MECHANIKA TEORETYCZNA
I STOSOWANA
Journal of Theoretical
and Applied Mechanics
2, 35, 1997

MODELLING AND TRIANGULATION OF GEOMETRICALLY COMPLICATED PLANE DOMAINS AND SOME APPLICATIONS TO FLUID MECHANICS

JAN KUCWAJ

Section of Applied Mathematics, Cracow University of Technology

An algorithm of modelling and triangulation of plain domains is presented. For the purpose of geometrical modelling of 2D domains appriopriate topology is defined. B-spline curves represent the boundary. An algorithm of generation of graded grids on B-spline curves is worked out. The internal points are generated by using the advancing front technique. The triangulation is performed by means of the Delaunay advancing front method. Numerical examples present meshes over airfoils.

Key words: grid generation, Delaunay triangulation, geometric modelling

1. Introduction

The paper is focused on modelling and triangulation of 2D domains of complex geometry. It especially concerns the profiles (airfoils), which are commonly met in fluid mechanics. An important class of them is the NACA (with or without a flap) family of wings or Korn's airfoil profile (see Derivieux et al. (1989)). Boundaries of these domains are described by sets of points, which are approximated by B-spline functions. Having the domain modelled in this way, we triangulate it using the algorithm being the subject of the paper. Triangulation of such domains is necessary for the FEM (finite element method) or the FDM (finite difference method) computations in fluid mechanics. In the case of airfoil profiles it is necesary to generate graded finite element meshes of the size of the order $10^{-\sqrt{\mathrm{Re}}}$ along the boundary, where Re is the Reynolds number, and gradually increasing when reaching the part of the boundary (on which Sommerfeld's conditions are imposed) defining the computational domain. It is assumed, that there is a prescribed mesh density function defining the sizes of elements and the problem is how to generate a finite element mesh according to the given density.

In the paper, numerical algorithms for mesh generation are proposed and their computer implementations are presented. In conclusion some numerical examples are presented. The mesh generation procedure with the mesh density function can also be used as a remeshing algorithm (see Zienkiewicz and Zhu (1991)), where according to the local error estimate, a completely new mesh is generated resulting in a better approximation of the solution. In the present approach, the idea of mesh generation is a continuation of the approach for uniform meshes (see Kucwaj (1992), (1993), (1995)). The algorithm of mesh generation and triangulation can be divided into the following steps:

- Boundary points generation with a given mesh density function ρ
- Internal points generation according to a given mesh density function ρ
- Triangulation of the domain on the previously obtained grid points
- Laplacian smoothing of the mesh.

In author's opinion new contributions of the paper are as follows:

1. Internal and boundary points generation with the aid of advancing front technique
2. The modelling of boundary by B-spline curves (see Bartels et al. (1989))
3. Computer code and data structure for generation of points and triangulation with a mesh density function
4. Generation of graded meshes on B-spline curves
5. Numerical examples illustrating the algorithm.

Further developements will be connected with: improvements of points generation, extension of the code by new algorithms, generation of grid points on surfaces with a given mesh density function, application of the method to remeshing algorithms, construction of a mesh density function for some fluid mechanics problems.

2. Main assumptions about the boundary representation

Topology and appriopriate data structure for a plane domain triangulation are connected with different types of curves which represent pieces of the boundary (cf Kucwaj (1995)). A curve is represented by its ends and the type. Therefore all the curves available having a representation in the computer data structure are topologically equivalent to a straight line segment.

It is assumed, that the domain $D \subset \mathbb{R}^2$ is multiconnected, say, k-connected with a finite number of internal contours. The boundary of the domain is represented as a union of closed curves, without multiple points. Every loop is represented as a sequence of curves, appearing in the order, in which the end of any curve is the beginning of the next one. In a special case the end of the last curve is the beginning of the first one. As it was assumed every curve was topologically equivalent to a straight line segment with the ends belonging to a fixed set of points $\mathcal{P}$. The curves are counter-clockwise with respect to the domain.

In the computer data structure the set of points $\mathcal{P}$ and the set of curves $\mathcal{C}$ are introduced

$$\mathcal{P} := \{P_{jk} \ : \ j = 1, \ldots, N_l, \quad k = 1, \ldots, N_p\}$$

where

N_l – number of loops

N_p – number of points in the jth contour

$$\mathcal{C} := \{C_i \ : \ i = 1, \ldots, N_c\}$$

where

N_c – number of curves

The following types of curves are available: a straight line segment, an arch of circle, a B-spline curve.

The computer code is open, so any new type of curve can be introduced.

It is assumed, that the curves do not intersect, i.e.: $\forall i, j = 1, \ldots, N_c$, $i \neq j$, $C_i \cap C_j = \emptyset$.

An additional assumption about the curve is: $\forall i = 1, \ldots, N_c$ $\exists$ *integers* l_1, l_2, $1 \leq l_1, l_2 \leq N_p$, such, that P_{l_1}, P_{l_2} are the endpoints of the curve C_i.

For the boundary $\partial\Omega$ of the domain Ω it is assumed, that there exists a set of numbers $k_1, k_2, \ldots, k_n$, such that

$$\partial\Omega = \bigcup_{s=1}^{n} C_{k_s} \tag{2.1}$$

On the other hand, $\forall i, 1 \leq i \leq N_p\ \exists j, 1 \leq j \leq N_c$, such that point P_i is an endpoint of the curve C_k.

It is not necessary to assume, that every curve of the set C must be a piece of $\partial\Omega$. It means that it is possible to define the curves lying in the interior of Ω and to generate points on it before generation of internal points.

