Algebraic Solution of the Load-Flow Problem for a four Nodes Electrical Network

Antonio Montes
Universitat Politècnica de Catalunya
E-mail: montes@ma2.upc.es

Abstract

The load-flow problem for an electrical network is formulated as a system of polynomial equations in several variables. In present Electrical Engineering it is solved using numerical methods. But the system contains parameters and must very often be solved in real time in order to simulate events. It would be, in principle, useful to reduce the system to echelon or triangular form, using algebraic techniques, as, for instance, Gröbner bases, in order to obtain the solution once for all and then use it in any simulation. In this paper, algorithms to triangulate the load-flow equations of a four nodes electrical network are presented. These algorithms have been implemented in Maple, and simulations using them are given. The advantages of the algebraic solution compared to the numerical one are discussed. In particular, the algebraic solution allows us to compute, in a very simple way, derivatives and the related numerical conditioning of the problem and also the numerical conditioning of the algorithm.

1 Introduction

In Electrical Engineering, the load-flow problem must very often be solved as the charges connected to an electrical network vary. Nowadays this problem is solved using numerical methods. Nevertheless it would be, in principle, very useful to obtain an algebraic solution that is computed only once for a given network and then used to obtain any particular solution. This is our objective.

In previous articles, we applied the ordinary methods of Computer Algebra to the load-flow problem. In [4] we studied the numerical conditioning of triangular polynomial systems produced by elimination algorithms. In [5] we solved what we call the not-matching 3-nodes problem using Gröbner bases and then we computed numerical conditioning both of the problem and the algorithm. In [6] we used the relations provided by comprehensive Gröbner bases (C.G.B.) [7] to simplify the previous results.

In these papers, we used the classical load-flow equations, also used in any numerical approach, that are expressed in terms of real variables and coefficients, and we only applied standard computer algebra algorithms. With these tools we only were able to solve the not-matching nodes problem of a 3-nodes network.

We now use a formulation of the equations in terms of complex variables and coefficients which is better adapted to an algebraic solution. Instead of using standard algorithms, we explore different possibilities and adopt now simpler eliminations. So, we are able to solve the not-matching 4-nodes problem and the matching 3-nodes problem. This approach also gives some insight about:

(i) Systems of equations in several complex variables in which both z_i and its complex conjugate $\bar{z}_i$ occur.

(ii) Elimination of variables by combining partial Gröbner bases, dialytic elimination or resultants and a degree-reduction method described in this paper.

(iii) Numerical conditioning of the load-flow problem and of the elimination algorithm.

2 The load-flow equations

An electrical energy distribution network is characterized by a set of lines connecting different nodes. The state variables describing the network are the voltages E_i and the total injected powers P_i (generated power minus connected charges) at each node. These quantities are complex (called phasors in Electrical Engineering). The equations are formulated (see [3]) equating the sum of power flux outgoing from a node through all the lines connected to it and the total injected power P_i at node i. The result is a system of n equations in complex variables and coefficients (one for each node) of the form:

$$E_i \sum_{j=1}^{n} a_{ij} \bar{E}_j - P_i = 0 \qquad 1 \leq i \leq n \tag{1}$$

The a_{ij} form a symmetric matrix (not an hermitian one), and are determined by the conductances and susceptances of the network. If, as usual, we model a line connecting node i and node j by the 3 real parameters c_{ij}, s_{ij} and b_{ij}, representing respectively the conductance, the susceptance and the susceptance to earth, then the formulas giving a_{ij} as functions of the conductances are:

$$\begin{aligned} a_{ii} &= \sum_{k\neq i}\left(c_{ik} + I\left(s_{ik} + \frac{b_{ik}}{2}\right)\right) \\ a_{ij} &= -c_{ij} - I s_{ij} \qquad \text{for} \quad i \neq j \end{aligned}$$

