CMES, vol.19, no.2, pp.135-143, 2007

The Parallel Mechanism of Node-Based Seamless Finite Element Method

Y.F. Nie[1], **S. Chang**[1] **and X.K. Fan**[1]

Abstract: A new parallel mechanism for Node-based Seamless Finite Element Method was proposed in this paper, which possessed the following three prominent points: realizing the workload balance for the parallel processes naturally, achieving synchronization of all the schedules under complex parallel environment, and filling up the gap between pre-processing and main processing. To support the scheme, three specific solutions of the parallel mechanism were proposed in this paper, all of which achieved the highly efficient parallel seamless connection between the FEM mesh generation process and structure analysis process. Two of the three schemes, i.e. dynamic scheme and self-adaptive scheme, enabled both schedules achieving synchronization of the whole process counting from FEM mesh generation to global stiffness matrix computation. The good tolerance of the two schemes to processor failure was also pointed out here. The parallel efficiency of the schemes was shown by both the theory analysis and initial parallel numerical examples results.

Keyword: Grid generation, Seamless connection, Finite element method, Speedup ratio, Parallel efficiency.

1 Introduction

As the increasing scale of the finite element computing [Bielak etc (2005), Hassan etc (2004)] and rapid development of parallel computers, the parallel techniques of finite element method (FEM) have been put into unprecedented attention. In the aspect of parallel FEM structure analysis, much work has been done, some of which is amazing [Li (1995), Ruan (2005)]. A review paper about mesh quality is given by Du etc (2006). Meanwhile, the pre-processing of FEM, especially parallel mesh generation techniques has turned into hot topic, which has already been discussed and researched widely and deeply [Mackerle (2003), Nave etc (2004), Chernikov etc (2006)].

The parallel Delaunay Triangulation of very Large Scale (50 million elements) was achieved in reference Said, Weatherill etc. (1999). Chew etc (1997) proposed parallel constraint Delaunay triangulation which could efficiently reduce the amount of communication. An adaptive analysis method for parallel auto generation of the unstructured grid was proposed by Shephard (1997) and Yuan etc (2006). However, the structure analysis process of the traditional FEM can not be done until the completion of the FEM grid generation. This kind of serial characteristic inevitably seriously restricts the parallel efficiency and becomes one of the bottlenecks of the parallel FEM computing. Compared with meshfree method which can be design as a parallel system very naturally especially as MLPG are used to solve partial differential equations [Atluri (2004), Cartwright (2003), Ma etc (2007)], FEM need to be developed to break the seam of grid generation and stiffness evaluations.

Node-based Finite element method [Yagawa (2004)] provided the possibility of achieving seamless connection between FEM structure analysis process and FEM grid generation process. Nie & Chang (2006) and Chang & Nie (2005) deeply discussed Node-based Finite element method (NB_FEM) as well as proposing node-based local grid generation algorithm, which is free of grid inconsistency. The node-based domain-partitioning algorithm that utilized constraint Delaunay paths to partition arbitrary

[1] Department of Applied Mathematics, School of Science, Northwestern Polytechnical University, Xi'an 710072, PR China. E-mail: Yfnie@nwpu.edu.cn

2D Domain was proposed in Chang & Nie (2005). This algorithm achieved zero-communication in the process of the parallel grid generation, thus the parallel efficiency of the parallel FEM grid generation was enhanced. However, as to complex domain, it still needs great efforts to insure the synchronizing completion of the grid generation in each sub-region. Some work has been given in Nie, Fan & Chang (2006).

In this paper a new mechanism of parallel FEM is proposed on the basis of our former work, which is able to achieve the synchronization of every computing process naturally, including realizing workload balance of different schedule naturally, synchronizing completion of all the schedules naturally under complex parallel environment (for example, processors are asked to deal with processes from different jobs). In addition, the new mechanism can break off the defects of the present parallel algorithms that the whole domain mesh generation process must be carried out firstly, which the structure analysis is waiting for. Considering the core of the mechanism is nodes, we called it the parallel schemes of Node-based Seamless Finite Element Method (NBS_FEM). The mechanism is carried out with three following specific schemes: the static scheme, the dynamic scheme and the self-adaptive scheme. The high performance parallel seamless connection between the FEM mesh generation process and the FEM structure analysis process is attributed to this new mechanism which is able to realize the load balance and minimization of the parallel communication cost.