3. Generation of points on curves

In this section the algorithms of generation of points on curves with a given mesh density function are presented. Particular algorithms correspond to the type of the curve. In the case of a straight line segment the technique after Lo (1991) is applied and for an arch this algorithm is adapted as well. In the case of B-spline curves a technique based on points generation on a control polygon and then projection of them on the curve is worked out.

3.1. Generation of points on a straight line segment

The problem of points generation with a mesh density function is not new and was a matter of interest of many authors (cf Knupp (1991), Lo (1992)). In this paper the procedure of points generation on a straight line segment follows the technique presented by Lo (1991). The only difference between the non-uniform points generation and the present algorithm lies in the fact, that there is a value dx defining the length of the increment dx partitioning the considered straight line segment into pieces of equal size to interpolate the mesh density function with piecewise linear function. It means, that the increment dx must be given. The procedure of the paper is simplified in comparison with that given by Lo (1991), since the uniform grid has been used to the mesh density function approximation over the straight line segment.

Consider the straight line segment $[a,b]$ (Fig.1). Let us introduce the following notations

$$m = E\Big(\frac{b-a}{dx} + 0.5\Big) \tag{3.1}$$

where $E(x)$, means the integer part of x,

$$\Delta x = \frac{b-a}{m} \tag{3.2}$$

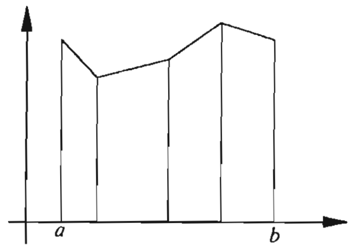

Fig. 1. The interpolation of the mesh density function

where a, b are the endpoints of the straight line segment,

$$P_i = a + i\Delta x \qquad i = 0, 1, \ldots, m \tag{3.3}$$

Additionally, it is assumed that, the mesh density function $\rho : [a, b] \mapsto \mathbb{R}$ is given. Then the algorithm of points generation looks like follows:

the first point of the considered grid is P_0. Let us assume, for an induction step, that A is just generated, the next point B should satisfy the following condition

$$|AB| = \frac{\rho(A) + \rho(B)}{2} \tag{3.4}$$

Consider the following cases:

- Case (i)

A, B lie in the same interval $[P_{i-1}, P_i]$, (see Fig.2), let

$$\begin{aligned} A &= P_{i-1} + \mu(P_i - P_{i-1}) \\ B &= P_{i-1} + \lambda(P_i - P_{i-1}) \end{aligned} \tag{3.5}$$

where

$$0 \leq \lambda \qquad \mu \leq 1 \tag{3.6}$$

The condition (3.6) is satisfied by B if and only if $B \in [P_{i-1}, P_i]$. The problem is how to find λ.

Taking into account Eqs (3.4) and (3.5), the following equations are satisfied

$$|AB| = |(\lambda - \mu)||P_{i-1}P_i| = \frac{\rho(A) + \rho(B)}{2} \tag{3.7}$$

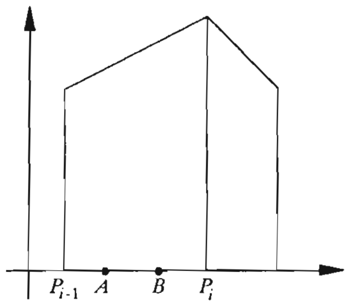

Fig. 2. Case (i)

On the other hand, the linear interpolation $h(x)$ of the mesh density function is

$$h(x) = \frac{h_i - h_{i-1}}{P_i - P_{i-1}}(x - P_{i-1}) + h_{i-1} \tag{3.8}$$

where $h_i = \rho(P_i)$, $i = 0, 1, \ldots, m$. Substituting of Eq $(3.5)_2$ into Eqs (3.8) we have

$$h(B) = h_{i-1} + \lambda(h_i - h_{i-1}) \tag{3.9}$$

Introducing Eq (3.9) into Eq (3.7) and the formula for $h(B)$ instead of $\rho(B)$, the following equation is obtained

$$(\lambda - \mu)|P_{i-1}P_i| = \frac{\rho(A) + h_{i-1} + \lambda(h_i - h_{i-1})}{2} \tag{3.10}$$

From Eq (3.10) by introducing the notation $r_i = |P_{i-1}P_i|$ we obtain

$$2\lambda r_i - 2\mu r_i = \rho(A) + h_{i-1} + \lambda(h_i - h_{i-1}) \tag{3.11}$$

then

$$\lambda(2r_i + h_{i-1} - h_i) = \rho(A) + h_{i-1} + 2\mu r_i \tag{3.12}$$

hence

$$\lambda = \frac{\rho(A) + h_{i-1} + 2\mu r_i}{2r_i + h_{i-1} - h_i} \tag{3.13}$$

The number λ should satisfy the following condition

$$0 \leq \lambda \leq 1 \tag{3.14}$$

if it does not satisfy the inequalities then the next case should be considered:

- Case (ii)

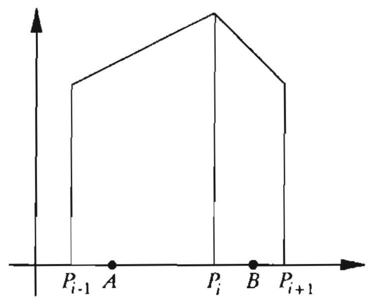

Fig. 3. Case (ii)

Points A, B are not lying the same straight line segment (see Fig.3). Strictly speaking the point A lies in the interval $[P_{i-1}, P_i]$, but the point B lies in $[P_i, P_{i+1}]$.

