Triangulation of two-dimensional multiply connected domain

Pyataev S.F.

Table of Contents

Triangulation of two-dimensional multiply connected domain with concentration and rarefection of grid

Pyataev S.F.

Triangulation of two-dimensional multiply connected domain with concentration and rarefection of grid

Pyataev S.F.

Introduction

Wide use of the finite element method for solution of various kinds of problems raises the requirements to the level of automation of domain fragmentation. There are algorithms and programs allowing to construct uniform grids on simply connected domains [1-4]. The advantages of the algorithms are their universality with respect to the shape of boundary of the domain as well as the possibility to triangulate simply and multiply connected domains with concentration and rarefaction of grid; the latter is attained by division of the initial domain into a number of simply connected domains and fragmentation or consolidation of one-dimensional final elements on boundaries of some subdomains. An obvious disadvantage of the triangulation algorithms for simple connected domains when applying to multiply connected domains or concentration of the grid is a great amount of handwork: division of the domain into subdomains, fragmentation of each boundary, input of information, etc. The idea of triangulation algorithm for multiply connected domain with concentration of grid described in [5]. It avoids the necessity of division of the domain into a collection of subdomains and at the same time retains the disadvantage connected with hand fragmentation of each contour (with the exception of the simplest elements of the contours: linear regions and arcs). Except that, indistinctness of the introduced in the paper requirements with respect to the properties of a new node being constructed (proximity to previously constructed node, proximity to one-dimensional finite element, simultaneous proximity to the node and the element, etc.) makes the programming considerably more difficult and forces the user either do develop conditions for a new node being constructed or quite reject the algorithm.

For the purpose of constructing a completely automated process of triangulation of arbitrary two-dimensional multiply connected domains, an algorithm of fragmentation for arbitrary piecewise smooth closed boundary contours is developed in the present paper.

In the third section, on the basis of the scheme proposed in [4, 5] and representing a consecutive filling of domain with triangular elements, the process of triangulation of the domain is constructed. The process of filling starts from the boundary which is preliminarily fragmented into one-dimensional finite elements. In the course of construction of triangular elements the boundary of the domain being not yet triangulated (following [4], we will call it *current grid boundary*, CGB) represents a number of continuous closed piecewise curves with possible self-intersections. A detailed description of construction of new nodes and elements is given in this section; in particular, the criteria of selection of previously constructed node (or construction of a new one) are given. And as a consequence, the criteria of construction of an element are given in the case when some regions of CGB close in. In the course of fragmentation of the boundary of domain and its triangulation a function of steps is used which adjusts the sizes of one-dimensional and triangular finite elements according to their position in the domain. Any positive function can appear as the function of steps; the principles of its construction are given in section 2.

Presentation of both the algorithms is given in a form convenient for programming. In appendices some auxiliary procedures are given, which are necessary for the work of the program and which, apparently, should be designed as subroutines.

1 Some recommendations on choice of the function of steps

In many problems one can beforehand make certain assumptions on subdomains of large gradients of the sought functions, appearing, as a rule, in the locations of concentrators of different kind, on lines of jump of coefficients of the problem, due to singularities in some points of boundary conditions, in the points of sharp change in the character of the boundary, etc. For concentration of the grid in such subdomains a necessity appears to construct finite elements with a step less than the basic step h_0 used for larger part of the domain Ω. Since during triangulation the triangular elements are constructed successively, their size can be determined according to their locations, by means of certain positive function of steps $h(x, y)$ with parameters responsible for the "centers" and "sizes" of the subdomains of concentration. These parameters should be chosen so that on leaving the subdomain the sizes of triangle elements would be of the order h_0.

