of those steps is found in Annex B.

In this method, we will use a Backward Forward Sweep to treat every feeder separately and a Newton-Raphson algorithm to treat the meshed part of the network.

The flow chart of this method is:
Developing a harmonic power flow software in distributed generation systems

Figure 14 – BFS Flowchart

Treat Switch and number the nodes

Find nodes that belongs to a cycle

Find nodes that belongs to each radial part

Eliminate small radial parts

Compute whole system $Y_{bus}$

Calculate $I$, for each nodes

For each radial part

Calculate $V$ for each node with the BFS algorithm

Calculate $V$ for each node that belongs to a cycle with a Newton-Raphson algorithm

If $\Delta V < \xi$

Yes

End

No
Choice of Harmonic Algorithm:

Iterative harmonic penetration load flow with $Y^h$ inversion (cf. 6.2.2):

The first harmonic load flow method that we implemented is the iterative harmonic penetration where we solve harmonic voltages by simply inverting the Ybus matrix at the corresponding frequency. Indeed, we need to have a simple harmonic method in order to test our software. However, this method does not converge in some network configuration. Thus, if there is no convergence after a definite number of iterations, we should use a more complex method.

Here is a flow chart of this method.
Figure 15 – Iterative Harmonic Load Flow Algorithm

1. Treat Switch and number the nodes.
2. For \( h \) from 1 to \( h_{\text{max}} \):
   - Compute \( Y_{\text{bus}}^h \), for each harmonic.
   - Run Fundamental Load flow.
   - Guess \( V^{\delta=0} \) for harmonics. Calculate \( I^h \).
   - For \( h \) from 1 to \( h_{\text{max}} \):
     - If \( h > h_{\text{max}} \), Yes:
       - Solve \( V^h = (Y^h)^{-1} \cdot I^h \).
     - Update \( I^h \) for each harmonic.
   - If \( \Delta V^h < \xi, \forall h \)
     - Yes:
     - No:
       - End.
Iterative Harmonic Load Flow with Gauss-Seidel (cf. 6.2.2):

The second method that will be used is the iterative harmonic load flow with Gauss-Seidel. This method will be implemented by the next person that will work on this software.

The flowchart is similar to the preceding one, but instead of solving $V^h = (Y^h)^{-1}J^h$ for each harmonic, the system $J^h = Y^h V^h$ is solved by using a Gauss-Seidel algorithm for each harmonic.

Harmonic Backward Forward Sweep (cf. 6.2.5):

As the fundamental backward forward sweep, I will not implement this algorithm. However, this method will be implemented by the next person that will work on this software. As the fundamental algorithm, it will combine radial treatments and a meshed part treatment.

Harmonic Newton-Raphson (cf. 6.2.4):

Ultimately we will use the Harmonic Newton-Raphson method if we have not found results with the two preceding methods. This method may be implemented by the next person that will work on this software, if it is needed.

7. Architecture, Software engineering

In the preceding parts, we described the needs and the requirements of the software, and we realized a theoretical study in order to determine the algorithm we wanted to implement.

Now we will describe the main elements of the software architecture.

7.1. Specifications

In this part, we will describe the functionalities that will be part of the final software, thanks to a use case diagram. With this diagram and its explanations, one should be able to understand the main actions the user shall be able to do and the how those actions are done.

First we must identify the users, also called actors:

Actors:

There are two different actors as regards the use cases: the user and the external module that realizes time domain simulations. The user can be OpenTURNS or the human user because they have the same relation with the software. The external module is used during the harmonic load flow. This actor just answers to request made by the system.
Use Case Description:

**Load a Network:**

**Purpose:** The user asks the system to load a new network described in an input file.

**Process:**
- Check for the input file’s validity
- For each read component:
  - Identify the generic component
  - Record the component in the symbolic representation

**Create an intermediary file:**

**Purpose:** When a new network is loaded, a Complete Intermediary File describing the same network is created. Thus, openTURNS would be able to read it.

