A parametric method for unstructured mesh generation

Jan Kucwaj, Andrzej Karafiat[*]

Institute of Computer Modelling, Cracow University of Technology,
ul. Warszawska 24, 31-155 Cracow, Poland

Abstract

The paper presents a method of mesh generation on the surfaces based on the parametric representation of these surfaces and mesh construction in 2-D reference domains. A density function is involved. The method is illustrated by some examples.

1. Introduction

Unstructured mesh generation is the object of interest of many researchers and programmers [1-3]. It is the first necessary step for the application of the Finite or Boundary Element Method for solving many technical problems. The size of the problems actually considered brings about an automatic generation of meshes. A variable density function allows for mesh concentration in the areas where a solution changes rapidly or loses its regularity. It can be especially effective in adaptive techniques. Unstructured mesh is a tool for the discretization of complicated surfaces too.

The presented approach can be applied to the surfaces which may be divided into regular parts of the form $S = x(D)$ where D is a bounded domain of the plane. The mesh is first generated on the edges dividing these parts. Each part the mesh is constructed in D using the advanced front technique and next mapped onto S.

The method is defined in Section 2. In Section 3 mesh generation on boundaries is presented. Generation of the mesh inside a surface is discussed in Section 4. Mesh examples are shown in Section 5.

The presented method is based on the authors' previous results [4]. For a review of meshing methods refer to [5-6].

---

[*] Corresponding author: *e-mail address:* akaraf@usk.pk.edu.pl

2. Parametric representation

We begin with a parametric representation of the surface

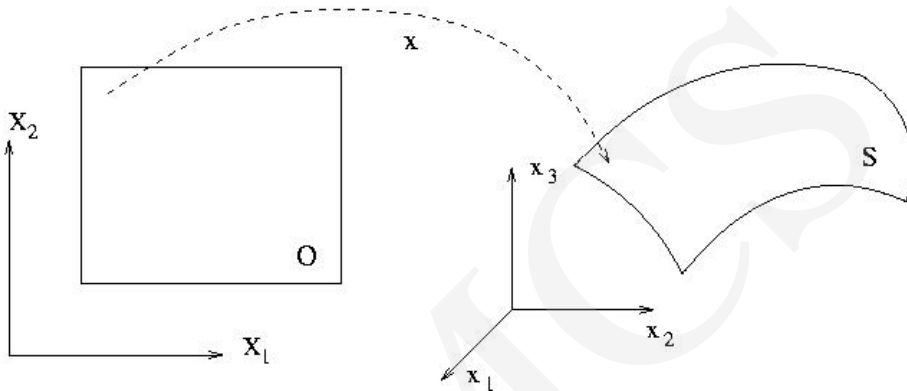$$x = x(X), \quad x = (x_1, x_2, x_3), \quad X = (X_1, X_2). \tag{1}$$

Fig. 1. Parametric representation of a surface

We assume that the representation is regular enough to perform all the calculations below.

The basic idea is to generate with reference to domain Ω and next to map it onto surface S. As an element of mesh generation we take the density function

$$r = r(x),$$

which in practice is replaced by the distance function

$$d(x) = r^{-1}(x).$$

The constructed mesh is expected to satisfy the condition

$$| \overline{ab} | = 0.5[d(a) + d(b)], \tag{2}$$

for the neighbouring nodes a, b on S where $|\overline{ab}|$ is the distance between the points a and b. This distance is expressed using the known differential geometry formula

$$| \overline{ab} | = \int_{t_a}^{t_b} \sqrt{g_{ij}(X(t)) \, dX_i\{t\} \, dX_j(t)} \, dt, \tag{3}$$

where

$$a = x(X(t_a)), \quad b = x(X(t_b)), \quad g_{ij}(X) = x_{,i}(X) \, x_{,j}(X) = \frac{\partial x}{\partial X_i}(X) \frac{\partial x}{\partial X_j}(X) \quad \text{is}$$

the metric tensor and the summation convention is used.