Introducing the following notations

$$l_i = |AP_i| \qquad\qquad r_i = h_{i+1} - h_i \tag{3.15}$$

then for some $0 \le \lambda \le 1$

$$|P_iB| = h_i + \lambda(h_{i+1} - h_i) \tag{3.16}$$

The point B should satisfy the condition

$$|AB| = \frac{1}{2}[h(A) + h(B)] \tag{3.17}$$

where h is defined by the piecewise linear approximation of the mesh density function ρ, so we have

$$h(B) = h_i + \lambda(h_{i+1} - h_i) \tag{3.18}$$

Then by the inserting Eq (3.18) into Eq (3.17) we obtain

$$h(A) + h_i + \lambda(h_{i+1} - h_i) = 2l_i + 2\lambda r_i + 2h_i \tag{3.19}$$

From the last equation the value λ is

$$\lambda = \frac{h(A) + h_i - 2l_i}{r_i} \tag{3.20}$$

If $\lambda > 1$ then the next segment to P_iP_{i+1} is considered, and so on.

If all the straight line segments are taken into account and all values of λ are greater than 1 then the last point of the mesh is just A, and the length of the interval $[A, P_{m+1}]$, where m is the number of generated points, is proportionally distributed among the previously defined segments according to their lengths. Let us introduce the following notation

A_j, $j = 0, 1 \ldots, m$ – the generated set of points,

$\delta_j = |A_jA_{j+1}|$, $j = 0, 1, \ldots, m-1$.

A new set of points A_j^*, $j = 0, 1, \ldots, m-1$ is defined by using the recursive formula as

$$A_0^* \doteq A_0$$
$$A_{j+1}^* \doteq A_j^* + \delta_j^* \qquad j = 0, 1, \ldots, m-1$$

Let $r = |AP_{m+1}|$, and δ_j be the length of the ith line segment, then new lengths of the straights line segments can be defined by the following formula

$$\delta_j^* = \delta_j + r\frac{\delta_j}{\delta} \qquad j = 0, 1, \ldots, m-1 \tag{3.21}$$

where

$$\delta = \sum_{j=0}^{m-1} \delta_j \tag{3.22}$$

3.2. Generation of points on an arch of circle

A circle as every curve in the computer data structure is defined by its ends and, additionally, by the center of the circle containing the arch. Let us introduce the notation

S – centre of the circle

R – radius of the circle

$\boldsymbol{A}, \boldsymbol{B}$ – ends of the arch.

The idea of the points generation on the the circle is as follows:

1. The arch is mapped onto the interval, the length of which equals to the length of the arch
2. A new mesh density function is defined by using the density function given on the arch
3. The points on the interval with the newly given density function are generation

4. The obtained nodes are remapped onto the curve.

The introduced interval is equal to $[0, M]$, where

$$M = |\breve{AB}| \tag{3.23}$$

The mapping of the arch onto the interval is defined as follows

$$K : \breve{AB} \longmapsto [0, M] \tag{3.24}$$

where for every point $P \in \breve{AB}$, $K(P) = |\breve{AP}|$. The mesh density function on the interval $[0, M]$ is defined as

$$\rho_M(x) := \rho\Big(K^{-1}(x)\Big) \tag{3.25}$$

The points of the interval $[0, M]$ are defined by the technique presented in the previous section.

3.3. Generation of points on the B-spline curve

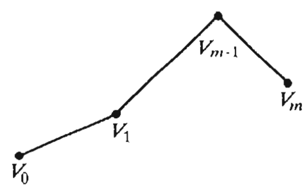

Fig. 4. Control polygon

A B-spline curve is defined by a set of control vertices $V_0, V_1, \ldots, V_m$, (see Fig.4) and the formula below (cf Bartels et al. (1989), Dierckx (1995)). The set of control vertices forms the control polygon. The B-spline curve is defined by the formula

$$C(t) = \sum_{i=0}^{m} V_i B_{i,p}(t) \tag{3.26}$$

where $B_{i,p}$ are B-spline functions of order p, defined by the recursive formula

$$B_{i,p} = \frac{(t - t_i)B_{i,p-1}(t)}{t_{i+p-1} - t_i} + \frac{(t_{i+k} - t)B_{i+1,p}(t)}{t_{i+p} - t_{i+1}} \tag{3.27}$$

for $p > 0$, and

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

if $p = 0$.

The B-spline functions are defined on the following knot vector (cf Bartels et al. (1989), Dierckx (1995))

$$V = \{\underbrace{t_0, \ldots, t_0}_{p+1}, t_1, \ldots, t_n, \underbrace{t_{n+1}, \ldots, t_{n+1}}_{p+1}\} \tag{3.28}$$

where

$$t_0 < t_1 < \ldots < t_n < t_{n+1} \tag{3.29}$$

and $m = n + 2p + 1$.

The generation of points on the B-spline curve starts with the generation of points on the control polygon and then the generated grid points are projected onto the B-spline curve to obtain the final grid. The projection is done by using a parametric equation of the curve, where points at first are generated on the interval, and then mapped onto the curve.

The proposed approach may be summarized as follows:

1. Define the interval $[0, M]$, where

$$M = \sum_{i=0}^{m-1} |V_i V_{i+1}| \tag{3.30}$$

Let C^* denote the control polygon defined by V_i, $i = 0, \ldots, m$. Then a mapping

$$K : C^* \longmapsto [0, M] \tag{3.31}$$

is defined as follows:

let $P \in C^*$ then

$$K(P) = \sum_{i=0}^{k-1} |V_i V_{i+1}| + |PV_{k+1}| \tag{3.32}$$

where k is such that $P \in V_k V_{k+1}$. It is easy to show, that $K(P) \in [0, M]$.

