A New Asynchronous Parallel Load Flow Calculation Algorithm

Shengtao Fan[1], Xiaqing Li[2], Li Zuo[2], Yunhua Li[1]

1. School of Automation Science and Electrical Engineering
Beijing University of Aeronautics and Astronautics
Beijing, China

2. Department of Electrical Engineering
Beijing Institute of Petrochemical Technology
Beijing, China

***Abstract*—A new asynchronous parallel load flow calculation algorithm is presented, and an improved algorithm is also advanced based on the analyzing of the basic algorithm. In order to verify the validation, the IEEE-30 and IEEE-57 systems are used, the results indicate that the basic algorithm can obtain the load flow of electrical power networks and the improved algorithm can improve the convergence and convergence rate of the exterior iterations. Moreover, the Ward equivalent taking the unbalance power distribution into account is used to further implement the asynchronous parallel calculation of dynamic load flow.**

***Keywords*—asynchronous parallel computing, dynamic load flow, Ward equivalent**

I. INTRODUCTION

The scale and complexity of power systems are increasing drastically in the past few decades, which requires more powerful and flexible tools for power systems analysis and computation. As a promising technology, great efforts have been made on the study of parallel processing. Solution of most power system problems is based on the solution of the linear algebraic problem and much work has been done on algorithms for parallel triangular factorization and forward and backward substitution [1-5]. Partitioning networks into subgroups among processors is the common way to achieve parallelism. The main goal of network partition is to divide the computation through decomposition into the bordered block diagonal form [6-7] or elimination trees [8]. The thought of the network partitioning method is: Firstly partition the large-scale network into several less-scale networks, perform equivalent calculation on the boundary of the partitioning, and then calculate the coordination variables, at length calculate the variables of the sub-networks and accordingly get the solution of the original network. If the order of the rows and columns of the corresponding network admittance matrix are arranged to follow the cluster configuration, the network admittance matrix will have a bordered block diagonal form (BBDF) and the network partitioning method is used to solve the correction equations in every step of the iteration [9]. However, the algorithm is synchronous which can't meet the requirement for distributed load flow calculation of large-scale power systems. A new asynchronous iterative algorithm is proposed for decomposition and coordination dynamic power flow in [10]. Interior and exterior iterations are designed in the algorithm. The former is used to achieve an isolated dynamic power flow solution for a subsystem and the latter is designed to achieve coordination between the two levels. By modifying the equivalent injection power on the boundary buses during the exterior iteration, a consistent power flow result with that for whole system can be obtained. However, the method needs all the subsystems to get the equivalent of the external network, which demands all the subsystems to track alterations of the external network; it still has some inconvenience in application.

II. A NEW ASYNCHRONOUS PARALLEL LOAD FLOW CALCULATION ALGORITHM

A. Definitions

For the convenience of description, the nodes of networks are divided into three kinds. The nodes of the first kind can be specified factitiously and the principle is: For a given network, if some nodes of the first kind are torn, the original network should be divided into several less-scale networks. The nodes in the original network, which directly connect to the nodes of the first kind, are nodes of the second kind. All the nodes in the original network except for the nodes of the first and second kinds are nodes of the third kind. In Fig. 1, nodes 5, 6, 7, 8, 9 are specified as nodes of the first kind (bold line), if tear node 5, 6, 7, 9, then the original network is divided into four less-scale networks. When the nodes of the first kind are specified, nodes 4, 10, 11 are accordingly nodes of the second kind, and nodes 1, 2, 3 are nodes of the third kind.

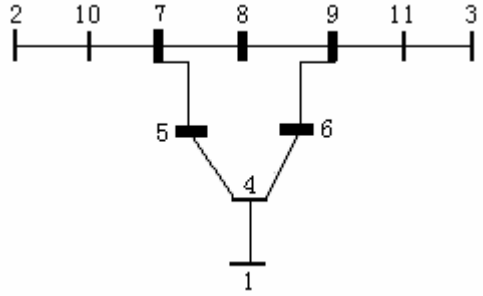

Figure 1. A simple 11-nodes system

If some nodes of the first kind are torn, the original network can be divided into several less-scale networks. For every less-scale network, remove the nodes of the first kind and the branches connecting to the nodes of the first kind, the remainder is called a sub-network of the original network. For example: If tear node 7 of the first kind in Fig. 1, a less-scale

This work was supported by Beijing Municipal Education Commission and Beijing Natural Science Foundation, China, under Grant KZ200510017012.

network can be obtained which consists of nodes 2, 7,10 and branches 2-10, 7-10; remove node 7 and branch 7-10, the remainder made up of node 2, 10 and branch 2-10 is called a sub-network of the original network.