We use relative units. For example, if the base voltage of the network is $T_B = 220$ MV and the base power $P_B = 100$ MVA, the conductance base unit is

$$X_B = (100\,\mathrm{MV})/(220\,\mathrm{MVA})^2.$$

A scheme of the complete four nodes network is given in fig. 1

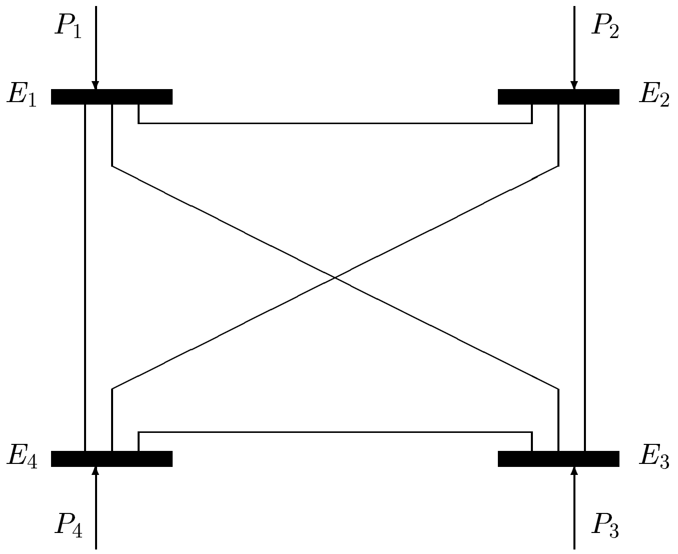

Fig. 1: The complete 4-nodes network

The load-flow problem is to compute the needed incoming powers in order to maintain the voltages near the base voltage 1. One generally formulates the problem starting from an observed state of the variables of the network. Suppose that we observe a deviation of the voltage E_1 from the base voltage 1. The problem is to compute what must the new injected power be at node q, without modifying the other powers in order to rebalance the voltage at node 1 to $E_1 = 1$.

We talk about a *matching* nodes problem when the node q is precisely the observed node, and about a *not-matching* nodes problem otherwise. In the second case one of the equations is linear and the algebraic problem is easier to solve.

For the not-matching nodes problem, the system of equations is:

$$\begin{aligned} a_{11} + \sum_{j=2}^{n} a_{1j}\bar{E}_j - P_1 &= 0 \\ a_{i1}E_i + E_i\sum_{j=2}^{n} a_{ij}\bar{E}_j - P_i &= 0 \qquad 2 \le i \le n-1 \end{aligned} \tag{2}$$

and after solving this system, the power P_n is computed by:

$$P_n = a_{n1}\bar{E}_n + E_n\sum_{j=2}^{n} a_{nj}\bar{E}_j$$

For the matching nodes problem, the corresponding equations are:

$$a_{i1}E_i + E_i\sum_{j=2}^{n} a_{ij}\bar{E}_j - P_i = 0 \qquad 2 \le i \le n \tag{3}$$

and for P_1:

$$P_1 = a_{11} + \sum_{j=2}^{n} a_{1j}\bar{E}_j$$

The P's are to be considered as parameters and the a_{ij} as constant parameters.

3 Equations in z and $\bar{z}$.

The simplest system (3) for $n = 2$ has the following form:

$$az\bar{z} + bz + c = 0 \tag{4}$$

with $a \neq 0$. This equation is quite simple to solve, but its discussion is illustrative. The idea is to take its complex conjugate and consider both equations, and then treat z and $\bar{z}$ as independent variables:

$$\begin{aligned} a z_1 z_2 + b z_1 + c &= 0 \\ \bar{a} z_1 z_2 + \bar{b} z_2 + \bar{c} &= 0 \end{aligned} \tag{5}$$

Solutions of (4) produce always solutions of (5). In the other direction, solutions of (5) will only be solutions of (4) if they verify the conjugacy condition $z_1 = \bar{z_2}$.

A simple computation leads to the following discussion:

- Case $b = 0$
  - Subcase c/a is not real. There is no solution.
  - Subcase c/a is real: The system (5) has infinitely many solutions. For all $z_1 \neq 0$ we have $z_2 = -c/az_1$. These solutions only provide solutions of (4) if $c/a < 0$. Then the solution of (4) is:

    $$z = \sqrt{-\frac{c}{a}}\, e^{i\varphi}$$

    for all φ.
- Case $b \neq 0$. Let us denote

  $$d = (\bar{a}c + a\bar{c} - |b|^2)^2 - 4|ac|^2.$$

  Now, system (5) always has 2 solutions, namely:

  $$(z_1, z_2) = (\frac{-b\bar{b} + a\bar{c} - \bar{a}c \pm \sqrt{d}}{2\bar{a}b}, \frac{-b\bar{b} - a\bar{c} + \bar{a}c \pm \sqrt{d}}{2a\bar{b}})$$

  These values are complex conjugates $-$and consequently solutions of (4)$-$ if and only if $d \geq 0$ (d is always real). Therefore, if $d \geq 0$, equation (4) has two solutions (one double solution if $d = 0$):

  $$z = \frac{-b\bar{b} + a\bar{c} - \bar{a}c \pm \sqrt{d}}{2\bar{a}b}$$

  and if $d < 0$ it has none.