Apparently, exact recommendations on construction of the function of steps cannot be given owing to the absence of exact definition of the notion of the domain of concentration. Therefore let restrict ourselves to formulation of general principles of construction of these functions, extending descriptive ideas of one-dimensional case to two-dimensional one. Let in one-dimensional case a qualitative graph of the function of steps is represented on Fig. 1

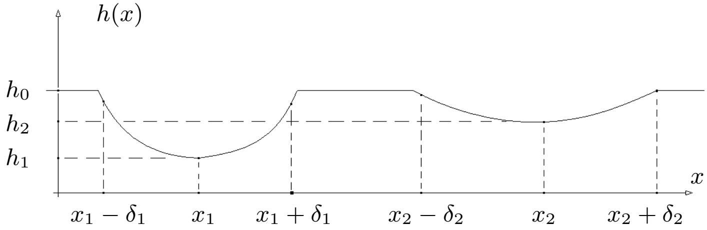


Fig. 1: An example of graph of the function of steps.

It is convenient to represent the function of steps in the form of a sum

$$h(x) = h_0 + \sum_{i=1}^{n} (h_i - h_0) f_i(x, x_i, \delta_i) ,$$

where δ_i is "characteristic size" of the $i-$th domain of concentration; x_i is center of domain of concentration; f_i is equal to 1 in the point x_i is of zero order outside its domain of concentration. In this case an approximate graph of the function f_i can be represented like on Fig. 2

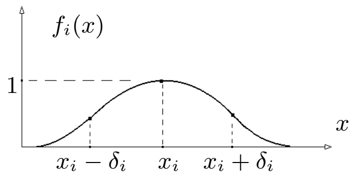


Fig. 2: An approximate graph of the function $f_i(x)$.

So, one can take $f_i(x)$ as one of the variants:

$$\mathrm{ch}^{-n_i}\left(\frac{x - x_i}{\delta_i}\right) ; \quad \left(1 + \left|\frac{x - x_i}{\delta_i}\right|^{n_i}\right)^{-1} ; \quad \exp\left\{-\left|\frac{x - x_i}{\delta_i}\right|^{n_i}\right\} ;$$

or

$$f_i(x) = \begin{cases} 1 - \left| \dfrac{x - x_i}{\delta_i} \right|^{n_i}, & \text{if } x \in (x_i - \delta_i, x_i + \delta_i)\,, \\ 0, & \text{if } x \notin (x_i - \delta_i, x_i + \delta_i). \end{cases}$$

Here the degree n_i is positive and characterizes the value of gradient of the function f_i.

However, an expansion of these variants over two-dimensional case by direct introduction of the second coordinate would not embrace the cases when the domain of concentration is stretched not along one of the axes but along some direction determined by a vector $(\cos\alpha, \sin\alpha)$. To eliminate this shortcoming, equip every such domain with a local coordinate system, in which the direction of stretching of the domain coincides with one of the new axes. Evidently, this transformation of coordinate system should take into account transfer and rotation, i.e.,

$$\tilde{x_i} = (x - x_i)\cos\alpha_i + (y - y_i)\sin\alpha_i,$$

$$\tilde{y_i} = -(x - x_i)\sin\alpha_i + (y - y_i)\cos\alpha_i,$$

where $(x_i,\ y_i)$ are coordinates of the center of i-th concentration; α_i is angle of rotation of the axes of i-th concentration.

Then the functions of steps in two-dimensional case by analogy with one-dimensional case can be taken in the form

$$h(x, y) = h_0 + \sum_{i=1}^{n} (h_i - h_0) f_i(x, y, x_i, y_i, \alpha_i, \beta_i, \delta_i),$$

where β_i, δ_i are "characteristic sizes" of the domain of concentration, and f_i can be presented, for instance, as

$$2\left(\mathrm{ch}^{n_i}\left(\frac{\tilde{x_i}}{\beta_i}\right) + \mathrm{ch}^{m_i}\left(\frac{\tilde{y_i}}{\delta_i}\right)\right)^{-1}; \qquad \left(1 + \left(\frac{\tilde{x_i}}{\beta_i}\right)^{n_i} + \left(\frac{\tilde{y_i}}{\delta_i}\right)^{m_i}\right)^{-1};$$

