A Review on Eigen Analysis of Power System Using Parallel Computing

Sonali Pandey
Central College of Engineering and Management
Dept. of Computer Science and Engineering
Raipur, Chhattisgarh, India

Dr. P. Udaya Kumar
Central College of Engineering and Management
Dept. of Computer Science and Engineering
Raipur, Chhattisgarh, India

Abstract

In this paper, a new parallel procedure is proposed to accelerate the solution of linear systems required by iterative Eigen analysis methods. Now days Eigen analysis of power systems is frequently used to study the effect and tune the response of existing controllers, to guide the design of new controllers. Recent developments in the area lead to the necessity of studying larger power system models, resulting from the interconnection of transmission. These models include new types of controls, mainly based on power electronic interfaces, which are expected to provide significant support in the future. The consequence is that the size and complexity of these models challenge the computational efficiency of existing Eigen analysis methods. In this paper, a procedure is proposed that uses domain decomposition and parallel computing methods, to accelerate the computation of Eigen values in a selected region of the complex plane with iterative Eigen analysis methods. The proposed algorithm is validated on a small transmission system and its performance is assessed on a large-scale combined transmission and distribution system.

Keywords— *power system model, Eigen, analysis, domain decomposition, parallel computing, Open mp.*

I. INTRODUCTION

Power systems, is essential for the design, coordination, and integration of controllers. The modern day electric power system is responsible for generating, transmitting and delivering more than one third of the total consumed energy. With this progress, the complexity of the system has grown. To manage this complex system, monitoring, control and operation functions are computer assisted. The systems for computer control of electric power systems have evolved.

Eigen analysis is central to the mathematical discipline of linear (matrix) algebra, and a thorough understanding of ordination methods requires training in linear algebra. Eigen analysis is a mathematical operation on a square, symmetric matrix. A square matrix has the same number of rows as columns. A symmetric matrix is the same if you switch rows and columns. Eigen analysis consists of a series of Eigen values and eigenvectors. Each Eigen value has an Eigenvector, and there are as many Eigen vectors and Eigen values as there are rows in the initial matrix. Eigen values are usually ranked from the greatest to the least. The first Eigen value is often called the "dominant" or "leading" Eigen value. Eigen values are also often called "latent value.

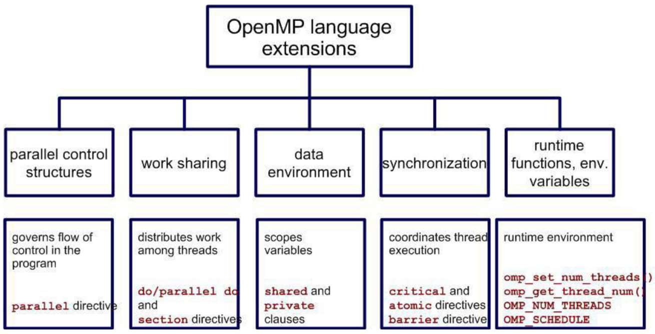
In mathematics, numerical analysis, and numerical partial differential equations, domain decomposition methods solve a boundary value problem by splitting it into smaller boundary value problems on sub domains and iterating to coordinate the solution between adjacent sub domains. A coarse problem with one or few unknowns per sub domain is used to further coordinate the solution between the sub domains globally. The problems on the sub domains are independent, which makes domain decomposition methods suitable for parallel computing. Domain decomposition methods are typically used as preconditioners for Krylov space methods. Parallel computing is a type of computation in which many calculations or the execution of processes are carriedout simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time. There are several different forms of parallel computing: bit-level, instruction-level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling. As power consumption (and consequently heat generation) by computers.

In computing, a pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion; in that case, some amount of buffer storage is often inserted between elements. A form of computer organization in which successive steps of an instruction sequence are executed in turn by a sequence of modules able to operate concurrently, so that another instruction can be begun before the previous one is finished.

The proposed procedure exploits the sparsity of the power system matrices and accelerates the solution by parallelizing the factorization. The consequence is that the size and complexity of these models challenge the computational efficiency of existing Eigen analysis methods. The decomposition of the system allows to skip some unnecessary computations. The simulation models in this paper are general and applicable to predict the performance of a multi-core.

