The Construction of Large Unstructured Grids by Parallel Delaunay Grid Generation

N. P. Weatherill, O. Hassan, K. Morgan and R. Said
Department of Civil Engineering, University of Wales Swansea, UK

1. Introduction

Present trends in computational engineering point to a requirement for more multi-disciplinary analysis. The coupling of fluids and structures, or thermal with fluids with structures are good examples. Furthermore, the inter-dependency of many engineering designs on such factors as fluids, structures and possibly electromagnetics also points to more challenging computational simulations. We are possibly not far away from a requirement to generate a grid outside, inside and through an aircraft for multi-disciplinary simulation. Such requirements will place considerable demands on mesh generators and, inevitably, large meshes will result.

In the field of computational electromagnetics in three dimensions, the size of mesh required for a particular simulation using standard algorithms is a function of the cube of the wavelength. Hence, if the wavelength is halved, the mesh with the same relative resolution, will require eight times the number of elements. Given this requirement, the simulation of high frequency electromagnetic scattering around typical aerospace configurations will necessitate meshes consisting of 10's of millions of elements. In our own work to-date, we have had a requirement for an unstructured mesh of 100 million elements.

1.2 Strategies for the Generation of Large Meshes

In our work three approaches have been used to construct large meshes. The first two involve the application of the standard h-refinement technique to a generated mesh. The first approach refines a mesh with h-refinement with a new node added to each edge in the mesh and the existing tetrahedra then sub-divided and new elements formed by the appropriate reconnection of nodal points. A second approach performs domain decomposition prior to h-refinement. A disadvantage of these two methods is that the user forfeits the ability to determine the location of the available nodes on the finest mesh. An additional complication is that, when boundary edges are considered, the added nodes need to be located on the boundary surface. The third approach is to adopt a parallel approach to the generation of the mesh.

In more detail we consider,

- *H-refinement*

  In this approach a new mesh is produced by h-refinement of a generated mesh. The RSB method is then used as the domain decomposition strategy [4,5]. The size of mesh that can be handled could be limited by the memory demands of the RSB procedure.

- *Decompose the mesh followed by h-refinement*

  In this approach a mesh is generated and decomposed within the limits imposed by the available computational resource. The mesh is decomposed into the same number of regions as there are processors available to perform the equation solution. Then a mesh of the desired size is produced by h-refinement of each of the decomposed regions separately. In this way, no further domain decomposition is required and the associated memory constraints are, therefore, removed. Theoretically, subject to the drawbacks of the h-refinement technique previously mentioned, this approach could be used to produce meshes of any desired size.

- *Parallel unstructured mesh generation*

  An alternative strategy is to generate the mesh in parallel and, if possible, preclude the requirement for domain decomposition. If this approach is followed, there are two obvious procedures. Firstly, large meshes could be generated by the development of parallel algorithms equivalent to the sequential methods already available, and secondly, existing algorithms could be used within a framework that allows the generation of grids in parallel. Table 1.1 highlights some of the advantages and disadvantages seen for the two approaches.

Parallelise the grid generation algorithm	
Advantages	High speed performance is expected
Disadvantage	Size of grid could be limited by available memory Performance dependent upon number of processors Domain decomposition is to be applied on a large grid Communication costs could be high A new algorithm must be developed

Parallelise using geometrical partitioning	
Advantages	Size of grid not limited by available memory No need for any further domain decomposition Minimum communication One processor can generate arbitrary large grids General framework could be used with different types of grid generators No need to parallelise the grid generation algorithm
Disadvantage	Poor load balancing The adverse effect on grid quality of inter-domain grids

Table 1.1 Comparison between the two different approaches

We have chosen to address the problem of the generation of large grids on computer platforms with modest resources by using a geometrical partitioning approach and a Delaunay based algorithm [1-3]. As will be described, this approach has already proved particularly effective in the generation of grids to 100 million elements.

2. General Principles of Parallel Mesh Generation

2. 1 Geometrical Partitioning

The approach adopted for parallel mesh generation is based upon geometrical partitioning of the domain. To generate a grid in parallel, the complete domain is divided into a set of smaller sub-domains, and a grid generated in every sub-domain independently. A combination of the sub-domain grids forms the final grid of the total domain. A manager/worker model is employed, in which the initial work is performed by the manager who then distributes the grid generation tasks to the workers. The manager can recombine all the sub-domain grids or, if the grid is particularly large, leave the partitioned grid on disc.

The entire procedure can be grouped into four stages:

Stage 1: Apply the geometrical partitioning scheme on the computational domain.

Stage 2: Generate meshes on the sub-domain boundaries.

Stage 3: Use the dynamic load balancing scheme to generate grids in each sub-domain.

Stage 4: Post-process the grid, including node smoothing of inter-domain boundaries, work load redistribution, and build the inter-domain communication table.

Figure 2.1 shows the general procedure in schematic form

- The geometry is point discretised, Figure 2.1a.
- The boundary points are connected using a Delaunay algorithm to produce an initial triangulation, as shown in Figure 2.1b. This step is performed sequentially.
- A greedy algorithm, with an area criterion, is employed to give a number of equally-sized sub-domains, Figure 2.1c. This step is performed sequentially.
- The inter-domain boundaries are discretised leading to a set of independent grid generation tasks, Figure 2.1d. This step is performed in parallel.