$$\exp\left\{-\left(\frac{\tilde{x_i}}{\beta_i}\right)^{n_i} - \left(\frac{\tilde{y_i}}{\delta_i}\right)^{m_i}\right\}; \qquad \begin{cases} 1 - \left|\dfrac{\tilde{x_i}}{\beta_i}\right|^{n_i} \cdot \left|\dfrac{\tilde{y_i}}{\delta_i}\right|^{m_i}, & \text{if } (x, y) \in \Omega_i, \\ 0\,, & \text{if } (x, y) \notin \Omega_i. \end{cases}$$

where $\Omega_i = (x_i - \beta_i, x_i + \beta_i) \times (y_i - \delta_i, y_i + \delta_i)$, the degrees n_i, m_i as before are positive and characterize the gradients of the functions f_i along the direction $(\cos\alpha_i\,, \sin\alpha_i)$ and orthogonal to it.

2 Fragmentation of the boundary of multiply connected domain

Let the boundary of a multiply connected domain be formed by N piecewise smooth closed contours given in some Cartesian coordinate system $0xy$ in parametric form.

Consider a piecewise smooth contour Γ (its index is omitted) formed by L smooth curves $\gamma_n, \quad n = 1, \ldots, L$, whose parametric equations are

$$(x(t),\ y(t)) \equiv \boldsymbol{x}(t) = \boldsymbol{x}_n(t), \quad t_n^- \le t \le t_n^+, \tag{2.1}$$

where t_n^-, t_n^+ are the limits of variation of the parameter t for γ_n. From the conditions of continuity and closeness of the contour Γ it follows that

$$\boldsymbol{x}_n(t_n^+) = \boldsymbol{x}_{n+1}(t_{n+1}^-), \quad n = 1, \ldots, L-1,$$

$$\boldsymbol{x}_1(t_1^-) = \boldsymbol{x}_L(t_L^+).$$

Parametrization (2.1) must be such that for the inner contour Γ the direction of encircling under increase of the parameter t would be clockwise, and for the external contour would be counterclockwise.

Fragmentation of Γ is performed successively, starting from the first smooth curve $\gamma_1 : \ \boldsymbol{x} = \boldsymbol{x}_1(t)$. The first node on Γ is $\boldsymbol{y}_1 = \boldsymbol{x}_1(t_1^-)$. Assume that $l-1$ first curves of the contour Γ are fragmented already, the last constructed node on these curves is $\boldsymbol{y}_{n_{l-1}} = \boldsymbol{x}_{l-1}(t_{l-1}^+) = \boldsymbol{x}_l(t_l)$ and a part of the curve γ_l is fragmented, with the last node $\boldsymbol{y}_{n_{l-1}+k} = \boldsymbol{x}_l(t_k^l)$ where t_k^l is the value of the parameter t for the last node, $t_k^l \in [t_l^-, t_l^+)$. Then we construct next node of the curve γ_l by 3 steps.

Step 1.

Denote by $s_l(t_k^l, t)$ the length of a part of the curve γ_l, corresponding to the values $t_k^l,\ t:$

$$s_l(t_k^l, t) = \int_{t_k^l}^{t} \mid \dot{\boldsymbol{x}}(t) \mid dt, \quad t \in [t_k^l, t_l^+],$$

where $\dot{\boldsymbol{x}}(t)$ is derivative of $\boldsymbol{x}(t)$ with respect to t.

