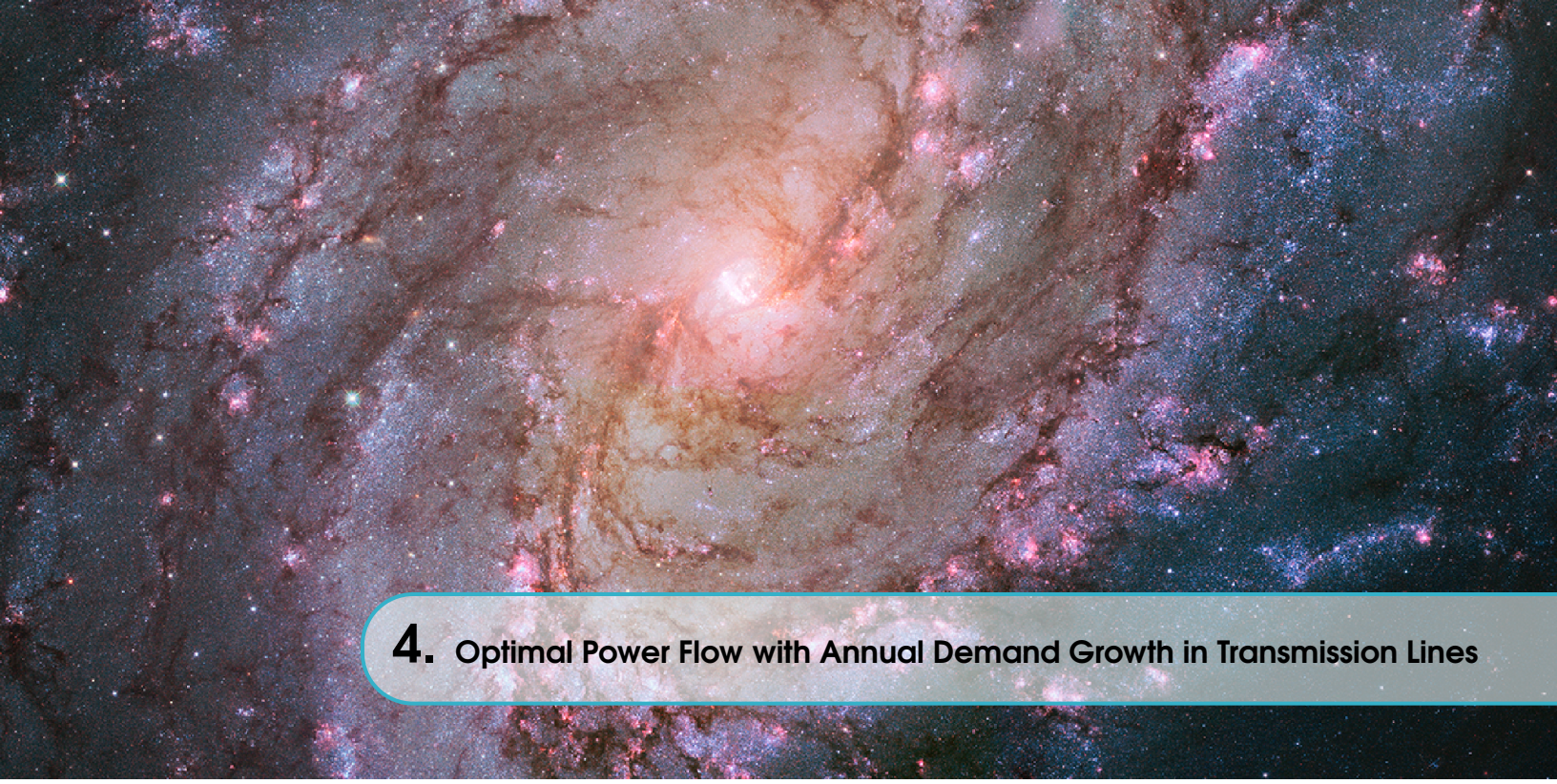

4. Optimal Power Flow with Annual Demand Growth in Transmission Lines

Ph.D. Diego Carrión Galarza is Professor of the Master's Degree Program in Electricity at Universidad Politécnica Salesiana (Ecuador)
Email: dcarrion@ups.edu.ec
https://orcid.org/0000-0002-0704-700X
DOI: 10.17163/abyaups.44.349

4.1 Introduction

Electrical power systems (EPS) are very complex systems due to the large number of elements involved in their operation; they have a large number of generation plants of different technologies, meshed connection systems at high voltage levels, bidirectional distribution systems, and end users with greater demands and iteration with the electricity market. With all this, the network topology of the electrical system is not so simple to analyze, which is why it is necessary to have new strategies for its control (1)–(6).

With the background above, new concepts have been created based on economic aspects, becoming an optimization problem, calling it now an optimal power flow problem (OPF). Since then, several resolution techniques have been developed (7).

Optimization plays an important role when planning the operation and managing to analyze the dynamism of the electricity market; therefore, the search for the optimal solution for the operation and management of energy systems has become a necessity, considering the high costs of different technologies, that energy resources are limited and the growing demand for electricity energy (8).

The objective of the OPF is to find the ideal configuration for optimal operation while minimizing operating costs and maximizing reliability. For an OPF, constraints involving operating costs, system losses, voltage level, transmission line capacity, equipment operating limits, and load demand are considered (9), (10).

According to the selected objective function and constraints, different mathematical formulations exist to solve an OPF. They can be classified as follows:

- Linear problems in which the objectives and constraints are given in linear forms with continuous control variables.
- Nonlinear problems, where the objectives and constraints; or both combined are nonlinear with continuous control variables.
- Nonlinear mixed integer problems with discrete and continuous control variables.

Instead, if analyzed from the point of view of the main attributes, they can be divided as follows:

- Lambda iteration method, in which the losses can be represented by a matrix (B), or the penalty factors can be calculated separately by a power flow. It forms the basis of many standard online economic dispatch programs.
- Gradient method is a slow convergence method and is difficult to solve in the presence of inequality constraints.
- Newton's method, whose convergence is very fast, can give problems with inequality constraints.
- Linear programming method, which is a method that can work with inequality constraints, functions, and nonlinear objective constraints handled by linearization.
- Interior point method: Another fully developed and widely used method for OPF. It easily handles inequality constraints.

In (11), the authors have used the optimal AC power flow model (OPF-AC) to study the behavior with high penetration of renewable energies by comparing the model with others and verifying its accuracy. In contrast, (12) shows the optimal DC power flow model (OPF-DC) applied to the analysis of power systems with the centralized and decentralized generation, which details an approximation of the EPS behavior. The OPF-AC problem for OPF calculation based on the gradient and Newton methods consists of finding the active and reactive power values, the voltage magnitudes of any generator unit to minimize the operating cost, and the fulfillment of various constraints. While OPF-DC is applied to calculate the load flow in the model, it is characterized by ignoring the losses, and the focus is only on the real power. This solution is usually used as the OPF-AC approximation to something more straightforward as a linear circuit analysis problem. Although an optimal AC power flow is a more accurate calculation of the steady-state

behavior of the EPS (13), correlation parameters can be obscured due to the complexity and nonlinearity of its characteristic equations. On the other hand, OPF-DC methods and their software requirements are simple; their models can be optimized efficiently, and the required minimum network data need little execution time (14). The optimal DC power flow approximation has been widely used in literature for nodal pricing and bottleneck analysis across transmission lines (15).

Another technique in which reactive power and EPS losses are not neglected to find an approximate solution to the optimal power flows is the linearization of the optimal AC power flows (LOPF-AC), for which numerical methods are used to obtain an approximation of the variables that have a power different from unity (16).

Due to the mathematical complexity of the optimization models that allow finding the solution to the OPF, the use of specialized software is necessary. This chapter deals with the help of Matlab and GAMS programs for modeling and optimization, respectively, but for this purpose, an interface has been created that makes GAMS a Matlab tool. With it, using the solution to the power flows, any other study can be performed, such as N-1 contingency analysis, reliability studies, and even stability studies.

