

RECENT ADVANCES IN UNSTRUCTURED MESH AND POINT GENERATION

R. Löhner¹

¹*School of Computational Sciences and Informatics
M.S. 4C7, George Mason University, Fairfax, VA 22030-4444, USA*

SUMMARY: Recent advances in mesh and point generation are reviewed. These include: meshing of discrete surfaces, parallel advancing front methods, improvements in RANS gridding via directional enrichment and tighter sphere packing for discrete particle methods. Several examples are included to illustrate the effectiveness of the developed techniques.

KEYWORDS: Unstructured Grid Generation, Parallel Meshing, Point Generation

1. INTRODUCTION

Field solvers based on unstructured grids play an ever increasing role in physics and engineering. Presently, all of the large-scale commercial software packages in Computational Structural Dynamics (CSD), Computational Fluid Dynamics (CFD) and Computational Thermodynamics (CTD) are based on finite element or finite volume methods operating on unstructured grids. The relentless advance in the physical complexity that may be modeled by these codes, together with increases in algorithmic efficiency and computer power, have placed a premium on the reliable, automatic generation of grids for complex geometries that are suitable for the problem at hand. The present paper reviews some recent developments in this direction. The first topic discussed is surface meshing, a key bottleneck faced in many industries as the virtual prototyping process is automated. The second topic is concerned with fast meshing for very large problems, focusing on parallel meshing. The third topic is reliable gridding of regions that require elements with extreme stretching, such as boundary or shear layers in flows. Finally, as a fourth topic, the generation of suitable clouds of points for the finite point or discrete particle methods is discussed.

2. DISCRETE SURFACE MESHING

The rapid, user-friendly definition of the surfaces defining the computational domain has been an important goal during the last decade. Surfaces can be defined either analytically (using B-Splines, NURBS, Coon's patches, etc.) or via triangulations. The latter option is particularly interesting for data sets stemming from remote sensing data (e.g. geographical data) or medical imaging [Ceb01]. An interesting observation made over the last years is that an increasing number of data sets used to define the geometry of CFD domains is given in the form of triangulations, even though the

CAD data is available. The reason for this shift in data type is that a watertight triangulation defines in a unique way the domain considered, and does not require any further geometric cleanup operations. This is not the case with native CAD datasets, in which we frequently encounter very large numbers of patches, overlapping patches, gaps, and other geometric pathologies that require user intervention. These developments have renewed the interest in robust surface meshing of so-called discrete surfaces (DS). Of the many innovations introduced during the last years, we mention:

- Automatic preprocessing/improvement of the DS;
- Introduction of a visibility horizon filter for close points/ sides;
- Strict enforcement of continuous topology;
- Improved 2D cross-check; and
- Adaptive background grid element size definition.

In the sequel, we expand on a few of these. For a detailed description, see [Til02].

Visibility Horizon: The advancing front method adds a new surface triangle by removing a side from the active front. Among the decisions required is whether to take an existing point to form the new triangle, or to introduce a new point. The list of close (i.e. possible) points is obtained from a proximity search. This list of possible close points is reduced by several tests (visibility, angles, etc.). Perhaps the most important validation test is based on the neighbour to neighbour search on the given DS. The starting face for the search is given by the underlying DS face at the midpoint of the side being removed from the active front. The direction is given by the close point. Any close point that can not be reached on the given DS using the neighbour to neighbour search is removed from the list. A similar procedure is used to filter close sides, which are required to test if the new triangle crosses the existing active front of sides.

Continuous Topology: A typical neighbour to neighbour search will not stop at internal DS geometry lines (given by sharp edges). Therefore, a so-called visibility horizon was introduced for the neighbour to neighbour search. All neighbour to neighbour edges given by internal geometry lines or angles beyond a certain tolerance are marked. In this way, the neighbour to neighbour search can recognize them. The neighbour to neighbour search stops at these internal geometry lines. The close point is marked as unreachable and removed from the list of candidates.

Adaptive Background Grid Based on DS: For complex geometries, the specification of desired element size can be a tedious, time-consuming process. Adaptive background grids [Löh96, Löh97] offer the possibility to reduce drastically the required level of human input. DS offer, by their way of defining the surface, a natural way to refine the background grid and to define the mesh size required for a proper definition of the geometry.

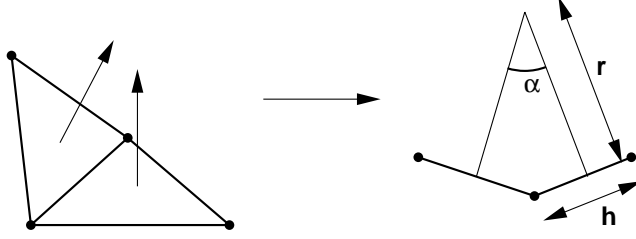


Figure 1 Measuring Surface Curvature

With the notation defined in Figure 1, the angle between two faces is given by

$$\frac{h}{2r} = \tan\left(\frac{\alpha}{2}\right) . \quad (1)$$

This implies that for a given element size h_g and angle α_g , the element size for a prescribed angle α_p should be:

$$h_p = h_g \frac{\tan\left(\frac{\alpha_p}{2}\right)}{\tan\left(\frac{\alpha_g}{2}\right)} . \quad (2)$$

For other measures of surface accuracy, similar formulae will be encountered. Given a prescribed angle α_p , the point-distances of the given DS surface triangulation are compared to those obtained from Eqn.(2) and reduced appropriately:

$$\delta_i = \min(\delta_i, h_p) . \quad (3)$$

These new point-distances are then used to adjust and/or refine the background grid. As an example of the effective use of adaptive background grids, we consider the air flow in the bronchii and lungs. A segmented image, together with the cuts at the extremities of the smaller branches, is shown in Figure 2a. The mesh sizes are automatically obtained from an adaptive background grid with 6 levels of refinement. This produced the surface mesh shown in Figure 2b. One can discern the smaller elements in regions of higher curvature and smaller vessel diameter. The volume mesh had approximately 1 million elements. In this first study, only the steady airflow was considered. The results obtained can be seen in Figures 2c,d, which show surface pressures and iso-surfaces of constant absolute value of velocity.