Here we are discussing about Open Mp Tools. Open MP stands for: Open Multi-processing or Open specifications for Multi-Processing viacol laborative work between interested parties from hardware and software industry, government and academia.

Open MP multithreading, a method of parallelizing whereby a master thread (a series of instructions executed consecutively) forks a specified number of slave threads and the system divides a task among them. The threads then run concurrently, with the runtime environment allocating threads to different processors.


OpenMP language extensions
parallel control structures
work sharing
data environment
synchronization
runtime functions, env. variables
governs flow of control in the program
parallel directive
distributes work among threads
do/parallel do and section directives
scopes variables
shared and private clauses
coordinates thread execution
critical and atomic directives
barrier directive
runtime environment
omp_set_num_threads()
omp_get_thread_num()
OMP_NUM_THREADS
OMP_SCHEDULE

II. Literature Survey

A . Edge -Disjoint Hamiltonian Cycles in Transposition Networks and Hypercube Networks.

Ruo-Wei Hung, Shang-Ju Chan, and Chien-Chih Liao proposed this paper to presence of edge-disjoint Hamiltonian cycles provides an advantage when implementing algorithms that require a ring structure by allowing message traffic to bedspread evenly across the network. Edge-disjoint Hamiltonian cycles also provide the edge-fault tolerant Hamilton city of an interconnection network. Two node-disjoint cycles in a network are called equal if the number of nodes in the two cycles are the same and every node appears in one cycle . The presence of two equal node-disjoint cycles provides algorithms that require a ring structure to be performed in the network simultaneously.

In this paper, we present linear time algorithms to construct two edge-disjoint Hamiltonian cycles and two equal node-disjoint cycles in an *n*-dimensional twisted cube. Among those proposed interconnection networks, the hypercube is a popular interconnection network with many attractive properties such as regularity, symmetry, small diameter, strong connectivity, recursive construction, partition ability, and relatively low link complexity. The topology of an interconnection network is usually modeled by a graph, where nodes represent the processing elements and edges represent the communication links.

Method Used - Linear Time Algorithm

Mathematical Expression-

Let $G = (V,E)$ be a graph with node set V and edge set E.

A (simple) path *P* of length _ in *G*, denoted by *v*0 →*v*1 → · · · →*v*_−1 →*v*_, is a sequence (*v*0, *v*1, · · ·, *v*_−1, *v*_) of nodes .

The first node *v*0 and the last node *v*_ visited by *P* are denoted by *start* (*P*) and *end*(*P*), respectively. Path *v*_ →*v*_−1 → · · · →*v*1 →*v*0 is called the *reversed path*, denoted by *P*rev,of path *P*. That is, path *P*rev visits the nodes of path *P* from *end*(*P*) to *start*(*P*) sequentially. In addition, *P* is a cycle if |*V* (*P*)| _ 3 and *end*(*P*) is adjacent to *start*(*P*).

A path *P* = *v*0 →*v*1 → · · · →*v*_−1 →*v*_ may contain anothersubpath*Q*, denoted as *v*0 →*v*1 → · · · →*vi*−1 →*Q* →*vj*+1 → · · · →*v*_−1 →*v*_, where *Q* = *vi* →*vi*+1 → · · · →*vj*for 0 _ *i*_ *j* _ _.

Conclusion - In this paper, we construct two edge-disjoint Hamiltoniancycles (paths) of a*n*-dimensional twisted cubes *TQn*, forany odd integer.

B. *Statistical Definition of an Algorithm in PRAM Model.*

Sunil kumarPanigrahi proposed this paper to introduce the statistical definitions of an algorithm in Parallel Random Access Machine Model (PRAM).We present a concept of statistical analysis of 2 × 2 matrix multiplication and its summation will be executed in 2n processors. It shows conceptually how a communication delaysTaking an important role in parallel time complexity. WeCompare communications delay with networks topology such asring, mesh and hypercube.This is an aggressive research approach to measuring timeComplexity of an algorithm in parallel model. TheMathematical bound is not sufficient to compute the timeComplexity of an algorithm in parallel model.

In generalwe define a statistical bound for parallel model and it derivedfrom the defining of statistical bound for sequential algorithm. The PRAM model is suitable for study of statisticalbound; in general, the number of processors on a real parallelarchitecture is limited. There are several definitions providefor the parallel algorithm complexity.