4.2 Configuration

To configure the Matlab – GAMS interface, the first thing to do is to configure GAMS to allow its variables to belong to the computer's environment variables; for this purpose, when installing GAMS, the advanced installation mode option must be selected, see figure 4.1, so that after selecting the installation folder and other options, the environment variables can be enabled, see figure 4.2.

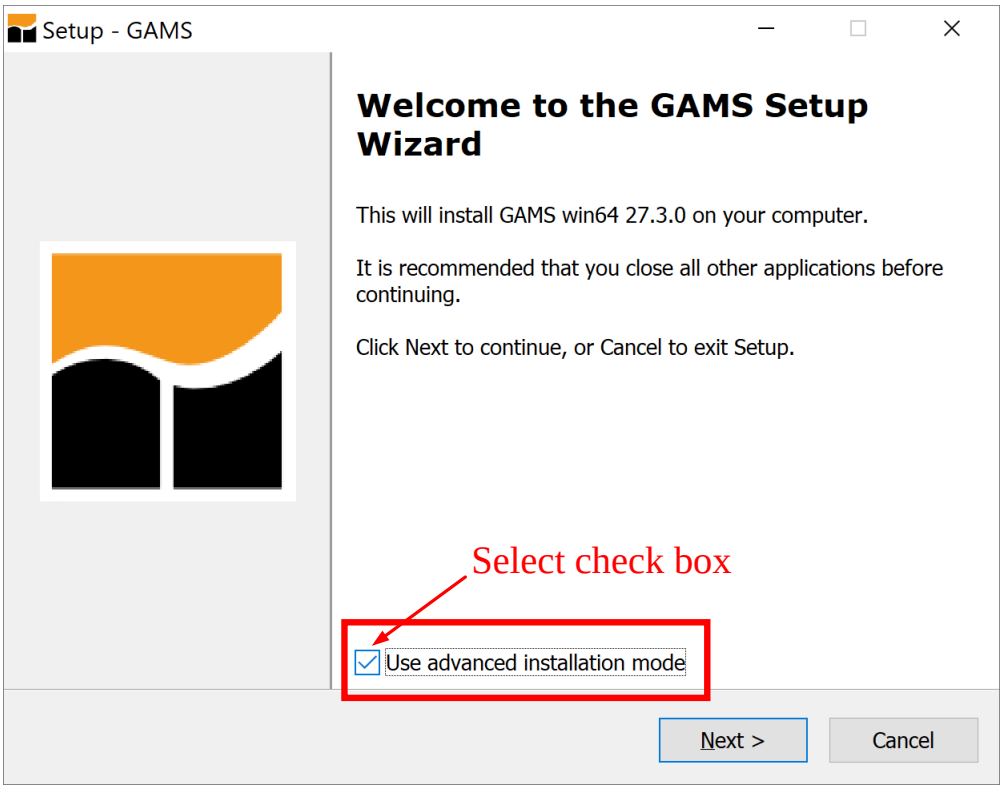

Figure 4.1: Initialization of GAMS installation process

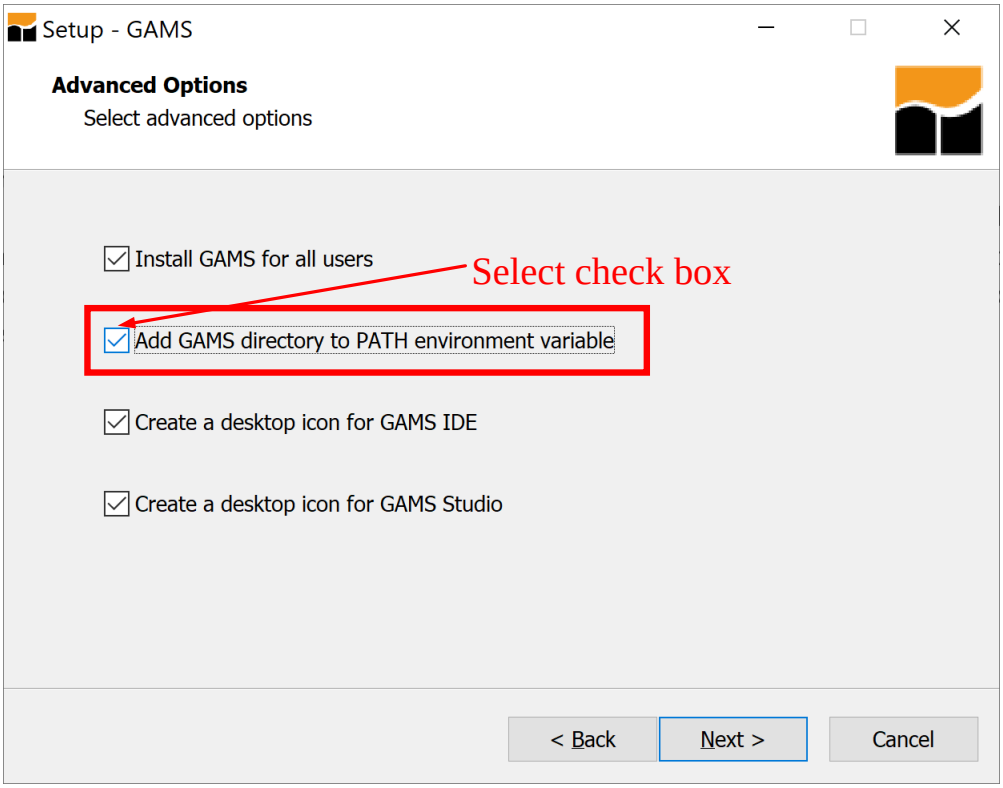

Figure 4.2: Enable GAMS path environment variable

Once GAMS is installed with this procedure, the environment variables of the computer must be configured, for which in Windows Explorer it must be made right click on "This PC" and from the menu select "Properties," there a system configuration window will be displayed, and in the right side it must be chosen "Advanced system settings," see figure 4.3; after that in the window that is displayed the option of environment variables must be selected, see figure 4.4. There it will allow configuring the environment variables for the user and those of the whole system; it is recommended to make the configuration for the two territories; for which the environment variable path must be edited and there to place the location where GAMS was installed, see figure 4.5.