If perform Ward equivalent on the original network, eliminate the nodes of the third kind, reserve the nodes of the first and second kinds, the equivalent network is called the coordination network of the original network. Because the sub-networks have been decoupled, if every sub-network performs Ward equivalent separately, eliminate the nodes of the third kind in every sub-network, reserve the nodes of the first and second kinds, the coordination network can also be obtained.

B. Basic Algorithm

The thought of the new algorithm follows the network partitioning method: Firstly perform Ward equivalent on every sub-network synchronously, eliminate the nodes of the third kind, thereby obtain the coordination network; and then perform centralized load flow calculation on the coordination network, thus can get the voltage of the nodes of the first kind; at length the state of the sub-networks can easily be obtained with the voltage of the nodes of the first kind. Because Ward equivalent grounds on some assumptions, which actually can't be met exactly, the voltages of the nodes of the first kind are not accurate. If set an exterior iteration which can reduce the error step by step, then the accurate voltage of the nodes of the first kind can be obtained, and farther get the load flow of the whole network.

Assume there are m nodes of the first kind, and the voltage is $\dot{U}_F$, which is an m dimension complex vector.

Algorithm F

F1. [Calculate the initial equivalent networks of the sub-networks] Choose an initial state of the whole network; the subsystem corresponding to every sub-network synchronously calculates the equivalent network of this sub-network using steady-state Ward equivalent method.

F2. [Calculate the voltage of the nodes of the first kind] The coordination system gets the coordination network with the equivalent networks of all the sub-networks. The coordination system calculates $\dot{U}_F$ using centralized load flow calculation algorithm. If $\dot{U}_F$ satisfies the convergence precision, current $\dot{U}_F$ is the voltage of nodes of the first kind, and the algorithm is over.

F3. [Get the state of the sub-networks] The coordination system sends $\dot{U}_F$ to corresponding sub-networks. The sub-networks obtain their state using centralized multi-$V\theta$ -node load flow calculation algorithm based on $\dot{U}_F$.

F4. [Recalculate equivalent networks of the sub-networks] Every subsystem recalculates the equivalent network using steady-state Ward equivalent based on the state of the sub-network. Go to F2.

C. Analysis of Algorithm

If communication failure occurs between the coordination system and one subsystem, the coordination system can use the equivalent network of the sub-network obtained in last exterior iteration, and the calculation can continue without influence, so the algorithm F is an asynchronous parallel algorithm.

We can get a sequence $\dot{U}_F^0, \dot{U}_F^1, \dot{U}_F^2 \cdots \dot{U}_F^k \cdots$ in algorithm F, and $\dot{U}_F^{k+1}$ is uniquely determined by $\dot{U}_F^k$, namely $\dot{U}_F^{k+1} = \varphi(\dot{U}_F^k)$. If the sequence is convergent, assume the limit is $\dot{U}_F^*$, considering the property of Ward equivalent "If the injection current of the external network keep the same, the Ward equivalent is accurate" [13], it is easy to see $\dot{U}_F^*$ is the accurate value of the voltage of the first kind nodes. Obviously algorithm F needs sequence $\{\dot{U}_F^k\}$ to be convergent; which generally implies that $\varphi(\bullet)$ is deflation beside $\dot{U}_F^*$, however, $\varphi(\bullet)$ may not actually satisfy the condition.

We actually extract the root of the equation $F(X) = \varphi(X) - X = 0$ using the fixed point iteration method in algorithm F. Compared with the Newton method; its convergence condition is more rigorous. However, $\varphi(\bullet)$ is implicit, it's hard to get the explicit formulation which brings on the difficulty of calculating the partial derivative of $\varphi(\bullet)$ when using Newton method. Is there a method that its convergence condition is less rigorous like Newton method, at the same time, can also avoid the difficulty of calculating the partial derivative of $\varphi(\bullet)$? The answer is surely true, and there are a few. We chose a method named the improved n-point secant method [11].