This example shows how to handle systems like (2) or (3). We consider at the same time the given equations and their complex conjugates and then E_i and $\bar{E}_i$ are treated as independent variables. A system with double the number of equations and variables is obtained with the possibility of extraneous solutions. But in the case that E_i and $\bar{E}_i$ are complex conjugates then they are solutions of the original system. This generally depends only on a real quantity that must be positive. The advantage of formulating the load-flow problem in terms of complex variables and its complex conjugates instead of real and imaginary parts is that the system already contains equations with eliminated variables and has a greater symmetry.

4 Elimination method

The objective is to determine an algebraic solution of systems of type (2) or (3) for fixed n. We want to transform the system to triangular form, eliminating all the variables except one, and then express the other variables in terms of the last one. There exist different eliminating methods. The best one from an algebraic point of view is the *lex* Gröbner basis. Let I be the ideal generated by the polynomials of the system. The result of computing the *lex* Gröbner basis of I is a set of Gröbner bases for every elimination ideal $I_k = I \cap \mathbb{C}[x_{k+1}, \dots, x_n]$. For zero dimensional ideals, the minimum polynomial in the last variable $-$that is a basis of $I_{n-1}-$ will be obtained. If the solutions are in general position and the ideal is radical, then the shape lemma ensures that all other variables become isolated in terms of the last one. This will be the most common situation for systems of load-flow type with floating point coefficients (reduced to rationals for computations). The difficulty lies in the increase of the complexity of the computation when the number of variables and equations increases.

We can, alternatively, generate polynomials of the different I_k combining pairs of polynomials in I_{k-1}. We can use three methods:

(1) *lex* Gröbner basis of two polynomials.

(2) Dialytic elimination and resultants.

(3) Degree reduction.

Let us consider the computation of the *lex* Gröbner basis with respect to the variables order $x_1 > x_2 > \dots > x_n$ of an ideal $I = < f_1, f_2 >$ generated by two polynomials. By the elimination theorem we know that it contains bases of the different elimination ideals, I_k. So, either it contains a polynomial not involving x_1 or the ideal I does not contain any polynomial with x_1 eliminated. In the last case, the resultant $\mathrm{Res}(f_1, f_2, x_1) = 0$ because it is a polynomial in I_1. This shows that f_1 and f_2 have a common factor of positive degree in x_1 and therefore its g.c.d. is not constant. It contains some factor of positive degree in x_1. Let $p = \text{g.c.d.}(f_1, f_2)$. We will have $f_1 = ph_1$ and $f_2 = ph_2$ and, for the corresponding varieties, $\mathbf{V}(f,g) = \mathbf{V}(p) \cup \mathbf{V}(h_1, h_2)$. Let $\{g_1, \ldots, g_s\}$ be the Gröbner basis of $< h_1, h_2 >$. It must contain some polynomial not involving x_1 because h_1 and h_2 do not have any common factors, and the above reasoning ensures it. It is now seen at once that the Gröbner basis of $< f_1, f_2 >$ is in this case $\{pg_1, \ldots, pg_s\}$.

If the Gröbner basis of $< h_1, h_2 >$ is 1 then I is the principal ideal $I = (p)$ generated by p and the polynomials f_1 and f_2 are not "independent". If the Gröbner basis of $< h_1, h_2 >$ is not 1, we can factor and compute the reduced varieties.

If we choose carefully the polynomials f_1 and f_2 in the load-flow elimination, common factors will not occur and Gröbner-basis will eliminate the principal variable.

The second method is dialytic elimination [2]. This method linearizes a polynomial system introducing the monomials as new variables and adding equations. In most cases dialytic elimination reduces to computing resultants. Although it does not lead necessarily to the elimination ideal basis, in most practical cases the result does not differ from the first method. But the size of the determinants to be computed can constitute a problem.

