Solving the load flow problem using the Gröbner basis

Antonio Montes
Dep. of Appl. Math 2

Jordi Castro
Dep. of Stat. & Op. Research

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

Abstract

In Electrical Engineering one of the most important problems to be solved for electrical networks is the load flow problem [6] [3]. Currently numerical solutions are provided by Newton's method, which involves recomputing the solution whenever the input data change. Given that this problem must be solved very often with different input data, the Gröbner basis can be an interesting approach since it can, in principle, provide a more algebraic solution of the input parameters and has to be solved completely only once, supplying formulas to compute single simulations. In this paper the basic ideas and a practical example are reported. Nevertheless, the required computations for practical network sizes are very complex and cannot yet be solved with the present algorithms.

1 General Formulation.

An electrical network can be represented by a graph. At each vertex k there is a global injected power (P_k, Q_k) (real and reactive part), which is the difference between the injected power arising from the generators connected to this vertex and the total outgoing charges. Two other pairs of variables associated to the vertex k are the real and imaginary part of the voltage (e_k, f_k). They are the state-variables of the network. A line connecting two vertices $k - l$ is characterized by constant values of its conductance and susceptance (c_{kl}, s_{kl}) (complex inverse of the impedance) and the susceptance with earth b_{kl}. It is of maximum interest in Electrical Engineering, to maintain the state-variables in a narrow tolerance band around the basic voltage that is taken to be (1,0), using adequate units. If the voltages go out of the band, the electrical engines connected to it could be damaged or not work correctly. In order to maintain the state of the network, it is necessary to connect or disconnect generators. The objective of the load flow problem is to analyze the state of the voltages (e_k, f_k) when the input charges vary, in order to control them.

The network is described mathematically by a system of quadratic equations relating the above quantities. Let us call $p_{kl} + jq_{kl}$ the complex power flowing from vertex k to vertex l. It depends only on the voltages at vertices k and l and on the parameters c_{kl}, s_{kl}, b_{kl} of line $k - l$,

$$p_{kl} = \frac{1}{2}(e_k, e_l, f_k, f_l) \begin{pmatrix} 2c & -c & 0 & -s \\ -c & 0 & s & 0 \\ 0 & s & 2c & -c \\ -s & 0 & -c & 0 \end{pmatrix} \begin{pmatrix} e_k \\ e_l \\ f_k \\ f_l \end{pmatrix}$$

$$q_{kl} = \frac{1}{2}(e_k, e_l, f_k, f_l) \begin{pmatrix} 2s-b & -s & 0 & c \\ -s & 0 & -c & 0 \\ 0 & -c & 2s-b & -s \\ c & 0 & -s & 0 \end{pmatrix} \begin{pmatrix} e_k \\ e_l \\ f_k \\ f_l \end{pmatrix} \quad (1)$$

The equations of the network in the so-called bus reference are obtained considering the power balance at the nodes: the total injected power at vertex k must be equal to the sum of power flowing from vertex k to all connected vertices.

$$\begin{aligned} \sum_{l=lines\ at\ vertex\ k} p_{kl} &= P_k \\ \sum_{l=lines\ at\ vertex\ k} q_{kl} &= Q_k \end{aligned} \quad (2)$$

A network with n vertices gives rise to a system of $2n$ quadratic equations in the $2n$ state-variables $e_1, f_1, e_2, f_2, \ldots, e_n, f_n$, having as second member the $2n$ input-parameters $P_1, Q_1, P_2, Q_2, \ldots, P_n, Q_n$. If we intend to solve this system for arbitrary given parameters (P_k, Q_k), $k = 1 \div n$, it will not be compatible. The input powers cannot all be fixed. This is natural from a physical point of view, because total energy must be conserved and the amount of energy used by the lines in the network depends on the state-variables. But the load flow problem has a different formulation.

At the dispatching controlling the network, all quantities (e_k, p_k) and (P_k, Q_k), $k = 1 \div n$, are known at any instant, and they vary with time as functions of the charge demands connected to the nodes. Let us suppose that the voltages are observed to be wrong at some vertex, decreasing or increasing too much. In this case, an order must be given to connect, disconnect or vary the power production at some generator.