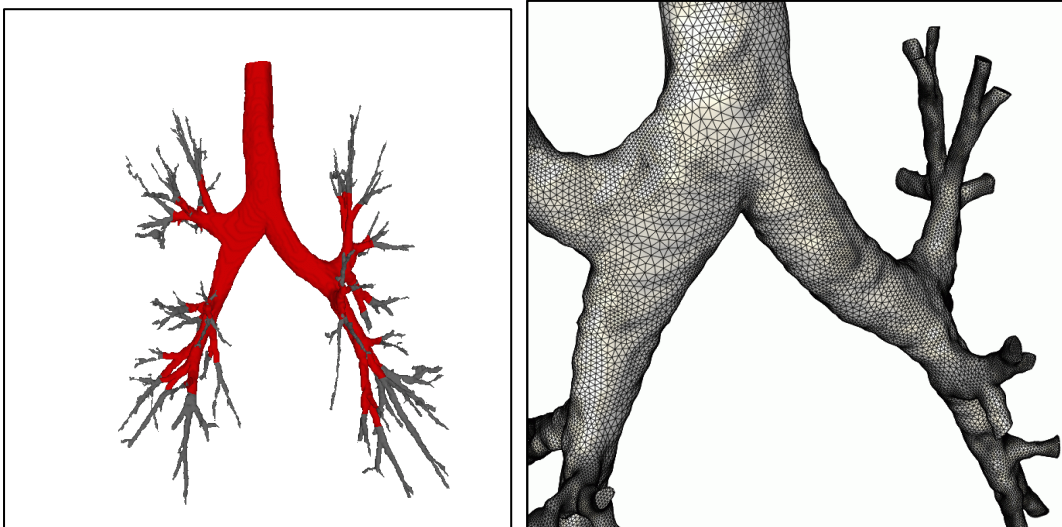


Figure 2a,b Thorax: Segmented Image and Surface Mesh

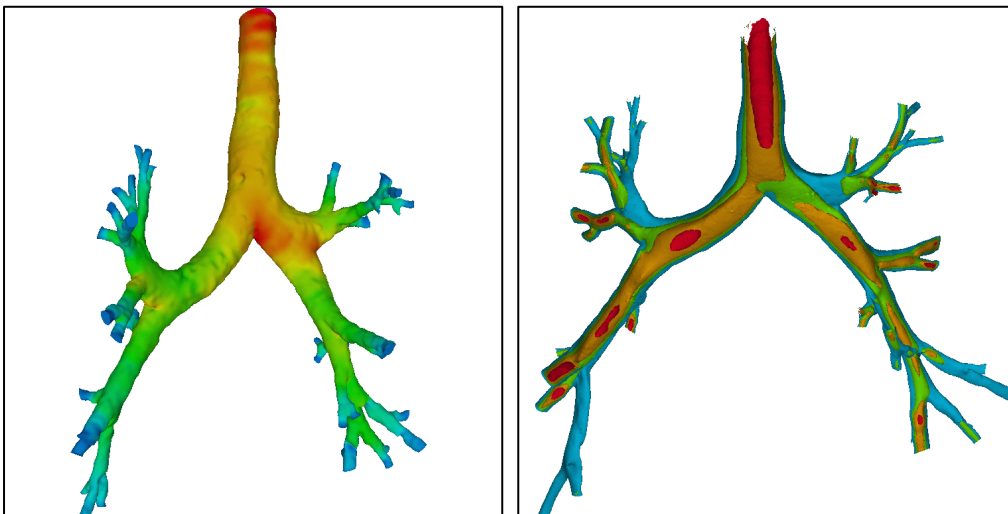


Figure 2c,d Thorax: Surface Pressure and Isosurfaces of Speed

3. PARALLEL GRID GENERATION

Over the last decade, major efforts have been devoted to harness the power of parallel computer platforms. While many CFD and CSD solvers have been ported to parallel machines, grid generators have lagged behind. For applications where remeshing is an integral part of simulations, e.g. problems with moving bodies [Löh90, Mes93,

Mes96, Bau96, Kam96, Löh98b, Has98] or changing topologies [Bau98, Bau99], the time required for mesh regeneration can easily consume more than 50% of the total time required to solve the problem. Faced with this situation, a number of efforts have been reported on parallel grid generation [Löh92, dCo94, Sho95, dCo95, Oku96, Che97, Oku97, Sai99].

The two most common ways of generating unstructured grids are the Advancing Front Technique (AFT) [Per87, Per88, Löh88a,b, Per90, Per92, Jin93, Fry94, Löh96] and the Generalized Delaunay Triangulation (GDT) [Bak89, Geo91, Wea92, Wea94, Mar95a]. The AFT introduces one element at a time, while the GDT introduces a new point at a time. Thus, both of these techniques are, in principle, scalar by nature, with a large variation in the number of operations required to introduce a new element or point. While coding and data structures may influence the scalar speed of the ‘core’ AFT or GDT, one often finds that for large-scale applications, the evaluation of the desired element size and shape in space, given by background grids, sources or other means [Löh96] consumes the largest fraction of the total grid generation time. Unstructured grid generators based on the AFT may be parallelized by invoking distance arguments, i.e. the introduction of a new element only affects (and is affected by) the immediate vicinity. This allows for the introduction of elements in parallel, provided that sufficient distance lies between them.