There is a third method that allows us to successively reduce the degree of the polynomials in one variable. We compute S-polynomials in the principal variable, i.e. a kind of initial step in the Wu-pseudo-division. This is interesting by itself, as it generally produces better numerical conditioning and can suppress extraneous solutions. This process will end when the principal variable is eliminated. Let us see how it works.

Let $f_1, f_2 \subset \mathbb{C}[x_1, x_2, \ldots, x_n]$ be the two polynomials. We consider them as polynomials in $\mathbb{C}[x_2, \ldots, x_n][x_1]$ and carry out a type of division. Let

$$\begin{aligned} f_1 &= x_1^{M_1} \tilde{f}_1(x_2, \ldots, x_n) + \text{ terms in } x_1 \text{ of degree less than } M_1; \\ f_2 &= x_1^{M_2} \tilde{f}_2(x_2, \ldots, x_n) + \text{ terms in } x_1 \text{ of degree less than } M_2. \end{aligned}$$

Let $M_1 \geq M_2$. Then, the polynomial

$$f = f_1 \tilde{f}_2 - x_1^{M_1 - M_2} f_2 \tilde{f}_1$$

has degree in x_1 less than M_1. Repeating this process with the polynomial of lesser degree in x_1 we can continue reducing the degree until a linear polynomial is obtained and then, finally, eliminating x_1. If f_1 and f_2 do not generate any polynomial not involving x_1, then the process will finish with the polynomial 0. But in the general case, when f_1 and f_2 do not have common factors of positive degree in x_1, we will obtain a polynomial of minimum degree in x_1 and a polynomial not involving x_1. This process approaches the elimination structure given by the "generalized shape lemma" [1], and we shall refer to it as the *degree reduction method.*

We shall combine the three techniques to solve the load-flow problem. With their help we find an algebraic solution for the complete 3-nodes problem and for the not-matching 4-nodes problem. We also find the numerical conditioning both of the problem and of the algorithm.

5 The 4-nodes network.

For the complete not-matching 4-nodes network, the polynomial ideal is

$$\begin{aligned} I = \quad & < a_{11} + a_{12}\,\bar{E}_2 + a_{13}\,\bar{E}_3 + a_{14}\,\bar{E}_4 - P_1, \\ & \bar{a}_{11} + \bar{a}_{12}\,E_2 + \bar{a}_{13}\,E_3 + \bar{a}_{14}\,E_4 - \bar{P}_1, \\ & a_{12}\,E_2 + a_{22}\,E_2\,\bar{E}_2 + a_{23}\,E_2\,\bar{E}_3 + a_{24}\,E_2\,\bar{E}_4 - P_2, \\ & \bar{a}_{12}\,\bar{E}_2 + \bar{a}_{22}\,\bar{E}_2\,E_2 + \bar{a}_{23}\,\bar{E}_2\,E_3 + \bar{a}_{24}\,\bar{E}_2\,E_4 - \bar{P}_2, \\ & a_{13}\,E_3 + a_{23}\,E_3\,\bar{E}_2 + a_{33}\,E_3\,\bar{E}_3 + a_{34}\,E_3\,\bar{E}_4 - P_3, \\ & \bar{a}_{13}\,\bar{E}_3 + \bar{a}_{23}\,\bar{E}_3\,E_2 + \bar{a}_{33}\,\bar{E}_3\,E_3 + \bar{a}_{34}\,\bar{E}_3\,E_4 - \bar{P}_3 > \end{aligned} \tag{6}$$

and the formula expressing the power P_4 in terms of the solution is

$$P_4 = a_{14}\,E_4 + a_{24}\,E_4\,\bar{E}_2 + a_{34}\,E_4\,\bar{E}_3 + a_{44}\,E_4\,\bar{E}_4 - P_3$$