D. The Improved n-point Secant Method

Assume $F(X) = 0$ is a nonlinear equation, $F: D \subset R^n \to R^n$ is the nonlinear mapping from the open domain C in R^n to R^n, $f_i (i = 1, \cdots, n)$ is the component function of the vector function $F(X)$. The improved n-point secant method is an iteration method, the iteration formula is:

$$\begin{cases} X^{k+1} = X^k - A_k^{-1} F(X^k) \\ A_k = (a_{ij}^k)_{n\times n} \\ a_{ij}^k = \begin{cases} a_{ij}^{k-1} & j \neq k (\mathrm{mod}\, n) \\ \dfrac{f_i(X^k + h\|F(X^k)\| e_j) - f_i(X^k)}{h\|F(X^k)\|} & j = k (\mathrm{mod}\, n) \end{cases} \\ k = 1,2,\cdots \quad i = 1,\cdots,n \quad j = 1,\cdots,n \end{cases} \tag{1}$$

Assume the starting value of the iteration is X^0, $A_0 = (a_{ij}^0)$ can be taken as:

$$a_{ij}^0 = \frac{f_i(X^0 + h\|F(X^0)\| e_j) - f_i(X^0)}{h\|F(X^0)\|} \quad i = 1,\cdots,n, \quad j = 1,\cdots,n \tag{2}$$

where e_j is the coordinate basis vector whose jth component is 1, and h is a positive constant.

The improved n-point secant method has the following convergence theorem: If $F(X^*)=0$, $F: D \subset R^n \to R^n$ is continuous and differentiable in the open neighborhood $S_0 = S(X^*, \delta_0) \subset D$ of X^*, and $F'(X^*)$ is nonsingular; then exist $S = S(X^*, \delta) \subset S_0$, for any $X^0 \in S$ and $(X^0 + h\|F(X^0)\|e_j) \in S$, the sequence $\{X^k\}$ generated by (1) and (2) converges superlinearly at X^*. If exist $\gamma > 0$, so as to $\|F'(X) - F'(Y)\| \le \gamma \|X - Y\|, \quad \forall X, Y \in S_0$ comes into existence, then $P \ge \tau$, where P is the convergence order of $\{X^k\}$ and τ is the unique positive root of the equation $x^n - x^{n-1} - 1 = 0$.

We can draw the following conclusions from above discussion:

- The improved n-point secant method only needs to calculate $F(X)$ and $F(X + \Delta X)$, thereby can avoid the difficulty of calculating the partial derivative of $\varphi(\bullet)$.
- Compared with the fixed point iteration method, its convergence condition is less rigorous, which can be seen from its convergence theorem.
- Compared with the fixed point iteration method, it has a higher convergence rate. It has a superlinear convergence rate, while the fixed point iteration method has a linear convergence rate.

III. Numerical Tests and Results

The IEEE-30 and IEEE-57 system are used to test the algorithm, the specification of the first kind nodes is shown in the appendices and the results are showed in Table I.

TABLE I. Results of IEEE-30 and IEEE-57 System

	IEEE-30 system	**IEEE-57 system**
A1	Converge after 37 times exterior iteration.	Diverge, the computation can't continue because of overflow after 20 times exterior iteration.
A2	Converge after 9 times exterior iteration.	Converge after 5 times exterior iteration.
A3	Converge after 5 times exterior iteration.	Converge after 3 times exterior iteration.
A4	Converge after 5 times exterior iteration.	Converge after 3 times exterior iteration.

A1: Use the basic algorithm, and the PV nodes of the second kind are transformed into PQ nodes after performing Ward equivalent.

A2: Use the basic algorithm, but the PV nodes of the second kind are kept PV nodes after performing Ward equivalent.

A3: Use the improved n-point secant method, but the PV nodes of the second kind are transformed into PQ nodes after performing Ward equivalent.

A4: Use the improved n-point secant method, and the PV nodes of the second kind are kept PV nodes after performing Ward equivalent.

