Solving Electrical Power Load Flow Problems Using Intervals

Eustaquio A. Martínez[1], Benjamín Barán[2], and Tiaraju Asmuz Diverio[3]

[1]Facultad Politécnica, Universidad Nacional del Este, Ciudad del Este, Paraguay, amartinez@politec.une.edu.py
[2]Centro Nacional de Computación
Universidad Nacional de Asunción
San Lorenzo, Paraguay, bbaran@cnc.una.py
[3]Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brasil, diverio@inf.ufrgs.br

Due to its importance to the planning and maintenance of large electrical power distribution systems, the electrical power load flow problem has been exhaustively studied by electrical engineers. Taken in this context, the idea of an interval approach is particularly interesting, if one considers the possibility of finding all the solutions within a domain. This advantage is specially important given that non-linearity of the problem give rise to at least one operating point (feasible solution) for which traditional controls drive the system to a collapse, producing an undesirable blackout as the ones produced in Brazil and the United States of America. It also allows the representation of the problem's parameters as intervals, based on the physical values and their respective tolerance intervals, aiming to the application of the method in the sensitivity analysis of an electrical power system.

This work first presents the use of interval arithmetic to solve the electrical power flow problem in a sequential environment. However, the sequential approach requires expensive computational resources (as supercomputers) not always available to solve the problem in workable time. Consequently, a new parallel asynchronous approach running on a local area network of personal computers is presented to speed up the process with resources already available in most organizations. This novel approach is based on the interval Newton/Generalized Bisection algorithm, that is parallelize using an asynchronous communication techniques that let each processor of an heterogeneous network of computers to work at its own speed, sharing results and workload with other processors of the network, to accomplish the calculation goal in workable time.

This way, the method may be scaled with a number of available processors, to solve problems of greater magnitude.

The electrical power load flow problem can be formulated as a quasi-linear system of equations

$$Yx = I(x) \tag{1}$$

where Y is the admittance matrix, $Y = \{y_{ki}\} \in C^{nxn}$, with $y_{ki} = G_{ki} + B_{ki} \in C$; $x \in C^n$ represents the (usually unknown) voltage vector (therefore, n is the problem size), and $I(x)$ is the current vector, $I \in C^n$. To facilitate control of operational restrictions (usually on the voltage magnitude), the problem is mostly solved in polar coordinates as:

$$P_k = V_k \sum_{i \in K} V_i (G_{ki} \cos \Theta_{ki} - B_{ki} \sin \Theta_{ki}) \tag{2}$$

$$Q_k = V_k \sum_{i \in K} V_i (G_{ki} \cos \Theta_{ki} - B_{ki} \sin \Theta_{ki}) \tag{3}$$

where $\Theta_{ki} = \Theta_k - \Theta_i \forall k \in \{1, \ldots, n\}$, K is the group of the bus bars adjacent to k and k itself.

The use of an interval approach for the solution of a non-linear system as the one above, brings along some interesting advantages, such as high accuracy and self-validation, as well as proof of root existence and uniqueness of solutions. The interval approach allows us to find the solutions by estimating an interval (or union of intervals) which is expected to contain one or more solution. The method then will indicate if such solutions exist or not. Observe that point methods used at present, such as the *Newton-Raphson*, do not posses these important features.

In this context, a non-linear system can be written:

$$F(X) = (f_1(X), \ldots, f_2(X))^T = 0 \tag{4}$$

where $F : R^n \to R^n$, $X = (x_1, x_2, \ldots, x_n)^T \in R^n$ and $\underline{x}_i \leq x_i \leq \overline{x}_i$ for $1 \leq i \leq n$, $\underline{x}_i$ y $\overline{x}_i$ are the lower and upper bounds of x_i . The interval Newton method for non-linear equation systems has quadratic convergence can be used to solve the problem. The system (4) can be written as a linear interval system:

$$\mathbf{F}'(\mathbf{X}^k)(\tilde{\mathbf{X}}^k - X^k) = -F(X^k) \tag{5}$$

where $\mathbf{X}^k \in IR^n$ is the interval vector where the solution $X^\star \in R^n$ is expected to be found; $X^k \in R^n$ is an inner vector of $\mathbf{X}^k$, i.e. $X^k \in \mathbf{X}^k$ (usually the midpoint of $\mathbf{X}^k$) is the unknown interval vector which is expected to contain the solution $X^\star$; $\mathbf{F}'(\mathbf{X}^k) \in IR^{nxn}$ is the interval extension of the Jacobian matrix of $Fin\mathbf{X}^k$. $\tilde{\mathbf{X}}^k$ can be calculated by solving equation (5). The iterative formula for a system with n variables results in:

$$\mathbf{X}^{k+1} = \mathbf{X}^k \cap \tilde{\mathbf{X}}^k \tag{6}$$

If $\mathbf{X}^{k+1} = \phi$ (empty interval) then the non-existence of a solution in $\mathbf{X}^k$ is proved. To compute $\tilde{\mathbf{X}}^k$ solving (5), any known method, such as Gauss elimination Method or Gauss-Seidel interval Method can be used. In this work it is used the latter.