Before giving the order, numerical simulations of the effect of proposed corrections should be done. A simulation is performed in the following way. Suppose we try to correct the voltage at node 1 by acting on the injected power at another vertex (perhaps the same). So we set vertex 1 to the base voltage $e_1 = 1$, $f_1 = 0$ in equations (2) and leave free the equations of the vertex where we prevent modification of power production, say vertex n. The system to be solved now will consist of a set of $2(n-1)$ quadratic equations in the state-variables $e_2, f_2, \ldots, e_n, f_n$ whose second members are the input-parameters $P_1, Q_1, \ldots, P_{n-1}, Q_{n-1}$. This system is now expected to be compatible, as we have left the power injection at vertex n free in order to allow conservation of energy. The equations to be solved for the simulation will be

$$\begin{array}{rcll} F_k(1,0,e_2,f_2,\ldots,e_n,f_n) & = & P_k & \\ G_k(1,0,e_2,f_2,\ldots,e_n,f_n) & = & Q_k & \quad k = 1 \div n-1 \end{array} \tag{3}$$

For any solution of this system the values of P_n, Q_n will be given by the two equations in (2) that have been already separated,

$$\begin{array}{rcl} P_n & = & F_n(1,0,e_2,f_2,\ldots,e_n,f_n) \\ Q_n & = & G_n(1,0,e_2,f_2,\ldots,e_n,f_n) \end{array} \tag{4}$$

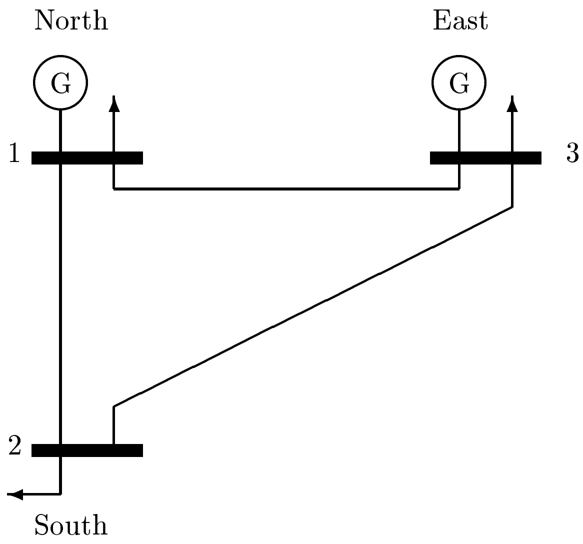
This kind of system must be solved many times for different values of the inputs (P_k, Q_k)'s. There are very efficient approaches for solving these equations using the Newton-Raphson method with sparsity techniques [7]. Out of the multiple solutions to the system of nonlinear equations only those close to $e_i = 1, f_i = 0$, for $i = 2 \div n$ are physically relevant, so this point is the usual starting point for iterative methods.

Nevertheless, it would be of interest if more algebraic solutions could be found, avoiding recomputation of the whole numerical solution for each simple variation of an input. In principle, this can be done by computing the Gröbner basis [1] in lex-order of the state-variables, taking the input variables $P_1, Q_1, \ldots, P_{n-1}, Q_{n-1}$ as parameters. As we are dealing with a zero-dimensional ideal, the resulting basis will be in triangular form [4], and it will allow computation of all the state variables by back substitution. Substituting them in (4) provides an explicit formula for the new total power input needed at vertex n to produce the solution.

In computing the Gröbner basis, the order of the variables is instrumental, and the complexity of the triangularized basis depends on a good choice. A presumable good order choice can be obtained following a Hamiltonian cycle in the network if it exists. Let us consider as an example a three vertex network.

2 Three vertex network.

Consider the triangle graph network in the picture



with line-constants given below

line	b	c	s
1-2	0.05	12.0	110.0
1-3	0.25	2.0	10.0
2-3	0.05	4.0	20.0

Line conductances

2.1 The Gröbner basis

In order to compute the algebraic solution it is necessary to convert the line parameters c, s, b to rationals, as it is pointless to compute the Gröbner basis in floating point arithmetic. We write equations (2) for vertex $1, 2$ as a list of polynomials:

$$F := [p_{12} + p_{13} - P_1, \quad q_{12} + q_{13} - Q_1, \quad p_{23} + p_{21} - P_2, \quad q_{23} + q_{21} - Q_2] \tag{5}$$

using (1) with $e_1 = 1, f_1 = 0$, and we reserve the equations of vertex 3, to evaluate P_3 and Q_3 once the solution has been obtained.

