

A reduced gradient method with variable base using second order information, applied to the Constrained- and Optimal Power Flow

Etienne Jamouille, *Senior Member, IEEE*, and Gauthier Dupont, *Member, IEEE*

Abstract-- This paper briefly analyses the history of the constrained and optimal power flow solutions as found in the literature along two of its main development channels, i.e. the fixed base and penalty factors approach and the variable base or Generalized Reduced Gradient (GRG) approach. This analysis serves as an introduction to the very objective of the paper, which is to show that GRG is perfectly suited to accommodate second order information allowing for the development of a very stable and efficient solution algorithm. This algorithm is presented and commented. A didactic case study is used to illustrate its particularities and efficiency. The mathematical development and a demonstration supporting the method are to be found in appendices.

Index Terms--Constrained power flow (CPF), optimal power flow (OPF), reduced gradient method.

I. INTRODUCTION

THE analysis of the power flows in an electrical network is an essential component of the expansion and operation planning studies of a transmission and distribution system.

This explains the numerous publications on the subject describing, from the beginning of the computer age, the evolution of the mathematical and computer techniques used in this issue.

From this point of view, one may distinguish in the power flow analysis three sub problems of growing complexity:

- The solution to the set of non-linear power flow equations (The Power Flow problem or PF);
- To search for a solution, if existing, to the PF problem taking a set of upper and lower bounds and possibly some complementary relations with reference to some problem variables. (The Constrained Power Flow problem or CPF);
- To search for the minimum of a given cost function, for all possible solutions of the CPF problem (The Optimal Power Flow problem or OPF).

A. A brief history of power flows

The PF problem was solved during the sixties, as soon as

performance of the available computers, in addition to the improvement of the numerical techniques of linear algebra, allowed for efficient factorization of large sparse matrices.

That made possible the factorization of Jacobian for networks of several hundreds of nodes and in turn the use of the Newton method. References [1] and [2] are rightly considered by the electricians as the founding papers of this technique in Europe and the USA.

The two other sub-problems, CPF and OPF, are both related to the same mathematical technique of non-linear programming. They share a common history.

The first generation of computer programs that aimed at a practical solution of the OPF problem did appear at the end of the sixties. Most of these programs used a gradient method, i.e., they calculated the first total derivatives of the cost function related to the independent variables of the problem. These derivatives, known as the gradient vector, determine the direction of the move to be given to the current solution in order to improve it. The procedure is iterative and stops when a sufficiently small gradient is obtained.

From the very beginning, this gradient method followed two distinctive development channels.

The first one, which the founding paper is presented in [3], is based on a fixed separation of the problem variables in dependent and independent variables. This fixed separation automatically entails a different treatment of the bounded variables based to their belonging to the dependent or to the independent set of variables. It is indeed easy to limit the displacement of the independent variables inside their feasible range. On the opposite, the control of the move of the dependent variables is only possible in an indirect way. This control is done by adding to the cost function some penalty terms whose values are proportional to the dependent variables violation bounds, multiplied by a set of *penalty factors*. The a priori knowledge of active bounds and the right choice of penalty factors are two well-known difficulties of this approach.

The second development channel, called GRG for (Generalized Reduced Gradient or originally in French "Gradient Réduit Généralisé"), was presented in English in 1969 in [4]. It is the generalization of Wolfe's reduced gradient method to a set of non-linear constraints.

E. Jamouille and G. Dupont are with SystemsEurope S.A., Bd. de l'Humanité 415, 1190 Brussels, Belgium - www.systemseurope.be (e-mail: etienne@jamouille.com or g.dupont@systemseurope.be).

As in the first approach, the GRG method moves the current feasible solution in the direction of the gradient vector in such a way as to satisfy, in the end, the optimality criteria. This second approach, born from the Linear Programming (LP) techniques, makes use of the some very useful concepts utilized in LP, i.e.

- The set of dependent variables is considered as the *base* (in LP terms) of the linearized system of equation around the current operating point;
- The concept of *current system base* and the need to change this base when a dependent variable reaches one of its limits, i.e. becomes "active".

References [5] and [6] were the first to show the interest of this approach as well as its operational efficiency. Reference [5] applies the GRG method to the general economic dispatching problem in which the security constraints of type (n-1) are explicitly introduced, while [6] applies the GRG to the classical OPF.

Around the same period, reference [7] also proposed a reduced gradient method with variable basis showing great similarities with GRG. In addition, reference [7] also presents a complete formulation of the CPF, the solution of which is presented as a prerequisite to find the first feasible solution required by OPF. The solution technique proposed for the CPF is the dual simplex algorithm of Wagner, progressing into the unfeasible solution domain.

At first sight, this second development channel unifying the treatment of the constraints, seems to present a clear advantage over the first one which uses a fixed base together with penalty factors. However, it should be noted that both channels are using the reduced gradient as a direction for the move, but this one is calculated with only the first derivatives of the cost- and constraints functions. Therefore both approaches were initially suffering two major drawbacks:

- A often very slow convergence due to a zigzagging direction provided by a gradient that naturally depends on the variables scaling;
- A moving direction satisfying the constraints only at the first order. This made it necessary to limit the amplitude of the move by some empiric way in order to avoid departing too much from the feasibility region, and to be able to re-enter this region without difficulty through the Newton method.

B. The rise of the second order information

[8] provides an interesting review of the problem state of the art in the late seventies. The interest has shifted toward methods involving second order information corresponding to the non-linearity of the cost- and constraints functions. Taking into account the non-linearity of the cost function is relatively easy and already done in the original GRG method. However the introduction of the non-linearity of the constraints requires the factorization of the Hessian of the cost function augmented with the penalty terms corresponding to the constraints.

There were numerous works published during the eighties and beginning of nineties which followed this approach. To simplify - and since we do not intent to be exhaustive - we will consider all these works as improvements by the addition of second order information to the approach opened by [3].