3. Boundary mesh

To construct the mesh on the boundary of *S* we use its parametrization
$$x = x(X(t)), \quad t \in [t_0, t_e].$$

Beginning from $x(t_0) = x(X(t_0))$ we look for such $t_1 \in (t_0, t_e]$ that
$$| x(X(t_1)) - x(X(t_0)) | = 0.5[d\ (x(t_0)) + d\ (x(t_1))]. \tag{4}$$

We approximate the above distance by the formula (3) and the unknown value $d(x(t_1))$ is interpolated linearly

$$\sqrt{g_{ij}(X(t_0))\frac{dX_i}{dt}\frac{dX_j}{dt}}(t_1 - t_0) = d(x(t_0)) + 0.5\frac{\partial d}{\partial x_i}\frac{dx_i}{dt}(t_1 - t_0), \tag{5}$$

then

$$t_1 = \frac{\sqrt{g_{ij}(X(t_0))\dfrac{dX_i}{dt}\dfrac{dX_j}{dt}}\,t_0 + d(x(t_0)) - 0.5\dfrac{\partial d}{\partial x_i}\dfrac{dx_i}{dt}t_0}{\sqrt{g_{ij}(X(t_0))\dfrac{dX_i}{dt}\dfrac{dX_j}{dt}} - 0.5\dfrac{\partial d}{\partial x_i}\dfrac{dx_i}{dt}}. \tag{6}$$

Point t_2 is obtained analogously from t_1 and so on. After some steps we may get $t_n > t_e > t_{n-1}$. We evaluate then
$$e = t_n + t_{n-1} - 2t_e.$$

If $e < 0$, i.e. $e = t_n - t_e < t_e - t_{n-1}$ we define
$$r_t = \frac{t_n - t_0}{t_e - t_0}.$$

In the opposite case we remove point t_n and define
$$r_t = \frac{t_{n-1} - t_0}{t_e - t_0}.$$

The nodes $t_1, ..., t_{n-1}(t_n)$ are replaced by $t'_1, ..., t'_{n-1}(t'_n)$, where $t_i' = t_0 + (t_i - t_0)r_t^{-1}$, $i = 1, ..., n-1\,(n)$.

In this way a mesh on each part of the boundary is fixed.

4. Internal mesh generation

For mesh generation inside the reference domain Ω the advancing front technique is used. Nodes on the surface are obtained from Ω by the mapping x, i.e. $a = x(A)$, $b = x(B)$, $c = x(C)$.

An initial front consists of the nodes and segments on the boundary. On the front of each node a an angle between two edges $\overline{ab}, \overline{ac}$ is calculated from the formula

$$\cos a = \frac{g_{ij}(A)dX_i^B dX_j^C}{|\overline{ab}||\overline{ac}|} \tag{7}$$

where $dX^B = \overline{AB}$, $dX^C = \overline{AC}$ (cf. Fig.2). Nodes of the front are divided into three groups with respect to the angle prescribed

1. $a > 2p/3$;
2. $p/3 \le a \le 2p/3$;
3. $a < p/3$.

Nodes in groups are consecutively considered.

For nodes belonging to the first group a new node $D \in \Omega$ is chosen by the formulas

$$\cos a_0 = \frac{g_{ij}(A)dX_i^B dX_j^D}{|\overline{ab}||\overline{ad}|}, \tag{8}$$

$$\sin a_0 = \frac{\sqrt{g}\left|dX_1^B dX_2^D - dX_2^B dX_1^D\right|}{|\overline{ab}||\overline{ad}|}, \tag{9}$$

where $a_0 = P\!\big/\!{3}$, $g = \begin{vmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{vmatrix}(A)$ and $dX^D = \overline{AD}$ (unknown).
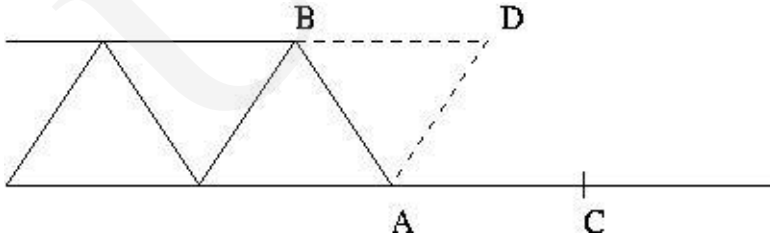

Fig. 2. New node construction when α>2π/3

We interpolate value of $d(x(D))$

$$|\overline{ad}| = 0.5[d(a) + d(d)] = d(a) + 0.5\frac{\partial d}{\partial X_i}dX_i^D, \tag{10}$$

and obtain a system of equations (we assume the orientation of the plane like in Fig. 2).

$$\left(g_{11}dX_1^B + g_{21}dX_2^B - 0.5\cos a_0 |\overline{ab}|\frac{\partial d}{\partial X_1} \right)dX_1^D$$
$$+ \left(g_{12}dX_1^B + g_{22}dX_2^B - 0.5\cos a_0 |\overline{ab}|\frac{\partial d}{\partial X_2} \right)dX_2^D = \cos a_0 |\overline{ab}|d(a) \tag{11}$$

$$\left(-\sqrt{g}\,dX_2^B - 0.5\sin a_0 \,|\,\overline{ab}\,|\,\frac{\partial d}{\partial X_1}\right)dX_1^D$$
$$+\left(\sqrt{g}\,dX_1^B - 0.5\sin a_0 \,|\,\overline{ab}\,|\,\frac{\partial d}{\partial X_2}\right)dX_2^D = \sin a_0 \,|\,\overline{ab}\,|\,d(a) \tag{12}$$

From this system of equations we obtain dX^D and the point D.