2. The knot vector

$$V = \{\underbrace{0, \ldots, 0}_{p+1}, t_1, \ldots, t_n, \underbrace{t_{n+1}, \ldots, t_{n+1}}_{p+1}\} \tag{3.33}$$

where

$$0 < t_1 < \ldots < t_n < t_{n+1} \tag{3.34}$$

is defined as

$$t_i = K(V_i) \qquad i = 0, \ldots, n+1 \tag{3.35}$$

3. Assume that a mesh density function

$$\rho : D \longmapsto \mathbb{R} \tag{3.36}$$

is given, where D is a 2D domain containing B-spline curve C, and control polygon C^*. Then the mesh density function

$$\rho_M : [0, M] \longmapsto \mathbb{R} \tag{3.37}$$

is defined by

$$\rho_M(t) = \rho(C(t)) \qquad t \in [0, M] \tag{3.38}$$

Finally, the points of the interval $[0, M]$ are generated using the algorithm given in Section 3.1. In this way a set of points $P_0, P_1, \ldots, P_l$ is obtained.

4. The points P_i, $i = 0, \ldots, l$ are mapped onto the B-spline curve $C = C(t)$ by using Eq (3.26). It is recommended that the obtained points be projected orthogonally on the control polygon, and then again onto the interval $[0, M]$ by using the transformation $K(P)$. In such a way a new set of points P_i', $i = 0, \ldots, l$ is transformed on the B-spline curve.

In the algorithm it is assumed, that the control polygon is close enough to the B-spline curve.

4. Generation of internal points

As in the proposed approach the advancing front technique (the Delaunay advancing front technique, Lo (1989)) is applied to the triangulation of the domain, the same approach is applied to the internal points generation. A majority of methods (cf Lo (1991), Morgan et al. (1994)) use the advancing front technique to both grid generation and triangulation. In this paper the idea of internal points generation and then the Delaunay triangulation is preferred. The advantages of this approach, in author's opinion, are as follows:

- A possibility of situating the points on the front in the way enabling the mesh density function requirements to be satisfied, which is impossible in the case of the advancing front approach with simultaneous grid generation and triangulation.

- In the case when a newly generated point is very close to the front an ill-conditioned triangle may appear, when advancing front technique performs triangulation while points are inserted (in Fig.5 both the meshes are obtained on that same grid points), what is avoided in the case of Delaunay triangulation performed after the grid generation.

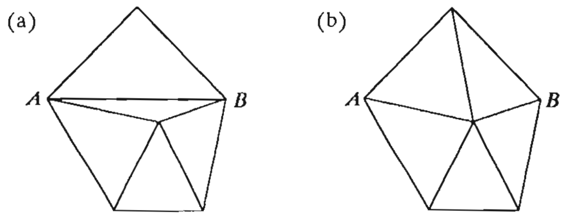

Fig. 5. (a) – Triangulation during points insertion, (b) – Delaunay triangulation after grid points are generated

One may consider the method using two runs of the advancing front procedure to be more time consuming. However there are two reasons, for which such an approach is more effective than the method with simultaneous point generation and triangulation.

The first approach is shorter because no triangles are formed. The second is the Delaunay triangulation on the given set of points. It is well known (cf Lo (1989)) that this is a very fast algorithm because there is no need to check the intersections with the so called *Delaunay part of the front*, which decreases as the number of triangles increases. It is rather difficult to decide, which method is faster, but the previous subsections indicate that the approach (in author's opinion) presented in this paper is superior.

It is assumed, that the mesh density function

$$\rho : \overline{\Omega} \longmapsto \mathbb{R} \tag{4.1}$$

is given. It should satisfy the condition

$$\rho(A) > 0 \qquad \forall A \in \overline{\Omega} \tag{4.2}$$

where $\overline{\Omega}$ denotes the closure of Ω. The algorithm starts with the front, which is a sequence of straight line segments positively oriented with respect to the domain. The set of such segments is obtained by points generation on the curves consituting the boundary of the domain. The novel idea we propose is to prescribe to every straight line segment of the front a value of 0 or 1 (what is interpreted as marked or not marked), at the beginning all the straight line segments will be marked as 1.

Starting with any straight line segment AB of the front a new point C is added if it lies on the left hand side of the segment AB and the sides of the triangle $\boldsymbol{ABC}$ satisfy the following conditions

$$|AC| = \rho(A) \qquad |BC| = \rho(B) \tag{4.3}$$

As a matter of fact the point C should satisfy the following conditions

$$\begin{aligned} \frac{\rho(A)+\rho(C)}{2} &= dist(A,C) \\ \frac{\rho(B)+\rho(C)}{2} &= dist(B,C) \end{aligned} \tag{4.4}$$

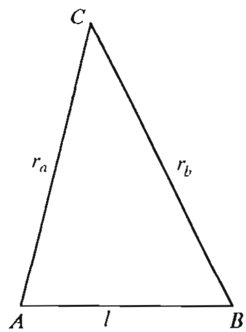

Fig. 6. Illustration of the conditions (4.6)

The point C satisfying Eqs (4.3) is an approximation of the point which should satisfy Eqs (4.4). In our first approach the point, that is obtained from the conditions (3.38), (4.1) is taken into account. In the further approach the point C will be treated as a good approximation of the point satisfying Eqs (4.3), which can be solved by the Newton-Raphson method. The presented approach works correctly, when the numbers $|AB|$, $\rho(A)$, $\rho(B)$ can create a triangle. Using the notation

$$r_a = \rho(A) \qquad r_b = \rho(B) \qquad l = |AB| \tag{4.5}$$

the numbers r_a, r_b, l form a triangle is if the following inequalities are satisfied