**Process:**
- Create the file.
- For each read component:
  - Write in the file the component Id, its type, its parameters, the nodes it is connected.
Create the numerical representation:

Purpose: When the user wants to calculate a load flow, a numerical representation of the network is created in order to be able to make the calculations.

Process:
- Creation of the fundamental numerical representation
  - For each junction element of the symbolic representation
    - Find in the components library the corresponding model.
    - Calculate the value of the model's parameters with the components parameters.
    - Add this model into the global fundamental numerical representation.

- Creation of the harmonic numerical representation for each frequency
  - For each junction element of the symbolic representation
    - Find in the components library the corresponding model.
    - Calculate the value of the model's parameters with the components parameters and the frequency.
    - Add this model into the global numerical representation corresponding to this frequency.

Modify components' parameters:

Purpose: When the user wants to modify the network description, the software has to read the modification file and modify the symbolic representation.

Process:
- Read the modification file: for each read component:
  - Modify the value of components parameters.

Undo modifications:

Purpose: When the user wants to undo the modifications, the software must get back to the initial values of parameters.

Process:
- For each modified component:
  - Get back to the components parameters initial values.

Calculate a fundamental Load Flow:

Purpose: The software must be able to realize a fundamental load flow with a chosen algorithm.

Process:
- If the algorithm has been changed, or junction components have been modified, or the network has just been loaded: creation of the numerical representation.
- Calculation of the load flow.
Calculate a harmonic load flow:

**Purpose:** The software must be able to realize a harmonic power flow with a chosen algorithm.

**Process:**
- If the algorithm has been changed, or junction components have been modified, or the network has just been loaded: creation of the numerical representation.
- Calculation of the load flow.
  - If it is needed, get a non linear load response with the external module

Create an output file:

**Purpose:** At the end of a load flow calculation, the software must be able to write the results in an output file.

**Process:**
- If the file does not exists:
  - Create the file
  - Write the header
- Write a line in the file for each component we want to observe.

Get a non linear load response:

**Purpose:** During a harmonic load flow, the software must be able to obtain a non linear load response

**Process:**
- Send a simplified version of the network for the non linear load.
- Send the harmonic voltages at that load.
- Obtain the harmonic currents in the load

### 7.2. Software architecture

In this part, we present the main packages of the software that has been created. Thus, one can know how the software is mainly made.

- The Files package deals with Input and Output.
- The Library package contains the models of the network components.
- The Client-Server Interface deals with interactions with the users.
- The Calculator contains the functions related to the internal network representation and load flow calculations.

In the following figure, we can how those packages are connected together. The detailed description of elements that belongs to each package is described in ANNEX C.
8. Results and comparison

We made tests on a specific network in order to validate the implemented fundamental methods. The network that we will use is a simple single-phase one:

![Test network diagram]

**Figure 18 – Test network**

With this network, we validated the Gauss-Seidel method and the Newton-Raphson method.

9. Conclusion

This document presents the researches and development I have done in order to develop a harmonic power flow software in distributed generation systems.

During this internship, I analyzed the different load flow and harmonic load flow algorithms that are presented in the literature and compared them together in order to determine the methods that should be implemented in the future software. Then I wrote the specifications and created a modular architecture of the software, which describes its implementation.

The implementation of this first version was a necessary step in order to develop the new
software. Indeed, we cannot overlook the fact that the conception of software is an iterative process which cannot be seen as linear one. At the beginning of this internship, specifications were not complete. Thus, I had to begin the development in order to really understand the issues and the constraints of this project, and improved the initial specifications.

The main contribution of this internship is the precise expression of the needs and goals of the software, and an implementation of its structure.

There will be future studies concerning this project. Indeed, the development stage will be done in a more global way with the assistance of contractors and computer science specialists from EDF R&D.

A synthesis of different algorithm will be done in order to get a software that automatically adapts its algorithm to the given network and the results of first iterations instead of choosing a specific algorithm at the beginning of calculations. Furthermore, studies will be carried out about parallelization of algorithms and software optimization, in order to have a software as efficient and fast as possible.