Next it is to verify whether new edges $\overline{AD}, \overline{BD}$ cross the existing front edges. In this case node D is moved to the nearest front node.

In the second group of nodes, when $p/3 \le a \le 2p/3$ we use the same construction of the node D taking $a_0 = 0.5a$ defined in (7). The value of the error

$$e = \left\{|\overline{bc}| - 0.5[d(b) + d(c)]\right\} + \left\{|\overline{ad}| - 0.5[d(c) + d(d)]\right\}$$

is evaluated. If $e > 0$, then the node D with the edges $\overline{AD}, \overline{BD}$ is included in the front. In the other case, the node D is not accepted and the new triangle ABC is built.
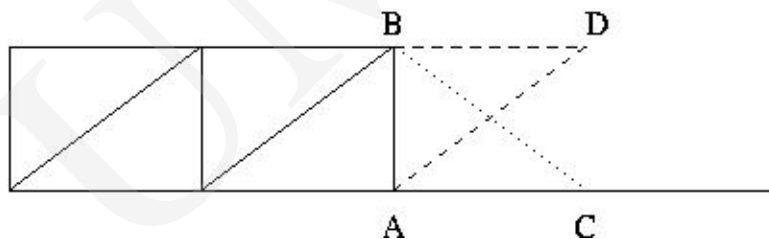

Fig. 3. New node/triangle construction when π/3<α2π/3

In the last group, when $a < p/3$ for each node A the triangle ABC is constructed and the edge $\overline{BC}$ becomes a new front edge.
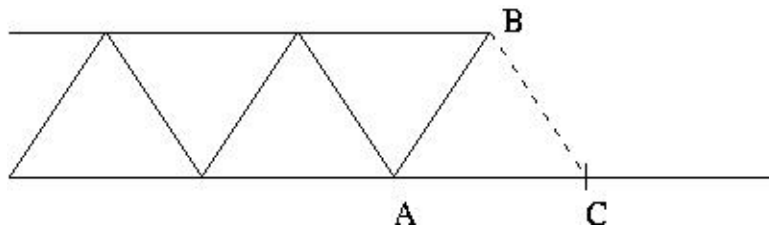

Figure 4. New triangle construction when α<π/3

This method is repeated in next steps until the front reduces to the empty set. The mesh is then smoothed, using the Laplacian method. In general smoothing algorithm passes 5-10 times the whole mesh.
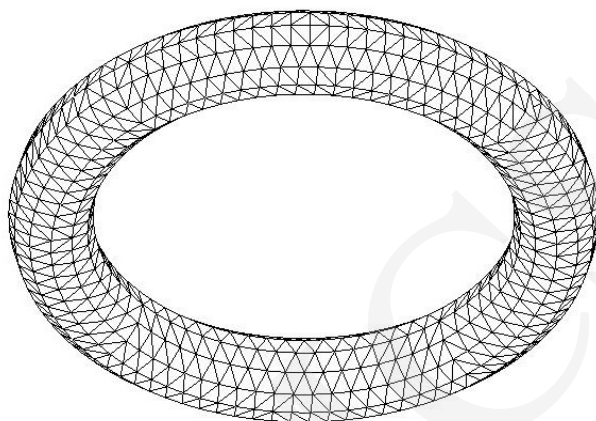
5. Examples

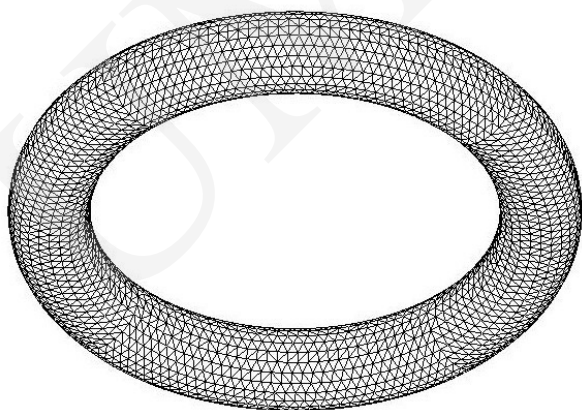Fig. 5. Unstructured mesh on a torus – coarse mesh