The current parallel computers can be divided into Shared Memory-MIMD system and Distributed Memory MIMD system [Du (2001)]. The Massively Parallel Processing connected by the special network coupling with a number of computing nodes (MPP), the Cluster Of Workstation that is made up of high performance processors via computer network connection (COW), are both classic distributed memory computer system possesses, which has great expansibility and high computing performance [Chen (1999)], Both MPP and COW have gradually become hot research area in the fields of high performance computer. The discussion in this paper will focus on Distributed Memory MIMD system.

This paper is organized as follows. In Section 2, NB_FEM method is introduced as well as NLMG algorithms which are the foundation of parallel scheme of NBS_FEM. In Section 3, NBS parallel mechanism of FEM is proposed with three specific parallel solutions. The dynamic scheme achieves the load balance of processor computation, and all processors almost simultanously complete the computation from FEM grid generation to computation of stiffness matrixes. In addition, the fault-tolerance is also shown in the dynamic scheme. As for static scheme, it achieves zero communication cost among processors, while adaptive scheme is a compromise between the parallel efficiency and fault-tolerance between dynamic solution and static solution. Some further discussion of NBS parallel scheme and numerical examples are given in the subsequent two sections. At last conclusion is reached.

2 Node-based local mesh generation algorithm

The current parallel mesh generation algorithms can mainly be classified into three types [De Cougny (1999)] as follows: (1) the boundaries of the sub regions are meshed in advance. The algorithm proposed by Chang, Nie (2005) and Blelloch, Miller, Tlmor (1996), which utilized Constraint Delaunay paths to partition the domain belongs to this type. (2) The boundaries of the sub regions are meshed later, which need more efforts on the merge of the sub region boundary meshes very carefully [Guan and Song (2003)]. (3) Sub region mesh generation proceeds simultaneously with boundary mesh generation. The first two types of algorithms must take more extra parallel spending to partition sub region boundaries or deal with the merge of boundary meshes so that the parallel efficiency is restricted seriously while the third will take more parallel spending on communications among those nodes. The available parallel mesh generation algorithms aren't able to connect seamlessly with FEM structure analysis process of being an integrated body. FEM structure analysis process has to wait until mesh generation process finished, thus the entire parallel efficiency of the FEM parallel computing is not so efficient.

For node-based finite element method (NB_FEM), the elements of every row in global stiffness matrix are determined by corresponding element patches which are generated by a central node as well as the corresponding neighbor nodes (the satellite points) [Nie and Chang (2006)]. So one of the important techniques in NB_FEM algorithm is the local mesh generation, i.e. the triangular element patches are generated in the local areas of each central node independently. In order to ensure the mesh quality, another important point is we expect the union of all element patches generated by NLMG algorithm is a kind of optimum mesh for the global domain in some sense. In order to realize the above mentioned goals and avoid mesh inconsistency phenomenon [Yagawa (2004)], Nie and Chang (2006) show us a kind of partition method to find the optimal searching circle radius for the local neighborhood region which has an initial searching circle radius, and the method facilities the NLMG algorithm to obtain a kind of high quality mesh, i.e. the union of the whole element patches generated in every local neighborhood region is equal to the Delaunay partition of the global domain. Even though how to give the initial searching circle radius of each central node becomes a crucial step of NLMG algorithm, because of the restriction of article length, we will discuss it combined with bucket technique [Bentley, Weide and Yao (1980)] in another paper.

3 NBS parallel mechanism

In this section, we put forward the new parallel mechanism of FEM based on NLMG algorithm. The new mechanism adopts node-based task allocation solutions, which are free of domain partition. The process shown in Figure 1 is called a Node Processing Unit (NPU). The parallel algorithm, which takes NPU as a base unit, is named as Node-Based Seamless Finite Element Method. According to the node allocation method, we present three specific implementing schemes, that is, dynamic allocation, static allocation and selfadaption allocation.

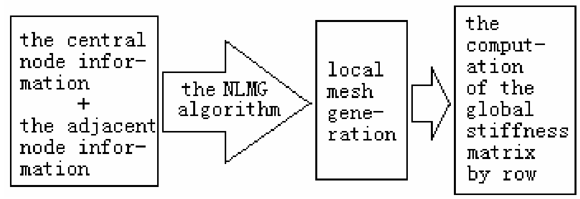

Figure 1: Node Processing Unit

3.1 Dynamic allocation scheme

The task scheduling of the NBS dynamic allocation adopts manager/worker mode.