C. The penalty approach

Among the various papers following this path, three of them seem to us good milestones to quote here.

Reference [9], the first using the Hessian, adds to the cost function the square of the mismatch of the equality and inequality constraints that are assumed actives, multiplied by some penalty factors. Then, by the optimization version of the Newton method, the minimum of the unconstrained augmented function is sought. This needs the direct calculation of the Hessian and its inverse by Gaussian elimination, as well as the use of sparse matrix inversion techniques. Unfortunately, this approach very soon showed to be quite limited due to the ever-increasing ill conditioning of the Hessian with the necessary increase of the penalty factors in the neighborhood of the solution.

Both [10] and [11] belong to the class of algorithms based on the Lagrange multipliers. Here it is the Lagrangien that is augmented with the penalty factors similar to those of [9]. The severity of these factors expresses the extent to which the minimization of the augmented Lagrangien approximates the original problem. Both references propose updating rules for the penalty factors aiming at the efficient convergence of the algorithm and the correct treatment of the binding inequality constraints at the optimum.

Finally, among the latest developments of methods using penalty terms, must be noted the use of the presently very fashionable barrier and interior point method in the OPF problem. Will this method, developed initially for large LPs, provide for better OPF solution algorithms than those already existing? This is far from certain.

D. The GRG approach

This being said on the first development channel, that we might call the *fixed base and penalty factors* approach, it should be well noted that the second channel of development, the GRG or "variable base" approach, despite its apparent advantage of being capable to deal directly with any type of constraints, binding or not, has to our knowledge generated complementary publication only in *Electricité De France* (EDF) and by its principal author.

In [12] will be found the status at the end of the eighties of the method called *differential injections* that was successfully applied by EDF to the problem of economic dispatching with explicit security constraints.

In the Anglo-Saxon world, to our knowledge, the GRG method has not generated any new publication on possible improvements since the original one.

Reference [13] however, using the *Sequential Quadratic programming* (SQP) technique, builds a sequence of sub

problems, each time linearising the constraints at the system operating point and finding the move that minimizes the sub problem Lagrangien developed to the second order. These linéarisation, added to the concept of the optimization of the next move from the current solution, give to [13] a family likeness to the method presented in this paper. However, in [13], the concept of variable base is totally absent. It is therefore not very well suited to deal with inequality constraints unless the binding ones are known a priori. However, it is interesting to note in [12] that SQP had already been identified as an interesting complement to the utilization of the GRG technique.

One could wonder if the relative absence of research aiming at the improvement of the GRG method is due to the fact that the introduction of second order information in the GRG seems to be less obvious than in the penalty factors method, for which the definition of the Lagrangien and its Hessian was offering a quite classical way of approach.

E. Paper objectives

This paper aims to show that, on the contrary, the GRG method is perfectly suited to accommodate second order information that provides for a stable and efficient convergence.

We will deal first with the CPF problem, as simpler and moreover providing to the OPF with its feasible initial solution. The main steps of the algorithm are presented and commented. Then the same guidelines are applied to the OPF. To illustrate the concepts and the performance of the method, a small case study is presented and its results are commented. Finally one shows the suitability of the method to deal with a most important problem for the planning engineer, i.e. the analysis of the sensitivity of the solution to the various problem parameters.

In the paper, we privilege the operational aspects of the presented matters rather than their pure mathematical aspects. All the mathematical developments are therefore to be found in appendices.

The underlying mathematical bases of our method are quite classical. They can be found in any good textbook of infinite analysis. In [14] we quote a textbook that has been used in Belgium and France by generations of engineers. In this paper, we have tried to use as much as possible the more modern non-linear programming concepts and notations found in the excellent textbook quoted in [15].

II. CONSTRAINED POWER FLOW

The objective of the CPF is to verify the existence of a technically acceptable power flow in a given transmission and generation system. Planning engineers are very often faced to this type of problem. In investment planning studies, they need to verify the operational realism of hypothetical investments. In operation planning studies, they want to verify consistency between measured currents and voltage values

with the parameters describing an existing system (lines, transformers, etc.).

Let's consider a network of n nodes. Mathematically the CPF problem consists in verifying if there exists a solution to the system (1) of $2n$ non-linear equations expressing the active and reactive power balances at each of the network node. This first system is completed by a second system (2) carrying some lower and upper bounds of the variables.

$$g(x, u) = 0 \quad (1)$$

$$\underline{x} \leq x \leq \bar{x} \quad (2)$$

$$\underline{u} \leq u \leq \bar{u}$$

These $2n$ equations may possibly be complemented by a set of r equality or inequality relations that may exist between the problem variables. For instance, taking into account transit constraints on the network lines is an example of such complementary relations. For sake of simplicity and without loss of generality in what follows, it is assumed that $r = 0$.

The number of problem variables is $2n + m$. They generally include:

- The module and voltage angle at each network node;
- The active and reactive generation at each network node;
- The transformer taps;
- Possible additional variables corresponding to HVDC equipment.

In the set of system variables one distinguishes m independent variables of type u , and $2n$ dependant variables of type x , which can be considered as implicit functions of the u variables through the system (1).

It is rather easy to transform this problem of existence of a solution into a non-linear programming problem, just by defining a cost function

$$Z = \sum c_j u_j + \sum c_i x_i = f(x, u) \quad (3)$$

that penalizes variables out of their bounds. In our case, these penalty costs are defined as linearly depending on the bounds overshoot (Fig 1).

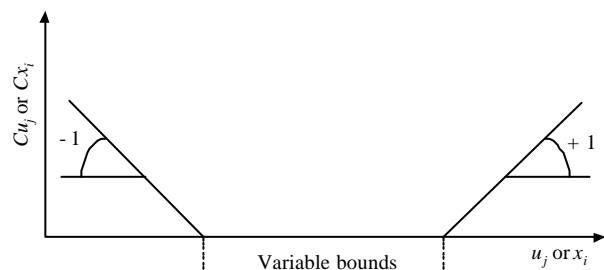


Fig. 1. Penalty costs for the CPF.