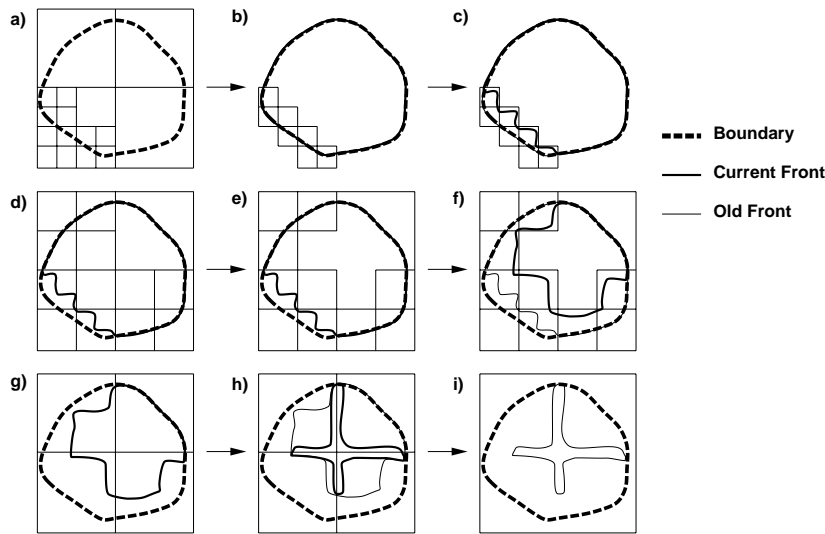


Figure 3 Parallel Grid Generation

A convenient way of delimiting the possible zones where elements may be introduced by each processor is via boxes. These boxes may be obtained in a variety of ways, i.e. via bins, binary recursive trees, or octrees. We have found the octree to be the best of these possibilities, particularly for grids with a large variation of element size. In order to recover a parallel gridding procedure that resembles closely the advancing front technique on scalar machines, only the boxes covering the active front in regions

where the smallest new elements are being introduced are considered. This has been shown schematically in Figure 3a,b for a simple 2-D domain. After these boxes have been filled with elements (Figure 3c), the process starts anew: a new octree is built (Figure 3d), new boxes are created (Figure 3e) and meshed in parallel (Figure 3f). This cycle is repeated until no faces are left in the active front (Figures 3g-i).

At the end of each parallel gridding pass, each one of the boxes gridded can have an internal boundary of faces. For a large number of boxes, this could result in a very large number of faces for the active front. This problem can be avoided by shifting the boxes slightly, and then regridding them again in parallel, as shown in Figure 4. This simple technique has the effect of eliminating almost all of the faces between boxes with a minor modification of the basic parallel gridding algorithm.

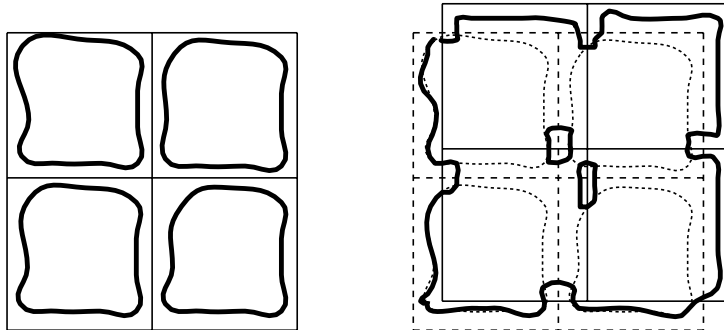


Figure 4 Shift and Regrid Technique

If we define as d_{min} the minimum element size in the active front, and as s_{min} the minimum box size in which elements are to be generated, the parallel AFT proceeds as follows:

WHILE: There are active faces left:

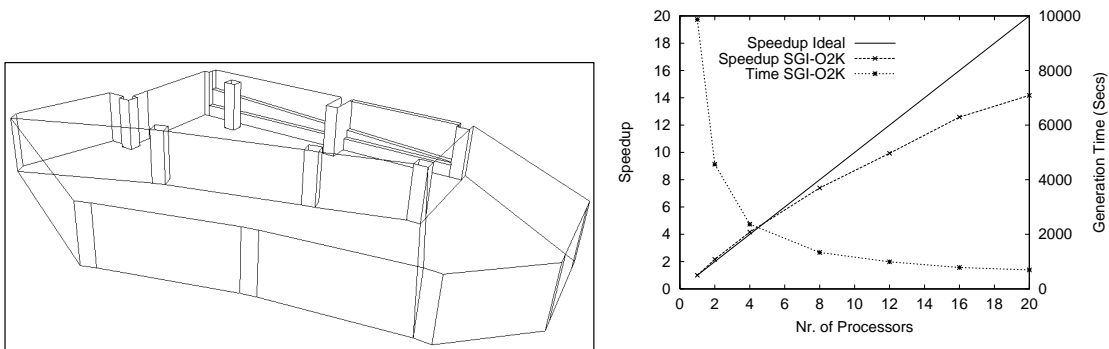
- Form an octree with minimum octant size s_{min} for the active points;
- Retain the octants that have faces that will generate elements of size d_{min} to $c_l \cdot d_{min}$;
- If too many octants are left: agglomerate them into boxes;
- DO ISHFT=0, 2:
 - IF: ISHFT.NE.0:
 - Shift the boxes by a preset amount;
 - ENDIF
 - Generate, in parallel, elements in these boxes, allowing only elements up to a size of $c_l \cdot d_{min}$;
- ENDDO
- Increase $d_{min} = 1.5 * d_{min}$, $s_{min} = 1.5 * s_{min}$;

ENDWHILE

The increase factor allowed is typically in the range $c_l = 1.5-2.0$. Recent improvements to the described parallel advancing front grid generator include:

- A more reliable workload estimation for spatial domains;
- The introduction of a limit for the number of elements a processor may generate (to prevent one processor from consuming much more CPU time than all the others);
- A reduction in the number of processors used if the number of remaining faces is too small (so-called 'scalar endgame'); and
- A reduction of processors used in the post-generation/improvement phase if the number of elements modified is too small.

We include two examples to demonstrate the effectiveness of parallel mesh generation. Both examples were obtained on an SGI Origin 2000 running in shared memory mode. Figures 5a,b show the outline and timings obtained for the garage of an office complex. The mesh had approximately 9.2 million tetrahedra. As one can see, although not perfect, speedups are comparable to those of production CFD codes.



