To appear in the proceedings of *Parallel Computational Fluid Dynamics' 93, Paris, France.*

PARALLEL GRID GENERATION ON DISTRIBUTED MEMORY MIMD MACHINES FOR 3-DIMENSIONAL GENERAL DOMAINS

Nikos Chrisochoides,* Geoffrey Fox,* Joe Thompson**
*Northeast Parallel Architectures Center, Syracuse University
111 College Place, Syracuse, NY, 13244-4100
**Engineering Research Center for Computational Field Simulation, Mississippi State University
P.O. Box Drawer 6176, Mississippi State, MS 39762
corresponding author: Nikos Chrisochoides (nikos@npac.syr.edu)

Extended Abstract

The mapping of sequentially generated irregular meshes onto distributed memory MIMD/SIMD machines has been studied extensively in the scientific computing literature; a list of very interesting results, which is by no means complete, appear in [Bokh 81], [Fox 86], [Morr 87], [Sada 87], [Farh 88], [Simm 91], [Lori 90], [Stev 92], [Mans 92], and [Chri 92]. In this paper we study the *parallel generation of structured grids for 3-dimensional complex domains.* Specifically, we present a parallel grid generation method based on *composite block structures*, and a strategy for the mapping of the computations associated to the parallel grid generation onto distributed memory MIMD machines (multiprocessor systems).

The parallel grid generation method for a given 3-dimensional domain Ω is described by the following four steps :

1. Decompose the physical domain, Ω, using the NGP interactive environment (see in [Thom 91]), into a small number of contiguous hexahedron subregions, Ω_i, that can be mapped into a rectangular computational blocks B_i which form an initial composite block structure $C_0(\Omega) = \{B_i\}_{i=1}^{N}$.

2. Generate sequentially an Algebraic grid that provides an explicit control of the physical grid shape and requires a minimal number of grid points. This grid is used as a background for the finer composite block structure, $C_f(\Omega) = \{B'_i\}_{i=1}^{N'}$ that satisfies the following three properties :
$$| C_0(\Omega) | < | C_f(\Omega) |$$
$$\forall B_i \in C_0(\Omega)\ \exists I_i \subset \aleph\ \ni: B_i = \cup_{j \in I_i} B'_j,\ B'_j \in C_f(\Omega)$$
$$|B'_i| = |B'_j|\ \ \forall B'_i, B'_j \in C_f(\Omega)$$

3. Generate in parallel a finer Algebraic grid, using $C_f(\Omega)$, that provides an explicit control of the physical grid spacing.

4. Generate in parallel the final grid using the fine Algebraic grid of step 3 as an initial solution to start the iterative Elliptic grid generation.

Steps 1 to 3 require limited amount of computation while the step 4 is the most time-consuming. The computation required for the parallel Elliptic grid generation can be described by a loosely synchronous computational model (i.e., a sequence of computation-communication steps that almost periodically require a global synchronization). A careful mapping of the composite block structure $C_f(\Omega)$ can reduce the communication overhead and minimize the overall execution time for the parallel grid generation.

The mapping of such computations onto distributed memory MIMD machines is a difficult combinatorial optimization problem. Strategies for the mapping of such computations onto distributed memory machines can be applied to different levels (i.e., geometric, algebraic) of the grid generation process. We propose and study geometry based mapping heuristics based on a composite block structure. Our goal is to map these computations onto the processors of a distributed memory MIMD machine so that the workload of the processors is balanced and the required communication and synchronization among the processors is minimum. Such mappings will be used to parallelize the grid generation modules of the National Grid Project [Thom 91].

The basic idea of the grid generation using a composite block structure is based on the decomposition of the physical domain Ω into contiguous hexahedron subregions Ω_i which are mapped to rectangular computational blocks, B_i (see in [Thom 84] for a comprehensive survey of this method). In each of the computational blocks an independent curvilinear coordinate system (ξ^1, ξ^2, ξ^3) is generated. Note that (i) the size (i.e., the number of grid points) of those computational blocks may vary and (ii) the number of the subregions and thus of the computational blocks usually is smaller than the number of the available processors for large and massively scale parallel machines. The grid of the full domain is generated by composing "properly" the separate coordinate systems of the computational blocks. This composition requires an interaction between adjacent rectangular blocks.

For each block B_i an independent curvilinear coordinate system can be generated in parallel. The degree of continuity of grid lines across the interfaces of adjacent curvilinear systems requires either the specification of grid points at the same fixed locations on both of the adjacent coordinate systems (case of discontinuous grid line slope) or the treatment of grid lines as a branch cut on which the generation system is solved just as it is in the interior of the blocks (case of continuous grid line slope). In this case the interface locations are determined by the grid generation system.

The continuity of grid line slope is handled by providing an extra layer of points (outer-layer) surrounding each block. The interface and outer-layer grid points of a block are forced to coincide with the interface and interior grid points of an other adjacent block. This coincidence of the points is maintained during the course of an iterative solution of an elliptic system over all blocks. This suggest that a local synchronization among the processors that process adjacent blocks is required at the end of each iteration.

