

# Principles of Rectangular Mesh Generation in Computational Physics

Vladislav Ermolaev<sup>1</sup>, Evgeniy Odintsov<sup>2</sup>, Alexander Sobachkin<sup>2</sup>,  
Alexey Kharitonovich<sup>3</sup>, Marina Bevzushenko<sup>4</sup>, and Sergei Zorin<sup>5</sup>.

<sup>1</sup> LTX Corp., Boston, MA, USA  
vermolaev@mail.ru

<sup>2,3</sup> NIKA GmbH., Moscow, Russia

<sup>3</sup> alkharit@nika.sokol.ru

<sup>4</sup> Boston University, Boston, MA, USA  
marinavb@math.bu.edu

<sup>5</sup> Parametric Corp., Boston, MA, USA  
szorin@ptc.com

**Abstract.** Many real world problems may be represented with mathematical models. These models often set complex mathematical, namely partial differential equation (PDE) problems hard to solve analytically and will often require computational approach. Definition of such computational problems will usually imply having a geometry model and initial conditions set on, in or around this model. Computational techniques have to deal with discrete space and time in order to approximate large and complex PDEs with ready to calculate simple arithmetical equations. Discontinuous or discrete space is called mesh. The scope of this article is problems of mesh generation and ways of their solution.

## 1 Introduction

Whether we have to deal with bioscience, fluid dynamics, electromagnetism or any other field where real world problems require computational approach it will often be a PDE problem. Its definition will usually imply having a geometry model, computational domain around it and boundary or initial conditions set in this domain. In order to apply computational approach for PDE solving it is necessary to convert continuous space into discontinuous or so-called mesh. This conversion results in partial loss of initial data or accuracy as geometry model and initial conditions in discontinuous space become a finite element model (FEM) that is merely an approximation of its prototype in continuous space. The finer the mesh the closer the approximation is. On the other hand fine mesh means great number of finite elements or cells every of which is participating in calculation thus consuming computational power and increasing overall calculation time.

Process of optimal mesh generation is very complex and will often consider specifics of the scientific field in general as well as specifics of every particular problem (adaptive mesh refinement) and sometimes even workflow of a problem solution (solution adaptive mesh refinement). The scope of this article is principles of rectangular mesh generation by the example model in which one of many 3D modelling CAD products is used as a basis for problem definition tool.

## 2 Mesh Generation

### 2.1 Basic Mesh

We will be discussing rectangular mesh. It is formed by intersection of three sets of planes that are parallel to each other within every set and sets being orthogonal to each other thus forming a Cartesian frame. Spans between adjacent parallel planes may vary to better resolve geometry model and flow features. This mesh is called *basic mesh* and is usually too coarse to yield any reasonable results.

Every cell of basic mesh (*basic cell*) may be split into eight equal rectangular cells. These smaller cells are *children* with respect to basic cell, their *parent*. This process is called *refinement*. Similarly, every group of eight children may be merged back to one cell. This is *unrefinement* process. Every child in turn may be split further to eight new cells and so on. *Level of refinement* or simply *level* of cell is non-negative number that shows how many times parents of the cell were refined to build this cell. Thus, level of any basic cell is zero, level of their children is one and further refinement yields cells of level two and so on. There is a rule requiring level difference of adjacent cells be not higher than one. This requirement is held for smoother and more accurate solution.

### 2.2 Initial Mesh

Mesh built before calculation is started has special name *initial*. This is because it may change further during calculation. Initial mesh is constructed in several stages:

- constructing basic mesh for a specified number of cells and stretching or contracting it locally to better resolve geometry model and flow features
- splitting basic mesh cells either to capture small solid features or to resolve curvature of interfaces between other materials with different physical properties
- narrow channel refinement - refinement of obtained mesh to better resolve narrow channels and conductors
- irregular cells refinement - if irregular cells appear, they are split to maximum allowed level or until become regular

First step to obtain optimal mesh is to build basic mesh wisely. We may contract it at regions of most interest or stretch somewhere to sacrifice accuracy for calculation time. This is done by means of *control planes*. They are set by engineer who defines problem and are part of basic mesh. Every such plane has associated number that tells mesh generation algorithm how much should mesh be stretched or contracted around this plane. This number  $N$  is dimension multiplier of basic cell of a uniform mesh. In other words, cells around this plane will be  $N$  times the size of cells of simple uniform mesh built in the same region and amount of cells per its dimension. Size of neighbour cells changes linearly from control plane to computational domain boundaries or to another control plane forming gradually contracted or stretched mesh.