Table-1 Comparsion between Mathematical Bounds and Statistical bounds.

Mathematical Bounds	**Statistical Bounds**
Operations are counted.	Operations are weighted.
They are system independent.	They are only system variant.
Theoritical Derivable.	They are Conceptual.
They are idea for analyzing worst case behavior(as the bounds have gurantee).	They are suitable for average case.

Method Used –PRAM model (Parallel Random Access Machine Model)

Mathematical Expression- Analysis of the algorithm executed n number steps over p number processors and havingcommunication delayd. then Tp(n) = □(n\p +d).If communication delay is negligible then the complexity isTp(n)= □ (n/p)And when n=p then Tp(n) = □(1) which is not true.

The communication delay can be evaluated as

Cj= Tmsg+ Tband+ Tprotocols

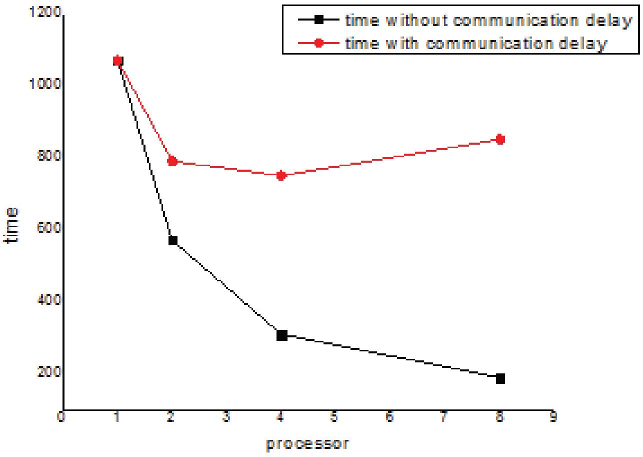


Figure-Time line chart between with and without communication delays

Conclusion - In this paper author outlined a new approach to the Statistical analysis of parallel algorithm in PRAM, and defining of several concepts. It seems that when processors increase the complexity of algorithm its increase or decrease depends upon the number of communication delays. Here we have derived the literature survey of the algorithm from KaiWang's Advance Computer Archietecture.

For further research this approach can also be applied for several algorithms. The conceptual results from the PRAM models is "worst case" results where nothing is known on the type of problem at hand; better results may be obtained with more knowledge by considering various types of PRAM models and communication networks.

C. Global commutative and associative reduction operations in faulty SIMD hypercubes.

C. S. Raghavendra and M. A. Sridhar, proposed this paper, he consider the problem of global commutative and associative operation, also known as semi group operation (such as addition and multiplication) on a faulty hypercube. In particular, we study the problem of performing such an operation in n- dimensional SIMD hypercube, Qn with up to n-1 node and/or link faults in a SIMD hypercube ,during a communication step nodes ca exchange information with their neighbour only across a specific dimention. Such algorithms are very useful in mission critical environments including medicine and space exploration.

We consider the problem of computing a global commutative and associative operation, also known as semi-group operation, (such as addition and multiplication) on a faulty hypercube. In particular, we study the problem of performing such an operation in an n-dimensional SIMD hypercube, Q, with up to *n* - **1** node and/or link faults.

The hypercube is a regular and symmetric structure that has proved very popular for parallel processing applications. Several hypercube based machines are commercially available which in-clude machines from Intel, Ncube, CM-2, and Paralex. An important area of research is to obtain efficient algorithms for parallel processing applications that operate gracefully even when some of the nodes of the cube fail. Such algorithms are very useful in mission critical environments including medicine and space exploration. Fault tolerance in machines such as hypercubes is important in order to achieve sustained high performance computing. There-fore, it is necessary to compute important primitive functions even in the presence of faults. The hypercube network is quite robust at least *n* faults are needed to disconnect *Q,* intotwo components.

Method Used - SIMD hypercube

Algorithm: The dimensions of Q, can be ordered as (d,, d,, . . ., d,) suck

that for every k, 2 5 k 5 n, every subcube induced by the dimensions

(d,, d,, . .., dk) contains at most k - 1 faulty nodes.