A solution is obtained if the Z value can be brought to zero. The algorithm is presented in Fig. 2 and its main steps are briefly explained below.

A. Step 1: Initialization

Initialization bears on the selection of the $2n$ dependent

variables. The Jacobien corresponding to these variables in system (1) must be invertible.

In the load flow problem, it is easy to define such a set of $2n$ variables. At each network node, one has only to select as dependent variables 2 out of the 4 variables that are normally attached to that node, i.e., for the first dependent variable, either the active generation or the voltage angle and for the second variable, select either the reactive generation or the voltage amplitude. Note that at least one active and one reactive generation variables shall be included in the dependent variables set. This normally guarantees a Jacobien, i.e. an initial invertible system base.

The remaining variables are the independent or out of base variables. The initial values of these variables are fixed at their nominal value for the voltages and transformer taps, while the generation is selected such as to respect approximately the global active and reactive system balance.

B. Step 2: Solution of the PF equations

This step inherits from step 1 or step 6 a set of independent variables and their values. One builds the current base and solves system (1) by the equations version of Newton's method. This defines the current solution (x_c, u_c) .

C. Step 3: Computation of the reduced gradient

It is first verified if, for the current solution, the cost function $Z=0$. In this case the problem is solved and an exit condition of the CPF has been reached.

If $Z \neq 0$, the current solution will be improved by moving the solution in the direction given by the reduced gradient vector G (appendix I). It should be noted however that this reduced gradient direction depends on the particular scaling given to each component u_j . The choice of this scaling being arbitrary, the only real information obtained from the gradient is the interest to move up or down the variables or to freeze them if some limit prevents their move in the indicated direction.

To cope herewith, one introduces a set of coefficients p_j defining for each variable its particular scaling factor. These scaling factors can be optimized using the second order information. This optimization will be discussed in the OPF case. For the simpler CPF case, experience shows that the normally used *per unit* system leads to natural scaling factors that are found to be satisfactory. Therefore, for the CPF, all the p_j are taken equal to 1.

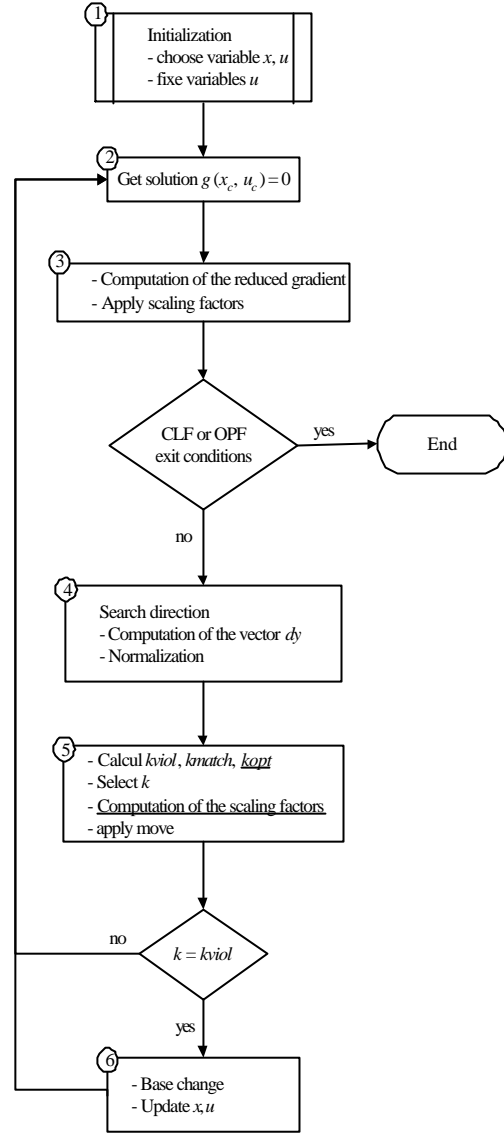


Fig. 2. CPF and OPF algorithm (underlined steps are for OPF only).

The move of the independent variables may then be written as

$$du_j = -G_j \cdot p_j$$

if the variable is free to move or

$$du_j = 0 \quad (4)$$

if the variable is frozen by a bound in the direction of the move.

If $Z \neq 0$ and all du_j equal zero (or are smaller in absolute value than an arbitrary small scalar $\epsilon \geq 0$), the CPF shall exit *without solution*. The cause of the problems' unfeasibility, and possible remedial action, can then be studied by analysis of the violated or limiting constraints.

D. Step 4: Search the direction in the space of feasible variations

Once du_j has been selected, one calculates the corresponding variations of the dependent variables in such a way as to keep the first total differential equal zero (appendix I):

$$dg = 0 \quad (5)$$

This allows for the definition of the dy vector that characterizes the search direction and - through the scalar factor k - the amplitude of the move.

$$dy = k \begin{pmatrix} du \\ dx \end{pmatrix} \quad (6)$$

Due to the introduction of the scalar factor, the components of the dy vector can be normalized. This is important from a numerical computation point of view. Factor k will be chosen in the subsequent step taking into account the normalization.

E. Step 5: Calculation of $kviol$, $kmatch$ and choice of k

There are two limitations to the value of k . The first one comes from the inequality constraints and is represented by the following rule in the algorithm.

“No move can make a variable goes beyond its nearest limit encountered in the direction of the move”. These limit points are discontinuities in which the left- and right gradient must be re-calculated in anyway. When applied to each x or u type variable, this rule leads to the definition of a first maximum value of k , called $kviol$.

A second limitation comes from the linear approximation that was done when computing dx as a function of du . This approximation can be estimated by studying the second term of the Taylor expansion of (1).