The mapping of a composite block structure $C_f(\Omega)$ onto distributed memory MIMD machines so that the workload of the processors is balanced and the required communication and synchronization among the processors is minimum, can be formulated by :

$$\min_{m} \max_{1 \leq i \leq P} \{ \; W(m(D_i)) + \sum_{D_j \in C_{D_i}} C(m(D_i), m(D_j)) \; \} \qquad (O)$$

where $D_i = \{ \, B_j | \; B_j \in C_f(\Omega) \; \textit{and assigned to processor} \; m(D_i) \, \}$ is the set of blocks (subdomain) that are assigned to the same processor, C_{D_i} is the set of the subdomains that are adjacent to the subdomain D_i, $m : \{D_i\}_{i=1}^{P} \rightarrow \{P_i\}_{i=1}^{P}$ is the mapping of the subdomains to processors, $W(m(D_i))$ is the computational load of the processor $m(D_i)$, which is analogous to $\sum_{B'_j \in D_i} |B'_j|$, and $C(m(D_i), m(D_j))$ is the communication required (for one iteration) between the processors $m(D_i)$ *and* $m(D_j)$.

There are two approaches for the solution of the optimization problem (O). The first approach is based on the approximation of the objective function of (O) by another function which is smoother, more robust and suitable for the existing deterministic and stochastic (physical) optimization methods [Fox 86], [Flow 88], [Will 91] and [Mans 92]. The second approach is based on the splitting of the optimization problem into two distinct phases [Simo 91], [Chri 92] corresponding to the *partitioning and allocation* of the composite block structure. In the *partitioning phase* we decompose the composite block structure in to a prespecified number (usually equal to the number of processors) of subdomains such that the following criteria are approximately satisfied:

(i) the subdomains have the same number of blocks,

(ii) the ratio interface surface to volume of the subdomains is minimum

(iii) the number of adjacent subdomains is minimum, and

(iv) each subdomain is a connected domain.

In the *allocation phase* the objective is to assign these subdomains to processors, such that the following objective is satisfied:

(v) the communication requirements of the underlying computation between the processors of a given architecture are minimum.

For a given physical domain Ω and an initial composite block structure $C_0(\Omega)$ (see [Thom 84]), the partitioning of a finer composite block structure $C_f(\Omega)$ into P non-overlapping subdomains $\{D_i\}_{i=1}^{P}$ is characterized in terms of the set of geometrical adjacent subdomains C_{D_i} to subdomain D_i and the number of the outer-layer interface grid points, $c(D_i, D_j)$, between the subdomains D_i *and* D_j with $c(D_i, D_j) = \mid \cup(\overline{B_i^{'}} \cap B_j)\mid$, where $B_i^{'} \in D_i$ and $B_j \in D_k$, & $D_k \in C_{D_i}$ (see Appendix). Then, the optimal partitioning, as defined by criteria (i) to (iii), can be viewed as the one with :

$$\lfloor N/P \rfloor \leq |D_i| \leq \lceil N/P \rceil \qquad i = 1, \ldots, P \qquad (O1)$$

$$\min \max_{1 \leq i \leq P} \sum_{D_j \in N(D_i)} \frac{c(D_i, D_j)}{|D_i|} \qquad (O1)$$

$$\min \max_{1 \leq i \leq P} |C_{D_i}| \qquad (O2)$$

where $|D_i|$ is the size of the subdomain D_i and it is defined as the cardinality of the set of grid points that belong in D_i.

The composite block structure $C_f(\Omega)$ can be created by an Algebraic grid (see in [Smit 83]), with the minimum number of grid points, that is based on the initial composite block structure and provides explicit control of the physical grid shape. Such a grid can be generated relatively easy using the interactive environment of the National Grid Project.

We solve the optimization problem (O0-O2) by first, defining an Euclidean graph $G(V, E)$ where the vertices (V) correspond to blocks $B_i^{'} \in C_f(\Omega)$ (the coordinates of the vertices are defined to be equal to the coordinates of the mass centers of the computational blocks, note that these coordinates are integers numbers) and the edges (E) indicate the connectivity of the blocks $B_i^{'}$ with their neighbor blocks, and then applying graph partitioning algorithms that use the geometric information that is associated to the vertices of the graph [Chri 92]

The determination of an optimal allocation m is equivalent to minimizing the communication overhead. The overhead due to communication of non-local data can be modeled in terms of the length of the interfaces between the subdomains and a function f_d. Mathematical models for the allocation phase are described by the following two minimization expressions :

$$\min_{m} \frac{1}{2} \sum_{i=1}^{P} \sum_{j=1}^{P} c(D_i, D_j) f_d(m(D_i), m(D_j)) \quad (O3)$$

or

$$\min_{m} \max_{1 \leq i \leq P} \{ \sum_{D_j \in N(D_i)} c(D_i, D_j) \times f_d(m(D_i), m(D_j)) \} \qquad (O4)$$

where $f_d(m(D_k), m(D_\ell))$ is a function that depends on the distance (Hamming) between the two processors $m(D_k), m(D_\ell)$ to which the subdomains have been allocated, the interface components (hardware/software) between the processors and the network, the router, and the architecture of the machine. An accurate analytic form of the function f_d is not known even for the commercially available well known distributed memory MIMD machines; An alternative solution to this problem is the use of approximations based on assumptions that simplify some of the above factors.