Under all the conditions, the input data and the convergence precision are the same. If the exterior iteration converges successfully, the maximum error of the amplitude and angle (rad) of the voltage is less than 10^{-6}, taking the results of the centralized load flow calculation algorithm as criterion.

We can draw the following conclusions from the above data:

- The improved n-point secant method can improve the convergence and convergence rate of the exterior iteration.
- If PV nodes of the second kind are kept PV nodes after performing Ward equivalent, can also improve the convergence and convergence rate of the exterior iteration.

IV. Asynchronous Parallel Calculation of Dynamic Load Flow

The above algorithms can't calculate dynamic load flow because Ward equivalent don't take the unbalance power distribution of the system into account.

A. Ward Equivalent Taking the Unbalance Power Distribution into Account

After performing Ward equivalent, the injection power on the boundary nodes $\dot{S}_B$ becomes $\tilde{\dot{S}}_B$, and we have $\tilde{\dot{S}}_B = \dot{S}_B - D\dot{S}_E$, where $D = \dot{E}_B \hat{Y}_{BE} \hat{Y}_{EE}^{-1} \dot{E}_E^{-1}$. When the electrical power network generates unbalance active power because of disturbance, the unbalance power should be distributed to all the nodes of the system according to a certain proportion. Assume the distribution proportion of the ith node is β_i, and the total unbalance power of the system is $\sum \Delta P_i$. If take the unbalance power distribution into account, $\dot{S}_B$ becomes $\dot{S}'_B = \dot{S}_B - \beta_B \sum \Delta P_i$, and $\dot{S}_E$ becomes $\dot{S}'_E = \dot{S}_E - \beta_E \sum \Delta P_i$, thereby $\tilde{\dot{S}}_B$ accordingly becomes:

$$\begin{aligned} \tilde{\dot{S}}'_B &= \dot{S}'_B - D\dot{S}'_E \\ &= \dot{S}_B - \beta_B \sum \Delta P_i - D(\dot{S}_E - \beta_E \sum \Delta P_i) \\ &= (\dot{S}_B - D\dot{S}_E) - (\beta_B - D\beta_E) \sum \Delta P_i \\ &= \tilde{\dot{S}}_B - \beta'_B \sum \Delta P_i \end{aligned} \tag{3}$$

where $\beta'_B = \beta_B - D\beta_E$. It is easy to see from (3) that β'_B is the unbalance power distribution proportion of the boundary nodes after equivalence.

B. *Asynchronous Parallel Calculation of Dynamic Load Flow Algorithm*

The asynchronous parallel calculation of dynamic load flow can be implemented with some modification of the algorithm F.

Algorithm D

D1. [Calculate the initial equivalent networks of the sub-networks] Choose an initial state of the whole network; the subsystem corresponding to every sub-network synchronously calculates the equivalent network of this sub-network using the Ward equivalent taking the unbalance power distribution into account.

D2. [Calculate the voltage of the nodes of the first kind] The coordination system gets the coordination network with the equivalent networks of all the sub-networks. The coordination system calculates $\dot{U}_F$ and the unbalance power of the system $\sum \Delta P_i$ using centralized dynamic load flow calculation algorithm. If $\dot{U}_F$ satisfies the convergence precision, current $\dot{U}_F$ is the voltage of nodes of the first kind, and the algorithm is over.

D3. [Get the state of the sub-networks] The coordination system sends $\dot{U}_F$ and $\sum \Delta P_i$ to corresponding sub-networks. The sub-networks firstly adjust the active power of the nodes based on $\sum \Delta P_i$, and then calculate their state using centralized multi-$V\theta$ -node load flow calculation algorithm based on $\dot{U}_F$.

D4. [Recalculate equivalent networks of the sub-networks] Every subsystem recalculates the equivalent network using the Ward equivalent taking the unbalance power distribution into account based on the state of the sub-network. Go to D2.