Remember that we take E_i and $\bar{E}_i$ as independent variables. We set $\mathbf{a} = \{a_{ij}, \bar{a}_{ij} : i \leq j,\ 1 \leq j \leq 4\}$, and $\mathbf{P} = \{P_i, \bar{P}_i : \ 1 \leq i \leq 4\}$. With this notation, we consider I as an ideal in $\mathbb{C}(\mathbf{a}, \mathbf{P})[\bar{E}_4, \bar{E}_3, \bar{E}_2, E_4, E_3, E_2]$ or, fixing $\mathbf{a}$, an ideal in $\mathbb{C}(\mathbf{P})[\bar{E}_4, \bar{E}_3, \bar{E}_2, E_4, E_3, E_2]$. This corresponds to fixing the impedances for a particular network. If we want to ensure that a general solution will continue to be a solution for any specialization of the parameters, the correct algebraic tool is the C.G.B. [7]. This algorithm indeed gives all of the conditions giving rise to cancelations for which a particular reduced Gröbner basis must be considered. We used such basis in [6] for the 3-nodes network and they allowed us to simplify our results. In practice it is generally sufficient to consider the parameters as simple constants not satisfying any other cancelation condition. Also, it is desirable not to fix $\mathbf{a}$ for as long as possible, so that the results are valid for any 4-nodes network.

Our strategy consists of eliminating variables considering $\mathbf{a}$ as a parameter so long as the formulas do not become too complex, and then fixing the impedances. We leave the use of C.G.B. for further computations, where they will provide possible simplifications of results.

Let us now see in detail the elimination method. For this purpose, we consider the variables order $\bar{E}_4 > \bar{E}_3 > \bar{E}_2 > E_4 > E_3 > E_2$. This order has the advantage of eliminating at once the $\bar{E}$'s variables whose only interest is to verify the conjugacy condition. This will result in a better numerical conditioning of the algorithm as it will only produce three linked variables, namely E_2, E_3, E_4. Moreover, if we are able to linearize all variables except E_2, we expect to obtain the best numerical condition numbers. A different strategy will consist in taking the order $\bar{E}_4 > E_4 > \bar{E}_3 > E_3 > \bar{E}_2 > E_2$. This order has the advantage that we can immediately recognize and eliminate the extraneous solutions not verifying the conjugacy condition. But this will result in a worse numerical condition numbers in addition to more complex computations, because we start with less polynomials in the elimination ideals. Consequently we choose the first strategy.

Denote by $I = < f_1, f_2, f_3, f_4, f_5, f_6 >$ the polynomials in (6). Using the given variables order we observe that $f_2 \in I_3$, $f_4 \in I_2$ and $f_6 \in I_1$. So we already have one polynomial in I_3, two in I_2 and three in I_1. We reasonably need two more "independent" polynomials in each of the ideals that will allow us to compute two polynomials in I_4 and one in I_5. This is our target.

We start with the computation of the Gröbner bases of $< f_1, f_3 >$ and of $< f_1, f_5 >$. Each basis contain one new and independent polynomial in I_1, say g_{11} and g_{22}. We choose f_1 in both computations because it is a linear polynomial. In Table 1 we show the scheme of eliminations, indicating the polynomials and the method used at each stage.

At level I_4, the complexity of the polynomials do not recommend us to continue without fixing $\mathbf{a}$. The polynomials g_{41} and g_{42} of I_4 have both 90 terms and degree two in E_3. Using method 3 we obtain g_{43} which is linear in E_3 but has 1712 terms. If we replace $\mathbf{a}$ by concrete values, then the number of terms reduces respectively to 33, 33 and 71.

We obtain g_5 as the resultant of g_{41} and g_{42}. It has 523 terms and degree 7 in E_2 and it does not have an independent term, providing the trivial solution $E_2 = 0$. This solution does not extend to a complete solution verifying the conjugacy condition. Consequently we obtain 6 solutions for E_2.

Elimination ideal	Basic polynomials			Other polynomials
I	f_1 (5,?)	f_3 (5,?)	f_5 (5,?)	f_6, f_4, f_2
method		$(f_1, f_3)_1$	$(f_1, f_5)_1$	
I_1	f_6 (5,?)	g_{11} (8,?)	g_{12} (8,?)	f_4, f_2
method		$(f_6, g_{11})_1$	$(g_{11}, g_{12})_1$	
I_2	f_4 (5,?)	g_{21} (26,?)	g_{22} (18,?)	f_2
method		$(f_4, g_{22})_1$	$(g_{21}, g_{22})_1$	
I_3	f_2 (5,?)	g_{31} (54,?)	g_{32} (54,?)	
method		$(f_2, g_{31})_2$	$(f_2, g_{32})_2$	$(g_{41}, g_{42})_3$
I_4		g_{41} (90,33)	g_{42} (90,33)	g_{43} (1712,71)
method			$(g_{41}, g_{42})_2$	$(g_{41}, g_{43})_3$
I_5			g_5 (?,523)	g_{51}

Table 1: Elimination scheme. Numbers in parenthesis represent number of terms not fixing $\mathbf{a}$ and fixing $\mathbf{a}$.