$$r_a + r_b > l \qquad r_a + l > r_b \qquad r_b + l > r_a \tag{4.6}$$

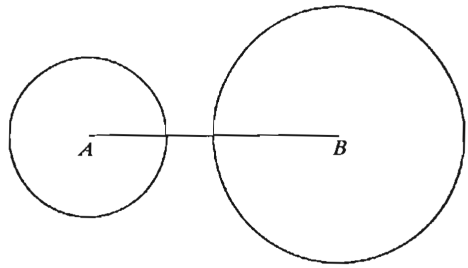

Fig. 7. Sample case of the condition $(4.6)_1$ not satisfied

If the condition $(4.6)_1$ (see Fig.7) is not satisfied (if r_a, r_b, l are positive both the conditions $(4.6)_{2,3}$ are automatically satisfied), then the point $\boldsymbol{C}$ is a midpoint of the straight line segment $\boldsymbol{AB}$ proportionally distributed according to the values r_a and r_b. It leads to the following formula

$$\boldsymbol{C} = \lambda \boldsymbol{A} + \mu \boldsymbol{B} \tag{4.7}$$

where

$$\lambda \geq 0 \qquad \mu \geq 0 \qquad \lambda + \mu = 1 \tag{4.8}$$

λ and μ are found from the proportions

$$\frac{\lambda}{l} = \frac{r_x}{r_x + r_y} \qquad \frac{\mu}{l} = \frac{r_y}{r_x + r_y} \tag{4.9}$$

The violation of Eq $(4.6)_2$ means, that $r_a + l \leq r_b$ (see Fig.8). In this case it is proposed to fix the point $\boldsymbol{C}$ on the line perpendicular to the straight line segment $\boldsymbol{AB}$ at the point $\boldsymbol{A}$ and lying on the left-hand side of $\boldsymbol{AB}$, and satisfying the proportionality

$$\frac{|CD|}{|CE|} = \frac{r_a}{r_b} \tag{4.10}$$

Introducing the following vectors

$$\boldsymbol{AB} = [a,b] \qquad \boldsymbol{v} = [-b,a] \tag{4.11}$$

and using the orthogonality of $\boldsymbol{v}$ and $\boldsymbol{AB}$ we obtain

$$\boldsymbol{C} = \boldsymbol{A} + \lambda \boldsymbol{v} \tag{4.12}$$

where

$$\lambda = \frac{|AC|}{|\boldsymbol{v}|} \tag{4.13}$$

As it is shown in Fig.8

$$|AC| = |AE| + |CE| \tag{4.14}$$

On the other hand, from equation (4.14) and Fig.8 we have

$$|CD| + |CE| = |AD| - |AE| \tag{4.15}$$

and

$$|AD| = \sqrt{r_b^2 - l^2} \qquad\qquad |AE| = r_a \tag{4.16}$$

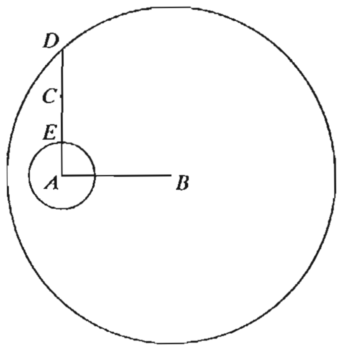

Fig. 8. Illustration of the violation of condition $(4.6)_2$

From the last two equations and Eqs (4.22) and (4.16) it is obtained

$$|CD| = \frac{|AD| - |AE|}{1 + \frac{r_b}{r_a}} \tag{4.17}$$

and eventually

$$\lambda = \sqrt{r_b^2 - l^2} - \frac{\sqrt{r_b^2 - l^2} - r_a}{1 + \frac{r_b}{r_a}} \tag{4.18}$$

In the case $r_b + l \leq r_a$ (see Fig.9) by the symmetry we obtain analogous formulas for C

$$C = B + \lambda v \tag{4.19}$$

where

$$\lambda = \sqrt{r_a^2 - l^2} - \frac{\sqrt{r_a^2 - l^2} - r_b}{1 + \frac{r_a}{r_b}} \tag{4.20}$$

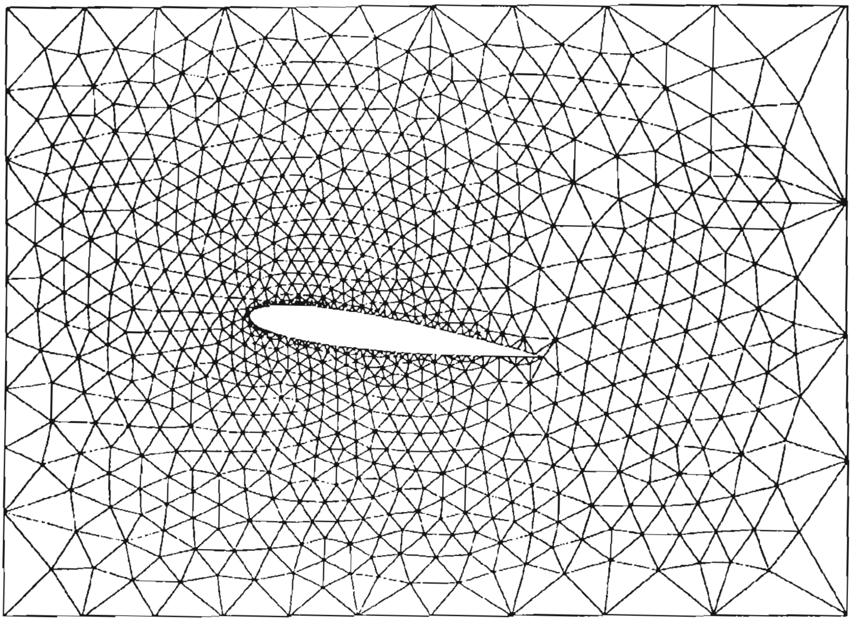

Fig. 9. NACA0012 airfoil, 1384 triangles and 736 nodes

4.1. The algorithm of internal points generation by the advancing front technique

As it was said in the previous subsection the advancing front method is used to the internal points generation and then the Delaunay-advancing front technique is applied to the triangulation.