Allocation heuristics for 2-dimensional graphs are presented in [Chri 92], easily these heuristics can be applied to 3-dimensional graphs. In our future work we will address the scheduling of the computations based on the partitioning and allocation of the composite block structure onto distributed memory MIMD machines. Also, we will solve the optimization problem (0) using an approximate objective function. Finally, an evaluation on the quality of the solution (i.e., minimization of the execution time of parallel Elliptic grid generation) for the two approaches will be performed.

Acknowledgements

The first author gratefully acknowledges the Alex G. Nason Foundation for the Nason Prize Award that supports him at NPAC.

Appendix

Let S be a solid and $C_0(S) = \{B_0, B_1, ..., B_N\}$ be a composite block structure. Then for each $B_i \in C_0(S)$ we define :

– $B_i^{'}$ to be the outer layer of B_i. The width of the outer layer $B_i^{'}$ is equal to the width of the half stensil used by the elliptic solver (see in [3]).

– $\overline{B_i}$ to be the closure of the block B_i, The closure $\overline{B_i}$ is defined to be equal to the union, $B_i^{'} \cup B_i$, of the outer layer $B_i^{'}$ and the block B_i.

– $C_{B_i} \mid_{C_0(S)} = \{B_j \mid B_j \in C_0(S) \wedge \overline{B_i} \cap B_j \neq \emptyset\}$ to be the set of blocks $B_j \in C_0(S)$ adjacent to the block $B_i \in C_0(S)$.

– $\mid C_{B_i} \mid_{C_0(S)} \mid$ to be the connectivity of the block B_i, which is equal to the cardinality of the set $C_{B_i} \mid_{C_0(S)}$.

– $\mid B_i \mid$ as the computational workload of an elliptic solver on B_i.

1 References

[Bokh 81] Shahid H. Bokhari, On the mapping problem, *IEEE Transactions on Computers*, Vol. C30, No. 3, 207-213, 1981

[Chri 92] N.P. Chrisochoides, *On the mapping of PDE computations to distributed memory MIMD machines*, Ph.D. Thesis, Purdue University, 1992.

[Fhar 88] C. Farhat, A simple and efficient automatic FEM domain decomposer, *Computers and Structures*, Vol. 28, 579–602, 1988.

[Flow 88] J. Flower, S. Otto and M. Salana, Optimal mapping of irregular finite element domains to parallel processors, In *Parallel Computers and Their Impact on Mechanics* (A.K. Noor, ed.), AMD, Vol. 86, 239–250, 1988.

[Fox 86] G.C. Fox, A review of automatic load balancing and decomposition methods for the hypercube, *Proceedings of the IMA Institute*, 63–76, 1986.

[Hamm 92] Hammond W. S., Mapping Unstructured Grid Computations to Massively Parallel Computers, Ph.D Thesis, Rensselaer Polytechnic Institute, Troy, NY.

[Lori 90] Loriot, M. and L. Fezoui, Mesh Splitting Preprocessor, Unpublished Manuscript.

[Mans 92] Mansour N., Physical Optimization Algorithms for Mapping Data to Distributed Memory Multiprocessors, Ph.D Thesis, Syracuse University, Syracuse, NY, 1992.

[Mori 87] Morison, R. and S. Otto, The scattered decomposition for finite elements, *Journal of Scientific Computing,*, Vol. 2, No 1, 1987

[Sada 87] P. Sadayappan and F. Ercal, Nearest-neighbor mappings of finite element graphs onto processor meshes, *IEEE Trans. on Computers*, Vol. 36, 1408–1420, 1987.

[Simo 91] Simon, H. D., Partitioning of Unstructured Problems for Parallel Processing, RNR-91-008, NASA Ames Research Center, Moffet Field, CA, 94035

[Smit 83] Smith, R. E., Three-Dimensional Algebraic Grid Generation, *AIAA Paper*, 83-1904, 1983.

[Thom 91] Thompson, J. F., National Grid Project, NSF Engineering Research Center for Computational Field Simulation, Mississippi State University.

[Thom 84] Thompson, J. F., A survey of composite grid generation for general three-dimensional regions. *Numerical Methods for Engine-Airframe Integration* S.N.B. Murthy and G. C. Paynter eds., 1984.

[Will 91] Williams, R.D., Performance of Dynamic Load Balancing Algorithms for Unstructured Mesh Calculations, *Concurrency Practice and Experience* Vol. 3, No. 5, 483-496, 1991