Figures 5a,b Garage: Wireframe and Speedups Obtained

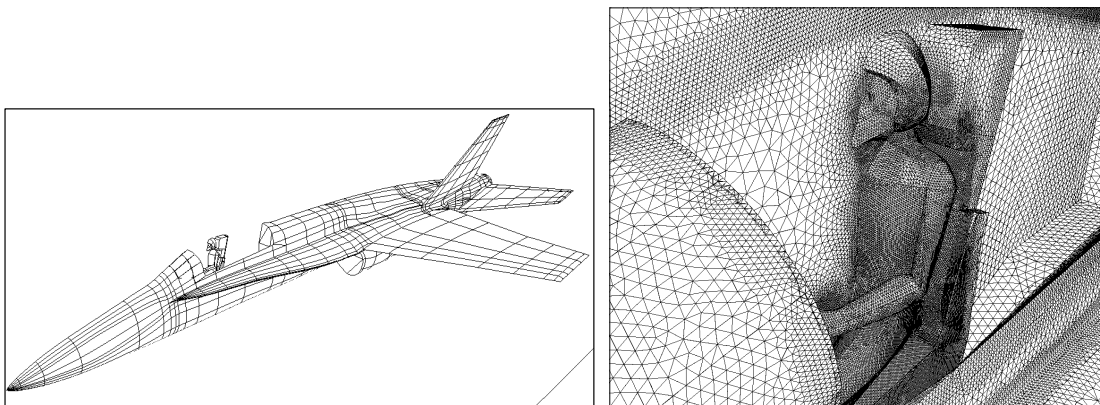


Figure 6a,b F18: Outline of Domain and Surface Mesh (Closeup)

Figure 6 shows a pilot seat ejection case, a typical problem with moving bodies that

requires several remeshings during the course of a simulation [Bau93, Bau95, Bau97, Sha00]. The outline of the domain is shown in Figure 6a. The surface triangulation of the final mesh, which had approximately 14 million tetrahedra, is shown in Figure 6b. The smallest and largest specified element side lengths were 0.65 cm and 250.00 cm respectively, i.e. an edge-length ratio of approximately $1 : 4 \cdot 10^2$ and a volume ratio of $1 : 5.6 \cdot 10^7$. The spatial variation of element size was specified via approximately 110 sources [Löh96a, Löh97].

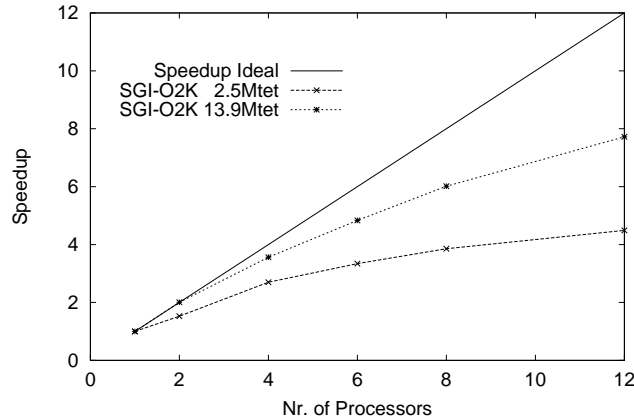


Figure 6c F18: Speedups Obtained

The speedup obtained for two different grid sizes is displayed in Figure 6c. One can see that the grid generator scales well with the number of processors and that scalability improves with the amount of work required. For more details on the parallel grid generator, see [Löh01].

4. IMPROVEMENTS IN RANS GRIDGING

The generation of high-quality grids suitable for RANS calculations of flows in and around complex geometries continues to be an active area of research. The two most common ways of generating highly stretched grids suitable for RANS calculations are the advancing layers technique (ALT) and the directional enrichment technique (DET). The ALT follows the spirit of the advancing front technique (AFT): starting from the ‘wetted surface’, add thin layers of elements until an isotropic mesh is achieved [Löh93, Pir94, Mar95b, Pir96]. From this point onwards, the mesh is completed with the AFT. The DET is an extension of the Generalized Delaunay Triangulation (GDT). At first, an isotropic mesh is generated (either via AFT or GDT). Then, the points in the regions where stretched elements are to be generated are removed (typically by collapsing edges). Thereafter, points are introduced in the near-wall region so as to obtain the desired RANS mesh. The newly introduced points are reconnected using the GDT [Per96, Löh99]. This latter technique, while general, can still generate bad grids

for complex geometries. A recent improvement was obtained by considering the layer-number of the points when performing diagonal/face swapping in 3-D. The quality of elements that cross several point-layers is reduced, leading to swaps that favour elements with minimal layer jump. The improvement is shown in Figure 7, where a corner is considered.

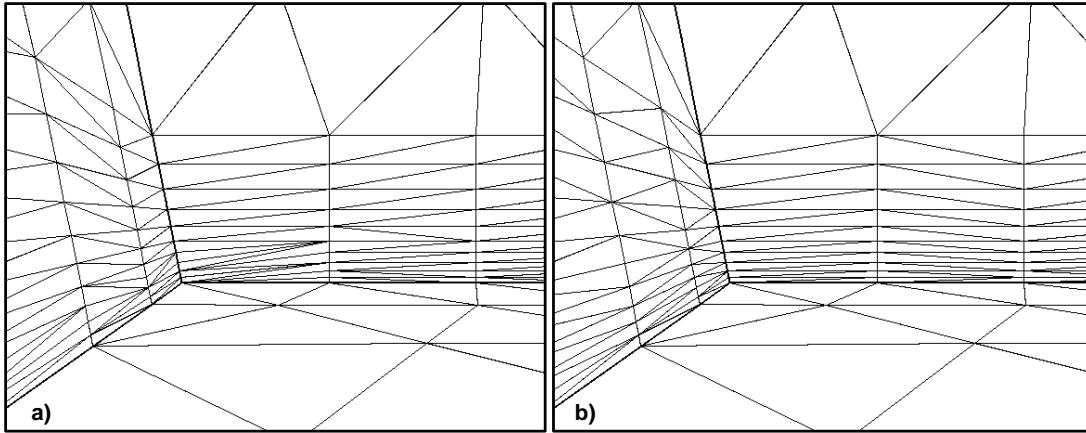


Figure 7 Corner: Surface Meshes Obtained

5. POINT GENERATION

Over the course of the last decade, a number of ‘gridless’, ‘mesh free’, or ‘discrete particle’ schemes have appeared in the literature (see, e.g. [Nay72, Bat93, Bel94, Oña96a, Oña96b, Liu96] and the references cited therein). The interest in these schemes stems from:

- a) The perceived difficulty of generating volume filling grids for problems characterized by complex geometries and/or complex physics;
- b) The perceived difficulty of producing high-order numerical schemes for complex geometries; and
- c) The perceived failure of continuum models to capture complex material behaviour, such as that encountered in concrete failure.