It can be demonstrated (appendix II) that the mismatch MM_i of the i^{th} equation of (1) at point

$$(x_c + k \cdot dx, u_c + k \cdot du) \quad (7)$$

can be written as

$$MM_i = \frac{1}{2} k^2 \cdot R_i \quad (8)$$

In this expression, R_i is independent of k and can be estimated by the calculation of a quadratic form involving the Hessian $\nabla^2 g_i$ at point (x_c, u_c) . However, since this analytic computation is rather time consuming, it is more efficient to evaluate the form numerically for a given but small value of $k = ktest$. For each equation, the mismatch at the point

$$(x_c + ktest \cdot dx, u_c + ktest \cdot du)$$

is therefore calculated, and from that mismatch one gets R_i from (8).

Then, by defining a maximum acceptable mismatch value $MMMax$ for the equations (1), one defines a $kmatch$ factor by the expression:

$$kmatch_i = \text{Min}_i \{kmatch_i\}$$

where, from (8),

$$kmatch_i = \sqrt{\frac{2 \cdot MMMax}{|R_i|}} \quad (9)$$

Finally k is selected as

$$k = \text{Min}\{kviol, kmatch\}$$

and the current operating point is moved by the quantity $k \cdot dy$.

F. Step 6: Change of base

This step first verifies if the proposed move brings some base variables to one of its limits. If not, one goes back to step 2 to calculate by the Newton method the exact values of the dependent variables corresponding to the new operating point.

If on the contrary, one of the base variables reaches one of its bounds, then $k = kviol$, and a change of base is necessary. This base change is similar to an LP simplex pivoting.

The pivot is on the line of the sensitivity matrix corresponding to the variable u_j whose move has been the most contributing to the variable x_i reaching one of its limits. The u_j variable will enter the base and the x_i variable will leave it. The new base and the corresponding new values of the independent variables are sent to step 2.

G. Convergence and efficiency of the algorithm

The CPF uses a gradient method in its simplest version, called the *steepest descent*. We hereafter discuss the efficiency of this approach when applied to the CPF problem.

Firstly, it is easy to give the algorithm a good starting point that at least satisfies system (1). For the adjustment of the active power balance, one only has to verify that the total available active power is greater than the sum of the loads at the nodes. If not, the problem has no solution.

The adjustment of the reactive power balance at the network nodes may be more complex because the network itself may be a source of reactive power. Therefore, in order to facilitate this adjustment, at those nodes where a difficulty might occur, we shall put in the set of dependent variables x the reactive power generating variable instead of the voltage variable. This allows to find easily an initial acceptable voltage profile.

A second potential obstacle of the steepest descent method is the difficulty to reach the exact minimum of Z when the problem is unfeasible, when $Z \neq 0$ and its minimum is obtained as a result of a balance between two variables that are both out of their limits. However, in our problem, this difficulty is without consequence because in that case the problem is known to have no solution. Thus, value given to ϵ is not critical.

A third potential difficulty is more delicate to cope with: it is the definition of the amplitude of the move vector. This vector defines a direction satisfying the constraints only at the first order, while it is obvious that system (1) must be kept verified with a given tolerance. This difficulty is solved by the utilization of second order information that allows for the calculation of $kmatch$ that will perfectly play its role as a *brake* when necessary.

III. OPTIMAL POWER FLOW

After the CPF, a feasible solution is available. With the OPF one intends to obtain a solution optimizing a given linear or quadratic cost function expressing a particular objective of the planning engineer. Here below is a non exhaustive list of some possible objectives:

- loss minimization;
- total generation cost minimization;
- analysis at the optimum of the active and reactive energy marginal costs at each node;
- optimal allocation of reactive power;
- analysis at the optimum of the generators' penalty factors;
- marginal interest of a line's reinforcement.

Whatever the quantity to optimize, the OPF offers the great advantage to give to the planning engineer a strong reference point from which he can investigate the sensitivity and security of the solution to any change in the parameters of his problem.

From a mathematical point of view, the OPF algorithm is very similar to the CPF. The cost function is different and may even show some discontinuities in the case of a power plant composed of various units with different characteristics. This however has no influence on the gradient method. This one only requires the existence of a partial first derivative of the cost function, at the current operation point, in the move direction of the variables.

Thus, the 6 steps of the OPF are identical to those of the CPF, except for:

- the addition of a limitation on the amplitude of the move, expressed by a new coefficient, $kopt$;
- the calculation of improved or quasi optimized scaling factors;
- specific exit conditions bearing on the value of the gradient component and the ratio between the expected gain of the next move and the actual value of the cost function (typically $= 10^{-8}$).

We first will discuss the calculation of the $kopt$ coefficient in step 5 after the calculation of $kviol$ and $kmatch$.

A. Computation of the optimal move coefficient

The optimal move coefficient $kopt$ takes into account the non-linearity of the dependent variables as a function of the move applied to the independent variables.

Let Δx be the vector representing the effective move of the x variables. The Taylor formula limited to the second order, applied to Δx viewed as a function of the independent variables du , can be written as:

$$\Delta x = k \cdot dx + \frac{1}{2} \cdot k^2 \cdot d^2 x \quad (10)$$

In this expression, dx is the vector of the first, and $d^2 x$ the vector of the second differentials of the x variables with respect to du at the current operating point (x_c, u_c).

By using (5), vector dx is expressed as a function of du .

The relation $d^2 g = 0$ allows for the calculation of $d^2 x$ by the following expression (appendix III):

$$d^2 x = - \left[\frac{dg}{dx} \right]^{-1} \cdot R \quad (11)$$

In this expression, the components of vector R are the constants R_i in (8). In the simple case where the cost function (3) is linear, its variation for a given coefficient k applied on du can be written as:

$$\Delta f(x, u) = \frac{df}{dx} \cdot \Delta x + \frac{df}{du} \cdot k du \quad (12)$$

The optimal value of k , or $kopt$, is found by equaling to zero the derivative of (12) in k and using (10) and (11):

$$kopt = \frac{\frac{df}{dx} \cdot dx + \frac{df}{du} \cdot du}{\frac{df}{dx} \cdot \left[\frac{dg}{dx} \right]^{-1} \cdot R} \quad (13)$$

In this expression, the numerator could be interpreted as the linear gain expected from a move that is proportional to the gradient, changed of sign.