Let $\boldsymbol{\Omega}$ be the considered domain, and

ρ be the mesh density function

$$\rho : \overline{\Omega} \longmapsto \mathbb{R} \qquad \rho > 0 \qquad \forall (x, y) \in \overline{\boldsymbol{\Omega}} \tag{4.21}$$

Let $\boldsymbol{\Gamma}$ be the generation front.

Then the algorithm of internal points generation can be summarized as follows

1. $\boldsymbol{\Gamma} \longleftarrow \partial \boldsymbol{\Omega}$, `(provided, that` $\partial \boldsymbol{\Omega}$ `is represented by a sequence of straight line segments determined by the points generated on it)`

```
2. IF (Γ ≠ ∅) THEN
   find the last straight line segment AB belonging to Γ
   ELSE
   finish the points generation
   ENDIF

3. IF (ρ(A) + ρ(B) < |AB|) THEN
   find the point C lying on the left-hand side of AB
   according to Eqs (4.3)
   ELSEIF (ρ(A) + |AB| < ρ(B)) THEN
   find the point C according to Eq (4.7)
   ELSEIF (ρ(B) + |AB|) < ρ(B)) THEN
   find the point C according to Eq (4.12)
   ELSE
   find the point C according to Eq (4.19)
   ENDIF

4. IF (∀ point P of the front dist(P, C) > (ρ(C)+ρ(P))/2) THEN
   IF (AC ∩ Γ ≠ ∅ and BC ∩ Γ ≠ ∅ THEN
   ·  add the point C to the set of internal points,
   ·  remove AB from the front,
   ·  add AC and CB to the front,
   ·  go to 2,
   ·  ELSE
   ·  mark the front segment AB as inactive,
   ·  go to 2,
   ·  ENDIF
   ELSE
   mark the front segment AB as inactive,
   go to 2
   ENDIF
```

The condition

$$dist(\boldsymbol{P},\boldsymbol{C}) > \frac{\rho(\boldsymbol{C}) + \rho(\boldsymbol{P})}{2} \tag{4.22}$$

is different from the condition presented by Lo (1991) i.e., for every segment $\boldsymbol{EF}$ of the front

$$dist(\boldsymbol{P},\boldsymbol{EF}) > \frac{\rho(\boldsymbol{C}) + \rho(\boldsymbol{P})}{2} \tag{4.23}$$

When the grid generation process is maintained together with the triangulation, then the violation of condition $(4.6)_1$ generates degenerated triangles. This will not happen in the case when the Delaunay triangulation is done after the grid generation process on the previously fixed set of points.

5. The algorithm of triangulation

The algorithm of triangulation is similar to the algorithm presented by Kucwaj (1992), (1993), Lo (1989). The only difference lies in the procedure of searching for the candidates for the given straight line segment of the front to form a triangle. In the previous case there was given one real value D defining the mesh density factor in the whole domain. The set of candidates was the set of the points lying in the interior part of the front or on the front and belonging to the appriopriate circle of the radius D. In the present algorithm this radius varies for different segments one straight line segment to another. It is equal to the length of the current straight line segment. Let us introduce the following notation:

$\partial\boldsymbol{\Omega}$ – boundary of the domain Ω
$\boldsymbol{\Gamma}$ – generation front
$\boldsymbol{\Gamma}_1$ – non-Delaunay part of the generation front
$\boldsymbol{\Gamma}_2$ – Delaunay part of the generation front.

Then the algorithm of triangulation can be written as follows:

1. $\boldsymbol{\Gamma}_1 \longleftarrow \partial\boldsymbol{\Omega}$

2. $\boldsymbol{\Gamma}_2 \longleftarrow \emptyset$

3. IF $(\Gamma_1 \neq \emptyset)$ THEN
   find the last segment AB belonging to Γ_1
   ELSE
   IF $(\Gamma_2 \neq \emptyset)$ THEN
   find the last segment belonging to Γ_2
   ELSE
   finish the triangulation
   ENDIF
   ENDIF

4. $\boldsymbol{R}_0 \longleftarrow$ the length of the segment AB

```
5. R ⟵ R_0

6. Specify the set S of all points belonging
   to Λ∪Γ_1∪Γ_2 lying in the interior of the circle defined by
   radius R and the centre in the midpoint of segment AB

7. Find the point C* ∈ Λ ∪ Γ_1 ∪ Γ_2, which satisfies
   AB × BC > 0, and C* ∈ C_ABC ∀C ∈ S

8. IF (there is no intersection with Γ_1) THEN
   form a triangle ABC* and add it to the list. Update Γ_1,
   Γ_2, Λ
   ELSE
   S ⟵ S − {C*}
   IF (S ≠ ∅) THEN
   go to 4
   ELSE
   R ⟵ R + R_0
   go to 3
   ENDIF
   ENDIF
```

6. The mesh density function in the interior of the domain

In various problems of mechanics it is necessary to define meshes which vary gradually from one piece of the boundary to another due to a nonuniform distribution of grid points on the boundaries. Following Lo (1991), the formula defining the mesh density function in the domain may be given as

$$\rho(x,y) = \frac{\sum\limits_{i=1}^{N} \frac{\rho_i}{d_i}}{\sum\limits_{i=1}^{N} \frac{1}{d_i}} \tag{6.1}$$

where

$$\begin{aligned} \rho_i &= |A_i\boldsymbol{B}_i| \qquad i = 1,\ldots,N \\ d_i &= |PA_i| \end{aligned} \tag{6.2}$$

and $\{\boldsymbol{A}_i\boldsymbol{B}_i;\ i = 1,\ldots,N\}$ is a set of boundary segments.