A new node $\boldsymbol{y}_{n_{l-1}+k+1}$ is constructed as follows: the value of the function of steps $h(x, y)$ is calculated in the last constructed node $\boldsymbol{y}_{n_{l-1}+k}$ and the solution $\tilde{t}_{k+1}^l$ of equation

$$s_l(t_k^l, t) = h(\boldsymbol{y}_{n_{l-1}+k}), \quad t \in [t_k^l, t_l^+], \tag{2.2}$$

is looked for. Suppose that solution of this equation exists (the contrary is considered in step 2). In this case it is unique due to positiveness of $\mid \dot{\boldsymbol{x}}_l(t) \mid$

(may be, with exception of finite number of points which do not influence uniqueness). According to the obtained value $\tilde{t}^l_{k+1}$ we calculate $\boldsymbol{x}_l(\tilde{t}^l_{k+1})$ and look for the solution t^l_{k+1} of the equation

$$s_l(t^l_k, t) = \frac{1}{2}[h(\boldsymbol{y}_{n_{l-1}+k}) + h(\boldsymbol{x}_l(\tilde{t}^l_{k+1}))]. \tag{2.3}$$

Suppose that this equation has a solution as well (the contrary is considered in step 3). Consider the inequality

$$\left| \frac{d - s_l(t^l_k, t^l_{k+1})}{s_l(t^l_k, t^l_{k+1})} \right| \leq \varepsilon, \quad d = \mid \boldsymbol{y}_{n_{l-1}+k} - \boldsymbol{x}_l(t^l_{k+1}) \mid, \tag{2.4}$$

the left-hand side of which is the relative difference between the length of arc and the length d of segment corresponding to the arc. The inequality characterises deviation of the arc from segment of stright line. Value of the parameter ε is specified by the user (for instance, $\varepsilon = 0.01$). If the inequality (2.4) is satisfied, then we declare the point $\boldsymbol{x}_l(t^l_{k+1})$ as a new node $\boldsymbol{y}_{n_{l-1}+k+1}$ and turn to construction of the next node. The declaration of the constructed point as a new node is substantiated by the fact that due to validity of equation (2.4) the one-dimensional finite element $[\boldsymbol{y}_{n_{l-1}+k}, \boldsymbol{y}_{n_{l-1}+k+1}]$ approximates the corresponding arc of curve good enough, and its length d under $h(x, y)$ smooth enough correlates with the average value of the function of steps over the ends of this element (see right-hand side of equation (2.3)). Otherwise, if the inequality (2.4) is not true, we successively decrease the right part of equation (2.3) by certain value (for example, by one tenth of the right-hand side) till the inequality (2.4) would be true. This situation appears when the length d of one-dimensional element calculated in accordance with the function of steps is "large" enough for acceptable approximation by this element of the arc which corresponds to it. Therefore successive decrease of this length is performed down to the value required by inequality (2.4). After that, we turn to construction of the following node $\boldsymbol{y}_{n_{l-1}+k+2}$.

Completion of the procedure of construction of new nodes on l-th curve of the contour Γ (and, respectively, on the whole contour Γ) is connected with the absence of solution of equation (2.2) and is described in step 2.

Step 2.

Now, consider the case when the equation (2.2) does not have solution, i.e.,

$$s_l(t^l_k, t^+_l) < h(\boldsymbol{y}_{n_{l-1}+k}).$$

This means that the last constructed node $\boldsymbol{y}_{n_{l-1}+k}$ is "close" to $\boldsymbol{x}_l(t^+_l)$, and construction of a new node by means of the function $h(x, y)$ is impossible.

Denote by δ_l the length of the remainder γ_l, and by d_k denote the length of the last constructed element:

$$\delta_l = s_l(t_k^l, t_l^+), \quad d_k = \mid \boldsymbol{y}_{n_{l-1}+k-1} - \boldsymbol{y}_{n_{l-1}+k} \mid .$$

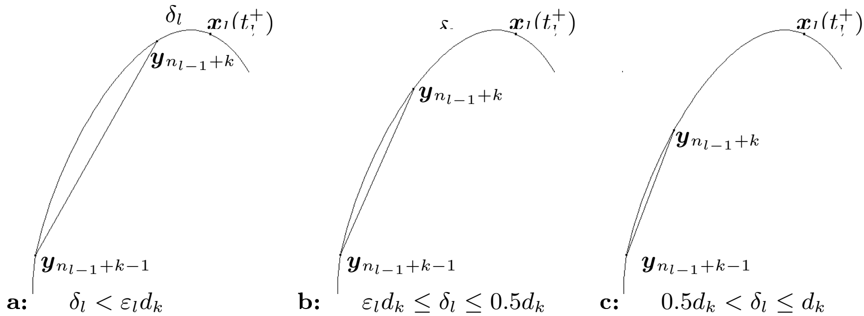


Fig. 3: All possible situations when $\delta_l \leq d_k$.