The triangular system obtained is formed by f_1, f_3, f_5 in I, f_6 in I_1, f_4 in I_2, f_2 in I_3, g_{41}, g_{42}, g_{43} in I_4 and g_5 in I_5.

As we have linear polynomials in the diagonal variable at all other elimination levels, the maximum number of solutions is 6. We just need one polynomial in each elimination ideal with non-zero diagonal coefficient to obtain the complete solution. Nevertheless we have to check the solution in all the equations of (6) not yet used, and also the conjugacy condition.

For concrete particular values, from the six solutions, only four verify the conjugacy condition, and only one verifies the technical requirements that the E's have values near 1 and the phases of E's and P's are not very high.

We note that if some lines in the network are missing, then the corresponding coefficients $a_{ij}, \bar{a}_{ij}$ are zero. In this case, even if the result would be simpler, we are not allowed to substitute $a_{ij} = \bar{a}_{ij} = 0$ in the elimination formulas. We must recompute the whole elimination method. The reason is that some leading terms of the polynomials involved can be missing, and the computations can fail. Automatic detection of these cases could be done using C.G.B.

6 Software design

Software designed in *Maple* uses the general elimination formulas described above in a routine LF4 having as input the network impedances. The output is a new routine, specific for the particular network. The routine LF4 is executed only once for a given network configuration, and it generates as output a new routine with an assigned name, containing the triangular system associated to the configuration. The output routine has as input P_1, P_2, P_3 and as output E_2, E_3, E_4 and P_4 and its only task is to solve the triangular system. As an option, this routine also outputs the condition numbers of the problem and of the algorithm. The principal work is performed by the LF4 routine which is to be executed only once. In any new simulation, only the generated routine, with the simple task, is to be used.

Let us now work out the numerical conditioning that has also been incorporated in the software.

7 Condition numbers of the problem and the algorithm

Let us separate real and imaginary parts of the powers and put $P_i = P_{ix} + I * P_{iy}$. The condition numbers of the problem are essentially given by the Jacobian matrix

$$J = \frac{\partial(E_2, E_3, E_4, P_4)}{\partial(P_{1x}, P_{2x}, P_{3x}, P_{1y}, P_{2y}, P_{3y})}$$

measuring the output variations with respect to the input parameters. (If we strictly want condition numbers we take relative variations and absolute values). The Jacobian can easily be computed by implicit differentiation on the triangular system and isolation of the corresponding terms:

$$\frac{\partial g_5}{\partial E_2}\frac{\partial E_2}{\partial P_{ix}} + \frac{\partial g_5}{\partial P_i} + \frac{\partial g_5}{\partial \bar{P}_i} = 0$$
$$\frac{\partial g_4}{\partial E_2}\frac{\partial E_2}{\partial P_{ix}} + \frac{\partial g_4}{\partial E_3}\frac{\partial E_3}{\partial P_{ix}} + \frac{\partial g_4}{\partial P_i} + \frac{\partial g_4}{\partial \bar{P}_i} = 0$$
$$\frac{\partial g_3}{\partial E_2}\frac{\partial E_2}{\partial P_{ix}} + \frac{\partial g_3}{\partial E_3}\frac{\partial E_3}{\partial P_{ix}} + \frac{\partial g_3}{\partial E_4}\frac{\partial E_4}{\partial P_{ix}} + \frac{\partial g_3}{\partial P_i} + \frac{\partial g_3}{\partial \bar{P}_i} = 0$$

$$\frac{\partial P_4}{\partial P_{ix}} = \frac{\partial P_4}{\partial E_4}\frac{\partial E_4}{\partial P_{ix}} + \frac{\partial P_4}{\partial \bar{E}_2}(\overline{\frac{\partial E_2}{\partial P_{ix}}}) + \frac{\partial P_4}{\partial \bar{E}_3}(\overline{\frac{\partial E_3}{\partial P_{ix}}}) + \frac{\partial P_4}{\partial \bar{E}_4}(\overline{\frac{\partial E_4}{\partial P_{ix}}})$$

The last formula assumes the validity of the conjugacy condition. Similar formulas are valid for the derivatives with respect to P_{iy}.