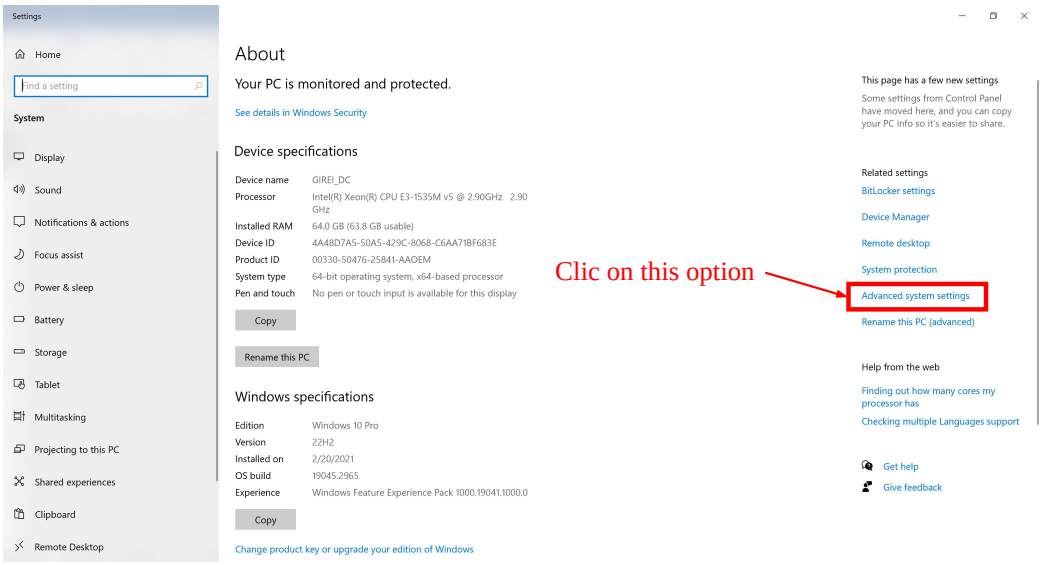

Figure 4.3: Configuration advanced system settings

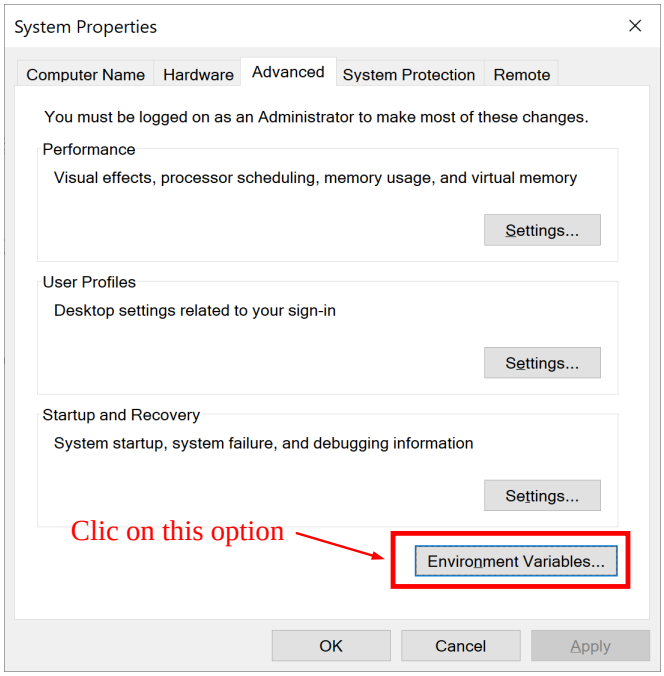

Figure 4.4: Choose environment variables

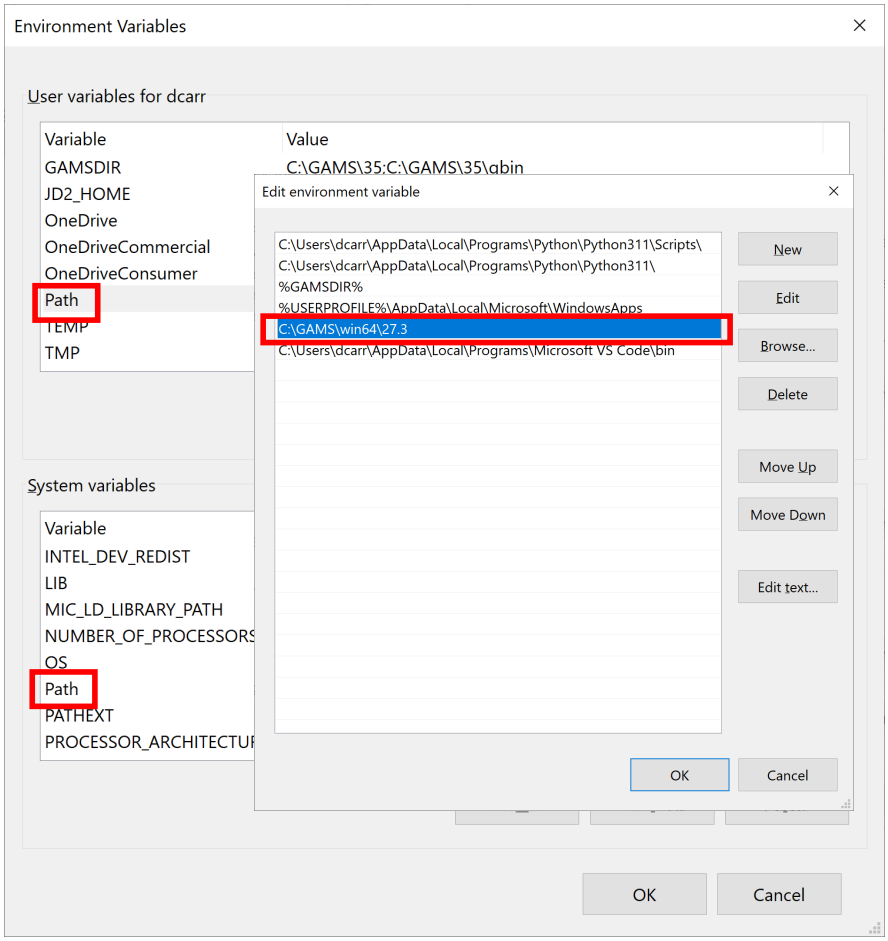

Figure 4.5: Set GAMS path

Once this process is completed, the environment variables must be configured in Matlab, for which the "Set Path" button must be selected in the toolbar of the "Home menu," see figure 4.6, and there is a similar process to the Windows configuration performed, a folder is added, and this will correspond to the location where GAMS was installed, see figure 4.7. Now you must verify that the process has been done correctly by writing the following code in the command window:

```
1 system 'gams ? lo=3'
```

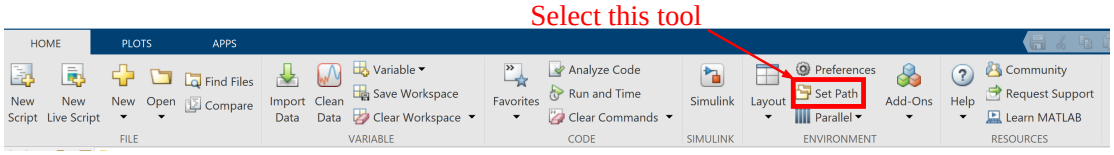

Figure 4.6: Set GAMS path

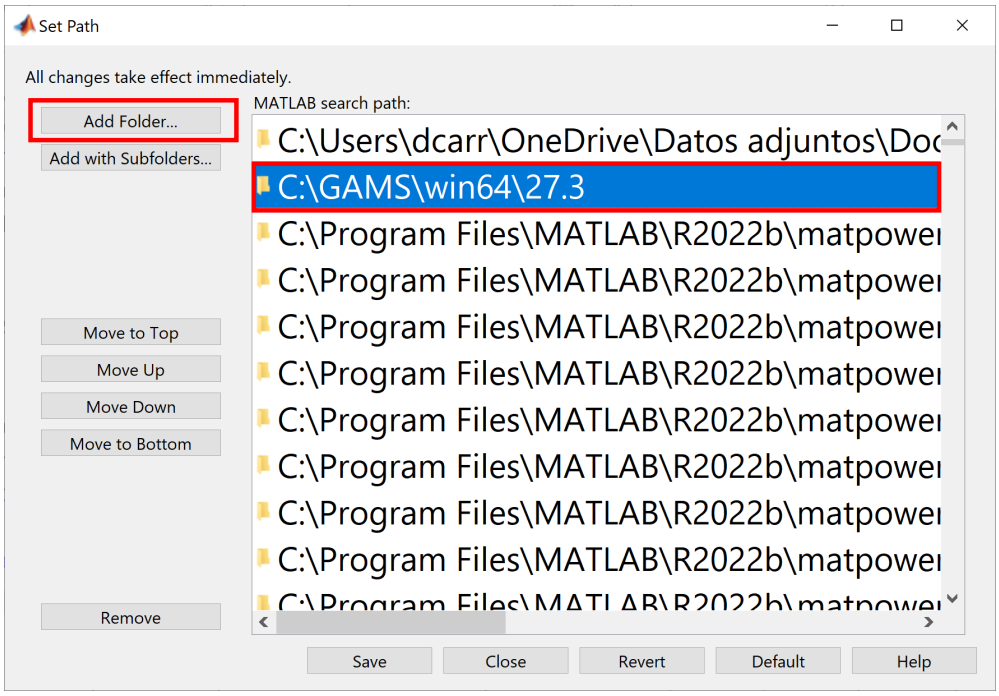

Figure 4.7: Configure GAMS path in Matlab

There you will see that the demo license information is reflected, execution time, and ans=0, see figure 4.8; if ans is equal to another value, it is recommended to review the configuration process because there may be an error and the environment variables have not been configured correctly.