The denominator equals the R vector, proportional to the active and reactive mismatch of (1), valued at the marginal active and reactive energy costs at the nodes.

In general, for any cost function that is convex and differentiable, appendix III presents the general formula for calculating $kopt$ and demonstrates that $kopt$ is necessarily positive around the optimum.

Finally, in the OPF at step 3, the selected k will be:

$$k = \text{Min}\{kviol, kmach, kopt\}$$

B. Calculation of the scaling factors

To optimise the scaling actors, one shall analyze the evolution of the cost function around the current operating point, as a function of the du moves.

Let Δf be the increase of the cost function for a move in the feasibility space of the independent variables such that

$$du_j = -G_j \cdot p_j \cdot k \quad (14)$$

The Taylor formula limited to the second order can be written as:

$$\Delta f = k \cdot df + \frac{1}{2} \cdot k^2 \cdot d^2 f \quad (15)$$

and for $k = kopt$ we have found a value of Δf , say Δf^* , that maximizes the decrease of the cost function. Now looking at Δf^* as a function of the variables p_j , we want to calculate the scaling factors $p_j > 0$, minimizing Δf^* for each non-frozen u_j variables. Let r be the number of frozen variables. A necessary condition for obtaining this minimum is the nullity of the first derivatives of Δf with respect to the p_j :

$$\frac{d(\Delta f^*)}{dp_j} = 0 \quad (16)$$

In principle, these $(m-r)$ conditions provide for a linear system of $(m-r)$ equations with $(m-r)$ unknowns, the solution of which gives a set of optimal scaling factors. This system is developed in appendix IV.

In practice, the matrix of the above linear system can be built from the Hessians of the cost function and of the equations (1). It is very sparse and its inversion is in principle possible. However, instead of this matrix construction and its inversion, we have preferred an iterative approach that is much simpler to implement: it uses the fact that (16) is linear in p_j .

Starting from an initial value given to the p_j , at each iteration, the algorithm calculates numerically, for each p_j , the value of the derivative of (16) at the right- and at the left side of the current p_j value.

These two values allow for the computation of the two coefficients of the linear expression of the derivative considered as a function of p_j . Then from that expression it is easy to calculate the p_j value that makes null the corresponding derivative of Δf^* or puts at its limit the u_j variable. This very simple method is an obvious choice when considering that a good approximation of the scaling factors is quite sufficient to improve significantly the convergence especially if the initial factors have been carefully selected.

C. Convergence and efficiency of the algorithm

This paragraph briefly discusses the convergence of the OPF algorithm and its ability to parameterize the results. First are commented the results obtained from a small didactic test network (Fig. 3).

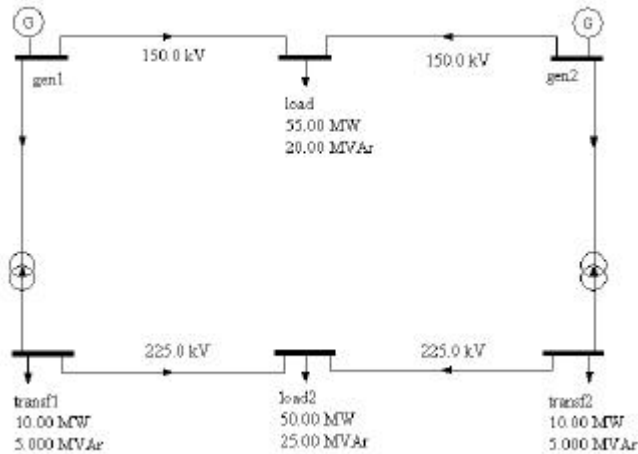


Fig. 3. Case study: Symmetrical loop 150/225 kV composed of 6 nodes with 4 loads, 2 variable generators and 2 controllable transformers.

The proposed test network is made of a symmetrical loop, linking two voltage levels (150 and 250 KV) through two controlled transformers. The system is fed by two generations of identical active- and reactive generating capacity and a cost function of which the quadratic coefficient equals 1% of the linear one. The system's optimal solution is obviously also

symmetrical. At the optimum, the two active and reactive generations and the transformers' ratios shall be equal, while the 4 voltages at each side of the transformers shall be at their upper bounds.

The initial solution is given by the CPF. In this first solution, one of the two generators is at its upper bound (70 MW) while the other is producing the missing active power. The voltages and transformer taps are at their nominal value.

The OPF algorithm needs 9 iterations to reach the optimum. Table 1 shows the evolution of some of its characteristic values such as the source (minimum of $kviol$, $kmatch$ or $kopt$) and the value of coefficient k giving the amplitude of the move and its expected gain.

One also observes the discontinuities in the evolution of the current system solution. These are marked by the changes of base or by an out-of-base variable hitting one of its bounds. In the table, one can see in the column nb the number of scaling factors p_j that have been updated in the iteration. These updates increase the calculation time at each iteration, but they drastically reduce the number of iterations required with a same base (generally not more than 3 or 4). However, it should also be noted that the calculation of the scaling factors aiming only at their improvement and not at their direct optimization, the choice of their initial value is of great importance on the number of necessary iterations.