The generation of an appropriate global cloud of points seems, at first, much simpler than the generation of points and elements. Three general ways of generating clouds of points have been reported to date:

- Generate a mesh using traditional grid generators, and remove the elements in a post-processing step;
- Generate a coarse mesh of elements, fill each element with an appropriate number of particles, and remove the elements in a post-processing step;
- Generate directly clouds of points using an advancing front point method [Löh98a].

The first approach defeats the purpose stated as one of the reasons for going to point methods. The second approach will have difficulties in regions where small elements are required due to geometrical constraints.

Only the third scheme allows for the direct generation of clouds of points with the same degree of flexibility as advanced unstructured grid generators [Löh88, Per88, Per90, Geo91, Jin93, Fry94, Löh96a, Löh97]. The mean distance between points (or, equivalently, the point density) is specified by means of background grids, sources and density attached to CAD-entities. In order not to generate points outside the computational domain, one assumes an initial triangulation of the surface that is compatible with the desired mean distance between points specified by the user. Starting from this initial ‘front’ of points, new points are added, until no further points can be introduced. Whereas the advancing front technique for the generation of volume grids removes one face at a time to generate elements, the present scheme removes one point at a time, attempting to introduce as many points as possible in its immediate neighborhood.

5.1 Advancing Point Generation Algorithm

Assume as given:

- A specification of the desired mean distance between points in space. This is done here through a combination of background grids, sources and mean distance to neighbours attached to CAD-data (see [Löh96a, Löh97] for more details).
- An initial triangulation of the surface, with the face normals pointing towards the interior of the domain to be filled with points.

With reference to Figure 8, the complete advancing front point generation algorithm may be summarized as follows:

- Determine the required mean point distance for the points of the triangulation;
- **while:** there are active points in the front:
 - Remove the point `ipout` with the smallest specified mean distance to neighbours from the front;
 - With the specified mean point distance: determine the coordinates of `nposs` possible new neighbours. This is done using a stencil, some of which are shown in Figure 9;
 - Find all existing points in the neighborhood of `ipout`;
 - Find all boundary faces in the neighborhood of `ipout`;
 - **do:** For each one of the possible new neighbour points `ipnew`:
 - If there exists a point closer than a minimum distance `dminp` from `ipnew`:
 ⇒ skip `ipnew`;
 - If the line connecting `ipout` and `ipnew` crosses existing faces:
 ⇒ skip `ipnew`;
 - Determine the required mean point distance for `ipnew`;
 - Increment the number of points by one;
 - Introduce `ipnew` to the list of coordinates;
 - Introduce `ipnew` to the list of active front points;
 - enddo**
- **endwhile**

Details on point stencils, boundary consistency checks and efficient data structures for the search operations required may be found in [Löh98a].