Put the sequence number of all the nodes into the public queue, which are saved in the memory of the manager processor so that they can be dispatch by the manager. After the data structure construction of both the manager processor and the worker processors are finished, each worker will ask for task assignment, meanwhile, once the manager receives the requests, he will send a package of central nodes (only one node on extreme) to the requesting worker. As soon as any worker processor receives a node list, the node-based local mesh generation of this node list and the computation of the rows of global stiffness matrix corresponding to this node list will be carried out, which is to say, a batch of NPU will be proceeded. Once the computation is finished, the worker will ask for another new task assignment. This process will be repeated until the public queue is empty.

Dynamic allocation solution (center in Figure 2) enable each processor always keep in a fully busy state during the whole FEM parallel computing, despite that the processing capacity of each processor varies or part of resources of the processor have been occupied or the mesh generation and the cost of computing and assembling element stiffness matrices with respect to each node is different. Theoretically, this solution can naturally achieve the real load balance that each processor almost finishes the computation simultaneously. The unique communication stems from the process which the manager dispatch the nodes information in the public queue to the workers. Hence, the amount of communication is very limited and is independent of the number of processors. For the sake of minimizing the communication time, and achieving the synchronization completion under complex parallel environment, the number of the central node that the manager sends each time need to be evaluated, which depend on the specific network environment.

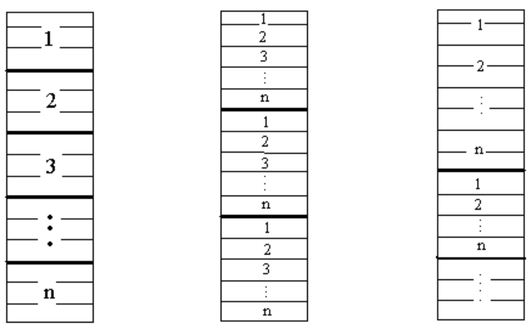

Figure 2: Task partition describing

We can compare this parallel scheme to the solution of ants carrying a pile of food. Even though the carry ability of the ants differs or the weight of each piece of food differs, these ants are able to keep the most efficient working state as long as each ant is in a busy state all the time and does useless work as little as possible. Hence, this pile of food can be carried away in a shortest time.

3.2 Static allocation scheme

The task scheduling of the NBS static allocation adopted non-manager/worker mode.

If the data and data structure of each processor is completed, all nodes will be allocated to each processor on average, that is to say, the number of difference among the number of nodes allocating to each processor is no more than one. Each processor will put the allocated nodes to its own task queue. When node allocation is finished, a node will be popped up from the task queue and the NPU process based on this node will be performed. Repeating this process until the task queue is empty.

As for the static allocation scheme (left in Figure 2), each processor is independent and need not any message communication during the whole process of local mesh generation and the element stiffness matrices computation and assemble. Thus zero-communication is achieved. However, due to the differences of the computation cost among those nodes and the differences of computational ability among processors, this scheme cannot insure the synchronizing computation of the sub structure. High efficiency can be achieved when the differences mentioned above are negligible.

3.3 Self-adaptive allocation scheme

Dynamic allocation scheme achieved load balancing among those worker processors, but we did it at the expense of assigning an extra manager processor, which is specially in charge of scheduling. (We will discuss the case that the manager takes a share in computation on the next section). Besides, if the number of worker processors is big, the frequent communications between the manager and the workers may become a bottleneck. With respect to the static allocation scheme, although it achieves zero-communications during the entire computation process, load imbalance will happen and the parallel speedup will decrease sharply when the number of the nodes allocated to each processor is nearly the same but the cost of computation varies a lot. Considering the aforementioned analysis, the expect of load balancing, the manager sharing computation and least communication as possible leads us to develop self adaptive allocation scheme which is a compromise of the dynamic scheme and the static scheme (right in Figure 2).

The new scheme can be described as follows. Firstly let us divide the public queue T into two parts, i.e. the upper part T1 and the lower part T2. Then T1 is allocated to each worker on average statically. As for T2, we can consider it as a new task queue. Now we divide T2 as two parts once again and deal with the two parts as the same like way, i.e. the upper part of T2 is allocated to those idle workers who have finished the previous static task while the lower part of T2 can be treated as a new queue. Continue the above-mentioned process until T2 task can be completed by one processor in a very short time.