TABLE I
CONVERGENCE OF THE CASE STUDY

Iter	$kviol$	$kmatch$	nb	$kopt$	Exp. Gain	Cost
1	0.16219	0.27690	0	0.1802	-1.012190	205.15907
Base Change : gen1 (Q) by gen1 (V)						
2	0.08345	0.07362	2	0.21321	-0.262280	204.16277
3	0.00975	0.08034	0	0.26347	-0.240597	204.00401
Base Change : gen1 (V) by gen2 (V)						
4	0.14093	0.33837	2	0.15872	-0.083469	203.98649
Base Change : gen2 (V) by gen1 (Q)						
5	0.06375	0.15962	3	0.01658	-0.002395	203.90296
6	0.00745	0.03119	3	0.00223	-0.000711	203.89972
7	0.00533	0.03133	0	0.15851	-0.026970	203.89866
Base Change : transf1 (V) by gen1 - transf1 (r)						
8	0.00030	0.28876	0	0.00276	-0.00002	203.89688
Base Change : transf2 (V) by gen2 - transf2 (r)						
9	0.07341	0.59721	0	0.00006	-0.000001	203.89687
OPF OK : By theoretical relative constraint						
Final Cost = 203,89687						
OPF did converge in 9 iterations						

Regarding the possibility of parameterization offered by the algorithm, it shall be noted that the final base and the optimum solution of a problem are an ideal starting point for a neighboring problem for which only one data element has been marginally modified. In this case the OPF generally requires only one or two extra iterations to reach the new optimum. This allows for reliability studies of type N-1 (single network or generator outage) or for the optimal choice of a network parameter. In this respect, Fig. 4 - shows the parameterization of

a capacitor bank located at node “load2” in the didactic test network.

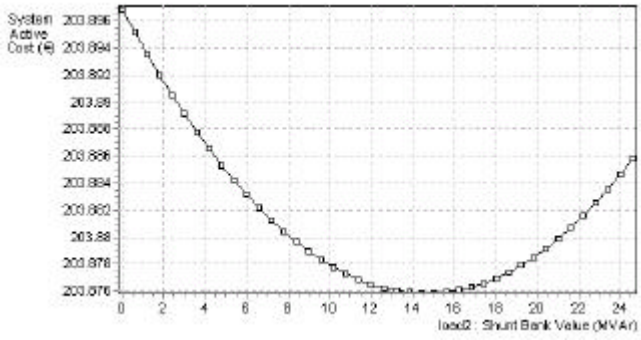


Fig. 4. Case study: Active generation system cost vs. shunt bank value.

On larger networks, the algorithm shows very similar characteristics to those shown in the case study. From user experience, the two following considerations can be noted:

- The number of network nodes is not the main problem's difficulty. Indeed, the factorization of a Jacobien corresponding to several hundreds or even thousands of network nodes is presently quite feasible
- The main problem difficulty is related to the number of control variables, their starting point, and their possible regulation range. These factors will determine the number of base changes requiring re-factorization of the Jacobien.

IV. SUMMARY AND CONCLUSIONS

The proposed method is general and in principle applicable to any convex, non-linear programming problem with equality and inequality constraints. It belongs to the GRG or reduced gradient with a variable base way of approach.

The method does not require the introduction of penalty terms in the cost function for the treatment of the binding inequality constraints, but allows for direct treatment of those constraints by a judicious change in the dependent and independent set of variables, when such a change is required by the move of the current solution. These moves, in the space of the feasible variations following the reduced gradient direction, are optimized at two levels.

At the first level, for given scaling factors, the move amplitudes of the independent variables are optimized. At second level, the scaling factors themselves are improved (optimized).

The first optimization level guarantees to stay into the feasible solution space at a given tolerance as well as a monotonous decreasing of the cost function, while the second level allows for obtaining a convergence close to the Newton type between two changes of base.

Both these two levels assume the possibility to develop the cost and the constraints functions up to the second order and therefore the evaluation of quadratic forms involving the Hessians of these functions.

However, the proposed method avoids the need to explicit

the Hessians of the constraints. The quadratic forms that contain them are efficiently evaluated using the mismatch generated by small first order feasible variations.

This technique makes the method very simple to program and applies particularly well to the CPF and OPF problems. Moreover the method is also very well suited for the sensitivity analysis of the solution to any problem parameter.

APPENDIX I - REDUCED GRADIENT AND SEARCH DIRECTION

A. Computation of the reduced gradient

Let $f(x,u)$ be a function subject to a set of $2n$ constraints expressed by the system (1) of $2n$ equations. One wishes to calculate the components of the vector

$$G = \frac{df}{du} \quad (17)$$

or reduced gradient of the function f with respect to each independent variable u_j ($j = 1..m$). It can be shown [14] that, if the Jacobien of system (1) corresponding to the x variables is invertible, the $2n$ variables x_i can be considered as implicit differentiable functions of the u variables. By applying the differentiation rules of the implicit functions, after total differentiation of the function f and the equations (1) with respect to the components u_j , one gets the two following matrix relations:

$$df = \nabla_x^T f dx + \nabla_u^T f du \quad (18)$$

$$dg = \nabla_x^T g dx + \nabla_u^T g du = 0 \quad (19)$$

In these expressions

- the symbols $\nabla_x^T f$ and $\nabla_u^T f$ represent the transpose of the vectors of the first partial derivatives of function f with respect to the x and u variables: i.e. the vectors df/dx and df/du ;
- the symbols $\nabla_x^T g$ and $\nabla_u^T g$ represent the matrices of the first partial derivatives of function g with respect to the x and u variables: i.e. the matrices dg/dx and dg/du .

It should be noted that dg/dx is the Jacobien of (1).

The elimination of the dx vector between (18) and (19) allows for calculation of vector G defined in (17) of which the j^{th} component can be written as:

$$G_j = \frac{df}{du_j} = \frac{df}{du_j} - \frac{df}{dx} \cdot \left[\frac{dg}{dx} \right]^{-1} \cdot \left[\frac{dg}{du_j} \right] \quad (20)$$

Note that if one selects a move $du_j = -G_j$ the first differential is always negative and can be written as:

$$df = -\sum_j G_j^2 \quad (21)$$

B. Linearized problem at the current operating point

At this stage, it is interesting to note that (20) can be interpreted as the simplex criteria of the linearized problem at the current operating point.