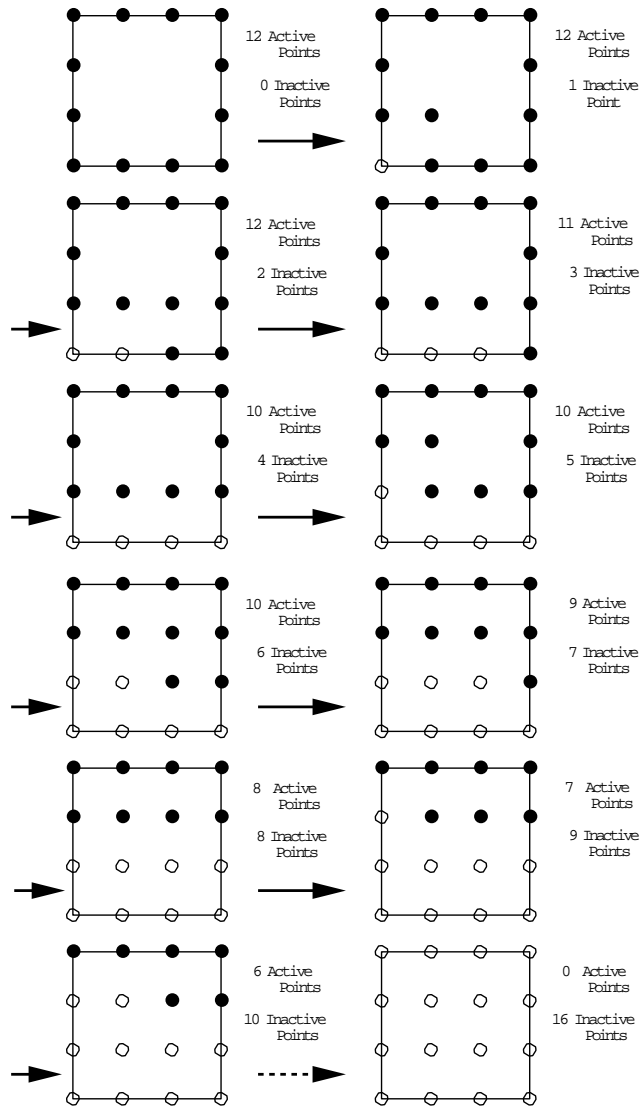


Figure 8 Advancing Front Point Generation

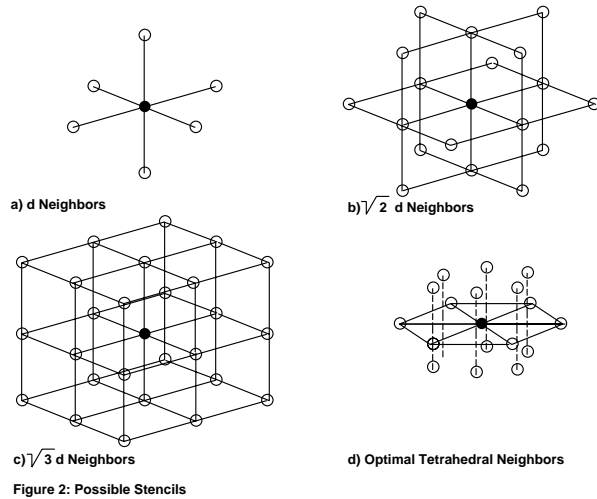


Figure 9 Stencils Used for Point Introduction

5.2 Tight Packing Option

The generation of clouds of **points** is less restrictive than the generation of clouds of **spheres**. In the latter, each sphere has a given radius, and the spheres may not interpenetrate each other. Several clouds of spheres generated with the technique described indicated that the packing obtained was below the expected maximum packing. This was particularly the case for clouds with a random variation of radius for the spheres. In order to increase the packing, the newly introduced sphere is moved as close as possible to the existing spheres (see Figure 10). An attempt is made to move the new sphere to the three closest spheres. Should this fail, the two closest spheres are considered. And should this also fail, the newly introduced sphere is moved as close as possible to the originating sphere. The spheres in question are assumed to form a spring system with contact forces. This spring system is relaxed until convergence to the closest possible position is achieved.

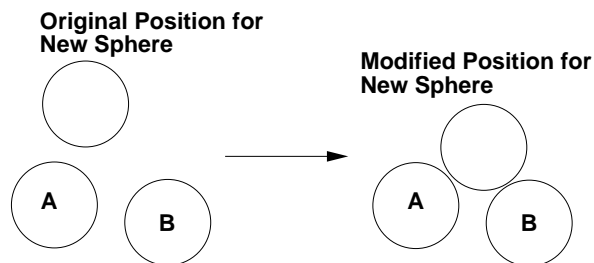


Figure 10 Movement of Spheres to Achieve Maximum Packing

It was found that this ‘movement to close spheres’ greatly increases the packing obtained for the final cloud of spheres. Figure 11 shows a typical discrete particle model for a concrete specimen. The domain is a cube, and the requirement was to generate spheres with a Gaussian size distribution and a diameter variation of 20%. The number of spheres generated was $n_{\text{poin}}=6,950$, of which $n_{\text{movp}}=6,889$ were moved during the generation process. The generated cloud of spheres had a volume fill fraction of $v_f = 0.485$. The average number of contacts was $n_c = 5.8$ per sphere.

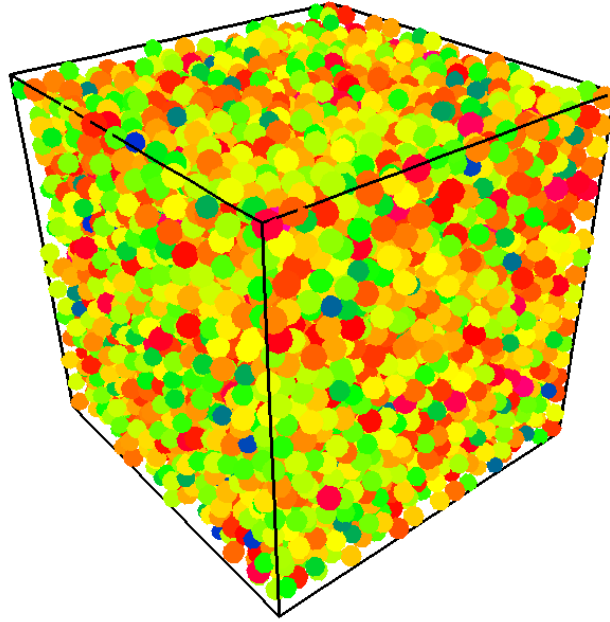


Figure 11 Cloud of Spheres for DMP Modeling of Concrete Specimen

Figure 12 shows snapshots of a sand-filling simulation with approximately $2 \cdot 10^5$ spheres. The sand was modelled by particles of approximately 2mm diameter. This is obviously larger than the actual sand, but nevertheless the material behaviour is captured. The figure shows the particles in a particular cut plane at different times during the filling and compaction process.