Control planes with multipliers may be rendered automatically given some specifics of the problem (field, geometry model, initial conditions, etc.). For instance, given a fluid dynamics problem of a projectile moving through atmosphere with relatively high Mach number there may be expected flow compression in the vicinity of projec-

tile's geometry and high turbulence trace behind it. Thus, problem definition contains enough information to automatically detect regions where physical parameters will most probably have large gradients and build more dense mesh in those regions.

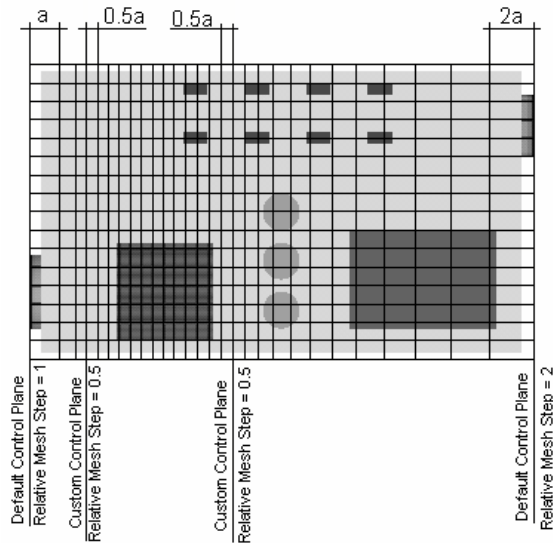


Fig. 1. Using control planes to vary mesh density

## 2.3 Geometry Capturing

For algorithm that computationally solves defined problem to run properly it has to have information about geometry in every cell of the mesh as well as initial conditions, if any, applied to that cell. Hence, next step after basic mesh is generated is to capture model geometry from CAD and refine basic mesh well enough not to lose important geometry features while keeping total number of cells low to save computational time.

Most of modern CAD products publish variety of tools in form of application programming interface (API) for third-party software to gain access to geometry model and operations on it. They may export different operations to determine relations between given point and the model, functions to determine all intersection points of given ray with the model and so on. However, the fastest way to work with geometry information proved to be by using parametric representation of surfaces that build up the model. This representation allows finding relations between every given cell of the mesh and geometry model.

**Cell types.** Depending on geometrical relation between a cell and model all cells are classified as *fluid*, *solid* or *partial* cells. Fluid cells have no common points with geometry model. Solid ones are those that completely belong to model, in other words fully occupied with solid body. Partial cells are those in between. They have intersections with model surfaces and situated partly inside and partly outside the model.

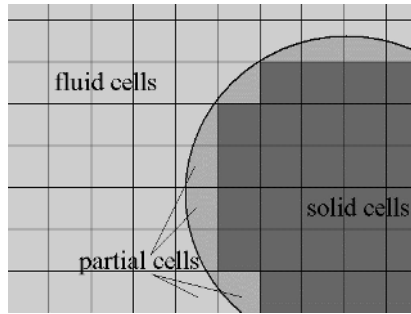


Fig. 2. Cell types

As solver considers every cell during calculation it has to know whether matter can flow in the cell and what valid directions of the flow are. Gas, liquid, heat, electrical current or other form of energy may flow through the cell. In case of pure fluid and solid cells this information is given by cell type itself as solver knows what matter can flow through every cell type. For example, gas and liquid flux is not limited by geometry in fluid cells and may flow in any direction through all cell facets. Same with heat flux in full solid cells, electricity in conductors, etc. Special case is partial cells however. Solver requires information about geometry in these cells to know what cell facets or parts of facets are valid for fluxes to pass. Since geometry inside partial cell may be very complex and solver refers to it very often during calculation it is virtually impossible to keep this information in original parametric form and have solver recalculate everything it needs every time. Instead, model surfaces inside every given cell are linearized, i.e. replaced with polygons based on points where cell edges intersect model surfaces. Not only such approximation reduces solver load, but it requires little memory to store geometry information inside cell also. It may be as little as coordinates of one point and normal vector to represent geometry model inside given cell.