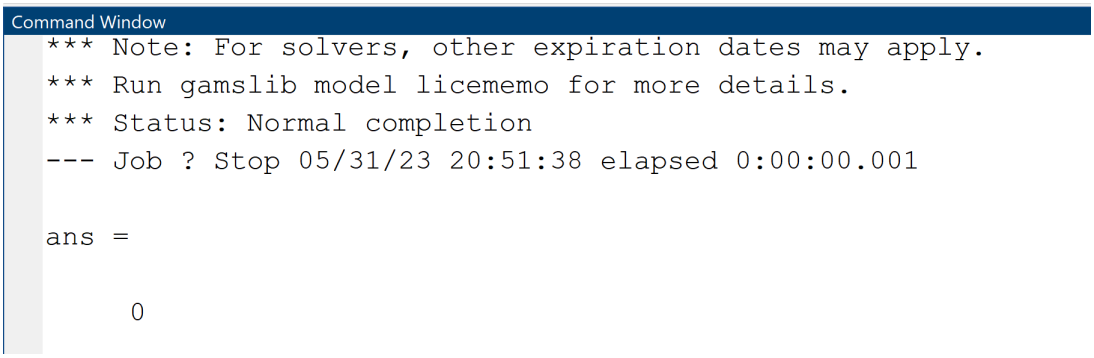

Figure 4.8: Configure GAMS path in Matlab

4.3 Optimal power flows

Power flows are a technique that allows quantifying each of the electrical variables in the operation of power systems, for which different styles such as Gauss – Seidel,

Newton − Raphson, and optimal power flows have been developed. The first two methods respond to iterative techniques, and the shortest convergence time of the two is Newton − Raphson; on the other hand, the optimal power flows respond to numerical optimization techniques that seek to minimize operating costs and maximize the technical resources of the power system.

Table 4.1: Variables related to the optimization model

Symbol	Description
i, j	Nodes index
C_g	Generation cost
P_g	Generator active power
Q_g	Generator reactive power
V_i	Nodes voltage
δ_i, δ_j	Nodes voltage angle
V_i^{max}	Upper voltage limit
V_i^{min}	Lower voltage limit
δ_i^{max}	Upper angular limit
δ_i^{min}	Lower angular limit
P_{i-j}	Active power flow
Q_{i-j}	Reactive power flow
S_{i-j}	Apparent power flow
SIL	Surge impedance load
B_{i-j}	Line susceptance
G_{i-j}	Line conductance
P_D	Demand active power
Q_D	Demand reactive power

4.3.1 DC Optimal power flow

The OPF-DC focuses on analyzing the EPS discarding reactive power, losses in the system, and that the magnitude of the voltage at each node is one pu; an OPF-DC seeks to minimize costs, which is similar to maximizing the welfare of society by obtaining a quality service at a lower cost. The objective function of the OPF-DC is defined in (4.1), which seeks to minimize production costs, which is subject to (4.2) that conditions the characteristics of the generation plants, (4.3) determines the maximum power flow that can circulate through the transmission lines, (4.4) the power balance between generation, demand, and power flow; (4.5) that determines the power flow that circulates through a transmission line depending on its susceptance; and (4.6) is the voltage angle limits (17)–(19).

$ObjetiveFunction$:

$$min\ Z = \sum_{g=1}^{N_G} C_g P_g \tag{4.1}$$

$Subjectto:$

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max} \tag{4.2}$$

$$-SIL \leq P_{i-j} \leq SIL \tag{4.3}$$

$$\sum_{i=1}^{n_{bus}} \sum_{j=1}^{n_{bus}} P_{i-j} = P_{G_i} - P_{D_i};\ \ \forall\, i,j\ \in\ N_B \tag{4.4}$$

$$P_{i-j} = B_{i-j}\,(\delta_i - \delta_j) \tag{4.5}$$

$$\delta_i^{min} \leq \delta_i \leq \delta_i^{max} \tag{4.6}$$

4.3.2 AC Optimal power flow

An OPF-AC, like an OPF-DC, aims to minimize the operating costs of the EPS, hence (4.7) describes the objective function, which is subject to (4.8) and (4.9) that determine the balance of active and reactive power between generation, demand, and power flow, by (4.10) and (4.11) that determines the active and reactive power flow that circulates through a transmission line depending on its susceptance and conductance, in (4.12) and (4.13) the active and reactive power limit restrictions of the generators are shown, (4.14) delimits the voltages in each node and the limits of the angular difference in each branch is (4.15) (20)–(22).

$ObjetiveFunction:$

$$min\ Z = \sum_{g=1}^{N_G} C_g P_g \tag{4.7}$$

$Subjectto:$

$$\sum_{i=1}^{n_{bus}} \sum_{j=1}^{n_{bus}} P_{i-j} = P_{Gi} - P_{Di}; \;\; \forall \, i, j \;\in\; N_B \tag{4.8}$$

$$\sum_{i=1}^{n_{bus}} \sum_{j=1}^{n_{bus}} Q_{i-j} = Q_{Gi} - Q_{Di}; \;\; \forall \, i, j \;\in\; N_B \tag{4.9}$$

$$P_{i-j} = V_i^2 G_{i-j} - V_i V_j \left[G_{i-j} cos(\delta_{i-j}) - B_{i-j} sin(\delta_{i-j}) \right] \tag{4.10}$$

$$Q_{i-j} = V_i^2 B_{i-j} + V_i V_j \left[G_{i-j} sin(\delta_{i-j}) - B_{i-j} cos(\delta_{i-j}) \right] \tag{4.11}$$

$$P_{Gi}^{min} \leq P_{Gi} \leq P_{Gi}^{max} \tag{4.12}$$

$$Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max} \tag{4.13}$$

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{4.14}$$

$$\delta_i^{min} \leq \delta_i \leq \delta_i^{max} \tag{4.15}$$

$$\delta_{i-j} = \delta_i - \delta_j \tag{4.16}$$

$$-SIL \leq S_{i-j} \leq SIL \tag{4.17}$$

4.4 Matlab - GAMS Coding and Results Analysis.

4.4.1 OPF-DC code

The following is the code developed in Matlab that allows the evaluation of the optimal DC power flows (OPF-DC), which has been divided into two parts. The first one is the main program in which the power system to be analyzed is configured, the sending and receiving of data between GAMS and Matlab, the file names to be generated, and the treatment of the OPF-DC results, which are Power flows in the lines, Node voltages, and Generation dispatch. The second part is the GAMS coding written in Matlab, which will be the OPF-DC code to be optimized.

Matlab main script code, written in a new script.

```
1  clc; clearvars; close all;
2  % Initialization of the interface with GAMS
3  GDXFileIN = 'Send'; %GDX File for send data to GAMS
4  GDXFileOUT = 'Get'; %GDX File for call data from GAMS
5  GMSFile = 'OPF_DC_30bus';
6  % The GAMS code for OPF-DC is written.
7  file='30BARRAS.xlsx'
8  writegams_OPFDC(GMSFile,file);
9  %% OPF-DC results are extracted
10 system(string(strcat('gams',{' '},GMSFile,' lo=3 gdx=',GDXFileOUT)))
       ;
11 %
       ************************************************************************

12 % Power flows in the lines
13 Pij.name = 'Pij';
14 Pij.compress = 'true';
15 Pij.form = 'sparse';
16 Pij = rgdx(GDXFileOUT,Pij);
17 %
       ************************************************************************

18 % Voltage angle at nodes
19 delta.name = 'delta';
20 delta.compress = 'true';
21 delta.form = 'sparse';
22 delta = rgdx(GDXFileOUT,delta);
23 %
       ************************************************************************

24 % Generation dispatch
25 Pgen.name = 'Pgen';
26 Pgen.compress = 'true';
27 Pgen.form = 'sparse';
28 Pgen = rgdx(GDXFileOUT,Pgen);
```

Line 7 of the main code represents the file in which the data is stored; for the case study, we have used as an example the data of the IEEE 30−busbar system in

an Excel file (30NODES.xlsx), which contains 3 sheets, called: data_gen, demand and lines, all sheets must start in cell A1. Electric demand is in the **demand** sheet and is shown in table 4.2, line data is in the **lines** sheet and is shown in table 4.3, and generator data is in the **data_gen** sheet, which is shown in table 4.4.