In order to solve (5) using the interval Newton Method, the applicable interval system may be written as:

$$\begin{bmatrix} \mathbf{H}^k & \mathbf{N}^k \\ \mathbf{J}^k & \mathbf{L}^k \end{bmatrix} \left(\begin{bmatrix} \tilde{\Theta}^k \\ \tilde{\mathbf{V}}^k \end{bmatrix} - \begin{bmatrix} \Theta^k \\ V^k \end{bmatrix} \right) = \begin{bmatrix} \Delta \mathbf{P}^k \\ \Delta \mathbf{Q}^k \end{bmatrix} \tag{7}$$

where $\begin{bmatrix} \mathbf{H}^k & \mathbf{N}^k \\ \mathbf{J}^k & \mathbf{L}^k \end{bmatrix} = \mathbf{F}'(\mathbf{X}^k)$; $\begin{bmatrix} \tilde{\Theta}^k \\ \tilde{\mathbf{V}}^k \end{bmatrix} = \tilde{\mathbf{X}}^k$; $\begin{bmatrix} \Delta \mathbf{P}^k \\ \Delta \mathbf{Q}^k \end{bmatrix} = F(X^k)$, and $\mathbf{H}$, $\mathbf{N}$, $\mathbf{J}$, $\mathbf{L}$ are the interval sub- matrices that depend of the problem, $\mathbf{V}$ and Θ are intervals vectors.

It is well established in the field that the search region for the load flow problem is:

$$\Theta = [-\Theta_{max}, \Theta_{max}] \tag{8a}$$

$$\mathbf{V} = [-\zeta + 1, \zeta + 1] \tag{8b}$$

where $\Theta_{max} \cong 10^\circ$ and $\zeta < 1$, according to heuristic recommendation.

System of equations (7) is first solved sequentially by using Interval Newton/Generalized Bisection. However, electrical systems are non-linear system of large dimensions and requires computational resources not always available. This fact motivated our studies of parallel techniques and algorithms in an asynchronous environment of personal computers to reduce processing times and to optimize the use of available computer networks. To solve the problem in a parallel asynchronous environment as a network of computers, the original problem should be partitioned in several smaller sub-problems, in such a way that each processor of the network, can work on its own sub-problems without much intervention of other processors.

A simple approach to partition the low flow problem in sub-problems when using the proposed interval method is by dividing the search domain in disjoint sub-domains. That way, each processor makes calculations on its assigned sub-domain without interfering with other processors. Of course, a master process is needed to manage the work of each processor. In this way, each processor carries out its search in a particular region, which is smaller that the global domain. At each processor, the algorithm detects whether a solution exists (or not) within its sub-domain and it communicates its finding to the master, which may assign a new sub-domain for each new available processor, managing load balancing, until the problem is completely solved. Of course, hard sub- problems may be further sub-divided using Generalized Bisection, reducing the total processing time.

In order to determine the advantages obtained by the proposed parallel method, a Sp (Speed-Up) measure of acceleration is defined as the relation between the sequential processing time and the parallel processing time.

In order to verify the proposition's validity, algorithms (both traditional point method and the proposed interval approach) were implemented in C language and several well known test problems were solved, as the IEEE 5 and 14 busbur paradigm, the Monticelli 30-busbar system and a 88-busbar electrical system. The authors decided to use an existing asynchronous communication facilities, already built in MPI (Message Passing Interface) software without any existing interval software, considering the difficulty of implementing a reliable parallel asynchronism in the latter. At the moment is in study the implementation of algorithms in c-xsc for Linux.

The computation environment was based on a 10 Mbps local area network of 5 personal computers with Pentium II processors of 400MHz and 32 Mb RAM, running a Linux Red Hat operating system. In the parallel implementation, one acts as the master, NFS and NIS server and has MPI installed. The others four work as slaves.

Experimental results show that the parallel approach is not only faster, but it also founds better results with smaller solution diameter and power mismatch (difference between actual and calculated power), thus offering an additional advantage.

In summary, a parallel asynchronous interval approach to the load flow problem seems beneficial in time reduction and quality of solutions and a conjecture of scalability of this advantages to larger problems with an even larger number of inexpensive computers in a local area network may be stated.