$$P_3 = p_{31} + p_{32} \equiv P_3(e_2, f_2, e_3, f_3), \quad Q_3 = q_{31} + q_{32} \equiv Q_3(e_2, f_2, e_3, f_3) \tag{6}$$

Now, we compute the Gröbner basis of F with respect to lex order with $e_2 > f_2 > e_3 > f_3$. We use *Maple* for this purpose. The result has the following form:

$$\begin{aligned} A\, f_3^2 + B\, f_3 + C &= 0 \\ U\, e_3 + V &= 0 \\ R\, f_2 + S &= 0 \\ M\, e_2 + N &= 0 \end{aligned} \tag{7}$$

where $A, B, C, U, V, R, S, M, N$ are polynomials with integer coefficients in the variables indicated below:

functions	depend on
A, B, C	P_1, Q_1, P_2, Q_2
V, N	P_1, Q_1, P_2, Q_2, f_3
U, M	P_1, Q_1
S	P_1, Q_1, f_3
R	constant

Base (7) is not only completely triangularized, as expected, but it has another very nice property: only one equation is quadratic in its main-variable, namely the first one, whereas the rest are linear. In this simple network the solutions can be completely expressed in explicit form using only one radical, namely

$$D = \sqrt{B^2 - 4AC}$$

Nevertheles the coefficients of the polynomial functions are big. Let us give the computed expressions of the coefficient functions obtained with *Maple*:

$$\begin{aligned}
A \ := \ & 664986038021686534400 - 858807722905600000\,P_1 \\
& -75081265175022080000\,Q_1 + 21470193072640000\,P_1^2 \\
& +21470193072640000\,Q_1^2
\end{aligned}$$

$$\begin{aligned}
B \ := \ & 869275603729979680000\,P_1 - 24522567908093923200\,Q_1 \\
& +40573425829794624000\,P_2 - 463762546990080000\,Q_2 \\
& +636603379775444583840 - 92772073474496000\,P_1^2 \\
& -968364090079360000\,Q_1\,P_1 + 216276151096512000\,Q_1^2 \\
& +5321804800000000\,P_1\,Q_2 + 2833147116800000\,P_1\,Q_1^2 \\
& +25914723328000000\,Q_1\,Q_2 - 559686416640000\,Q_1\,P_1^2 \\
& +2833147116800000\,P_1^3 - 559686416640000\,Q_1^3 \\
& -46559139744000000\,P_1\,P_2 - 226721435715840000\,Q_1\,P_2
\end{aligned}$$

$$\begin{aligned}
C \ := \ & -142597737984000000\,P_2\,Q_2 + 2922116889920000\,P_2\,Q_1^2 \\
& +623775980093760000\,P_2^2 - 3853890280000000\,P_2\,P_1^2 \\
& +22740058173551911400\,P_1 - 4332028461568937840\,Q_1 \\
& +19222978969681623200\,P_2 - 3950912494201440000\,Q_2 \\
& +739139880000000\,P_1^4 - 851326041360604919 \\
& -14030739078400000\,P_2\,Q_1\,P_1 + 8149606400000000\,Q_2^2 \\
& +225990854829153360\,P_1^2 - 1146174508352812000\,Q_1\,P_1 \\
& +136429144013221600\,Q_1^2 - 521086912224000000\,P_1\,Q_2 \\
& +10726873974320000\,P_1\,Q_1^2 + 117547559980800000\,Q_1\,Q_2 \\
& -24831978835552000\,Q_1\,P_1^2 + 4450318720000000\,P_1\,Q_1\,Q_2 \\
& -2222393979760000\,P_1^3 - 1127394311904000\,Q_1^3 \\
& +262981454085824000\,P_1\,P_2 - 632248804589024000\,Q_1\,P_2 \\
& -6490246899200000\,P_1^2\,Q_2 + 76737028160000\,P_1^2\,Q_1^2 \\
& -28897998400000\,Q_1\,P_1^3 - 28897998400000\,P_1\,Q_1^3 \\
& -626288064000000\,Q_1^2\,Q_2 + 2823040160000\,Q_1^4
\end{aligned}$$