Table 4.2: IEEE 30busbar system electrical demand data

	Pd	**Qd**
1	0	0
2	21,7	12,7
3	2,4	1,2
4	7,6	1,6
5	94,2	19
6	0	0
7	22,8	10,9
8	30	30
9	0	0
10	5,8	2
11	5	2
12	11,2	7,5
13	0	0
14	6,2	1,6
15	8,2	2,5
16	3,5	1,8
17	9	5,8
18	3,2	0,9
19	9,5	3,4
20	2,2	0,7
21	17,5	11,2
22	0	0
23	3,2	1,6
24	8,7	6,7
25	0	0
26	3,5	2,3
27	0	0
28	0	0
29	2,4	0,9
30	10,6	1,9

Table 4.3: IEEE 30busbar system lines data

	r	x	b	sil
1 . 2	0,02	0,0575	0	130,00
1 . 3	0,05	0,1852	0	130,00
2 . 4	0,06	0,1737	0	65,00
3 . 4	0,01	0,0379	0	130,00
2 . 5	0,05	0,02	0	130,00
2 . 6	0,06	0,1763	0	65,00
4 . 6	0,01	0,0414	0	90,00
5 . 7	0,05	0,116	0	70,00
6 . 7	0,03	0,082	0	130,00
6 . 8	0,01	0,09	0	32,00
6 . 9	0	0,208	0	65,00
6 . 10	0	0,556	0	32,00
9 . 11	0	0,208	0	65,00
9 . 10	0	0,11	0	65,00
4 . 12	0	0,23	0	65,00
12 . 13	0	0,14	0	65,00
12 . 14	0,13	0,2559	0	32,00
12 . 15	0,0662	0,1304	0	32,00
12 . 16	0,0945	0,12	0	32,00
14 . 15	0,221	0,12	0	16,00
16 . 17	0,0824	0,1932	0	16,00
15 . 18	0,107	0,2185	0	16,00
18 . 19	0,0639	0,1292	0	16,00
19 . 20	0,034	0,068	0	32,00
10 . 20	0,0936	0,209	0	32,00
10 . 17	0,0324	0,0845	0	32,00
10 . 21	0,0348	0,0749	0	32,00
10 . 22	0,0727	0,1499	0	32,00
21 . 22	0,0116	0,22	0	32,00
15 . 23	0,1	0,202	0	16,00
22 . 24	0,115	0,18	0	16,00
23 . 24	0,132	0,27	0	16,00
24 . 25	0,1885	0,3292	0	16,00
25 . 26	0,2544	0,38	0	16,00
25 . 27	0,1093	0,2087	0	16,00
28 . 27	0	0,369	0	65,00
27 . 29	0,2198	0,4153	0	16,00
27 . 30	0,3202	0,6027	0	16,00
29 . 30	0,2399	0,453	0	16,00
8 . 28	0,0636	0,2	0	32,00
6 . 28	0,0169	0,06	0	32,00

Table 4.4: IEEE 30busbar system generators data

	pmax	pmin	b	Qmax	Qmin
1	120	0	2	150	0
2	120	0	1,75	60	-20
5	100	0	1	44,7	-15
8	110	0	3,25	62,5	-15
11	60	0	3	40	-10
13	80	0	3	48,7	-15

Function writing GAMS code developed in Matlab, written in a new function called writegams_OPFDC.m.

```
function writegams_OPFDC(FileGMS,file)
if ~contains(FileGMS,'.gms')
    FileGMS = strcat(FileGMS,'.gms');
end
filesDir = dir;
s0 = 0;
for c0=1:length(filesDir)
    findGMS = strfind(string(filesDir(c0).name),FileGMS);
    if isempty(findGMS)&&s0~=1 %#ok<*STREMP>
        s0 = 0;
    else
        s0 = 1;
    end
end
% s0 = 0: File does not exist .gms
% s0 = 1: File exists .gms
%*********************************************************************
if s0 == 0
    fgms = fopen(FileGMS,'w');
    clc;
    fprintf(fgms,'$Title Optimal power flow DC - 30 Bus System \n\n'
        );
    fprintf(fgms,'$offsymxref \n');
    fprintf(fgms,'$offsymlist \n\n');
%*********************************************************************
    fprintf(fgms,'option limrow=0; \n');
    fprintf(fgms,'option limcol=0; \n');
    fprintf(fgms,'option solprint=on; \n');
    fprintf(fgms,'option sysout=off; \n\n');
%*********************************************************************
    fprintf(fgms,'option LP=CPLEX; \n');
    fprintf(fgms,'option MIP=CPLEX; \n');
    fprintf(fgms,'option NLP=CONOPT; \n');
    fprintf(fgms,'option MINLP=COINBONMIN; \n\n');
%*********************************************************************
    fprintf(fgms,'option OPTCR=0; \n');
    fprintf(fgms,'option OPTCA=0; \n\n');
%*********************************************************************
```

```
38      fprintf(fgms,'SCALAR \n');
39      fprintf(fgms,'sbase     /100/ \n');
40      fprintf(fgms,'pi        /3.141592654/; \n\n');
41  %*********************************************************************
42      fprintf(fgms,'set \n');
43      fprintf(fgms,'bus               /1*30/ \n');
44      fprintf(fgms,'slack(bus)        /1/; \n\n');
45      fprintf(fgms,'alias(bus,node); \n');
46  % i=M:\\OPFDC\\%s corresponds to the address of the file where the
47  % EPS data is stored
48  %*********************************************************************
49      fprintf(fgms,'TABLE    data_gen(bus,*)  generators data \n');
50      fprintf(fgms,'$call =xls2gms r=data_gen!A1:F7 i=M:\\OPFDC\\%s O=
            Buses.inc \n',file);
51      fprintf(fgms,'$include Buses.inc \n');
52      fprintf(fgms,'; \n\n');
53  %*********************************************************************
54      fprintf(fgms,'TABLE    demand(bus,*)  electrical demand data \n'
            );
55      fprintf(fgms,'$call =xls2gms r=demand!A1:C31 i=M:\\OPFDC\\%s O=
            Buses.inc \n',file);
56      fprintf(fgms,'$include Buses.inc \n');
57      fprintf(fgms,'; \n\n');
58  %*********************************************************************
59      fprintf(fgms,'TABLE    lines(bus,node,*)  lines data \n');
60      fprintf(fgms,'$call =xls2gms r=lines!A1:E42 i=M:\\OPFDC\\%s O=
            Buses.inc \n',file);
61      fprintf(fgms,'$include Buses.inc \n');
62      fprintf(fgms,'; \n\n');
63  %*********************************************************************
64      fprintf(fgms,'display lines, demand, data_gen; \n');
65  %*********************************************************************
66      fprintf(fgms,'lines(bus,node,''x'')$(lines(bus,node,''x'')=0) =
            lines(node,bus,''x''); \n');
67      fprintf(fgms,'lines(bus,node,''sil'')$(lines(bus,node,''sil'')
            =0) = lines(node,bus,''sil''); \n');
68      fprintf(fgms,'lines(bus,node,''bij'')$lines(bus,node,''sil'') =
            1/lines(bus,node,''x''); \n');
69  %*********************************************************************
70      fprintf(fgms,'Parameter connection_lin(bus,node); \n');
71      fprintf(fgms,'connection_lin(bus,node)$(lines(bus,node,''sil'')
            and lines(node,bus,''sil'')) = 1; \n');
72      fprintf(fgms,'connection_lin(bus,node)$(connection_lin(node,bus)
            ) = 1; \n');
73      fprintf(fgms,'display connection_lin, lines, demand, data_gen; \
            n\n');
74  %*********************************************************************
75      fprintf(fgms,'Variable Z, Pij(bus,node), Pgen(bus), delta(bus) ;
             \n');
76      fprintf(fgms,'Equation res1, res2, res3; \n\n');
77  %*********************************************************************
78      fprintf(fgms,'res1.. \n');
```

```
79      fprintf(fgms,'Z =e= sum((bus), Pgen(bus)*data_gen(bus,''b'')*
            sbase); \n\n');
80  %*********************************************************************
81      fprintf(fgms,'res2(bus,node)$connection_lin(bus,node).. \n');
82      fprintf(fgms,'Pij(bus,node) =e= lines(bus,node,''bij'')*(delta(
            bus)-delta(node)); \n\n');
83  %*********************************************************************
84      fprintf(fgms,'res3(bus).. \n');
85      fprintf(fgms,'Pgen(bus)$data_gen(bus,''Pmax'') - demand(bus,''pd
            '')/base =e= sum(node$connection_lin(node,bus), Pij(bus,node)
            ); \n\n');
86  %*********************************************************************
87      fprintf(fgms,'Model loadflow /all/; \n\n');
88  %*********************************************************************
89      fprintf(fgms,'Pgen.lo(bus) = data_gen(bus,''Pmin'')/sbase; \n');
90      fprintf(fgms,'Pgen.up(bus) = data_gen(bus,''Pmax'')/sbase; \n');
91  %*********************************************************************
92      fprintf(fgms,'delta.up(bus) = pi/2; \n');
93      fprintf(fgms,'delta.lo(bus) =-pi/2; \n');
94      fprintf(fgms,'delta.l(bus) = 0; \n');
95      fprintf(fgms,'delta.fx(slack) = 0; \n\n');
96  %*********************************************************************
97      fprintf(fgms,'Pij.up(bus,node)$((connection_lin(bus,node))) = 1*
            lines(bus,node,''sil'')/base; \n');
98      fprintf(fgms,'Pij.lo(bus,node)$((connection_lin(bus,node))) =-1*
            lines(bus,node,''sil'')/base; \n');
99      fprintf(fgms,'solve load flow min Z using lp; \n');
100 end
```