PROOFA.ssume that the fault set F is ordered arbitrarily, as F = (f,,

f,, . . ., fn-l). We will first prove another fact using induction:

that there is an ordering (zl, *i2,* , **zn-,)** of some ***IZ*** - 2 dimensionssuch that, for every 1,1 *5 1 5 n* - 2, no two faulty nodesamong*f*,, . . ., $+, agree in their values of the]-bit vector in positions *I*,, . .,*I,*

Define the Jth dimension *zi*as follows

a) Choose *i1* to be any dimension in which *f* ,disagrees

with*f*,.

b)For***I*t** 2, if fi+l agrees with some earlier faulty node *fi*

(I 5 J) in all the dimensions il, ***i2,*** ., *zi-**, then choose any

dmension*t*m whchh+l disagrees mth*fi,* and let *I,* = *t*.

c) If there is no such *fi,* then choose for z, any dimension that is not already chosen.

The base case, *I* = 2, is easy to verify. To show the inductivestep, note that when choosing *if,* if h+l disagrees with all of *f,,f,,* ...., ***4*** m the *(1* - 1)-bit vectors induced by bit positions zl,. , i,-,, then we can choose any dimension for ***il***that is notalready chosen, and our inductive hypothesis will be preserved.On the other hand, if there is some earlier fi withwhich*f,,,* agrees on these *1* - 1 bits, then there is at most onesuch*f,,* by the inductive hypothesis, and we only need tochoose some dimension for *zi*on which ***A+,*** and *fi*disagree.We now construct the ordering required by the theorem by essentially using the reverse ordering of the *zIs.*

Conclusion - A fundamental question he tried to address in this paper is the following –

In an n-dimensional SIMD hypercube with up to n-1 node or link faults .the faults being known a priority, what is the least number of time steps needed to acomplish afixed global semi group operation on all the faulty nodes .In this he describe three algorithm for this problem in case of node faults ,based upon a theorem concerning the existence of a certain ordering on the faulty cube’s dimension.

D. A new parallel symmetric tridiagonaleigensolver

Hiroyuki ISHIGAMI and Kinji KIMURA and Yoshimasa NAKAMURA Proposed this paper in order to accelerate the subset computation of Eigen pairs for real symmetric tridiagonal matrices on shared- memory multi-core processors, a parallel symmetric tridiagonal Eigen solver is proposed, which computes Eigen values of target matrices using the parallel bisection algorithm and computes the corresponding eigenvectors using the block inverse iteration algorithm with reortho globalization (BIR algorithm). The BIR algorithm is based on the simultaneous inverse iteration (SI) algorithm, which is a variant of the inverse iteration algorithm, and is introduced to a block parameter. The BIR algorithm is also expected to accelerate the computation of eigenvectors evenon massively parallel computers.

The proposed Eigensolver is expected to accelerate the computation of Eigen pairs even on massively parallel computers. Numerical experiments on sharedmemorymulti-core processors show that the BIRalgorithm is faster than the SI algorithm and achieves the good parallel efficiency.

Algorithm - Main routine of parallel bisection algorithm

```
$kf$= 1, $kl$= 1
repeat
    !$omp parallel do private( $j$, $r$, $c$t)
    do $k = kf, . . . , kl$
        $c$t = 0, $r = 1$
        do $j = 1, . . . ,n$
            $r = dj− e2j$
            $−1/r − μk(e0 = 0)$
            if $r ≤ p$min then
                $ck = ck + 1$, $r$ = min($r$, $−p$min)
    Check convergence & update $kf,kl$, μ
until all desired eigenvalues are computed.
```

Method Used - Bisection and inverse iteration algorithms.

Conclusion - In this he present the parallel bisection algorithm and BIR algorithm is based on the simultaneous inverse iteration algorithm and is introduced to block parameter. in addition, many cases of the numerical experiment also show that the proposed ,which is composed of the parallel bisection algorithm and the BIR algorithm, is more accurate than other Eigen solver such as QR iteration algorithm.

E. ***An Optical Multi-Mesh Hypercube.***

Ahmed Louri and Hongki Sung Proposed this paper This paper presents the OMMH topology, analyzes its architectural properties and potentials for massively parallel computing, and compares it to the hypercube. Moreover, it also presents a three-dimensional optical design methodology based on free-space optics.