Algorithm D can implement the asynchronous parallel calculation of dynamic load flow, and the experiments indicate that the convergence and convergence rate are almost the same with algorithm F. However, algorithm D can't utilize the improved n-point secant method to improve the convergence and the convergence rate, for the improved n-point secant method needs to calculate $F(\dot{U}_F + \Delta\dot{U}_F)$, while $\sum \Delta P_i$ isn't merely determined by $\dot{U}_F$, it also depends on the injection power of all the nodes of the coordination network, it is hard to estimate $\sum \Delta P_i$ of $\dot{U}_F + \Delta\dot{U}_F$ with the state of $\dot{U}_F$.

C. *Dynamic Load Flow Calculation in Especial Condition*

The improved n-point commonly can't be used to improve the convergence and the convergence rate. However, if there are many nodes in the system whose unbalance power distribution proportion β_i is zero, we can specify the small quantity of nodes whose distribution proportion is non-zero as the nodes of the first kind. Because the sub-networks don't actually share the unbalance power, the unbalance power of the system $\sum \Delta P_i$ is independent of the state of the sub-networks, and the sub-networks needn't adjust the active power of the nodes based on $\sum \Delta P_i$. The improved n-point secant method is still applicable in this especial condition.

V. CONCLUSIONS

A new asynchronous parallel load flow calculation algorithm is presented. The experiments show that the basic algorithm can obtain the load flow of electrical power networks; the improved n-point secant method and keeping PV nodes of the second kind as PV nodes after equivalence can improve the convergence and convergence rate of the exterior iterations. The Ward equivalent taking the unbalance power distribution into account is used to further implement the asynchronous parallel calculation of dynamic load flow.

VI. APPENDIX

The first kind nodes of the IEEE-30 system are node 6 8 9 10 11 21 22 24 (bold line) which are shown in Fig. 2. Given the first kind nodes, the second kind nodes, the third kind nodes, the sub-networks and the coordination network can be accordingly obtained.