In Linear Programming terminology, the dependent

variables are then called *basic* and the independent variables are called out-of-base. The Jacobien, or current system base, is noted B and the sub matrix of the independent variables is noted H . Similarly, the vectors \mathbf{df}/\mathbf{dx} and \mathbf{df}/\mathbf{du} are represented by the vectors Cu and Cx . With those notations, the simplex criteria's are written as:

$$\frac{df}{du_j} = Cu - Cx \cdot B^{-1} \cdot H_j$$

In this expression, from (20) the quantities:

$$\left[\frac{df}{dx} \right] \cdot \left[\frac{dg}{dx} \right]^{-1} = Cx \cdot B^{-1} \quad (21)$$

are the dual variables of the constraints (1) or the multipliers of Lagrange λ_i changed of sign (in our particular problem they can be interpreted as the marginal costs of the active and reactive energy at the network nodes), and

$$\left[\frac{dg}{dx} \right]^{-1} \cdot \left[\frac{dg}{du_j} \right] = B^{-1} \cdot H_j = -\frac{dx}{du_j} \quad (22)$$

is a column vector of which each component is the term corresponding to the variable u_j in the sensitivity matrix S of the linearized system at the current operating point.

C. First order variation move

Under these simplified notations, (5) is written as:

$$B \cdot dx + H \cdot du = 0$$

This leads to the definition of the dx vector as a function of the du vector by the expression

$$dx = B^{-1} \cdot H \cdot du = S \cdot du$$

Then one may formulate (6) as

$$dy = k \cdot \underline{S} \cdot du \quad (24)$$

In this expression, \underline{S} of dimension $(2n+m, m)$ is the sensitivity matrix S bordered on its upper side by a unit matrix.

Note that dy is a vector satisfying condition (5) and therefore belonging to the space of first order feasible variations around the current operating point. This vector provides for a search direction belonging to the feasible solution space.

APPENDIX II - MISMATCH EVALUATION

We write the second order Taylor series expansion of the vector function $g(x, u)$ around the current operating point (x_c, u_c) for a small displacement dy belonging to the subspace of first order feasible variations:

$$g(x_c + kdx, u_c + kdu) = g(x_c, u_c) + k \nabla_x^T g dx + k \nabla_u^T g du + \frac{1}{2} k^2 R \quad (25)$$

where R is a column vector of which the component

$$R_i = (dy)^T \nabla^2 g_i (dy) \quad (26)$$

is a quadratic form where

$$\nabla^2 g_i = \frac{d^2 g_i}{dx du}$$

stands for the Hessian matrix of the i^{th} constraint of (1).

Using the fact that the point (x_c, u_c) is by definition a

feasible solution of the system (1) and that the vectors dx and du belong to the subspace of first order feasible variations, i.e.

$$k \nabla_x^T g dx + k \nabla_u^T g du = 0$$

from (25), one may obtain:

$$MM = \frac{1}{2} k^2 R \quad (27)$$

where MM is the mismatch vector of which the component MM_i is the difference from 0 of $g_i(x + kdx, u + kdu)$, and R is a $2n$ vector depending on the search direction dy , but independent of k .

APPENDIX III - EVALUATION OF d^2f AND Δf

Let

$$\Delta f = kdf + \frac{1}{2} k^2 \cdot d^2 f \quad (28)$$

be the evolution of the cost function around the current solution (x_c, u_c) , for a move vector dy belonging to the space of first order feasible variations. In this expression $d^2 f$, that we want to calculate, takes into account the non-linear evolution of the functions f et g .

As in appendix I, we applied the rules for differentiating implicit functions. After differentiating two times (1) and then the function f (see [10], Chap. 4), one finds:

$$d^2 g = (dy)^T \nabla^2 g (dy) + \nabla_x^T g \cdot d^2 x = 0 \quad (29)$$

$$d^2 f = (dy)^T \nabla^2 f (dy) + \nabla_x^T f \cdot d^2 x \quad (30)$$

(26) and (29) allow for the calculation of vector

$$d^2 x = -(\nabla_x^T g)^{-1} R \quad (31)$$

Then replacing $d^2 x$ in the second expression (30) one finds $d^2 f$.

Using (30) and (31), (28) can be written as:

$$\Delta f = k df + \frac{1}{2} k^2 \left((dy)^T \nabla^2 f (dy) - \nabla_x^T f (\nabla_x^T g)^{-1} R \right)$$

Introducing the quantities A , B and C , we may write this as

$$\Delta f = k \cdot A + \frac{1}{2} k^2 (B - C) \quad (32)$$

Equaling to zero the derivative of (32) with regard to k , we find k_{opt} , i.e. the optimal k minimizing Δf :

$$k_{opt} = \frac{-A}{B - C} \quad (33)$$

The numerator A of k_{opt} is necessarily negative because the move du is chosen in the reverse gradient sense (see appendix I). Let's show that the denominator is necessarily positive around the optimum. From equation (21), we already have shown that the vector

$$\nabla_x^T f (\nabla_x^T g)^{-1}$$

gives the marginal cost of the constraints or the Lagrange multipliers λ_i with reverse sign.

Introducing these multipliers and replacing R by its equivalent quadratic form (26), the denominator $B - C$ can be written as:

$$d^2 f = dy^T \left(\nabla^2 f + \sum_{i=1}^{2n} I_i \nabla^2 g_i \right) dy \quad (34)$$

This expression, according to the optimum conditions of Lagrange [15], is necessarily greater or equal to zero at the optimum.

Finally, replacing $kopt$ by its value in (32), one gets:

$$\Delta f = -\frac{1}{2} \frac{A^2}{(B-C)} \leq 0$$

This expression guarantees the monotonous decrease of the cost function and therefore the convergence of the algorithm.