It is important to remember that the main script, the function, and the Excel file must be in the same location.

4.4.2 OPF-AC code

The following is the code developed in Matlab that allows the evaluation of the optimal AC power flows (OPF-AC), which has been divided into 2 parts. The first one is the main program in which the power system to be analyzed is configured, the sending and receiving of data between GAMS and Matlab, the file names to be generated, and the treatment of the OPF-AC results, which are Power flows in the lines, Node voltages, and Generation dispatch. The second part is the GAMS coding written in Matlab, which will be the OPF-AC code to be optimized.

Matlab main script code, written in a new script.

```
1  clc; clearvars; close all;
2  % Initialization of the interface with GAMS
3  GDXFileIN = 'Send'; %GDX File for send data to GAMS
4  GDXFileOUT = 'Get'; %GDX File for call data from GAMS
5  GMSFile = 'OPF_AC_30bus';
6  % The GAMS code for OPF-AC is written.
7  file='30NODES.xlsx'
```

```
writegams_OPF(GMSFil,file);
%% OPF-AC results are extracted
system(string(strcat('gams',{' '},GMSFile,' lo=3 gdx=',GDXFileOUT)))
    ;
%*********************************************************************
%  Power flows in the lines
Pij.name = 'Pij';
Pij.compress = 'true';
Pij.form = 'sparse';
Pij = rgdx(GDXFileOUT,Pij);
Qij.name = 'Qij';
Qij.compress = 'true';
Qij.form = 'sparse';
Qij = rgdx(GDXFileOUT,Qij);
%*********************************************************************
% Node voltages
V.name = 'V';
V.compress = 'true';
V.form = 'sparse';
V = rgdx(GDXFileOUT,V);
delta.name = 'delta';
delta.compress = 'true';
delta.form = 'sparse';
delta = rgdx(GDXFileOUT,delta);
%*********************************************************************
% Generation dispatch
Pgen.name = 'Pgen';
Pgen.compress = 'true';
Pgen.form = 'sparse';
Pgen = rgdx(GDXFileOUT,Pgen);
Qgen.name = 'Qgen';
Qgen.compress = 'true';
Qgen.form = 'sparse';
Qgen = rgdx(GDXFileOUT,Qgen);
```