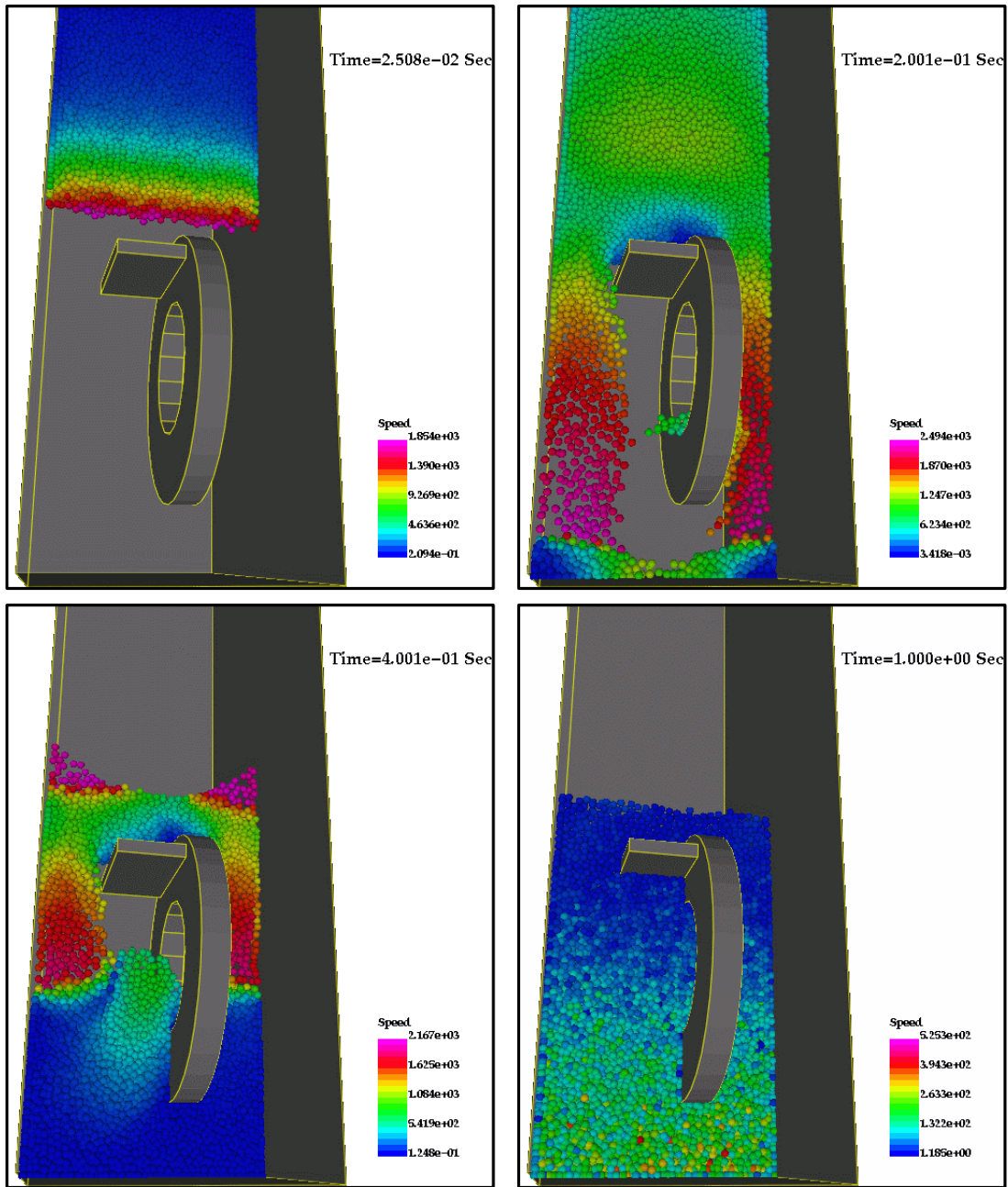


Figure 12 Sand-Filling Simulation With $2 \cdot 10^5$ Spheres

CONCLUSIONS AND OUTLOOK

Over the last decade, unstructured grid generation has become a focal point of research. The present paper has reviewed some recent developments in discrete surface gridding, parallel mesh generation, RANS gridding and the filling of space with a desired point density. Each one of these areas has seen notable advances.

As with any development of technology, the list of possible improvements is never ending. Among the areas of research in unstructured grid generation requiring immediate attention we mention:

- Automatic detection and proper meshing of ‘stiff’ geometric features, such as thin gaps;
- Scaling of parallel gridding to large numbers of processors (> 100);
- Improved RANS gridding, especially for separated, transient wake regions;
- Faster sphere generation to a prescribed volume fraction; and
- ‘Smoothing’ procedures for clouds of points.

REFERENCES

- [Bak89] T.J. Baker - Developments and Trends in Three-Dimensional Mesh Generation. *Appl. Num. Math.* **5**, 275-304 (1989).
- [Bat93] J. Batina - A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Aircraft Configurations; *AIAA-93-0333* (1993).
- [Bau93] J.D. Baum, H. Luo and R. Löhner - Numerical Simulation of a Blast Inside a Boeing 747; *AIAA-93-3091* (1993).
- [Bau94] J.D. Baum, H. Luo and R. Löhner - A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies; *AIAA-94-0414* (1994).
- [Bau95] J.D. Baum, H. Luo and R. Löhner - Validation of a New ALE, Adaptive Unstructured Moving Body Methodology for Multi-Store Ejection Simulations; *AIAA-95-1792* (1995).
- [Bau96] J.D. Baum, H. Luo, R. Löhner, C. Yang, D. Pelessone and C. Charman - A Coupled Fluid/Structure Modeling of Shock Interaction with a Truck; *AIAA-96-0795* (1996).
- [Bau98] J.D. Baum, H. Luo and R. Löhner - The Numerical Simulation of Strongly Unsteady Flows With Hundreds of Moving Bodies; *AIAA-98-0788* (1998).
- [Bau99] J.D. Baum, H. Luo, E. Mestreau, R. Löhner, D. Pelessone and C. Charman - A Coupled CFD/CSD Methodology for Modeling Weapon Detonation and Fragmentation; *AIAA-99-0794* (1999).
- [Bel94] T. Belytschko, Y. Lu and L. Gu - Element Free Galerkin Methods; *Int. J. Num. Meth. Eng.* **37**, 229-256 (1994).
- [Che97] L.P. Chew, N. Chrisochoides and F. Sukup - Parallel Constrained Delaunay Meshing; *Proc. 1997 Workshop on Trends in Unstructured Mesh Generation*, June (1997).
- [Ceb01] J.R. Cebal and R. Löhner - From Medical Images to Anatomically Accurate Finite Element Grids; *Int. J. Num. Meth. Eng.* **51**, 985-1008 (2001).