4 The further discussion of the NBS parallel scheme

It is well known that the key to obtain good parallel efficiency lies in how to realize load balance and minimize the parallel expense [Du (2001)]. This is because the parallel computation time is determined by the processor which finishes the computation task most slowly and the main parallel expenses is caused by the message passing in Distributed Memory MIMD system.

The key of the NBS dynamic scheme achieving high parallel efficiency attributed to the realization of the load balancing naturally. As to the former parallel computation of FEM, the parallel optimizing goal is the balance of the number elements or the nodes in the substructure that is allocated to each processor (which is equal to the static allocation scheme mentioned on last section). However, the load balance of the number of the nodes in each processor doesn't mean to the load balance of computation cost. The load balance of the number of the nodes can also cause distinct differences of computation time for every processor, so that the parallel efficiency is restricted. But as for the dynamic allocation solution, processor will have one node list delay on the worst cases. Even if this happened, it would not affect the entire parallel efficiency because comparing with run time of each process one node list delay is subtle.

As mentioned above, dynamic allocation scheme achieved the load balancing among those worker processors. But one processor has to be acted as the manager who is in charge of dispatching the tasks. On the one hand, if the manager only takes charge of allocating computation task and doesn't join any computation while it is vacant, there is no doubt that it would be a waste of resources, thus the entire parallel efficiency will decrease. On the other hand, if the manager takes too much computation tasks, the communications between the manager and the workers will inevitably be affected and thus, the wait and delays will happen. Accordingly, the manager should first respond to the requests of the workers and on this premise, the manager need to participate in computation as much as possible.

Fault-tolerance is needed to be considered for the distributed computation system. The so-called fault-tolerance or robust characteristic means that the designed algorithms are able to run even though some processors have already failed. In our computing environment COW(the Cluster Of Workstation), users share the cluster computation system by solution of the Myrinet network, when all the resources of processor are taken up by some users, this processor will be not available to the others and processor allocation failures will occur.

The fault-tolerance of the dynamic allocation scheme of the new parallel mechanism is satisfying. Certainly we must first insure that the manager worker has a good performance (and it is easy to be done). If an exception occurs in a worker processor (the resources is taken up or interrupted), it will not request for computation any more and the manager will not allocate new task to this worker. Due to each worker processor working independently, the worker, which is in exception, will not affect the communication and the computation of other good worker processors. And the normal workers will share the tasks, which should have been taken by the exceptional worker. On the other hand, if the exceptional worker becomes normal again during the computation, it can continue to request task. It is no other than the "dynamic character" of the dynamic scheme that enables this new scheme possessing great fault-tolerance.

As for the NBS static allocation scheme, the nodes are distributed to each processor preliminarily and uniformly. Processor achieves zero communications during the entire parallel process. If the amount of computation in each processor is nearly the same and the computing capacity of each processor is equivalent, this static solution will be better than the other two solutions, because of zero- communications and realization of the load balance. It should be noticed that the fault-tolerance of NBS static allocation scheme is not so good.

The adaptive allocation scheme of the NBS parallel mechanism, which allows the manager takes a share in computation and doesn't affect the communications between the manager and the workers on the mean time, compromises the virtues of the dynamic solution such as load-balancing and good fault-tolerance and the static solution such as few communications. Consequently, the performances of parallel efficiency and the fault-tolerance are between the dynamic scheme and the static scheme on general. But it is not absolute as there are many reasons that affect the parallel efficiency (seeing the numerical examples 1 and 2).

5 The parallel numerical experiments

The parallel numerical experiments are performed to test the parallel performances of these new FEM parallel schemes, including the parallel speedup ratio and the parallel efficiency. The parallel experiment environment is Hp rx 2600 cluster system. It is composed of 40 computing nodes and 2 managing nodes (each node is a HP rx2600 framework server), which possesses the high speed computing network via Myrinet network. We adopt the MPI as our parallel programming mode and C++ language programs.

We now take the two-dimensional Laplace equation as an example to illustrate that the new FEM parallel schemes achieving the seamless linking from the node-based local mesh generation to the global stiffness matrix assemble. As regards the seamless linking from the form of the global stiffness matrix to the solving of the global stiffness matrix, we will show it in another paper. The problem model is as follows:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0, & (x,y)\in\Omega \\ u|_{\partial\Omega}=0, & \Omega \text{ is an arbitrary two-dimen-} \\ & \text{-sional domain} \end{cases}$$

Example 1: Given 500 nodes generated randomly in the given multi-connected domain. The dynamic allocation and static allocation schemes are performed to realize the mesh generation and the forming of the global stiffness matrix. The parallel mesh grid is shown in Figure 3.