In Fig. 3 the situations are shown when $\delta_l \leq d_k$ and displacement of the last constructed node takes place into the last point $\boldsymbol{x}_l(t_l^+)$ of the curve γ_l. In Fig. 3a, or a redistribution of δ_l occurs over all the previous one-dimensional elements approximating γ_l proportionally to the lengths (Fig. 3b). In Fig. 3c a new one-dimensional element with the length d_k is constructed, and new residual $\delta_l - d_k$ is introduced and redistributed as above, over all the elements proportionally to their lengths.

Thus, if the residual δ_l satisfies the inequality

$$\delta_l < \varepsilon_1 d_k \tag{2.5}$$

where ε_1 is small enough, for instance, 0.1, then we displace the last constructed node $\boldsymbol{y}_{n_{l-1}+k}$ into the last point of γ_l :

$$\boldsymbol{y}_{n_{l-1}+k} = \boldsymbol{x}_l(t_l^+)$$

and turn to construction of nodes on the next curve γ_{l+1}. If (2.5) is not valid, we consider the inequality

$$\delta_l < 0.5 d_k. \tag{2.6}$$

If (2.6) is true, then the number of nodes on γ_l remains the same, and the residual δ_l is redistributed over all arcs constructed on γ_l proportionally to their lengths. Denoting by $\tilde{s}_l$ the length of that part of the curve γ_l which was passed when constructing the nodes:

$$\tilde{s}_l = \sum_{i=0}^{k-1} s_{i,i+1}^l, \quad s_{i,i+1}^l \equiv s_l(t_i^l, t_{i+1}^l), \quad t_0^l \equiv t_l^-. \tag{2.7}$$

Then the lengths of new arcs $s^{*l}_{i,i+1}$ are determined through the lengths of previous arcs $s^{l}_{i,i+1}$ according to the formulas

$$s^{*l}_{i,i+1} = s^{l}_{i,i+1}(1 + \delta_l/\tilde{s}_l), \quad i = 0, \dots, k-1. \tag{2.8}$$

Successively solving the equations

$$s_l(t^{*l}_i, t^{*l}_{i+1}) = s^{*l}_{i,i+1}, \quad t^{*l}_0 \equiv t^-_l, \quad i = 0, \dots, k-1, \tag{2.9}$$

we obtain new values t^{*l}_n and as well as new nodes on γ_l :

$$\begin{aligned} \boldsymbol{y}^*_{n_{l-1}+i} &= \boldsymbol{x}_l(t^{*l}_i), \quad i = 0, \dots, k-1, \\ \boldsymbol{y}^*_{n_{l-1}+k} &= \boldsymbol{x}_l(t^+_l). \end{aligned} \tag{2.10}$$

After that we turn to construction of nodes on the next curve γ_{l+1}.

If the residual δ_l does not satisfy the condition (2.6), then consider a new inequality

$$\delta_l \le d_k. \tag{2.11}$$

If this inequality is true, i.e., the length of the remainder part of γ_l is less than the length of the last constructed one-dimensional finite element but exceeds its half-length due to violation of (2.6), then the number of nodes on γ_l is increased by one, new residial is introduced

$$\delta_l = s_l - \tilde{s}_l$$

where s_l is lenfth of γ_l, and

$$\tilde{s}_l = \sum_{i=0}^{k-1} s^l_{i,i+1} + s^l_{k-1,k},$$

Then we come to (2.8)-(2.10) with new $\delta_l, \tilde{s}_l$ and with addition of one more arc, the length $s^l_{k,k+1}$ of which is equal to the length of the last constructed arc $s^l_{k-1,k}$. At that, it is necessary to increase the value of k by one in (2.8)-(2.10).