7. Numerical examples

In this section some numerical results are presented. In Fig.10 and Fig.11 the NACA012 airfoil is presented.

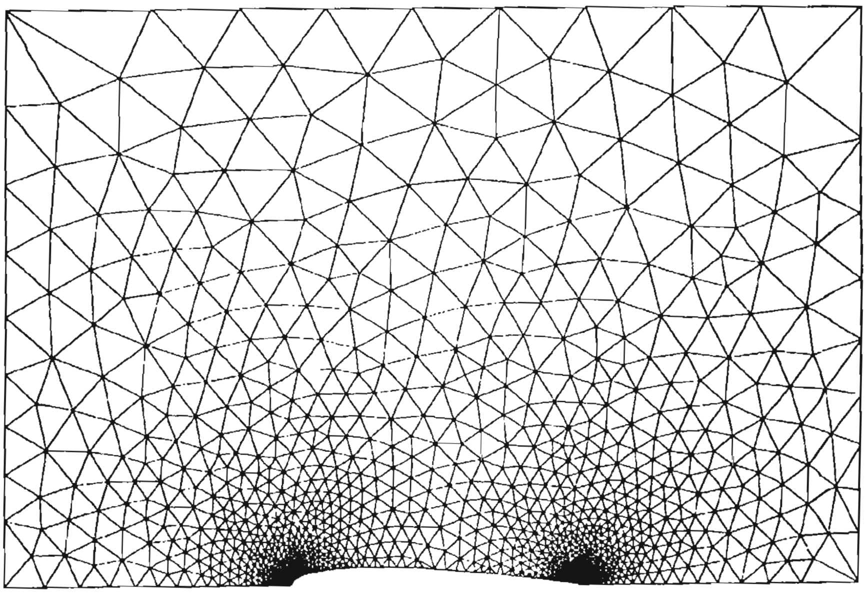

Fig. 10. A half of NACA0012 airfoil with 1678 triangles and 902 nodes

The mesh density function for the mesh shown in Fig.9 was assumed as

$$\rho(x,y) = \begin{cases} 0.4\sqrt{x^2+y^2}+0.2 & \text{if} \quad (x,y) \in EL \\ 0.05\sqrt{(x+0.01)^2+y^2} & \text{if} \quad (x,y) \in IL \\ Eq(6.1) & \text{if} \quad (x,y) \in D \end{cases}$$

where EL - external loop, ILS - internal loop and D - domain.

Thus the mesh density function for the mesh shown in Fig.10 is of the form

$$\rho(x,y) = \begin{cases} 0.1\sqrt{x^2+y^2}+0.007 & \text{if} \quad x < 0.5 \\ 0.1\sqrt{(x-1.008930411365)^2+y^2}+0.006 & \text{otherwise} \end{cases}$$

The mesh density function for the meshes in Fig.11 was taken as

$$\rho(x,y) = \begin{cases} 0.2\sqrt{x^2+y^2}+1.4 & \text{if} \quad x < 50 \\ 0.2\sqrt{(x-73)^2+(y+13)^2}+1.1 & \text{if} \quad 50 \le x < 78 \\ 0.2\sqrt{(x-88)^2+(y+20)^2}+1.1 & \text{if} \quad 78 \le x < 100 \\ 0.2\sqrt{(x-111)^2+(y+36)^2}+1.1 & \text{otherwise} \end{cases}$$

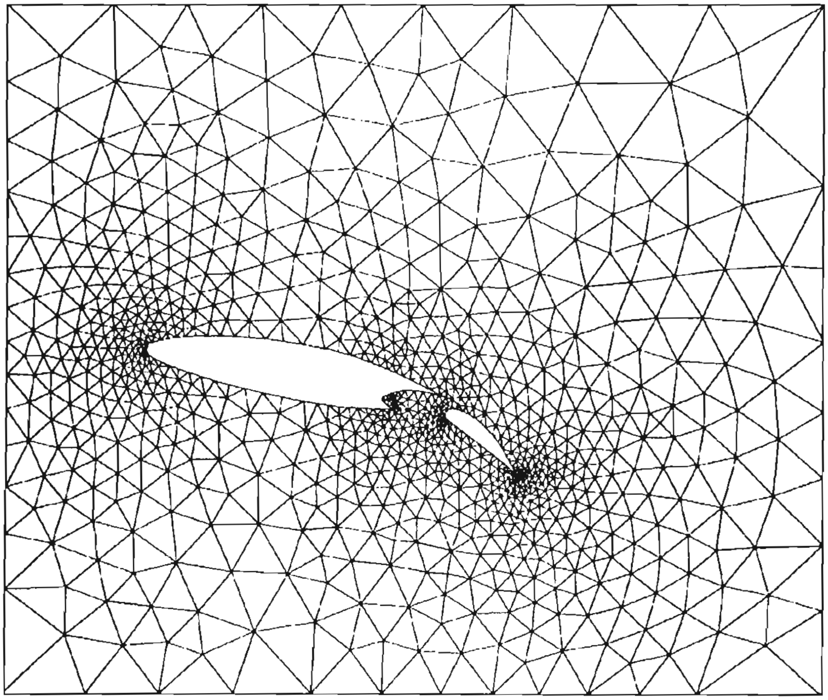

Fig. 11. NACA0066, 1740 triangles and 939 nodes

8. Final remarks

The paper presents the algorithm of points generation with a given mesh density function in 2D multiconnected domains. The algorithm based on the advancing front approach is worked out for both the grid generation and triangulation. The characteristic feature of the algorithm is that triangulation follows grid generation. It allows one to perform the Delaunay triangulation on the obtained grid points.