$$
\begin{aligned}
U &:= 12386511500 + 778687600\,P_1 - 159910000\,Q_1 \\
V &:= -401200\,P_1^2 + 4284787800\,P_2 - 12487396027 + 4300835800\,P_1 \\
&\quad +(139351726860 - 159910000\,P_1 - 778687600\,Q_1)\,f_3 \\
&\quad -349460360\,Q_1 - 401200\,Q_1^2 - 489760000\,Q_2
\end{aligned}
$$

$$
\begin{aligned}
R &:= 1238651150 + 77868760\,P_1 - 15991000\,Q_1 \\
S &:= -219980\,Q_1\,P_1 - 99700 - 795600\,Q_1 + 6968003\,P_1 \\
&\quad +400000\,Q_2 + 699900\,P_1^2 + 16000\,Q_1^2 - 3499500\,P_2 \\
&\quad +(-104000 + 7278960\,P_1 - 832000\,Q_1)\,f_3
\end{aligned}
$$

$$
\begin{aligned}
M &:= 24773023000 + 1557375200\,P_1 - 319820000\,Q_1 \\
N &:= 1600000\,P_1^2 - 24721116491 - 2463598300\,P_1 + 635470120\,Q_1 \\
&\quad +89920000\,Q_2 + 13678000\,Q_1\,P_1 - 2799600\,Q_1^2 - 786687600\,P_2 \\
&\quad +(-25787323120 + 16640000\,P_1 + 145579200\,Q_1)\,f_3
\end{aligned}
\tag{8}
$$

For given line-parameters and a choice of test and input vertices, we have to compute only one Gröbner basis. This has the following properties:

1. It provides explicit formulas for computing all output variables, namely f_3, e_3, f_2, e_2 and also P_3, Q_3 (by equation (6)) as functions of the input variables P_1, Q_1, P_2, Q_2.

2. Formulas provided by the Gröbner basis can be introduced in a computer routine, allowing computation of individual simulations of the load-flow problem with explicit formulas.

3. The analytic behavior of the solution can be studied, as we have explicit formulas. In particular, we shall compute the condition numbers of the problem in Subsection 2.2. Taylor developments can also be obtained.

4. For our particular 3-vertex problem, we need only one radical to compute f_3, the rest of voltages being linear in f_3 with rational functions of the parameters as coefficients.

5. For the 3-vertex network there exist also two and only two solutions of the equations for given input parameters, from which only one has a physical sense, as the resulting bus voltages are a long way outside their practical limits for the second solution.

Let us give here, as an example, the two solutions obtained for given input parameters.

P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-6.0	7.0	5.0	.06570788000	1.088343535	.05938500406
				-.5020041160	12.67980999	.01683041719
				e_2	P_3	Q_3
				1.052823641	1.105253801	1.458195082
				-.0159103917	932.5302466	4685.964628

As we can see, only the first one has physical sense.

Table 1 gives different simulations obtained by varying the inputs and using floating point arithmetic in the calculations. We only give there the physical interesting solution. The numerical stability of the formulas will be studied in Subsection 2.3.

2.2 Problem condition numbers

It is important to note that when computing the Gröbner basis, exact rational arithmetic must be used to provide formulas to be introduced in a computer routine, whereas floating point arithmetic can be used to evaluate particular solutions. It is important to discuss numerical conditioning of the algorithm, and also the problem condition numbers.

Let us begin with the study of the problem condition numbers. In mathematical terms, the problem is formulated by the function

$$\begin{array}{cccc} \phi : & \mathbb{R}^4 & \longrightarrow & \mathbb{R}^6 \\ & (P_1, Q_1, P_2, Q_2) & \longmapsto & \phi(e_2, f_2, e_3, f_3, P_3, Q_3) \end{array}$$

The problem condition numbers express how much relative variation of the outputs corresponds to little relative variation of the inputs. For physical reasons, we expect the problem to be well conditioned. Given that now we have explicit formulas for ϕ, we can compute exact formulas for the derivatives and for the condition numbers of the problem. The condition number of y_i relative to x_j is given by $C(y_i; x_j) = |x_j/y_i||\partial y_i/\partial x_j|$, as we have