In the case when (2.11) is not satisfied, we consider the equation

$$s_l(t^l_k, t) = d_k, \quad t \in [t^l_k, t^+_l], \tag{2.12}$$

which has a solution due to inequalities

$$s_l(t^l_k, t^+_l) \equiv \delta_l > d_k,$$

$$s_l(t^l_k, t^l_k) = 0 < d_k.$$

The value t^l_{k+1} obtained from (2.12) is tested for realization of the inequality (2.4) and further in accordance with the algorithm, with the difference that if (2.4) is not satisfied then not the right-hand side of (2.3) is successively decreased, but the right-hand side of (2.12).

Step 3.

Consider the case when the equation (2.3) have no solution. Then after obtaining $\tilde{t}^l_{k+1}$ from equation (2.2) we come to the inequality (2.4), in which t^l_{k+1} is substituted by $\tilde{t}^l_{k+1}$. If the inequality is true, we declare the point $\boldsymbol{x}_l(\tilde{t}^l_{k+1})$ as a new node $\boldsymbol{y}_{n_{l-1}+k+1}$ and turn to construction of new node on γ_l. Otherwise successively decrease the right-hand side of the equation (2.2) till (2.4) is satisfied, after that turn to construction of the next node.

The described algorithm allows to decompose the boundary of domain into one-dimensional finite elements (further called units), each of them being specified by a pair of integers n_1 and n_2 – numbers of its nodes – and their coordinates. The information on successive order of the units can be stored in two one-dimensional arrays K and M.

1) $k_i = K(i)$ is the number of units on the i-th contour of CGB, $i = 1, \dots, N$.

2) $m_j = M(j)$, $m_{j+1} = M(j+1)$ are the numbers of the first and second nodes of j-th unit, respectively, if $j \neq \sum_{i=1}^{n} k_i$ for all $n = 1, \dots, N$. Otherwise, i.e., there exists such n_* that $j = \sum_{i=1}^{n_*} k_i$, then the number of the first node of such a unit is $M(j)$ as before, and the number of the second node is $M(\sum_{i=1}^{n_*-1} k_i + 1)$.

It is necessary to stress that the length of the array K changes in the process of triangulation, what is connected with change of the number of connectedness of the domain being not triangulated yet. The length of M also is not fixed, since either M is supplemented with new units, or the exhausted units from M are removed (the units which are not included in CGB on the next stage of construction of element).

3 Triangulation of a domain

It was noted above that the triangulation algorithm is based on successive filling of the domain with triangular elements. When filling the domain with the elements, CGB changes and in general case represents a number of closed broken contours. The number of connectivity of the domain being triangulated and the number of units of CGB change and can exceed the initial quantities. Therefore in the program one should watch that the length of the arrays K and M (see section 3) would not exceeded the given one.

Under coming together of different parts of CGB or in the domains of sharp changes of the function $h(x, y)$ the the triangular elements being constructed can be of an elongated form, and that can result in considerable errors when using this grid in finite element method. In order to avoid this defect, after construction of the grid an improvement is made which little changes compact triangles and significantly changes the elongated ones. The improvement of the grid is performed by means of the relations

$$\boldsymbol{x}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{x}_{k_i}$$

where $\boldsymbol{x}_k$ is the node being corrected; n_k is the number of nodes surrounding the node $\boldsymbol{x}_k$; $\boldsymbol{x}_{k_i}$ are the surrounding nodes. The number n_k and nodes $\boldsymbol{x}_{k_i}$ are determined by means of the triangles possessing the common vertex $\boldsymbol{x}_k$.

The triangulation algorithm consists in the following.

Step 1.