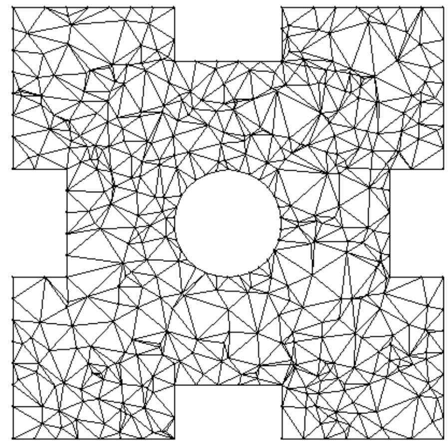

Figure 3: The parallel constraint Delaunay triangulation in the multi-connected domain

As for the dynamic allocation scheme (the manager joins in computation), 2 processors are assigned, the speedup ratio is 1.80351 and the parallel efficiency is 90.18%, As for the static allocation scheme and 2 processors assigned, the results are 1.82589 and 91.29% respectively.

Example 2: Given 10,000 uniformly distributed nodes randomly in the given rectangle domain. Three new schemes are applied respectively. We assign 2,4,6,8 processors with respect. As to the adaptive allocation solution, half of the nodes that is, 5000 nodes are to be allocated statically. The parallel computing results of the three solutions are presented in Table 1,2 and 3 as follows.

Example 3: Given 100,000 uniformly distributed nodes randomly in the given rectangle domain, Three new scheme solutions are applied respectively. 2,4,6,8 processors are assigned with respect. Similarly, as to the adaptive allocation solution, half of the nodes that is, 50,000 nodes are to be allocated statically. The parallel computing results of the three solutions are presented in Table 4,5 and 6 as follows.

Table 1: The dynamic allocation scheme

The number of processors	2	4	6	8
speedup ratio	1.91202	3.71995	5.53121	7.12608
parallel efficiency	95.60%	93.00%	92.17%	89.08%

Table 2: The static allocation scheme

The number of processors	2	4	6	8
speedup ratio	1.92950	3.87832	5.69413	7.24877
parallel efficiency	96.48%	96.96%	94.90%	90.61%

Table 3: The self-adoption allocation scheme

The number of processors	2	4	6	8
speedup ratio	1.91124	3.74393	5.55102	7.20344
parallel efficiency	95.56%	93.60%	92.52%	90.04%

Table 4: The dynamic allocation scheme

The number of processors	2	4	6	8
speedup ratio	1.95460	3.67781	5.32249	7.13953
parallel efficiency	97.73%	91.95%	88.71%	89.24%

Table 5: The static allocation scheme

The number of processors	2	4	6	8
speedup ratio	1.96341	3.70677	5.41936	7.35116
parallel efficiency	98.17%	92.67%	90.32%	91.89%

Table 6: The self-adaption allocation scheme

The number of processors	2	4	6	8
speedup ratio	1.95822	3.66035	5.32609	7.18981
parallel efficiency	97.91%	91.51%	88.77%	89.87%

6 Conclusion

The parallel mechanism of node-based seamless finite element method proposed in this paper is feasible which can naturally achieve the seamless connection from mesh generation to stiffness matrix assemble. Moreover, the workload balance in each process and almost synchronization of all the processes under complex parallel environment could be achieved naturally. The three parallel solutions to realize the new parallel mechanism are studied and compared. Each scheme has its own advantages and disadvantages on different conditions. And the adaptive allocation solution is much flexible and expandable. The preliminary parallel experimentations indicate that when the number of processors is 8, the parallel efficiency can still be up to 90% or so. The further research about this new mechanism is under way.

Acknowledgement: The main idea of this paper appeared during my visiting CARE, University of California at Irvine. This work was supported by Program for New Century Excellent Talents in University of China, Natural Science Basic Research Plan in Shaanxi Province of China (No. 2006A05), and National Natural Science Foundation of China (No. 90405016).