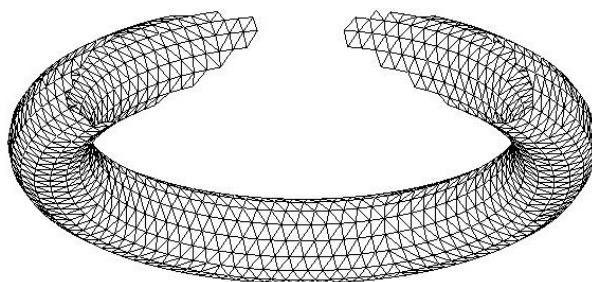Fig. 6. Unstructured mesh on a torus – fine mesh

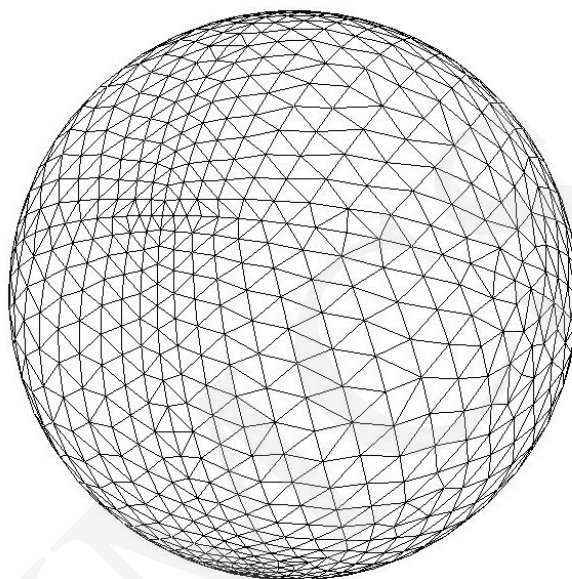Fig. 7. Unstructured mesh on a torus – mesh construction

Fig. 8. Unstructured mesh on a sphere

Generation of the mesh presented in Figure 6 (7 980 nodes and 15 960 triangles) needed 0.03 sec CPU time at the computer Pentium IV 1.5GHz.

6. Conclusions

The presented method guarantees that all nodes are placed on the surface. Triangles are an approximated surface which interpolates the real one. The distance between both can be easily estimated using the well-known interpolation inequalities [7]. If it must be bounded from above, appropriate surface parameters like the second quadratic form of the surface can be included into the density function.

In spite of the fact that the Advancing Front Technique is performed in a two-dimensional reference domain, the size of the mesh is controlled by the use of the metric tensor. Another way of surface measuring is presented in [8].

The CPU time even in the case of meshes consisting of many thousands of nodes is a few seconds. It means that it can be used in adaptive methods for solving boundary-value problems. Moreover, in [2] and [3] parallel versions of the Advancing Front Technique is applied.

The authors' purpose for presenting this method is the first step for the construction of unstructural three-dimensional mesh generator. Another application is an adaptive method for the Boundary Element method in 3-D. In

this application, mesh density function may be determined by an error indicator or by foreseen properties of the solution of the problem.

Further research combines the Delaunay Methods [1, 2] and AFT may lead to more efficient algorithms and improve the quality of the final grids.

References

[1] Frey P.J., *Delaunay Mesh Generation Coupled with an Advancing Front Approach*, Comput. Methods Appl. Mech. Eng., 157 (1998) 115.
[2] Lo H., *Generating Quadrilateral Elements on Plane and over Curved Surfaces*, Comput. and Struct,. 21 (1998) 421.
[3] Loehner J., *A Parallel Advancing Front Grid Generation Scheme*, Int. J. Numer, Methods Eng., 51 (2001) 663.
[4] Stasik J., Kucwaj J., Karafiat A., *Generation of Unstructured Meshes on Surfaces,* Proc. 2nd Europ. Conf. on Comp. Mechanics, June (2001), Cracow, Poland (CD-ROM).
[5] George P.L., *Automatic Mesh Generation*, Wiley, Chichester, (1991).
[6] Thompson J.F., Soni B.K., Weatherill N.P., *Handbook of Grid Generation*, CRC Press, Boca-Raton, (1999).
[7] Ciarlet P., *The Finite Element Method for Elliptic Prob.*, N.Holland, Amsterdam, (1978).
[8] Owen S.J., Saigal S., *Surface Mesh Sizing Control*, Int. J. Numer, Methods Eng., 47 (2000) 497.
[9] Mezentier A., Hamman O., Weatherill N., *Geometric Model Analyses for Unstructured Surface Mesh Generation*, Proc. 6-th Int. Conf. "Numerical Grid Gener. in Field Simulation and Related Fields", Mississippi St. Univ., (2000) 619.