Petr Kahánek, Alexej Kolcun

BRESENHAM'S REGULAR MESH DEFORMATION AND ANGLE CRITERIA

Abstract

In this paper, we investigate local deformations of structured meshes with respect to minmax angle and maxmin angle criteria. We introduce an effective algorithm to perform the predefined deformation (straight line) and we introduce a theorem according to which the generated triangulation is optimal with respect to both minmax and maxmin.

Keywords

Computational geometry, triangulations, structured meshes, minmax angle criterion

1 Introduction

In many areas of research, mathematical models of various real-world objects are used. By mathematical model, we often mean some system of partial differential equations (PDE) that have to be solved before any conclusion can be done. Generally, it is very difficult to solve a differential equation exactly; it is therefore common to use some numerical technique.

One of the most widely used tools in this area is undoubtly finite element method (FEM). Its simplest case is based on triangular elements with linear interpolation of the solution.

To obtain a "good numerical solution", i.e. "as close to the exact solution as possible", one needs to generate a "good triangulation". In this paper, we will consider two criterions to distinguish between "good" and "bad" triangulations: minmax angle criterion and maxmin angle criterion. However, it will be shown that the minmax approach is more suitable for linear interpolation purposes (see chap. 2) [2],[3].

We will focus on a special type of triangulations, called structured triangulations or simply structured meshes (chap.3).

The main goals of this paper are to introduce simple and efective technique of structured mesh deformation and to introduce a theorem that will evaluate quality of the resulting triangulation.

2 Triangulations

Let S be a finite set of points in the Euclidean plane. A **triangulation** of S is a maximally connected straight line plane graph whose vertices are the points of S. By maximality, each face is a triangle except for the exterior face which is the complement of the convex hull of S [1].

2.1 Delaunay triangulations

Triangulation of a set of points S is a Delaunay triangulation, if and only if the circumcircle of any triangle does not contain any other points of S in its interior.

It is Delaunay triangulation that is usually used when a triangulation is needed. It can be generated very effectively, in time O(n log n) for expected case, where n is the number of points to be triangulated. Moreover, a Delaunay triangulation maximizes the minimal angle. It even performs better than that: it lexicographically maximizes an angle vector V = (α_1, α_2, ..., α_{3t}), $\alpha_1 \leq \alpha_2 \leq ... \leq \alpha_{3t}$, where t is the number of triangles. [4]

A Delaunay triangulation is dual to another fundamental structure of computational geometry, Voronoi diagram. At this point, we will use Voronoi diagram to show the problem we are forced to deal with when using our mesh for interpolation purposes.

When the values are interpolated from the vertices to the interior area, it is suitable to have triangles of aproximately equilateral shape.Triangulation of a set of points S is a Delaunay triangulation, if and only if the circumcircle of any triangle does not contain any other points of S in its interior.

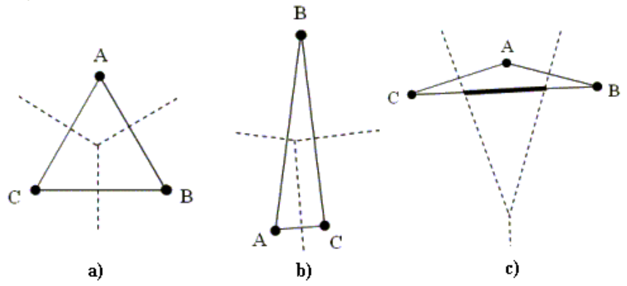

Figure 1: Voronoi diagrams for a) equilateral triangle, b) triangle with a "very sharp" angle, c) an obtuse-angle triangle

See Figure 1. On the left (a), we can see a triangle that is just perfect for interpolation - all its vertices "have the same influence" over the triangle.

For a triangle with one “very sharp” angle (b), the situation is not that good, but at least it is consistent in the following meaning: each vertex should affect primarily those points inside the triangle which are its close neighbours. Now, lets take a look at what happens when an obtuse angle appears (c).

One can see that vertex A **does not** affect big part of line BC, although it is much closer to it than any of vertices B and C. This may cause big interpolation errors. So we will want to avoid "big" angles [2].

Unfortunately, although a Delaunay triangulation can help us a little in this aspect , it is generally not the one that minimizes the maximal angle. An example that illustrates this is on the Figure 2.

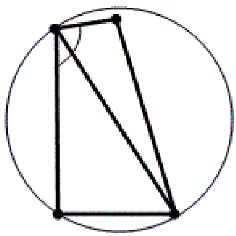

Figure 2: Minmax triangulation that is not Delaunay.