Although such geometry linearization has its advantages it is a source of certain difficulties as well. If cell intersects some fine geometry feature such as thin solid plate, or on the contrary, a thin crack in solid model, or some spire, it becomes very hard to approximate that geometry with one polygon. Such approximation may change physics of processes dramatically, e.g. allow fluids to flow where they should not. This problem may partly be solved by building base mesh wisely. By preliminary analysis of geometry model basic mesh may be projected so that opposite faces of thin solid plane occur inside two adjacent cells instead of one. Needle or spire may appear coaxial with cell edge occupying four adjacent cells and therefore be formed with four polygons. There are situations however when little can help but refining cell into smaller ones. Such cell where geometry can not be approximated with one polygon without principal changes to physics in the cell is called *irregular*. Irregular cells are usually subject to refinement to maximum allowed level or until become regular solid, fluid or partial cells.

## 2.4 Mesh Refinement

After base mesh is rendered every its cell undergoes a number of tests to decide should it be split into smaller ones. There are several reasons why it may be necessary to refine a cell. All these reasons are represented with some mathematical criterion thus forming a list of *refinement criteria*. Every cell is tested according to this list.

**Curvature criterion.** All geometry model surfaces are approximated with polygons inside mesh cells. If radius of surface curvature is comparable to characteristic size of cell then approximation of this surface with polygons inside cells will be very coarse. The greater the radius with respect to cell size the smoother approximation will be.

To control geometry approximation accuracy *curvature criterion* rule is applied to every partial cell. Normal to surface polygon of partial cell is compared by pairs with normals to surface polygons of immediate neighbour cells. These normal differences  $|n_1 - n_2|$  should be less than certain number  $c$ , curvature criterion. Otherwise the cell is split into next level cells to improve accuracy of geometry surface approximation.

**Passing areas ratio criterion.** Polygon approximating geometry surface divides some cell faces into two areas which belong to different media, e.g. solid and fluid, different solids, porous media, etc. Solver uses these areas during calculation when it needs information what part of every cell's face is passable for one or another flux, e.g. fluid flow, or heat flow in solids or electrical current in conductors.

Every cell face is common to two adjacent cells with the exception of computational region boundaries. Actual surface and polygons that approximate it generally intersect cell edges in different points. It is likely that polygons of two adjacent cells intersect common cell face differently. Thus, same cell face may have different ratio of two areas mentioned above depending on what cell is considered. For conservation laws to hold solver always uses same ratio for the same cell face regardless of cell it considers. What solver uses is some average of area ratios of two adjacent cells.

Even though conservation laws are not violated areas solver uses are not exact. They are different from what would real geometry surface produce. To control error related to this issue there is another refinement rule and criterion. Difference of area ratios  $|s_1 - s_2|$  given by considering adjacent cells should be less than some number  $s$ .

**Narrow channel resolution criterion.** Main focus of rules and criteria above is model geometry and its approximation in the mesh. This however is not the only concern in initial mesh generation. Physical behaviour in certain areas of computational region is sometimes predictable even before calculations. Thus, by initial conditions and geometry model base mesh is stretched and contracted in some places in anticipation of certain physical behaviour there. Another example when anticipated physical phenomena affect mesh generation is refinement of narrow channels or conductors.

Narrow channel is geometry feature with one linear dimension several times greater than any other. Pipes of various cross sections, cracks and narrow gaps between solid bodies, and so on, are common examples of narrow channels. If fluid can flow there it is likely to flow with high speed. The narrower the channel the higher flow speed can be expected. For high speed flows in narrow channels physical phenomena in boundary layers may have significant influence. Boundary layer is region of fluid close to solid body where flow speed is considerably lower than in the middle of the channel due to friction. The narrower the channel and higher the flow speed the more effect of

boundary layers is. Sometimes disregard of this effect may crucially change overall solution.

Similarly, high frequency currents will flow mostly through boundary layer of conductor due to skin effect. Energy flow occurs mostly on conductor surface exponentially diminishing with depth.