10. Reference


Developing a harmonic power flow software in distributed generation systems


[18] *ExperTEC – Analyse Fréquentielle de Réseau Electrique Multi-Phase - Partie 1 : Description Générale du Logiciel*, Internal report within EDF R&D, 2005 (in French)

[19] *ExperTEC – Analyse Fréquentielle de Réseau Electrique Multi-Phase - Partie 2 : Description des Composants*, Internal report within EDF R&D, 2005 (in French)


**ANNEX A : INPUT AND OUTPUT FORMAT EXAMPLES**

**CSV Format**

In a single phase network, the user wants to observe fundamental and third harmonic voltages for a component C1 and fundamental and fifth harmonic voltages for a component C2.

The file is:

```
Grid_Example.ASC ; fundamental_algorithm = Newton-Raphson ; harmonic_algorithm = Harmonic Newton-Raphson ; software v1.0 ; CSV format ;
```
Developing a harmonic power flow software in distributed generation systems

Simple CSV Format

The file is:

Grid_Example.ASC ; fundamental_algorithm = Newton-Raphson ; harmonic_algorithm = Harmonic Newton-Raphson ; software v1.0 ; CSV format ;

Simulation 1:

C1 ; \( V_{\text{fund}} = 210 \); \( \theta_{\text{fund}} = 0 \); \( V^3 = 2 \); \( \theta^3 = 0.52359 \);

C2 ; \( V_{\text{fund}} = 215 \); \( \theta_{\text{fund}} = 0.12456 \); \( V^5 = 1.3 \); \( \theta^5 = 0.47894 \);

Simulation 2:

ANNEX B: NETWORK SEPARATION AND PARALLELIZATION OF CALCULATIONS

B.1 - Cycles detection Algorithm

Description:

This algorithm allows us to know the nodes that are part of a cycle in a non-oriented graph. It uses a Breadth-First Search, in which we give to every node a level representing its distance from the initial node. The nodes are put in a FIFO and are read in the increasing level order. (cf. [12])

Algorithm:

\( E_{\text{nodes}} \) : Set of nodes
\( E_{\text{cycles}} \) : Set of nodes in the nodal part
\( E_{\text{final}} \) : Set of nodes at the end of radial part

\( \text{Neighbourhood}(n) \)
\( \text{Level}(n) \)

\begin{verbatim}
begin Find _cycles
Create Fifo
end
\end{verbatim}
Developing a harmonic power flow software in distributed generation systems

for $n \in E_{nodes}$ do
    Level($n$) = null;
end for

add Source to Fifo;

while Fifo Is_not_empty do
    get Current_Node from Fifo;
    remove Current_Node from Fifo;
    if $\text{Card}(\text{Neighbourhood}(\text{Current}_{\text{Node}})) = 1$ then
        Current_Node $\in E_{final}$;
    else
        for $n \in \text{Neighbourhood}(\text{Current}_{\text{Node}})$ do
            if Level($n$) = null then
                Level($n$) = Level(Current_Node) + 1;
                add $n$ to Fifo;
            end if
            if Level($n$) < Level(Current_Node) then
                null;
            end if
        end for
        if Level($n$) $\geq$ Level(Current_Node) then
            if Current_Node $\in E_{cycles}$ then
                add Current_Node to $E_{cycles}$;
                for $m \in \text{Fathers}(\text{Current}_{\text{Node}})$ do
                    if $m \in E_{cycles}$ then add $m$ to $E_{cycles}$; end if
                end do;
            end if
            if $n \not\in E_{cycles}$ then
                add $n$ to $E_{cycles}$;
                for $m \in \text{Fathers}(n)$ do
                    if $m \not\in E_{cycles}$ then add $m$ to $E_{cycles}$; end if
                end do;
            end if
        end if
    end if
end while

end Find_cycles

begin Fathers(Current_Node : node)
create List;
for $n \in \text{Neighbourhood}(\text{Current}_{\text{Node}})$ do


if \((Level(n) < Level(Current\_Node))\text{ and}(n \notin List)\) then
  add \(n\) to \(List\);
  \(Fathers(n)\);
end if
end for
return \(List\)
end \(Fathers\)

B.2 - Radial Part creation algorithm

Description:
With this algorithm, we can group together nodes which belong to the same radial part. We start the search at each ending node and go through the parents of this node until we detect a node which belongs to another radial part or to the nodal part.