The diagonal divides the largest angle, producing a minmax triangulation. But there also is a triangle, which fails Delaunay criterion, because its circumcircle is not empty.

2.2 Minmax angle triangulations

Minmax angle triangulation is the one that minimizes the minimal angle over all possible triangulations. It is therefore the one we would like to use for linear interpolation.

The bad news is that the best known algorithm for minmax triangulation needs time O(n^2 log n) in the worst case [1]. However, for some special cases, a Delaunay triangulation satisfies minmax criterion. We will use this observation later.

3 Structured meshes

Let us have a regular, orthogonal grid (its elements are identical squares). Now, lets deform the grid so that it respect the given geometry. The result of this deformation is a structured mesh. It is obvious (and important) that it is topologically equivalent to the original grid.

Structured triangulation is a structured mesh, whose elements (quadrilaterals) were triangulated, i.e. for every element, one of the two possible diagonals was chosen.

There is one great advantage of a structured mesh – we can obtain very efficient form of the incidence matrix of structured mesh. The incidence matrix of nodes plays a significant role for most kinds of simulations. It depends on the manner of node indexing. Fig.4 shows incidence matrix of the regular grid, where nodes are indexed a) randomly, b) in the "spiral way from the boundary to the interior" c) in the "natural way – row by row". We can see that the last method gives us the best form of the incidence matrix, where all nonzero elements are concentrated in the regular structure near the main diagonal. As a result, use of a structured mesh allows us to solve much larger problems (in terms of number of points), because effective methods for storage and manipulations with incidence matrix are offered in this case [2, 3].

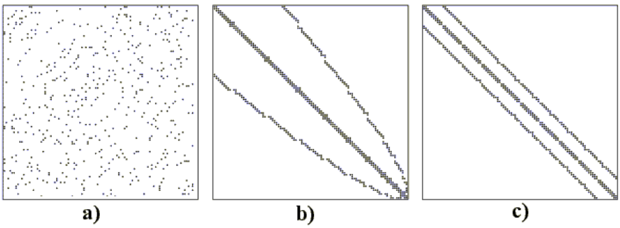

Figure 3: Incidence matrices for different indexing of nodes

On the other hand, there is a disadvantage, too: structured meshes are not suitable for refinement. However, the technique of nested meshes can solve this problem (Multigrid methods which use this idea are used for PDE solving) [3].

4 Structured triangulations with defined geometry

Before any calculations can be done, it is neccesary to deform the mesh so that it respects the geometry inherent to the given problem. In general, such geometry can be very complicated; it is, however, a common task to model some basic shapes, i.e. curves in 2D, surfaces in 3D. Here, we will deal with the simplest curve type in two dimensions, line segment.

Start in I_1
Until we have reached I_2
    Find indices *x*, *y* of the next node (via Bresenham)
    Move it so that it lies on the line
    If both *x* and *y* have changed (in Bresenham)
        Force corresponding diagonal

Algorithm 1: Local mesh deformation

There are different ways of mesh deformation [2, 3]. Usually, mesh generators use some kind of parametrization; they, in fact, divide the mesh into several smaller meshes, which are "glued" together with the modelled boundary.

Let us suppose that we have a line segment, whose ends are fixed in nodes of the original (not yet deformed) mesh, say I_1 and I_2. For the process of deformation, we will use an adaptation of the well-known Bresenham's algorithm. We will use it to find out which nodes should be moved during deformation. See Algorithm 1.

It is important to say that we move nodes only in direction of **one axis** (x or y). This is determined by slope α of the given line: for $0 \leq \alpha \leq \pi/4$, we will move the nodes only vertically, for $\pi/4 < \alpha \leq \pi/2$ horizontally.
After the deformation is done, we have to triangulate our mesh. But since we are in 2D, this is trivial - there are just two possible diagonals in each element. We will use the minmax criterion to choose one. The whole process of generating the required triangulation is illustrated on the Figure 4.

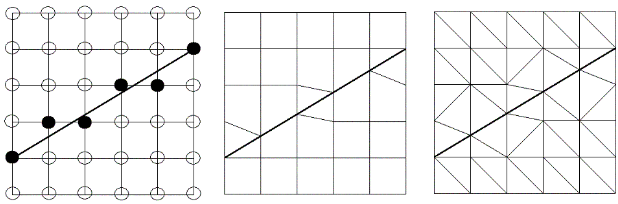

Figure 4: Mesh deformation (line segment modelling).