To take boundary layer phenomena into account there should be enough number of cells across every narrow channel. There is separate criterion for this purpose, average number of cells across any channel in the model. Regions inside all channels are refined until there are enough cells. This refinement is not linear and generates more cells close to channel walls and less in the middle of assumed stream. This pattern helps resolve future boundary layers while keeping overall number of cells minimal.

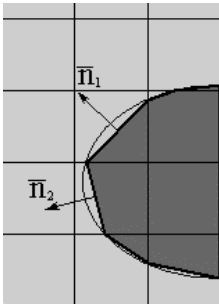


Fig. 3. Linear approximation of geometry

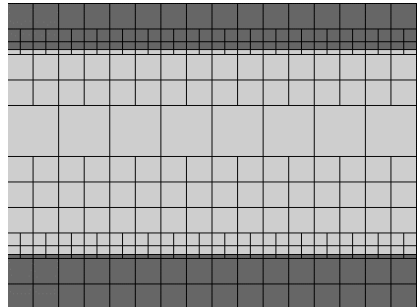


Fig. 4. Channel with eight cells fit across it

## 2.5 Local Mesh Settings

Great variety and complexity of real life problems makes it very difficult to automate all stages of mesh generation process. It is possible to recognize certain patterns of initial conditions and geometry in problems and build mesh accordingly. However, an engineer will sometimes know problem specifics better than any general purpose algorithm may detect. To give engineer some flexibility in these cases and therefore improve overall accuracy and timing performance *local mesh settings* are implemented.

Local mesh settings are non-overlapping regions of computational domain where all settings and criteria mentioned above have their local values. This is a tool to select specific region manually and demand higher geometry resolution in it or refine some solid body to certain level to better resolve heat flow in it or set different narrow channel resolution criteria for different channels, etc. All problem specifics known beforehand may be reflected with local mesh settings to improve calculation performance.

## 3 Refining Mesh during Calculations

Above principles help generate appropriate mesh to achieve good calculation accuracy and keep number of cells low to save calculation time. Also it is proved to be very efficient tactics to adjust mesh as problem solution develops.

These mesh adjustments or *solution adaptive refinements* may take place several times during calculation as solution picture changes. The idea of solution adaptive refinement is to locate regions with extreme gradients of physical parameters, either very low or conversely very high. Cells in regions with high gradients are subject to extra refinement to better resolve physical peculiarities such as pressure drops. At the same time cells of regions with almost constant parameters are merged into lower level cells.

Parameters to take into account may vary depending on cell medium and problem specifics. Pressure and velocity vector are important for gases and liquids; medium density for compressible flows; temperature is typical parameter for solids; wave magnitude and phase angle or field energy may be considered for electromagnetic fields.

In order to decide whether given cell should be split, merged or left untouched physical parameters in it and its immediate neighbours are compared. Differences of these values are used to calculate some weighted average that shows should this cell be split or merged if it should at all.

## 4 Summary

In this article we revealed basic principles of rectangular mesh generation. Problem of building optimal mesh to maximize ratio of achieved accuracy to computational time is very complex. One of the fundamentals is to use maximum information about every particular task to design mesh for its solution. This may help dramatically increase performance by concentrating on details of crucial importance while saving on others that may only contribute little error.

Different aspects of a problem may affect mesh generation. We have seen how model geometry features may influence this process as well as initial conditions and physical peculiarities anticipated in future solution. There are several steps in mesh design to take into account every such aspect of the problem.

Principles discussed here are not limited for use with PDE solvers and may be generalized for larger class of problems including integral equations.

## References

1. Joe F. Thompson, Bharat Soni and Nigel P. Weatherill: Handbook of Grid Generation. CRC Press Pub. (1998)
2. Vladimir D. Liseikin: Grid Generation Methods. Springer-Verlag (1999)
3. Graham F. Carey: Computational Grids : Generation, Adaptation, and Solution Strategies: Series in Computational and Physical Processes in Mechanics and Thermal Science. Taylor & Francis Pub. (1997)
4. Joe F. Thompson, Z. U. A. Warsi and C. W. Mastin: Numerical Grid Generation : Foundations and Applications. North Holland Pub. (1985)  
<http://www.erc.msstate.edu/publications/gridbook/>