Abstract—For power system, power flow analysis is very important to enhance the system security and efficiency. An algorithm based on distributed computing is proposed for power flow analysis. In this algorithm, the coefficient matrix transformed into a bordered block diagonal form (BBDF) matrix, and each block is sent to corresponding client to calculate. In IEEE standard power systems, this algorithm has been proved more effective than Newton-Raphson iteration and decoupled algorithm.

Index Terms—distributed computing, power flow, Jacobian matrix, matrix partitions, client

I. INTRODUCTION

In power engineering, the power flow analysis is an important tool involving numerical analysis applied to a power system. The great importance of power flow analysis is in the planning the future expansion of power systems as well as in determining the best operation of existing systems. So precise power flow analysis is foundation for power system planning and investment, and fast power flow analysis is precondition for power system structure and operation [1].

With the development of society and economy, and power grid becomes more and more huge, the power flow analysis is a difficult work because the computation is enormous. So the novel algorithms are needed to improve conventional power flow algorithm.

Distributed computing is a field of computer science that studies distributed systems. In distributed computing, a problem is divided into many tasks, each of which is solved by one computer [2].

Power system has many nodes, such as generator, substation, these nodes have many own computers, and these computers have been in a network [3]. Equipped with distributed program, these computers can make up a distributed system to calculate the power flow.

In power flow analysis, distributed computing has four methods: partition method, multiple factoring method, sparse vector method and inverse matrix method. For partition method, it has specific physical meaning, and its program is easy [4]. In this paper, the distributed computing used partition method is applied to the power flow algorithm, this algorithm has been proved more effective than Newton-Raphson iteration and decoupled algorithm in IEEE standard power systems.

II. BASIC PRINCIPLES

A. Power flow equations

The power equations give the relationships between bus powers and bus voltages in term of the admittance parameters of the transmission system. In power system, there are following types of buses: there are load (P, Q) buses, generator (P, |V|) buses, and a slack or swing bus.

Power flow equations have two variables, one is voltage magnitude |V|, and the other is voltage phase θ.

In polar coordinate, the power flow equation along a line (i, j) as follows:

\[ P_i = \sum_{j=1}^{n} |V_i||V_j| (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \]

\[ Q_i = \sum_{j=1}^{n} |V_i||V_j| (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \]

In (1), the \( G_{ij} \) is conductance, and \( B_{ij} \) is susceptance.

And the equations are nonlinear equations.

At the P, Q buses, the complex voltage is unknown because we do not know \( |V| \) and \( \theta \). At the P, |V| bus, Q and \( \theta \) are unknown. The \( |V| \) and \( \theta \) variables are implicit variables in the power flow equations and iterative solution methods are required.

B. Solution of power flow

There are several different methods of solving the resulting nonlinear system of equations. One method is Gauss iteration or its variation, Gauss-Seidel iteration. The complex form of the power flow equations is used. The iteration formula remains unchanged through the entire calculation.

Another method is Newton-Raphson iteration. The real form of the power flow equation is used. The iteration formula involves a Jacobian matrix that changes as the iterations proceed [5].

For computations involving power systems under the usual operating conditions, some simplifications of the Newton-Raphson scheme are usually possible. One of these modifications is called decoupled power flow. It still requires the updating of Jacobian matrices for each iteration, but the dimensionality of the computation is reduced. Another modification is called fast-decoupled power flow. In this case, the updating of matrices is no longer required and the computational burden is greatly reduced.
Actually the above methods can be described solving large-scale sparse linear equations, and expressed as following:

\[ Jx = b \]  \hspace{1cm} (2)

In (2), \( J \) is the nodal admittance matrix, and it is called Jacobian matrix which is multi-dimensional, nonsingular, symmetrical and sparse. \( x \) is an unknown vector, \( b \) is a known vector [6].