The numerical conditioning of the algorithm can also be computed determining upper bounds for the condition numbers of E_2, E_3, E_4 and P_4 from the triangular system. Condition numbers in a triangular system are linked one to another and the condition number of each new computed variable depends on the previous ones. For this reason, triangulation is not the best numerically conditioned method as condition numbers become amplified. Nevertheless, as we have been able to linearize in the diagonal variables we hope that the condition numbers will not increase much. The formula to compute an upper bound for the condition number [4] of a root x_n of a polynomial $P(x_1, \ldots, x_n) = \sum p_i$ where the $x_1, \ldots, x_{n-1}$ have already been numerically computed and have respectively condition numbers $C_1, \ldots, C_{n-1}$ is

$$\frac{1}{\epsilon}\left|\frac{\triangle x_n}{x_n}\right| \leq C_n \leq$$
$$\frac{\sum_i \left\{ |\, p_i \,| \left[1 + 2\deg(p_i) + \sum_{k=1}^{n-1} \deg(p_i, x_k)\, C_k \right] \right\}}{|\sum_i \deg(p_i, x_n) p_i|(1 - 0.005n)}$$

being ϵ the floating point precision we are using. These formulas can be included in the generated network routine.

8 Simulation

We consider the following network impedances for the simulation:

$$\begin{array}{llllll}
c_{12} = 150, & s_{12} = 10, & b_{12} = 1, & c_{13} = 200, & s_{13} = 12, & b_{13} = 1 \\
c_{14} = 200, & s_{14} = 15, & b_{14} = 2, & c_{23} = 300, & s_{23} = 18, & b_{23} = 3 \\
c_{24} = 100, & s_{24} = 8, & b_{24} = 1, & c_{34} = 400, & s_{34} = 20, & b_{34} = 3
\end{array}$$

We generate the network routine and begin the simulation with an example. The results are given in Table 2.

Input		Output	
$P_1 =$	$-9.9499 - 3.4454I$	$E_2 =$	$0.969846 - 0.000754I$
$P_2 =$	$-29.0819 - 4.4860I$	$E_3 =$	$1.030769 - 0.001731I$
$P_3 =$	$20.5736 - 1.6976I$	$E_4 =$	$1.041846 - 0.001454I$
		$P_4 =$	$20.816604 - 1.540049I$

Table 2: Simulation

The corresponding values of the Jacobian are

$$1000 \left(\frac{\partial(E_2, E_3, E_4)}{\partial(P_{1x}, P_{2x}, P_{3x}, P_{1y}, P_{2y}, P_{3y})} \right)^t =$$

$$\begin{pmatrix} -1.3466 - .0709I & -1.5088 - .1025I & -2.4576 - .1864I \\ 1.7293 + .0863I & .0008 - .0074I & -2.0284 - .1178I \\ .2470 + .0096I & .7340 + .0315I & -.9177 - .0511I \\ -.0920 + 1.2337I & -.1057 + 1.5546I & -.1675 + 2.4967I \\ .1116 - 1.5663I & -.0065 - .0009I & -.1380 + 1.9282I \\ .0135 - .2304I & .0334 - .7562I & -.0562 + .9275I \end{pmatrix}$$

$$\left(\frac{\partial P_4}{\partial(P_{1x}, P_{2x}, P_{3x}, P_{1y}, P_{2y}, P_{3y})} \right)^t = \begin{pmatrix} -1.07764 + .027348I \\ -1.70594 - .004203I \\ -1.02219 - .002522I \\ -.0083697 - .99828I \\ -.0183311 - 1.53753I \\ -.0011703 - 1.00007I \end{pmatrix}$$

We observe that the E's vary slightly as functions of the P's, so that the voltages are in a stable domain that corresponds well to physical acceptable values. The derivative of P_4 with respect to the real (imaginary) parts of the P's is near -1 (-I) in accord with conservation of power and little power consumed by the network. (In this case the sum of all powers must be near zero, and the powers not involved are fixed.)