- [dCo94] H.L. de Cougny, M.S. Shephard and C. Ozturan - Parallel Three-Dimensional Mesh Generation; *Computing Systems in Engineering* 5, 311-323 (1994).
- [dCo95] H.L. de Cougny, M.S. Shephard and C. Ozturan - Parallel Three-Dimensional Mesh Generation on Distributed Memory MIMD Computers; *Tech. Rep. SCOREC Rep. # 7*, Rensselaer Polytechnic Institute (1995).
- [Fry94] J. Frykestig - Advancing Front Mesh Generation Techniques with Application to the Finite Element Method; *Pub. 94:10*, Chalmers University of Technology; Göteborg, Sweden (1994).
- [Geo91] P.L. George, F. Hecht and E. Saltel - Automatic Mesh Generator With Specified Boundary; *Comp. Meth. Appl. Mech. Eng.* 92, 269-288 (1991).
- [Has98] O. Hassan, L.B. Bayne, K. Morgan and N. P. Weatherill - An Adaptive Unstructured Mesh Method for Transient Flows Involving Moving Boundaries; pp. 662-674 in *Computational Fluid Dynamics '98* (K.D. Papailiou, D. Tsahalis, J. Périaux and D. Knörzner eds.) Wiley (1998).
- [Jin93] H. Jin and R.I. Tanner - Generation of Unstructured Tetrahedral Meshes by the Advancing Front Technique; *Int. J. Num. Meth. Eng.* 36, 1805-1823 (1993).
- [Kam96] A. Kamoulakos, V. Chen, E. Mestreau and R. Löhner - Finite Element Modelling of Fluid/ Structure Interaction in Explosively Loaded Aircraft Fuselage Panels Using PAMSHOCK/ PAMFLOW Coupling; *Conf. on Spacecraft Structures, Materials and Mechanical Testing*, Noordwijk, The Netherlands, March (1996).
- [Liu96] W.K. Liu, Y. Chen, S. Jun, J.S. Chen, T. Belytschko, C. Pan, R.A. Uras and C.T. Chang - Overview and Applications of the Reproducing Kernel Particle Methods; *Archives Comp. Meth. Eng.* 3(1), 3-80 (1996).
- [Löh88a] R. Löhner - Some Useful Data Structures for the Generation of Unstructured Grids; *Comm. Appl. Num. Meth.* 4, 123-135 (1988).
- [Löh88b] R. Löhner and P. Parikh - Three-Dimensional Grid Generation by the Advancing Front Method; *Int. J. Num. Meth. Fluids* 8, 1135-1149 (1988).
- [Löh90] R. Löhner - Three-Dimensional Fluid-Structure Interaction Using a Finite Element Solver and Adaptive Remeshing; *Computer Systems in Engineering* 1, 2-4, 257-272 (1990).
- [Löh92] R. Löhner and J.D. Baum - Adaptive H-Refinement on 3-D Unstructured Grids for Transient Problems; *Int. J. Num. Meth. Fluids* 14, 1407-1419 (1992).
- [Löh93] R. Löhner - Matching Semi-Structured and Unstructured Grids for Navier-Stokes Calculations; *AIAA-93-3348-CP* (1993).
- [Löh96a] R. Löhner - Extending the Range of Applicability and Automation of the Advancing Front Grid Generation Technique; *AIAA-96-0033* (1996).
- [Löh96b] R. Löhner - Regridding Surface Triangulations; *J. Comp. Phys.* 126, 1-10 (1996).
- [Löh97] R. Löhner - Automatic Unstructured Grid Generators; *Finite Elements in Analysis and Design* 25, 111-134 (1997).
- [Löh98a] R. Löhner and E. Oñate - An Advancing Point Grid Generation Technique; *Comm. Num. Meth. Eng.* 14, 1097-1108 (1998).

- [Löh98b] R. Löhner, C. Yang, J. Cebral, J.D. Baum, H. Luo, D. Pelessone and C. Charman - Fluid-Structure-Thermal Interaction Using a Loose Coupling Algorithm and Adaptive Unstructured Grids; *AIAA-98-2419* [Invited] (1998).
- [Löh99] R. Löhner - Generation of Unstructured Grids Suitable for RANS Calculations; *AIAA-99-0662* (1999).
- [Löh01] R. Löhner - A Parallel Advancing Front Grid Generation Scheme; *Int. J. Num. Meth. Eng.* 51, 663-678 (2001).
- [Mar95a] D.L. Marcum and N.P. Weatherill - Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection; *AIAA J.* 33, 9, 1619-1625 (1995).
- [Mar95b] D.L. Marcum - Generation of Unstructured Grids for Viscous Flow Applications; *AIAA-95-0212* (1995).
- [Mes93] E. Mestreau, R. Löhner and S. Aita - TGV Tunnel-Entry Simulations Using a Finite Element Code with Automatic Remeshing; *AIAA-93-0890* (1993).
- [Mes96] E. Mestreau and R. Löhner - Airbag Simulation Using Fluid/Structure Coupling; *AIAA-96-0798* (1996).
- [Nay72] R.A. Nay and S. Utku - An Alternative for the Finite Element Method; *Variational Methods Eng.* 1 (1972).
- [Oku96] T. Okusanya and J. Peraire - Parallel Unstructured Mesh Generation; *Proc. 5th Int. Conf. Num. Grid Generation in CFD and Related Fields*, Mississippi, April (1996).
- [Oku97] T. Okusanya and J. Peraire - 3-D Parallel Unstructured Mesh Generation; *Proc. Joint ASME/ASCE/SES Summer Meeting* (1997).
- [Oña96a] E. Oñate, S. Idelsohn, O.C. Zienkiewicz and R.L. Taylor - A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow; *Int. J. Num. Meth. Eng.* 39,3839-3866 (1996).
- [Oña96b] E. Oñate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor and C. Sacco - A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems; *Comp. Meth. Appl. Mech. Eng.* 139, 315-346 (1996).
- [Per87] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz - Adaptive Remeshing for Compressible Flow Computations; *J. Comp. Phys.* 72, 449-466 (1987).
- [Per88] J. Peraire, J. Peiro, L. Formaggia K. Morgan and O.C. Zienkiewicz - Finite Element Euler Calculations in Three Dimensions; *Int. J. Num. Meth. Eng.* 26, 2135-2159 (1988).
- [Per90] J. Peraire, K. Morgan and J. Peiro - Unstructured Finite Element Mesh Generation and Adaptive Procedures for CFD; *AGARD-CP-464*, 18 (1990).
- [Per92] J. Peraire, K. Morgan, and J. Peiro - Adaptive Remeshing in 3-D; *J. Comp. Phys.* (1992).
- [Per96] J. Peraire and K. Morgan - Unstructured Mesh Generation Including Directional Refinement for Aerodynamic Flow Simulation; *Proc. 5th Int. Conf. Num. Grid Generation in CFD and Related Fields*, Mississippi, April (1996).
- [Pir94] S. Pirzadeh - Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method; *AIAA-94-0417* (1994).

- [Pir96] S. Pirzadeh - Progress Towards a User-Oriented Unstructured Viscous Grid Generator; *AIAA-96-0031* (1996).
- [Til02] R. Tilch and R. Löhner - Advances in Discrete Surface Grid Generation: Towards a Reliable Industrial Tool for CFD; *AIAA-02-0862* (2002).
- [Wea92] N.P. Weatherill - Delaunay Triangulation in Computational Fluid Dynamics; *Comp. Math. Appl.* 24, 5/6, 129-150 (1992).
- [Wea94] N.P. Weatherill and O. Hassan - Efficient Three-Dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints; *Int. J. Num. Meth. Eng.* 37, 2005-2039 (1994).