The computer code was developed and tested, and a variety of numerical tests were run. The results prove effectiveness and usefullnes of the presented method and the computer code.

The code is written in FORTRAN and the examples were run on a workstation. The generated meshes were used for solving the Navier-Stokes equations. The results show effectiveness of the proposed approach being considered as an initial step for adaptation.

References

1. BAEHMAN P.L., WITTCHEN S.L., SHEPARD M.S., GRICE K.R., YERRY M.A., 1983, Robust, Geometrically Based, Automatic Two-Dimensional Mesh Generation, *IJNME,* **24**

2. BARTELS R.M., BEATTY J.C., BARSKY B.A., 1989, *An Introduction to Splines and Their Use in Computer Graphics and Geometric Modelling,* Prentice-Hall

3. BLACKER T., STEPHENSON M.B., 1991, Parsing a New Approach to Automated Quadrilateral Mesh Generation, *IJNME,* **32**, 811-847

4. BRACKBILL J.U., SALTZMAN J.S., 1982, Adaptive Zoning for Singular Problems in Two Dimension, *Journal of Computational Physics,* **46**, 242-368

5. DEMKOWICZ L., ODEN J.T., RACHOWICZ W., HARDY O., 1989, Toward a Universal h-p Adaptive Finite Element Strategy, Part 1. Constrained Approximation and Data structure, *Comput. Meth. Appl. Mech. Engrg.,* **77**, 79-112

6. DEMKOWICZ L., RACHOWICZ W., BANAŚ K., KUCWAJ J., 1992, 2-D Adaptive Package, Cracow University of Technology, Report No. 4/1992, Section of Applied Mathematics

7. BAJER A., DEMKOWICZ L., 1993, Implicit Parametrization in Geometrical Modelling, *The XI Polish Conference on Computer Methods in Mechanics,* Kielce-Cedzyna, Poland, 61-68

8. DIERCKX P., 1995, *Curve and Surface Fitting with Splines,* Oxford Science Publications, Clarendon Press, Oxford

9. DERVIEUX A., VAN LEER B., PERIAUX J., RIZZI A. (EDIT.), Numerical Simualtion of Compressible Euler Flows, *Notes on Numerical Fluid Mechanics,* **26**, Viewweg, Braunschweig/Wiesbaden

10. FAUX I.D., PRATT M.J., 1979, *Computational Geometry for Design and Manufacture,* Ellis Harwood Limited - Publishers Chichester, John Wiley & Sons, New York-Chichester-Brisbane-Toronto

11. KNUPP P.M., 1991, The Direct Variational Grid Generation Method Extended to Curves, *Applied Mathematics and Computation,* **65**, 65-78

12. KUCWAJ J., 1992, Unstructured Grid Generation over 2-D Domains, report No 3/1992, Cracow University of Technology, Applied Mathematics Section

13. KUCWAJ J., 1993, Fully Automatic Generation of Unstructured 2-D Triangular Meshes Using Lo's Advancing Front Technique, *The XI Polish Conference on Computer Methods in Mechanics,* Kielce-Cedzyna, Poland, vol I, 473-484

14. KUCWAJ J., 1995, Automatic Grid Generation over Plain Domains and Surfaces, *Zeszyty Naukowe PK,* **5**, 61-82

15. LO S.H., 1985, A New Mesh Generation Scheme for Arbitrary Planar Domains, *Int. J. Numer. Meth. in Engng.,* **21**, 1403-1426

16. LO S.H., 1989, Delaunay Triangulation of Non-Convex Planar Domains, *Int. J. Numer. Meth. in Engn.*, **28**, 2695-2707

17. LO S.H., 1991, Automatic Mesh Generation and Adaptation by Using Contours, *Int. J. Numer. Meth. in Engn.*, **31**, 689-707

18. LO S.H., 1992, Generation of High-quality Gradation Finite Element Mesh, *Engineering Fracture Mechanics*, **41**, 2

19. MORGAN K., HASSAN O., PROBERT E.J., 1994, Unstructured Mesh Generation for Viscous High Speed Flows, *Proceedings of the IV-th Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, 779-793, Swansea, Wales

20. SALTZMAN J., 1985, Variational Methods for Generating Meshes on Surfaces in Three Dimensions, *Journal of Computational Physics*, **63**, 1-19

21. STEINBERG S., ROACHE P.J., 1986, Variational Grid Generation, *Numerical Methods for Partial Differential Equations*, **2**, 71-96

22. ZIENKIEWICZ O.C., ZHU J.Z., 1991, Adaptavity and Mesh Generation, *International Journal for Numerical Methods in Engineering*, **32**, 783-810

23. WARSI Z.U.A., 1986, Numerical Grid Generation in Arbitrary Surfaces Throught a Secong-Order Differential-Geometric Model, *Journal of Computational Physics*, **64**, 82-96

Modelowanie i triangularyzacja płaskich obszarów o złożonej geometrii i pewne zastosowania w mechanice płynów

Streszczenie

Praca przedstawia modelowanie geometryczne obszarów płaskich oraz ich triangularyzację. W celu uwzględnienia szerokiej klasy obszarów płaskich zostały odpowiednio zdefiniowane cechy topologiczne obszaru oraz ich wzajemne relacje. Krzywe reprezentowane są poprzez wykorzystanie funkcji typu B-spline. Opracowany został algorytm generowania punktów w obszarze z zadaną gęstością. Punkty wewnętrzne generowane są metodą postępującego frontu. Praca ilustrowana jest przykładami numerycznymi.

Manuscript received October 4, 1996; accepted for print November 28, 1996