P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-6.0	7.0	5.0	.06570788000	1.088343535	.05938500406
				e_2	P_3	Q_3
				1.052823641	1.105253801	1.458195082
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-7.9	-6.0	7.0	5.0	.06220654880	1.087548421	.05881452114
				e_2	P_3	Q_3
				1.052770028	1.002878872	1.443001877
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.1	-6.0	7.0	5.0	.06920934470	1.089138714	.05995547412
				e_2	P_3	Q_3
				1.052877248	1.207761393	1.474081044
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-5.9	7.0	5.0	.06627892690	1.084690045	.05946042811
				e_2	P_3	Q_3
				1.052265296	1.102728465	1.344320780
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-6.1	7.0	5.0	.06513669980	1.091996952	.05930959277
				e_2	P_3	Q_3
				1.053381991	1.107919915	1.572801253
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-6.0	7.1	5.0	.06277733675	1.087969521	.05965708279
				e_2	P_3	Q_3
				1.052834041	1.004541119	1.456482264
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-6.0	6.9	5.0	.06863834975	1.088719051	.05911291981
				e_2	P_3	Q_3
				1.052813102	1.206087242	1.460515839
P_1	Q_1	P_2	Q_2	f_3	e_3	f_2
-8.0	-6.0	7.0	5.1	.06618461400	1.085491708	.05936453147
				e_2	P_3	Q_3
				1.053089332	1.103875698	1.353482850

Table 1: Triangle network: set of input and output data

$$| \frac{\triangle y_i}{y_i} | \leq \sum_j | \frac{x_j}{y_i} \frac{\partial y_i}{\partial x_j} || \frac{\triangle x_j}{x_j} |$$

Computing the derivatives, from (7) we obtain

$$f_3 = \frac{-B \pm D}{2\,A}; \qquad \frac{\partial f_3}{\partial A} = -\frac{f_3}{A} \mp \frac{C}{D\,A}; \qquad \frac{\partial f_3}{\partial B} = \mp\frac{f_3}{D}; \qquad \frac{\partial f_3}{\partial C} = \mp\frac{1}{D}$$

Using chain derivation's rule, the above results and formulas (6), (7) and (8), we can compute ϕ'

$$\begin{aligned}
\frac{\partial f_3}{\partial P_1} &= \frac{\partial f_3}{\partial A}\frac{\partial A}{\partial P_1} + \frac{\partial f_3}{\partial B}\frac{\partial B}{\partial P_1} + \frac{\partial f_3}{\partial C}\frac{\partial C}{\partial P_1} \\
\frac{\partial e_3}{\partial P_1} &= (-\frac{1}{M}(\frac{\partial N}{\partial P_1} + \frac{\partial N}{\partial f_3}\frac{\partial f_3}{\partial P_1}) + \frac{N}{M^2}\frac{\partial M}{\partial P_1}) + \\
\frac{\partial f_2}{\partial P_1} &= (-\frac{1}{R}(\frac{\partial S}{\partial P_1} + \frac{\partial S}{\partial f_3}\frac{\partial f_3}{\partial P_1}) + \frac{S}{R^2}\frac{\partial R}{\partial P_1}) + \\
\frac{\partial e_2}{\partial P_1} &= (-\frac{1}{V}(\frac{\partial V}{\partial P_1} + \frac{\partial V}{\partial f_3}\frac{\partial f_3}{\partial P_1}) + \frac{V}{U^2}\frac{\partial U}{\partial P_1}) + \\
\frac{\partial P_3}{\partial P_1} &= \frac{\partial P_3}{\partial e_2}\frac{\partial e_3}{\partial P_1} + \frac{\partial P_3}{\partial f_2}\frac{\partial f_2}{\partial P_1} + \frac{\partial P_3}{\partial e_3}\frac{\partial e_2}{\partial P_1} + \frac{\partial P_3}{\partial f_3}\frac{\partial f_3}{\partial P_1}
\end{aligned}$$

We are able to obtain exact computable expressions for all condition numbers. We do not give explicit formulas here as they are very complex. Instead, a computer routine evaluates them as a function of the input parameters, giving the following results for a set of values:

$$C(f_3, e_3, f_2, e_2, P_3, Q_3;\ P_1, Q_1, P_2, Q_2)(-8.0, -6.0, 7.0, 5.0) =$$

$$= \begin{pmatrix}
4.262986746 & 0.5215022755 & 3.121930607 & 0.3627943229 \\
0.058448443 & 0.2014135808 & 0.024104358 & 0.1310496144 \\
0.768512496 & 0.0761986826 & 0.320715742 & 0.0172349622 \\
0.004073558 & 0.0318200059 & 0.000696100 & 0.0126211668 \\
7.414859137 & 0.1409119321 & 6.382348828 & 0.0649579717 \\
0.852539312 & 4.7006137320 & 0.096820290 & 3.6004484280
\end{pmatrix}$$