Now we would like to know whether our algorithm produces "nice" meshes in terms of angle criterions mentioned above. It shows that it does, since we can introduce this pleasing theorem:

Theorem. *Generated triangulation is optimal with respect to both minmax and maxmin angle criterions.*

Unfortunatelly, we don't have enough space to present the proof. It can be, however, found in [2].

By trivial observation, we can see that the theorem holds for the original (undeformed) grid - a square is an example of a quadrilateral, for which minmax and maxmin diagonal does not differ. The theorem also gives us an idea that with use of specific transformations, this relationship remains preserved. And what's more, the whole process is straightforward and easy to implement.

We can even weaken our assumption that the ends of the line are fixed in the nodes of the original mesh: for $0 \leq \alpha \leq \pi/4$, the ends can be allowed to change their y-coordinate (x-coordinate for $\pi/4 < \alpha \leq \pi/2$, respectively).

If we abandon the assumption completely, a little annoyance may occur, see Figure 5. The grey element obviously does not satisfy the minmax criterion. However, even with this possibility, our approach is worth fighting for: there at most two problematic places for one line segment (one by each end node) and the worst angle cannot be greater than $2\pi/3$.

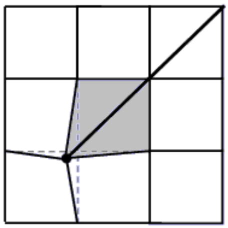

Figure 5: A possible problem when the assumption of fixed ends is abandoned.

5 Applications

As we mentioned before, there is a need of several meshes during FEM preprocessing, which vary in density a lot. However, their other attributes should be more or less the same: all of them should respect given geometry and all of them should fulfill certain quality criteria

Widely used parametric approach, in general, generates well-shaped triangles. But there is also a disadvantage: the generated triangulation is not homogenous in terms of density (and hence size of elements), see Figure 6.

This is not suitable for generating coarse meshes, since their elements should be approximately equally sized - a good reason to use local deformations. Our algorithm even produces minmax optimal triangulation, so there's no need to involve unstructured mesh.

In Table 1, we present results of experiments. We modelled line segments with three different slopes (5, 25 and 40 degrees) on a regular mesh of 10x10 nodes.

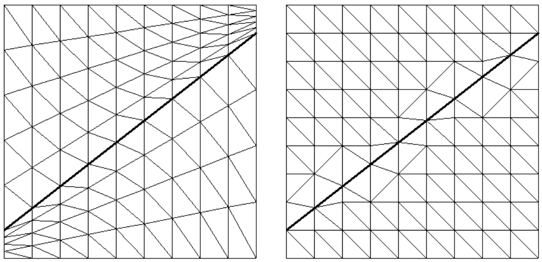

Figure 6: Parametrization vs. local deformations

We can see that for the 40 degrees case (which is also on Fig.14), parametric approach generates quite bad triangles.

Table 1: Experiments

Loc. def. / Param.	5^o	25^o	40^o
Max	90.0 / 90.0	90.0 / 90.0	103.0 / 113.0
Min	29.1 / 38.7	29.1 / 26.6	25.3 / 9.0
avg. max	85.9 / 86.8	79.3 / 78.1	83.0 / 80.4
avg. min	39.8 / 43.2	43.6 / 43.0	43.4 / 38.2

6 Conclusion

We showed that certain kind of mesh deformation produces triangulations optimal with respect to both minmax angle and maxmin angle criteria. We also implemented algorithm for line segment modelling.

In the future, we would like to continue in our efforts to investigate local deformations of regular meshes.

1) A question arises whether it is possible to preserve good properties of the resulting triangulation when predefined geometry doesn't touch the diagonals of grid cells.
2) Generalization of our approach to model more complex shapes is interesting.
3) Generalization of our approach to 3D is important.

7 References

[1] Edelsbrunner H., Tan T. S., Waupotitsch: An O(n^2 log n) time algorithm for the minmax angle triangulation, *SIAM J. Scientific & Statistical Computing*, 13 (4), 994-1008, 1992

[2] Kahánek, P.: Strategie minmax a maxmin v pravidelných sítích, Diploma thesis, The University of Ostrava, 2004 (In Czech)

[3] Kolcun, A: Preprocessing pre aplikaciu MKP v ulohach geomechaniky, disertation thesis, Institute of Geonics, AS CR, 1999 (In Slovak)

[4] Kolingerová, I: Rovinne triangulace, habilitation thesis, University of West Bohemia, Pilsen, Czech Republic, 1999 (In Czech)