The OMMH can maintain a constant node degree regardless of the increase in the network size. In addition, the flexibility of the OMMH network makes it well suited for optical implementations. This paper presents the OMMH topology, analyzes its architectural properties and potentials for massively parallel computing,

The proposed optical implementation has totally space-invariant connection patterns at every node, which enables the OMMH to be highly amenable to opticalimplementation using simple and efficient large space-bandwidth product space-invariant optical elements.

Method Used – Optical Multimesh Hypercube

OMMH network

with the wrap-around mesh can be described as follows:

Rule 1

ommh,,(z,j,k) = ((i+ 1)mod **1 , j , k**)

ommh,,(i,j,k) = ((I + i- 1)mod 1.j.k)

ommhm3(i,j,k) = (i, (j + 1)mod m , k)

ommhm4(i , jk**,**) = (i ,(m+ j - 1)mod m, k)

ommhcd(i , j ,5 -1 . . . kd+lkdkd-1 ' . ' k o) =

(Z , j , k,-l " . kd+lkdkd-l ... ko), ford = 0,1, ..', 72- 1,

whereknPl .' . kd+lkdk&l . . 'kois

representation of integer k.

a binary

The first four interconnection functions, **ommh,,** ,**ommh,,** ,**ommh m3,** and **ommh,,** , are for the four-nearest-neighbor connectionsincluding wrap-around connections and **ommhCd,** ford = 0, 1, . . . ,n - 1, determines the hypercube interconnection.

Conclusion - To overcome the lack of scalability in the regular hypercubenetworks, a new interconnection network topology, called an Optical Multi-Mesh Hypercube, is presented. The proposednetwork is a combination of hypercube and mesh topologies. The analysis and simulation results show that the new interconnectionnetwork is very scalable, meaning the configuration of the existing nodes is relatively insensitive to the growth of the network size.

III. CONCLUSION

We trying to Increasing the speed of single core processors created more heat and produce higher power consumption.

Multicore architectures are proposed for their capability to provide more processing power than single core processors, without increasing heat and power usage.

Eigen analysis methods have been used in power systems for stability analysis and the design, tuning, and coordination of controllers. At the same time, the mode shape and participation factors of these mode scan help identify the control parameters that need tuning or coordination.

It was implemented using the shared-memory parallel model with the use of OpenMP API and tested on a multi-core desktop computer.

When small and medium-scale systems are considered, full space methods are used to compute all the Eigen values and eigenvectors.

REFERENCE

[1] G. Angelidis and A. Semlyen, “Improved methodologies for the calculationof critical eigenvalues in small signal stability analysis,” IEEE Transactions on Power Systems, vol. 11, no. 3, pp. 1209–1217, 1996.

[2] S. K. Panigrahi, and S. Chakraborty, “Statistical definition of an algorithm in PRAM model & analysis of 2 x 2 matrix multiplication in 2^n processors using different networks,” Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014, pp. 717–724.

[3] X. Li, X. Yang, L. He, J. Zhang, and C. Yu, “Conditional diagnosability of optical multi-mesh hypercube networks under the comparison diagnosis model,” Theoretical Computer Science, vol. 531, 2014, pp. 47–53.

[4] J. Zhou, Z. Wu, S. Yang, and K. Yuan, “Symmetric property and reliability of balanced hypercube,” IEEE Transactions on Computers, vol. 64(3), 2013, pp. 876–881.

[5] C. S. Raghavendra and M. A. Sridhar, "Global commutative and associative reduction operations in faulty SIMDhypercubes," in IEEE Transactions on Computers, vol. 45, pp. 495-498, April 1996.

[6] X. Zhang and C. Shen, “A Distributed-computing-based Eigenvalue Algorithm for Stability Analysis of Large-scale Power Systems,” in 2006 International Conference on Power System Technology, oct 2006.

[7] Hiroyuki ISHIGAMI and Kinji KIMURA and Yoshimasa NAKAMURAA New Parallel Symmetric Tridiagonal Eigen solver based on Bisection and Inverse Iteration Algorithms for Shared-memory Multi-core Processors, 2015 10th International Conference on, Parallel, Grid, Cloud and Internet Computing, pp 216-223.

[8] AdibHaron, Jintao Yu, RazvanNane, MottaqiallahTaouil, Said Hamdioui, KoenBertels “Parallel Matrix Multiplication on Memristor Based Computation-in-Memory Architecture” 978-1-5090-2088-1/16/$31.00 ©2016 IEEE. PP.759-766.