References

Atluri S.N. (2004): The meshless methods (MLPG) for domain & BIE discretization. Tech. Science Press

Bentley J.L., Weide B.W., Yao A. C. (1980): Optimal expected time algorithms for closest point problems. *ACM Transactions on Mathematical Software*. Vol. 6, pp.563-580

Bielak J., Ghattas O., Kim E.-J. (2005): Parallel octree-based finite element method for large-scale earthquake ground motion simulation. *CMES - Computer Modeling in Engineering and Sciences*. Vol. 10, pp. 99-112

Blelloch G, Miller. G, Talmor D. (1996): Developing a practical projection-based parallel Delaunay algorithm . Proc. ACM Symposium on Computational Geometry.

Cartwright C.K. (2003): A Parallel Algorithm for Matrix Assembly in Meshfree Methods. PH. D. Thesis. The University of Iowa

Chernikov A.N., Chrisochoides N.P. (2006) : Parallel guaranteed quality Delaunay uniform mesh refinement. *SIAM Journal on Scientific Computing*.Vol.28, pp.1907-1926

Chang S, Nie Y F. (2005): Node-Based Local Mesh Generation Algorithm within An Arbitrary 2D Domain. *Acta Aeronautica et Astronautica Sinica*, Vol. 26, pp.556-561

Chen G.L. (1999): Parallel computation— structure, algorithm and programming. Beijing: Higher education publisher. pp.83-102, pp164-179

Chew L. (1997): Parallel constrained Delaunay meshing. American Society of Mechanical Engineers, *Applied Mechanism Division*, Vol. 220, Trends in Unstructured Mesh Generation, pp.89-96

De Cougny H L, Shephard M S. (1999): Parallel volume meshing using face removals and hierarchical repartitioning. *Computer Methods in Applied Mechanism and Engineering*. Vol.174, pp.275 $\sim$ 298

Du ZH. (2001): MPI Parallel program designing. Tsinghua University Press

Du Q.,Wang D.S. (2006): Recent progress in robust and quality Delaunay mesh generation. *Journal of Computational and Applied Mathematics*. Vol.195, pp.8–23

Guan Z.Q., Song C. (2003): The new research progresses on the finite element mesh generation. *Chinese Journal of Computer-Aided Design & Computer Graphics*. Vol. 15, pp.1-14

Hassan O.; Morgan K.; Jones J.; Larwood B.; Weatherill N.P. (2004): Parallel 3D time domain electromagnetic scattering simulations on unstructured meshes. *CMES - Computer Modeling in Engineering and Sciences*. Vol. 5, pp. 383-393

Li X.M. et al (1995): The parallel computing environment and finite element analysis parallel algorithm. *Chinese Science Fund*. Vol.1, pp.15-21

Ma J.F., Krishnaswami P., Xin X.J. (2007): A truly meshless pre- and post-processor for meshless analysis methods. *Advances in Engineering Software*. Vol. 38, pp.9-30

Mackerle J. (2003): FEM and BEM parallel processing: Theory and applications- a bibliography (1996-2002). *Engineering Computations* .Vol. 20, pp.436-484

Nave D., Chrisochoides N., Chew L.P. (2004): Guaranteed-quality parallel Delaunay refinement for restricted polyhedral domains. *Computational Geometry*. Vol.28, pp.191-215

Nie Y.F., Chang S. (2006): Node-based local mesh generation algorithm. *Chinese Journal of Computational Mechanism*. Vol. 23, pp.252-256

Nie Y.F., Fan X.K., Chang S., Yuan Z.B. (2006): New and efficient node-based local mesh generation parallel algorithm. *Journal of Northwestern Polytechnical University*. Vol.24, pp731-735

Ruan H.H., Yuan Y., Liu M. (2005): The progress of parallel finite element research based on the distributed Storage environmen. *Journal of Tongji University*. Vol. 33, pp.21-27

Said N.A., Weatherill R., Morgan N.P., Verhoeven K. (1999): Distributed parallel delaunay mesh generation. *Computer Methods in Applied Mechanism and Engineering.* Vol. 177, pp.109-125

Shephard M.S. (1997): Parallel automatic adaptive analysis. *Parallel Computing.*Vol. 23, pp.1327-1347

Yagawa G. (2004) Node-by-node parallel finite elements: A virtually meshless method. *International Journal for Numerical Methods in Engineering*, Vol. 60, pp.69-102

Yuan Y., Guo Q., Peng Z. (2006): Adaptive dynamic parallel Delaunay mesh generation with minimal communication load. *Chinese Journal of Electronics*. Vol.15, pp.399-402