Function writing GAMS code developed in Matlab, written in a new function called writegams_OPF.m, and the EPS data is the same file used in OPF-DC (30NODES.xlsx).

```
function writegams_OPF(FileGMS,file)
if ~contains(FileGMS,'.gms')
    FileGMS = strcat(FileGMS,'.gms');
end
filesDir = dir;
s0 = 0;
for c0=1:length(filesDir)
    findGMS = strfind(string(filesDir(c0).name),FileGMS);
    if isempty(findGMS)&&s0~=1 %#ok<*STREMP>
        s0 = 0;
    else
        s0 = 1;
    end
```

```
14 end
15 % s0 = 0: File does not exist .gms
16 % s0 = 1: File exists .gms
17 %*********************************************************************
18 if s0 == 0
19     fgms = fopen(FileGMS,'w');
20     clc;
21     fprintf(fgms,'$Title Optimal power flow AC - 30 Bus System \n\n'
           );
22     fprintf(fgms,'$offsymxref \n');
23     fprintf(fgms,'$offsymlist \n\n');
24     fprintf(fgms,'option limrow=0; \n');
25     fprintf(fgms,'option limcol=0; \n');
26     fprintf(fgms,'option solprint=on; \n');
27     fprintf(fgms,'option sysout=off; \n\n');
28     fprintf(fgms,'option LP=CPLEX; \n');
29     fprintf(fgms,'option MIP=CPLEX; \n');
30     fprintf(fgms,'option NLP=CONOPT; \n');
31     fprintf(fgms,'option MINLP=COINBONMIN; \n\n');
32     fprintf(fgms,'option OPTCR=0; \n');
33     fprintf(fgms,'option OPTCA=0; \n\n');
34 %*********************************************************************
35     fprintf(fgms,'SCALAR \n');
36     fprintf(fgms,'sbase    /100/ \n');
37     fprintf(fgms,'pi       /3.141592654/; \n\n');
38 %*********************************************************************
39     fprintf(fgms,'set \n');
40     fprintf(fgms,'bus              /1*30/ \n');
41     fprintf(fgms,'slack(bus)       /1/; \n\n');
42     fprintf(fgms,'alias(bus,node); \n');
43 % i=M:\\OPFAC\\%s corresponds to the address of the file where the
44 % EPS data is stored
45 %*********************************************************************
46     fprintf(fgms,'TABLE    data_gen(bus,*)  generators data \n');
47     fprintf(fgms,'$call =xls2gms r=data_gen!A1:F7 i=M:\\OPFAC\\%s O=
           Buses.inc \n',file);
48     fprintf(fgms,'$include Buses.inc \n');
49     fprintf(fgms,'; \n\n');
50 %*********************************************************************
51     fprintf(fgms,'TABLE    demand(bus,*)  electrical demand data \n'
           );
52     fprintf(fgms,'$call =xls2gms r=demand!A1:C31 i=M:\\OPFAC\\%s O=
           Buses.inc \n',file);
53     fprintf(fgms,'$include Buses.inc \n');
54     fprintf(fgms,'; \n\n');
55 %*********************************************************************
56     fprintf(fgms,'TABLE    lines(bus,node,*)  lines data \n');
57     fprintf(fgms,'$call =xls2gms r=lines!A1:E42 i=M:\\OPFAC\\%s O=
           Buses.inc \n',file);
58     fprintf(fgms,'$include Buses.inc \n');
59     fprintf(fgms,'; \n\n');
60 %*********************************************************************
```

```
61      fprintf(fgms,'display lines, demand, data_gen; \n');
62  %*********************************************************************
63      fprintf(fgms,'lines(bus,node,''x'')$(lines(bus,node,''x'')=0) =
            lines(node,bus,''x''); \n');
64      fprintf(fgms,'lines(bus,node,''r'')$(lines(bus,node,''r'')=0) =
            lines(node,bus,''r''); \n');
65      fprintf(fgms,'lines(bus,node,''b'')$(lines(bus,node,''b'')=0) =
            lines(node,bus,''b''); \n');
66      fprintf(fgms,'lines(bus,node,''sil'')$(lines(bus,node,''sil'')
            =0) = lines(node,bus,''sil''); \n');
67      fprintf(fgms,'lines(bus,node,''bij'')$lines(bus,node,''sil'') =
            1/lines(bus,node,''x''); \n');
68      fprintf(fgms,'lines(bus,node,''z'')$lines(bus,node,''sil'') =
            sqrt(sqr(lines(bus,node,''x'')) + sqr(lines(bus,node,''r'')))
            ; \n');
69      fprintf(fgms,'lines(node,bus,''z'')$(lines(bus,node,''z'')=0) =
            lines(bus,node,''z''); \n');
70      fprintf(fgms,'lines(bus,node,''th'')$(lines(bus,node,''sil'')
            and lines(bus,node,''x'') and lines(bus,node,''r'')) = arctan
            (lines(bus,node,''x'')/(lines(bus,node,''r''))); \n');
71      fprintf(fgms,'lines(bus,node,''ybus'')$lines(bus,node,''sil'') =
             1/lines(bus,node,''z''); \n');
72      fprintf(fgms,'lines(node,bus,''ybus'')$(lines(bus,node,''ybus'')
            =0) = lines(bus,node,''ybus''); \n\n');
73  %*********************************************************************
74      fprintf(fgms,'Parameter connection_lin(bus,node); \n');
75      fprintf(fgms,'connection_lin(bus,node)$(lines(bus,node,''sil'')
            and lines(node,bus,''sil'')) = 1; \n');
76      fprintf(fgms,'connection_lin(bus,node)$(connection_lin(node,bus)
            ) = 1; \n');
77      fprintf(fgms,'display connection_lin, lines, demand, data_gen; \
            n\n');
78  %*********************************************************************
79      fprintf(fgms,'Variable Z, Pij(bus,node), Qij(bus,node), Pgen(bus
            ), Qgen(bus), delta(bus), V(bus); \n');
80      fprintf(fgms,'Equation res1, res2, res3, res4, res5; \n\n');
81  %*********************************************************************
82      fprintf(fgms,'res1.. \n');
83      fprintf(fgms,'Z =g= sum((bus), Pgen(bus)*data_gen(bus,''b'')*
            sbase); \n\n');
84  %*********************************************************************
85      fprintf(fgms,'res2(bus,node)$connection_lin(bus,node).. \n');
86      fprintf(fgms,'Pij(bus,node) =e= (V(bus)*V(bus)*cos(lines(node,
            bus,''th'')) - V(bus)*V(node)*cos(delta(bus)-delta(node)+
            lines(node,bus,''th'')))*lines(node,bus,''ybus''); \n\n');
87  %*********************************************************************
88      fprintf(fgms,'res3(bus,node)$connection_lin(bus,node).. \n');
89      fprintf(fgms,'Qij(bus,node) =e= (V(bus)*V(bus)*sin(lines(node,
            bus,''th'')) - V(bus)*V(node)*sin(delta(bus) - delta(node) +
            lines(node,bus,''th'')))*lines(node,bus,''ybus'') - lines(
            node,bus,''b'')*V(bus)*V(bus)/2; \n\n');
90  %*********************************************************************
```

```
91      fprintf(fgms,'res4(bus).. \n');
92      fprintf(fgms,'Pgen(bus)$data_gen(bus,''Pmax'') - demand(bus,''pd
            '')/base =e= sum(node$connection_lin(node,bus), Pij(bus,node)
            ); \n\n');
93  %*********************************************************************
94      fprintf(fgms,'res5(bus).. \n');
95      fprintf(fgms,'Qgen(bus)$data_gen(bus,''Qmax'') - demand(bus,''qd
            '')/base =e= sum(node$connection_lin(node,bus), Qij(bus,node)
            ); \n\n');
96  %*********************************************************************
97      fprintf(fgms,'Model loadflow /all/; \n\n');
98  %*********************************************************************
99      fprintf(fgms,'Pgen.lo(bus) = data_gen(bus,''Pmin'')/sbase; \n');
100     fprintf(fgms,'Pgen.up(bus) = data_gen(bus,''Pmax'')/sbase; \n');
101     fprintf(fgms,'Qgen.lo(bus) = data_gen(bus,''Qmin'')/sbase; \n');
102     fprintf(fgms,'Qgen.up(bus) = data_gen(bus,''Qmax'')/sbase; \n\n'
            );
103 %*********************************************************************
104     fprintf(fgms,'delta.up(bus) = pi/2; \n');
105     fprintf(fgms,'delta.lo(bus) =-pi/2; \n');
106     fprintf(fgms,'delta.l(bus) = 0; \n');
107     fprintf(fgms,'delta.fx(slack) = 0; \n\n');
108 %*********************************************************************
109     fprintf(fgms,'Pij.up(bus,node)$((connection_lin(bus,node))) = 1*
            lines(bus,node,''sil'')/sbase; \n');
110     fprintf(fgms,'Pij.lo(bus,node)$((connection_lin(bus,node))) =-1*
            lines(bus,node,''sil'')/sbase; \n');
111     fprintf(fgms,'Qij.up(bus,node)$((connection_lin(bus,node))) = 1*
            lines(bus,node,''sil'')/sbase; \n');
112     fprintf(fgms,'Qij.lo(bus,node)$((connection_lin(bus,node))) =-1*
            lines(bus,node,''sil'')/sbase; \n\n');
113 %*********************************************************************
114     fprintf(fgms,'V.lo(bus) = 0.9; \n');
115     fprintf(fgms,'V.up(bus) = 1.1; \n');
116     fprintf(fgms,'V.l(bus) = 1; \n');
117     fprintf(fgms,'V.fx(slack) = 1; \n\n');
118     fprintf(fgms,'solve load flow min Z using nlp; \n');
119 end
```

In each of the codes in the main script, the treatment of the variables resulting from the optimization process, such as voltage, angle, and active and reactive power, is carried out at the end.

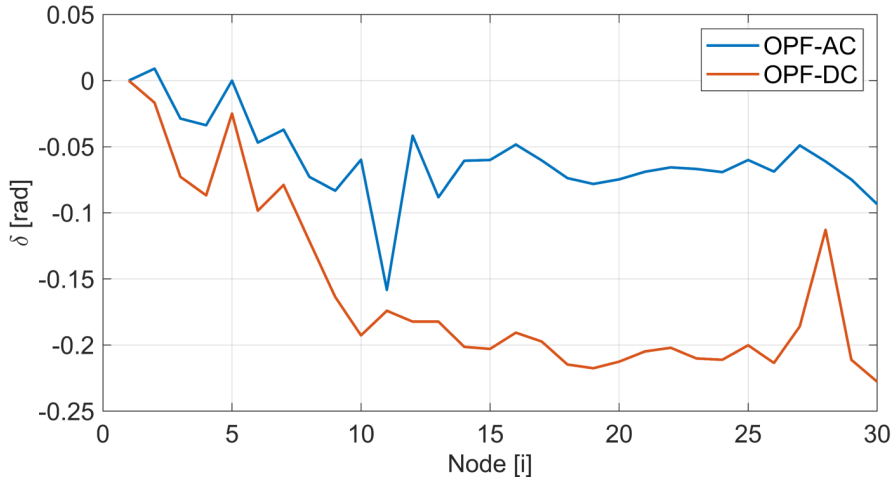

Figure 4.9: δ OPF$-$AC vs δ OPF$-$DC

Figure 4.9 shows the comparison of the results of the voltage angle optimization in each of the EPS nodes; as can be seen, there is much similarity in the results in terms of the trend, but because the OPF-DC gives approximate results, its fluctuations are more significant because there are fewer electrical parameters to analyze or that restrict the optimization model.