APPENDIX IV - COMPUTATION OF THE SCALING FACTORS

Let Δf^* be the increase of the cost function corresponding to a move of the independent variables, as in (14):

$$du_j = -G_j \cdot p_j \cdot kopt \quad (35)$$

From (28), one can write:

$$\Delta f^* = kopt \cdot df + \frac{1}{2} kopt^2 \cdot d^2 f \quad (36)$$

Let's develop the conditions:

$$\frac{d(\Delta f^*)}{dp_j} = 0 \quad \text{or equivalently} \quad \frac{d(\Delta f^*)}{d(du_j)} = 0 \quad (37)$$

Replacing dy by its expression (24) in (34) we find:

$$d^2 f^* = kopt^2 \cdot du^T \cdot \underline{S}^T \left(\nabla^2 f + \sum_{i=1}^{2n} I_i \nabla^2 g_i \right) \underline{S} \cdot du$$

Under the assumption that $kopt$ can be considered constant, in the domain of variation of du , the derivative of $d^2 f$ with respect to du can be written:

$$\frac{d^2 f^*}{du} = 2 \cdot kopt^2 \cdot \underline{S}^T \left(\nabla^2 f + \sum_{i=1}^{2n} I_i \nabla^2 g_i \right) \underline{S} \cdot du$$

The derivative of df with respect to du is known by (20)

Finally, the conditions (37) generate the matrix relation:

$$\frac{d(\Delta f^*)}{d(du)} = G + kopt \cdot \underline{S}^T \left(\nabla^2 f + \sum_{i=1}^{2n} I_i \nabla^2 g_i \right) \underline{S} \cdot du = 0$$

written more simply as

$$M \cdot du^* = -G$$

In principle, the inversion of the square matrix M of dimension $(n-r)$ with r as the number of frozen variables, allows for the computation of the vector

$$du^* = -M^{-1} \cdot G$$

and then the optimum scaling factors

$$p^* = -\frac{du^*}{G}$$

It should be noted that, when in the unconstrained case, this approach is totally equivalent to Newton's optimisation method.

REFERENCES

[1] J. Carpentier, "application of Newton method to load flow computations", Proc. PSCC 1, London, 1963.

- [2] W.F. Tinney and C.E. Hart, "Power flow solution by Newton's Method", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-86, n°11, Nov. 1967.
- [3] H.W. Dommel and W.F. Tinney, "Optimal Power Flow Solutions", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-87, Oct. 1968.
- [4] J. Abadie and J. Carpentier, "Generalization of the Wolfe Reduced Gradient method to the case of non linear constraints", *Revue Optimisation*, pp 37-47, Academic Press, 1969.
- [5] J. Carpentier, C. Cassapoglou and C. Hensgen, "Injections différentielles, une méthode de résolution générale des problèmes de dispatching économique sans variables entières utilisant le procédé du Gradient Réduit Généralisé", in *Proc. Symposium Helors-Ifors*, Athens, 1968.
- [6] J. Peschon, D.W. Bree and L.P. Haydu, "Optimal Solutions involving systems security", in *Proc. PICA*, 1971.
- [7] E. Jamouille, J. Velghe, "Optimisation sous variables bornées de la répartition des puissances dans les réseaux maillés à courants alternatifs", *Revue E*, supplément au Bulletin de la Société Royale Belge des Electriciens, vol. VI, n° 11, 1971.
- [8] J. Carpentier, "Optimal Power Flows", *Electrical Power & Energy Systems*, vol. 1, n°1, April 1979.
- [9] A.M. Sasson, F. Vioria and F. Haytes, "Optimal load flow solution using the Hessian matrix", in *Proc. PICA*, 1971, pp 203-209.
- [10] A. Santos, S. Deckman and S. Soares, "A Dual Augmented Lagrangian Approach for Optimal Power Flow", *IEEE Trans. Power Systems*, vol. 3, n°3, August 1988.
- [11] A. Monticelli, Wen-Hsiung and E. Liu, "Adaptive Movement Penalty Method for the Optimal Power Flow", *IEEE Trans. Power Apparatus and Systems*, vol. 7, n°1, February 1992.
- [12] J.L. Carpentier, A. Gillon and Y. Jegouzo, "Multimethod Optimal Power Flows at Electricité De France", IFAC, Seoul, 1989.
- [13] R.C. Burchett, H.H. Happ and D.R. Vierath, "Quadratically convergent optimal power flow", *IEEE Trans. Power Apparatus and Systems*, PAS-103, November 1984.
- [14] de La Vallée Poussin, *Cours d'Analyse Infinitésimale*, Louvain: Uystpruyst, Paris: Gauthier-Villars, 1943.
- [15] D. P. Bertsekas, *Non Linear Programming*, Athena: Scientific, 1999.

Etienne Jamouille received his degree in electrical engineering from the Université Catholique de Louvain, Belgium in 1959 and a complementary degree in Industrial management science in 1965 from the same university.

During 12 years, he was part of a Belgian electrical utility, firstly as operation engineer in a power plant then as head of the methods and models department of the company.

During this period, he pioneered various methods in generation and transmission systems planning and reliability analysis, including the Equivalent Load Duration Curve (ELDC) method for production costing and security assessment of a hydrothermal generation system.

In 1974, he founded Systems-Europe and headed the company until his retirement in 2000. He is presently external technical consultant of Systems Europe.

Gauthier Dupont received the B. Sc. degree in physics and the Licence Spéciale in geophysics from the Université Libre de Bruxelles, Belgium, in 1994 and 1995, respectively.

Since 1996, he was with Systems-Europe (SE), Brussels, Belgium, as a project engineer for planning studies as well as an analyst-developer specialized in Object Oriented Programming. Since 2000, he held the IT Department of SE

