SOLVING ELECTRICAL POWER LOAD FLOW PROBLEMS USING INTERVALS

Eustaquio A. Martínez[*] mailto:amartinez@politec.une.edu.py

Benjamín Barán[†] mailto:bbaran@cnc.una.py

Tiaraju Asmuz Diverio[‡] mailto:diverio@inf.ufrgs.br

August 14, 2003

Extended Abstract

Due to its importance to the planing 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. 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 proposes the use of interval arithmetic to solve the electrical power flow problem in a sequential and parallel context and demonstrates through results the viability of the method's application. Parallel solution of the problem is it makes in an asynchronous-parallel context of a personal computer network, based on the interval Newton/Generalized Bisection algorithm. This way it suggests that the method can be scaled with the number of 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 voltage vector, and $I(x)$ is the current vector, $I \in C^n$.

[*]Facultad Politécnica. Universidad Nacional del Este, Ciudad del Este - Paraguay

[†]Centro Nacional de Computación. Universidad Nacional de Asunción, San Lorenzo - Paraguay

[‡]Instituto de Informática. Universidade Federal do Rio Grande do Sul, Porto Alegre - RS - Brasil

In general terms, the problem to solve is:

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

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

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 ending criterion used is generally the power mismatch, which must be smaller than a tolerance ε specified.

The use of an interval approach for the solution of a non-linear system, 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 solution to the problem by estimating an interval which is expected to contain one or more solutions, or intervals whose union results in a search region. The method then will indicate if such solution exists or not, and whether it is the only one. Observe that punctual methods used at present, such as the *Newton-Raphson*, do not posses these important features.

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 is expected to be found $X^\star \in R^n$; $X^k \in R^n$ 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 Jacobean matrix of F in $\mathbf{X}^k$. $\tilde{\mathbf{X}}^k$ can be calculated by solving equation (5). The iterative formula for a system with n variables results :

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

If $\mathbf{X}^{k+1} = \phi$ (empty interval) then the inexistence of 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. This proposes uses the latter.

In order to solve the problem (5) using the interval Newton Method, the applicable interval system must been found and 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.

The solution search region is:

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

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

where $\Theta_{max} \cong 10$ y $\zeta < 1$, according heuristic recommendation.

The system (7) is solved sequentially by using Interval Newton/Generalized Bisection algorithm.

Preliminaries results of the sequential solution of the problem, using of interval arithmetic proved to be rather heavy, computational speaking. For that reason, is it proposed paralleling the solution within the realm of interval arithmetic, so as to reduce the processing time using a distributed computation environment.

In a distributed environment each processor carries out the search for solutions in a particular region, which is smaller that the original search region of the problem. This reduces the processing time. The algorithm employed in each processor detects whether a solution exists or not within in its particular search region. This enables the processor to be available for the another search region, thus balancing the workload.

In order to verify the proposition's validity, the algorithms where implemented in C language and interval operations, tested by solving, through the sequential and parallel methods IEEE's paradigm problem of 5 and 14 bus bars were used. Monticelli 30-busbar and the 88-busbar combined problem were used as well. The latter were also solved -on the same environment- sequentially trough the *Newton Raphson* (punctual) method, for the purpose of comparison. This analysis confirmed that the solution by interval method is computationally much heavier.

Comparing the processing times obtained with sequential solution, a considerable reduction it was observed when the processing was done in more than one processor. This confirms the advantage of using the parallel method proposed, rather than the sequential one.The Speed-Up obtained for each problem, shows that the processing time improves with the use of more processors.

Currently we are studying the implementation of other alternatives to paralleling the solution of the problem with the objective of found out a most efficient solution for each case.