These values should be compared with incremental quotients obtained from Table 1. For example

$$C(P_3;\ P_1)(-8.0, -6.0, 7.0, 5.0) = 7.414859137$$

is compared to the following quotient extracted from Table 1

$$|\frac{8.0}{1.105253801}\frac{1.002878872-1.207761393}{-7.9-(-8.1)}|=7.414858770$$

The agreement is excellent. Observing the condition number matrix we see that the problem is very well conditioned, since the coefficients are small.

2.3 Algorithm's numerical conditioning

Let us now discuss the numerical conditioning of the algorithm. From a numerical point of view it seems to be an ill-conditioned algorithm since the coefficients of the involved polynomials are very big. But this is only apparent, as we shall see.

For the purpose of computing the condition numbers of the root the general formulas described in [5] will be used

Applying them to the roots of the first equation (7) in order to compute f_3, we obtain

$$\begin{aligned}\frac{1}{\epsilon}|\frac{\triangle f_3}{f_3}| \le C_{f_3} = & \qquad (9)\\ = \frac{|\, Af_3^2 \,|(C_A+5)+|\, Bf_3 \,|(C_B+3)+|\, C \,|(C_C+1)}{|\; 2Af_3^2+Bf_3 \;|}1.006\end{aligned}$$

where C_A, C_B, C_C are the condition numbers for computing A, B, C as functions of the variables P_1, Q_1, P_2, Q_2. If we use rational arithmetic to compute A, B, C they are zero, but using floating point arithmetic these quantities can be evaluated following [5].

$$\frac{1}{\epsilon}|\frac{\triangle A}{A}| \le C_A = \frac{\sum_i |\,\mathrm{term}_i(A)\,|(2\ \deg(\mathrm{term}_i(A))+1)}{|\,A\,|}1.006 \qquad (10)$$

Upon substitution in (9) we compute the condition number C_{f_3}. Using it, we can compute again the condition numbers for e_3, f_2, e_2, P_3, Q_3. Results are given in Table 2.

The biggest condition number occurs for Q_3, which is of the order 10^4, so that 4 digits can be lost. This is quite acceptable as condition numbers have been overestimated.

$P_1 = -8.0$	$Q_1 = -6.0$	$P_2 = 7.0$	$Q_2 = 5.0$		
f_3 (C_{f_3})	e_3 (C_{e_3})	f_2 (C_{f_2})	e_2 (C_{e_2})	P_3 (C_{P_3})	Q_3 (C_{Q_3})
.06570788 (145.4)	1.0883435 (223.0)	.05938500 (38.11)	1.0528236 (35.77)	1.1052538 (5112.)	1.4581951 (17003.)
-.50200412 (36.07)	12.679810 (44.62)	.01683042 (165.4)	-.01591039 (2871.)	932.5302466 (103.7)	4685.964628 (101.1)

Table 2: Algorithm condition numbers using f.p.a.

3 Conclusions

We can conclude that the Gröbner basis approach can provide interesting analytical results that cannot be obtained by current numerical methods. Practical general formulas depending on parameters can be obtained that allow the computation of particular solutions and condition numbers for the algorithm and the problem. Condition numbers are not very high. However, we are still far from being able to obtain results for networks of a practical size.

References

[1] D. Cox, J. Little, D. O'Shea, "Ideals, Varieties and Algorithms", Springer-Verlag, 1991.

[2] L.L. Freris and A.M. Sasson, Investigation on the load- flow problem, *Proc. IEE (London)*, **115**, (1968), 1459-1470.

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

[4] D. Lazard, Solving Zero-dimensional Algebraic Systems, *Jour. Symb. Comp.*, **13**, (1992), 117-131.

[5] A. Montes, Numerical conditioning of zero dimensional Gröbner bases, *SIGSAM Bull.*, **27**, 1, (1993) 12-19.

[6] G.W. Stagg, "Computer Methods in Power System Analysis", Mc. Graw-Hill, 1968.

[7] W.P. Tinney and C.E. Hart, Power flow solution by Newton's method, *IEEE Trans on PAS*, **86**, (1967) 1449-1460.