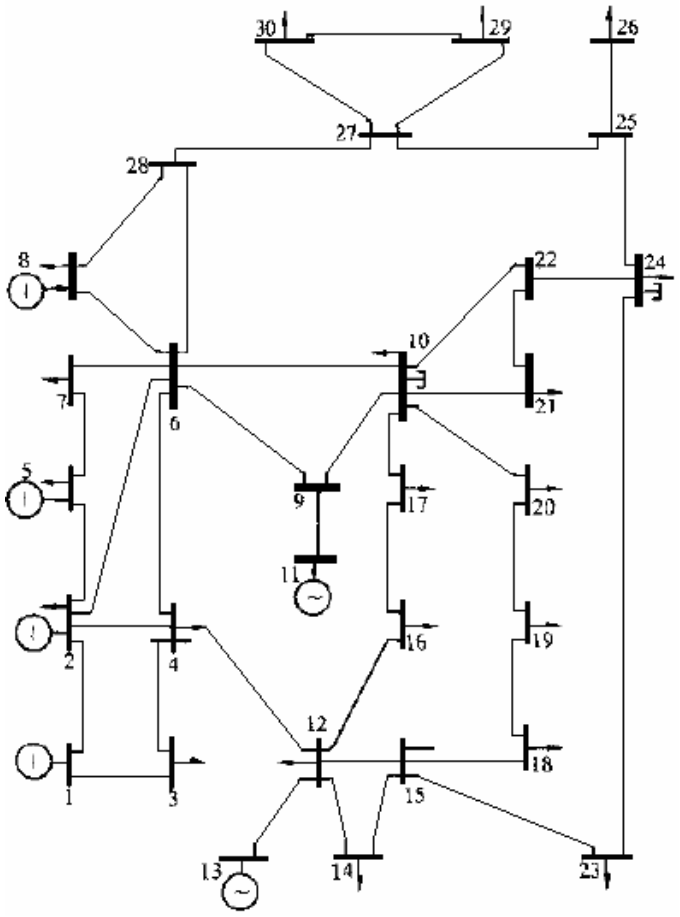

Figure 2. First kind nodes specification of the IEEE-30 system

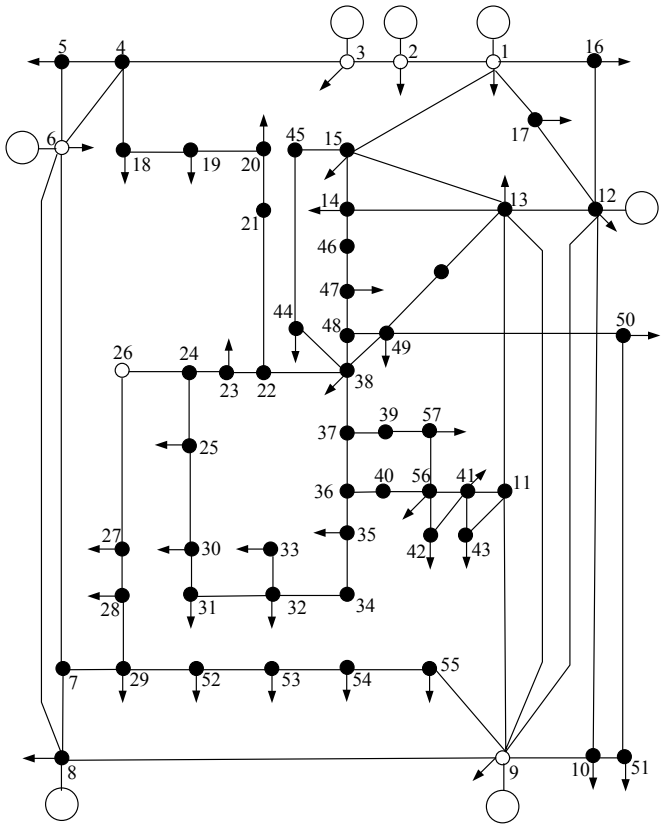

Figure 3. First kind nodes specification of the IEEE-57 system

The first kind nodes of the IEEE-57 system are node 1 2 3 6 9 26 (hollow circle) which are shown in Fig. 3.

VII. REFERENCES

[1] D.E. Barry, C. Pottle and K. Wirgau, "A Technology Assessment Study of Near Term Computer Capabilities and Their Impact on Power Flow and Stability Simulation Program," EPRI-Tps-77-749 Final Report, 1978.

[2] J. Fong and C. Pottle. "Parallel Processing of Power System Analysis Problems via Simple Parallel Microcomputer Structures," Exploring Applications of Parallel Processing to Power System Analysis Problems, EPRI-EL-566-SR, 1977.

[3] K. Lau, D.J. Tylavsky and A. Bose, "Coarse Grain Scheduling in Parallel Triangular Factorization and Solution of Power System Matrices," IEEE/PES Summer Meeting, Minneapolis, Jul. 1990.

[4] A. Abur, "A Parallel Scheme for the Forward/Backward Substitutions in Solving Sparse Linear Equations," IEEE Trans. on Power Systems, vol. 3, no. 4, pp.1471-1478, Nov. 1988.

[5] F.L. Alvarado, D.C. Yu and R. Betancourt, "Partitioned Sparse A^{-1} Methods," IEEE Trans. on Power Systems, vol. 5, no. 2, pp.452-459, May 1990.

[6] I.C. Decker, D.M. Falcao and E. Kaszkurewicz, "Conjugate Gradient Methods for Power System Dynamic Simulation on Parallel Computers," IEEE Trans. on Power Systems, vol. 11, no. 3, pp. 1218-1227, Aug. 1996.

[7] H.H. Happ, "Diakoptics - the Solution of System Problems by Tearing," IEEE Proceedings, vol. 62, no.7, pp. 930-940, Jul. 1974.

[8] F.L. Alvarado, W.F. Tinney and M.K. Enns, "Sparsity in Large-scale Network Computation," Control and Dynamic Systems, vol. 41, pp.207-272, 1991.

[9] Boming Zhang, Shousun Chen, "Advanced Power System Network Analysis", Beijing: Tsinghua University Press, 1996.

[10] Haibo Zhang, Borming Zhang, Hongbin Sun, "A Decomposition and Coordination Dynamic Power Flow Calculation for Multi-area Interconnected System Based on Asynchronous Iteration," Automation of Electric Power Systems, vol. 27, no. 21, Dec. 2003.

[11] Qingyang LI, Zizhong MO, Liqun QI, "Numerical Solutions for the Systems of Nonlinear Equations," Beijing: Science Press, 1987.