Algorithm:
\(E_{radial\_i}\) : Set of nodes that belongs to the radial part \(i\)

\(number\_radial\_part\) : number of radial part

For \(n \in E_{final}\) do
  create \(E_{radial\_current}\);
  add \(n\) to \(E_{radial\_current}\);
  \(Current\_node = n\);
  treat\((E_{radial\_current})\);
end do

Begin treat\((E_{radial\_current} : E_{radial})\)
  loop
    \(m = \text{Next.Neighbourhood}(Current\_node)\);
    exit when \(Level(m) < Level(n)\);  
  end loop
  \(Current\_node = m\);
  if \(Current\_node \in E_{cycle}\) then
    \(is\_regular\_node = false\);
  else
    for \(i\) in 1.. \(number\_radial\_part - 1\) do
      if \(Current\_node \in E_{radial\_i}\) then
        \(is\_regular\_node = false\);
        \(merge(E_{radial\_i}, E_{radial\_current})\);
        \(number\_radial\_part = number\_radial\_part - 1\);
      end if
    end for
    if \(is\_regular\_node = true\) then
add Current_node in E_radial_current;

treat(E_radial_current);
end if
end treat;

B.3 - Small radial part suppression algorithm

Description:
With this algorithm, we are able to detect and suppress the radial parts that have a small number of nodes.

Algorithm:

\[
\text{for } i \text{ in } 1.. \text{number_radial_part} \text{ do}
\]
\[
\text{if } \text{Card}(E_{radial,i}) < \text{min_number} \text{ then}
\]
\[
\text{merge}(E_{radial,i}, E_{cycles});
\]
\[
\text{end if}
\]
\[
\text{end do}
\]

ANNEX C : PACKAGE DIAGRAM
Figure 19 – Detailed package diagram

ANNEX D : LIST OF FIGURES AND TABLES
Developing a harmonic power flow software in distributed generation systems

FIGURE 1 - SOFTWARE DESCRIPTION
FIGURE 2 - THREE-PHASE COMPONENT EXAMPLE
FIGURE 3 - COMPONENTS LIBRARY
FIGURE 4 – LINE AND CABLE MODEL
FIGURE 5 – DIFFERENTIAL AND NODAL REPRESENTATIONS OF TRANSFORMER
FIGURE 6 – SINGLE-PHASE TRANSFORMER MODEL
FIGURE 7 – AUTOTRANSFORMER MODEL
FIGURE 8 – LINEAR BUS HARMONIC MODEL
FIGURE 9 - INPUT AND OUTPUT
FIGURE 10 - INTERMEDIARY FILE AND OPENTURNS ACTION
FIGURE 11 - EXAMPLE NETWORK
FIGURE 12 - BFS ILLUSTRATION
FIGURE 13 – BREAKPOINT REPRESENTATION
FIGURE 14 – BFS FLOWCHART
FIGURE 15 – ITERATIVE HARMONIC LOAD FLOW ALGORITHM
FIGURE 16 - USE CASE DIAGRAM
FIGURE 17 - PACKAGE DIAGRAM
FIGURE 18 – TEST NETWORK
FIGURE 19 – DETAILED PACKAGE DIAGRAM

List of figures

TABLE 1 – FUNDAMENTAL ALGORITHM COMPARISON
TABLE 2 - HARMONIC ALGORITHM COMPARISON

List of tables