C. Distributed computing for power flow

In order to perform distributed computing in a power system, the Jacobian matrix is transformed into a BBDF matrix [7]. Each block is sent to corresponding client to calculate. Therefore, the (2) can be formulated as follows:

\[
\begin{bmatrix}
J_{11} & \cdots & J_{1k} & J_{1n} \\
\vdots & \ddots & \vdots & \vdots \\
J_{k1} & \cdots & J_{kk} & J_{kn} \\
J_{n1} & \cdots & J_{nk} & J_{nn}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
\vdots \\
x_k \\
x_n
\end{bmatrix} =
\begin{bmatrix}
b_1 \\
\vdots \\
b_k \\
b_n
\end{bmatrix}
\]  \hspace{1cm} (3)

Where subscripts 1,\ldots, k, represent k client, and n represents the cutset block, respectively. Form (3), the unknown vector in the cutset block can be solved by the following equation:

\[ x_n = (J_{nn} - \sum_{i=1}^{k} J_{ni}J_{ii}^{-1}J_{ni})^{-1}(b_n - \sum_{i=1}^{k} J_{ni}J_{ii}^{-1}b_i) \]  \hspace{1cm} (4)

The unknown vector in each client can be solved by the following equation:

\[ x_i = J_{ii}^{-1}(b_i - J_{ni}b_n) \]  \hspace{1cm} (5)

The Jacobian matrix is divided with LU decomposition algorithm as following:

\[
\begin{bmatrix}
J_{11} & \cdots & J_{1k} & J_{1n} \\
\vdots & \ddots & \vdots & \vdots \\
J_{k1} & \cdots & J_{kk} & J_{kn} \\
J_{n1} & \cdots & J_{nk} & J_{nn}
\end{bmatrix} =
\begin{bmatrix}
L_{11} & \cdots & U_{1k} & U_{1n} \\
\vdots & \ddots & \vdots & \vdots \\
L_{k1} & \cdots & U_{kk} & U_{kn} \\
L_{n1} & \cdots & L_{nk} & L_{nn}
\end{bmatrix}
\]  \hspace{1cm} (6)

Substituting (6) into (4), the unknown vector in the cutset block can be rewritten as (7):

\[ x_n = (J_{nn} - \sum_{i=1}^{k} L_{ni}U_{ii}U_{ii}^{-1}L_{ni}U_{ini})^{-1}x = (J_{nn} - \sum_{i=1}^{k} L_{ni}U_{ii}U_{ii}^{-1}L_{ni}U_{ini})^{-1}b_n - \sum_{i=1}^{k} L_{ni}U_{ii}U_{ii}^{-1}b_i \]  \hspace{1cm} (7)

Substituting (6) into (4), the unknown vector in each client can be rewritten as (8):

\[ x_i = U_{ii}^{-1}(b_i - L_{ii}U_{ii}b_n) \]  \hspace{1cm} (8)

III. DISTRIBUTED COMPUTING FOR POWER FLOW

A. Structure of distributed computing in power system

The structure of distributed computing in power system is illustrated in Fig. 1.

In Fig.1, the computer in dispatching center is host, and the computers in substation are clients. The host and clients are connected with ethernet network. The host builds the nodal admittance matrix according to power grid, and the nodal admittance matrix is separated into several matrix partitions, and these matrix partitions are sent the client to calculate the power flow.

In separating of the nodal admittance matrix, not the host but the connection mode of the power grid decides the number of matrix partitions.

B. Flow chart of distributed computing in power system

In distributed computing, the flow chart is shown by Fig.2. First, the host forms the nodal admittance matrix, and then divides the matrix into several computers in substations, last, the client computers calculate and the result. The flow chart of distributed computing is shown by Fig.2.
Actually, in above program, a timer is set in order to avoid endless loop. The program will be exit if the timer achieves the preset number.

C. Simulation result

For the limit condition, this algorithm is simulated in Pentium (R) personal computer. In IEEE-30 bus and IEEE-39 bus standard power systems, Newton-Raphson iteration, decoupled and distributed computing is used to calculate the power flow [8].

![Flow chart of distributed computing in power system](image)

Fig. 3 presented three algorithms cost time in power flow analysis in two power system. In IEEE-30 bus, the time of Newton-Raphson is 0.125 second; the decoupled is 0.11 second, and the distributed computing is 0.96 second. And in IEEE-39 bus, Newton-Raphson is 0.151 second; the decoupled is 0.136 second, and the distributed computing is 0.118 second.

The simulation result showed that the distributed computing algorithm is effective than Newton-Raphson iteration and decoupled algorithm.

IV. CONCLUSION

Distributed computing becomes more and more popular in power flow analysis, and the power system is a symmetrical and sparse, so the BBDF is fit for the distributed computing in power system. This study presents a distributed computing algorithm with BBDF method. In this algorithm, the nodal admittance matrix is divided into several blocks with LU decomposition method, and each block is sent to corresponding client to calculate the power flow. At last, this algorithm is simulated in IEEE-30 bus and IEEE-39 bus standard power systems, the simulation result proved availability of the distributed computing algorithm.

REFERENCES