The computed condition numbers are

$$C_{E_2} = 101; \quad C_{E_3} = 211; \quad C_{E_4} = 291; \quad C_{P_4} = 36392$$

The condition number $C_{P_4} = 36392$ implies the loss of five significant digits in the computation of P_4. Even though we give upper bounds for the precision loss, these bounds are not very large, as we can easily work with ten digits to obtain five correct digits. Nevertheless this increase of the condition numbers can become a problem for wider networks.

Let us carry out a new simulation to show the usefulness of the Jacobian. We can compare how the use of the Jacobian in first order expansions is sufficient for the predictions of the output parameters. Suppose that after some time, the state variables have changed from the initial values given in Table 2, to the values given in Table 3, E_1 still continuing to be the reference voltage 1. The result of applying the routine with the new input parameters P_1', P_2', P_3' is given in Table 3.

Input		Output	
$P_1' =$	$-4.81810 - 3.107351I$	$E_2 =$	$0.962420 - 0.000750I$
$P_2' =$	$-29.07828 - 4.45128I$	$E_3 =$	$1.021530 - 0.001716I$
$P_3' =$	$18.60430 - 1.79946I$	$E_4 =$	$1.031000 - 0.001520I$
		$P_4 =$	$17.344727 - 1.662258I$

Table 3: Simulation varying P_1, P_2, P_3

We can compare the output parameters with first order expansions using the Jacobian. We have

$$\begin{array}{lcr}
E_2' \approx E_2 + \sum_{i=1}^{3} \left(\frac{\partial E_2}{\partial P_{ix}} \triangle P_{ix} + I \, \frac{\partial E_2}{\partial P_{iy}} \triangle P_{iy} \right) & = & 0.962427 - 0.000750I \\
E_3' \approx E_3 + \sum_{i=1}^{3} \left(\frac{\partial E_3}{\partial P_{ix}} \triangle P_{ix} + I \, \frac{\partial E_3}{\partial P_{iy}} \triangle P_{iy} \right) & = & 1.021542 - 0.001717I \\
E_4' \approx E_4 + \sum_{i=1}^{3} \left(\frac{\partial E_4}{\partial P_{ix}} \triangle P_{ix} + I \, \frac{\partial E_4}{\partial P_{iy}} \triangle P_{iy} \right) & = & 1.030979 - 0.001494I \\
P_4' \approx P_4 + \sum_{i=1}^{3} \left(\frac{\partial P_4}{\partial P_{ix}} \triangle P_{ix} + I \, \frac{\partial P_4}{\partial P_{iy}} \triangle P_{iy} \right) & = & 17.289834 - 1.683683I
\end{array}$$

There is a high coincidence in the voltages and a deviation of order 1 % in P_4. So triangulation allows us to know the local variation coefficients without new heavy computations. This can be of great practical interest for the simulations as it supplies a local linear expression to compute corrections.

References

[1] M.E. Alonso, E. Becker, M.F. Roy, T. Wormann, "Zeros, Multiplicities and Idempotents for Zerodimensional Systems", Algorithms in Algebraic Geometry and Applications, Progress in Mathematics, Birkhauser, 1996.

[2] O. Bottema, B. Roth, "Theoretical kinematics", Dover Publications, Inc., 1990.

[3] Ch. Gross, "Power System Analysis", John Wiley & Sons, 1986.

[4] A. Montes, "Numerical Conditioning of a System of Algebraic Equations with a Finite Number of Solutions using Gröbner Basis", *SIGSAM Bull.*, **27**, num. 1, January 1993.

[5] A. Montes, J. Castro, "Solving the Load-Flow Problem Using the Gröbner Basis", *SIGSAM Bull.*, **29**, num 1, January 1995.

[6] A. Montes, "Bases de Gröbner comprehensivas y el problema del transporte eléctrico", *Actas del Primer Encuentro de Algebra Computacional y Aplicaciones*, Ed. L. González-Vega, Universidad de Cantabria, 1995.

[7] V. Weisspfenning, "Comprehensive Gröbner Basis", *Jour. Symb. Comp.*, **14**/1, 1-29, 1992.

Aknowledgements

I express my gratitude to professor N. Nabona, at the Universitat Politècnica de Catalunya, for his useful comments and his patience discussing the load-flow problem with me. I am also indebted to professor V. Weisspfenning, at Passau Universität, and his student M. Pesch for their interest on this problem and their suggestion of using C.G.B.