This process can be carried out with each variable, and even more complex analyses can be started, such as studies of different stability indexes.

4.5 Conclusions

The use of computational tools for the study and analysis of power systems has led to the availability of new methodologies that can be more robust and that find solutions to problems that could not be solved before. For example, the use of OPF in large systems some years ago was solved by continuous power Flow, which is a modification of the classical Newton - Raphso model, and now it can be solved by linear or nonlinear optimization.

Having an interface between GAMS and Matlab facilitates the EPS analysis processes since one can generate different conditions for the model in Matlab and will always call the same optimization subroutine. This situation is very complex to perform directly in GAMS.

Having generic methodologies to solve optimal EPS power flows allows finding solutions that simulators do not yet have in their algorithm; for example, optimal switching of transmission lines is a new theory that is already applied to date in EPS in contingency scenarios and avoids possible blackouts by not having a

characterization of the operation in this type of scenarios and now having an answer based on optimization criteria the more reliable solutions.

The interface in Matlab and GAMS reduces the analysis and study times of the EPS, simplifies the traditional way of performing such research, and the reliability of not having errors in the manipulation of the information by not having external files, as a result of independent analysis, is increased.

Bibliography

[1] D. Carrión, J. W. González, G. J. López, and I. A. Isaac, "Alternative fault detection method in electrical power systems based on ARMA model", *2019 FISE-IEEE/CIGRE Conference - Living the Energy Transition, FISE/CIGRE 2019*, 2019. DOI: 10.1109/FISECIGRE48012.2019.8984981.

[2] J. Toctaquiza and D. Carrión, "Estado del arte modelo óptimo de operación posterior a ataques intencionales considerando conmutación de los sistemas de transmisión", *ITECKNE Innovación e Investigación en Ingeniería*, vol. 18, no. 2, p. 17, 2021, ISSN: 1692-1798. DOI: https://doi.org/10.15332/iteckne.v18i2.2559.

[3] D. Carrión, J. Francisco, and M. Paul, "Revisión para la restauración optima de la operación del sistema eléctrico basado en criterios de calidad de energía y estabilidad", *Revista de I+D Tecnológico*, vol. 17, no. 1, p. 9, 2021. DOI: 10.33412/idt.v17.1.2928.

[4] J. Pilatásig Lasluisa and D. Carrión, "Resiliencia de Sistemas Eléctricos de Potencia mediante la Conmutación de Líneas de Transmisión – Estado del arte", *I+D Tecnológico*, vol. 16, no. 2, 2020, ISSN: 1680-8894. DOI: 10.33412/idt.v16.2.2834.

[5] J. Ramirez, D. Carrión, and E. Inga, "Compensación reactiva en redes eléctricas de transmisión basado en programación no lineal considerando ubicación óptima de SVC", *Revista de I+D Tecnológico*, vol. 17, no. February, 2021. DOI: 10.33412/idt.v17.1.2918.

[6] J. Palacios and D. F. Carrión Galarza, "Estado del arte de la planeación de expansión de sistemas de transmisión", *I+D Tecnológico*, vol. 16, no. 2, pp. 1–8, 2020. DOI: 10.33412/idt.v16.2.2835.

[7] N. Gupta, R. Shekhar, and P. K. Kalra, "Computationally efficient composite transmission expansion planning: A Pareto optimal approach for techno-economic solution", *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 917–926, Dec. 2014, ISSN: 01420615. DOI: 10.1016/j.ijepes.2014.05.070. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S014206151400341X.

[8] A. R. Kumar and L. Premalatha, "Electrical Power and Energy Systems Optimal power flow for a deregulated power system using adaptive real coded biogeography-based optimization", *INTERNATIONAL JOURNAL OF ELECTRICAL POWER AND ENERGY SYSTEMS*, vol. 73, pp. 393–399, 2015, ISSN: 0142-0615. DOI: 10.1016/j.ijepes.2015.05.011.

[9] R. S. Ferreira, C. L. T. Borges, and M. V. F. Pereira, "A flexible mixed-integer linear programming approach to the AC optimal power flow in distribution systems", *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2447–2459, 2014, ISSN: 08858950. DOI: 10.1109/TPWRS.2014.2304539.

[10] T. Akbari and M. Tavakoli Bina, "Linear approximated formulation of AC optimal power flow using binary discretisation", *IET Generation, Transmission & Distribution*, vol. 10, no. 5, pp. 1117–1123, 2016, ISSN: 1751-8687. DOI: 10.1049/iet-gtd.2015.0388. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2015.0388.

[11] V. Sarkar and S. a. Khaparde, "Implementation of LMP-FTR mechanism in an AC-DC system", *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 737–746, 2008, ISSN: 08858950. DOI: 10.1109/TPWRS.2008.920202.

[12] A. Kargarian, J. Mohammadi, J. Guo, *et al.*, "Toward Distributed/Decentralized DC Optimal Power Flow Implementation in Future Electric Power Systems", *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2018, ISSN: 19493053. DOI: 10.1109/TSG.2016.2614904.

[13] "Pricing Energy and Ancillary Services in Integrated Market Systems by an Optimal Power Flow", *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 339–347, 2004, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2003.820701.

[14] B. Stott, J. Jardim, and O. Alsac, "DC Power Flow Revisited", *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2009.2021235.

[15] M. Bachtiar Nappu, A. Arief, and R. C. Bansal, "Transmission management for congested power system: A review of concepts, technical challenges and development of a new methodology", *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 572–580, Oct. 2014, ISSN: 13640321. DOI: 10.1016/j.rser.2014.05.089. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364032114004237.

[16] T. Akbari and M. Tavakoli Bina, "A linearized formulation of AC multi-year transmission expansion planning: A mixed-integer linear programming approach", *Electric Power Systems Research*, vol. 114, pp. 93–100, Sep. 2014, ISSN: 03787796. DOI: 10.1016/j.epsr.2014.04.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779614001539.

[17] P. Masache, D. Carrión, and J. Cárdenas, "Optimal Transmission Line Switching to Improve the Reliability of the Power System Considering AC Power Flows", *Energies 2021, Vol. 14, Page 3281*, vol. 14, no. 11, p. 3281, Jun. 2021. DOI: 10.3390/EN14113281. [Online]. Available: https://www.mdpi.com/1137108.

[18] S. Pinzón, D. Carrión, and E. Inga, "Optimal Transmission Switching Considering N-1 Contingencies on Power Transmission Lines", *IEEE Latin America Transactions*, vol. 19, no. 4, pp. 534–541, 2021. DOI: 10.1109/TLA.2021.9448535. [Online]. Available: https://latamt.ieeer9.org/index.php/transactions/article/view/3691/636.

[19] D. Carrión, J. Palacios, M. Espinel, and J. W. González, "Transmission Expansion Planning Considering Grid Topology Changes and N-1 Contingencies Criteria", in *Recent Advances in Electrical Engineering, Electronics and Energy*, Springer, Ed., Springer, 2021, pp. 266–279. DOI: 10.1007/978-3-030-72208-1_20. [Online]. Available: http://link.springer.com/10.1007/978-3-030-72208-1%7B%5C_%7D20.

[20] D. Carrión, E. García, M. Jaramillo, and J. W. González, "A Novel Methodology for Optimal SVC Location Considering N-1 Contingencies and Reactive Power Flows Reconfiguration", *Energies*, vol. 14, no. 20, pp. 1–17, 2021. DOI: 10.3390/en14206652.

[21] A. Lemus, D. Carrión, E. Aguire, and J. W. Gonz, "Location of distributed resources in rural-urban marginal power grids considering the voltage collapse prediction index", *Ingenius*, vol. 28, pp. 25–33, 2022. DOI: https://doi.org/10.17163/ings.n28.2022.02.

[22] F. Quinteros, D. Carrión, and M. Jaramillo, "Optimal Power Systems Restoration Based on Energy Quality and Stability Criteria", *Energies*, vol. 15, no. 6, 2022. DOI: 10.3390/en15062062.