Find an unit z_{min} of CGB which has the minimal length l_{min} and the nodes $\boldsymbol{x}^1_{min}, \boldsymbol{x}^2_{min}$. Denote by z^-_{min}, z^+_{min} the units preceeding and following z_{min}, respectively. Choose from z^-_{min}, z^+_{min} an unit z^*_{min} which forms with z_{min} the minimal angle β_{min} ($\beta_{min} = \min(\beta_1, \beta_2)$; the angles β_1 and β_2 are measured counterclockwise from z_{min} to z^-_{min} and from z^+_{min} to z_{min}, respectively). Denote the nodes of the choosen pair of nodes (they are either z^-_{min}, z_{min} or z_{min}, z^+_{min}) by $\boldsymbol{x}^{min}_1, \boldsymbol{x}^{min}_2, \boldsymbol{x}^{min}_3$. If $\beta_{min} \leq 80^o$ (otherwise we come to step 4), then go to step 2.

Step 2.

Make a test of getting into the triangle $\Delta(z_{min}, z^*_{min})$ (see Appendix 1) of the nodes of CGB, with exception of the nodes forming z_{min}, z^*_{min}. If there are no such nodes, come to step 3, otherwise from all the nodes got into $\Delta(z_{min}, z^*_{min})$ choose a node $\boldsymbol{y}_*$ closest to z_{min} (see Appendix 2) and go to step 12.

Step 3.

Consider a circle with radius $|\boldsymbol{x}^{min}_3 - \boldsymbol{x}^{min}_1|/2$ and the center $\boldsymbol{x}_c$ which is the midpoint of the third side z in $\Delta(z_{min}, z^*_{min})$, $\quad \boldsymbol{x}_c = (\boldsymbol{x}^{min}_1 + \boldsymbol{x}^{min}_3)/2$. If nodes of CGB do not get into the half of the circle external with respect to $\Delta(z_{min}, z^*_{min})$, then come to step 13. Otherwise from all the nodes choose the closest to z node $\boldsymbol{x}_m$ and divide the quadrangle $\Box(z_{min}, z^*_{min}, \boldsymbol{x}_m)$ so that the minimal angle of the resulting triangles would be maximal (the choice should be done from two variants of division of the quadrangle into two triangles). The consideration of the quadrangle is necessary in order to avoid constructing elongated triangle with the vertices $\boldsymbol{x}^{min}_1, \boldsymbol{x}^{min}_3, \boldsymbol{x}_m$, since

$\boldsymbol{x}_m$ can be located close enough to z. Further, declare both the obtained triangles as elements, remove z_{min}, z^*_{min} from CGB, determine connectivity of the domain, add two new units $[\boldsymbol{x}_1^{min}, \boldsymbol{x}_m], [\boldsymbol{x}_m, \boldsymbol{x}_3^{min}]$ (see Appendix 8) and go to step 1.

Step 4.
Construct the point

$$\boldsymbol{x}_* = \boldsymbol{x}^c_{min} + h_{cp}\boldsymbol{n}, \quad \boldsymbol{x}^c_{min} = \frac{1}{2}(\boldsymbol{x}^1_{min} + \boldsymbol{x}^2_{min}). \tag{3.1}$$

Here $\boldsymbol{n}$ is the normal to z_{min} directed inside the domain:

$$(\boldsymbol{x}^2_{min} - \boldsymbol{x}^1_{min}) \times \boldsymbol{n} = l_{min}\boldsymbol{e}_3, \quad \boldsymbol{e}_3 = (0, 0, 1);$$

h_{cp} is average value of the function of steps over the vertices of equilateral triangle constructed on the basis z_{min} :

$$h_{cp} = \frac{1}{3}\sum_{i=1}^{3} h(\boldsymbol{\xi}_i), \quad \boldsymbol{\xi}_j = \boldsymbol{x}^j_{min}, \quad j = 1, 2; \quad \boldsymbol{\xi}_3 = \boldsymbol{x}^c_{min} + \frac{\sqrt{3}}{2} l_{min}\boldsymbol{n}.$$

Under certain conditions described below, the point $\boldsymbol{x}_*$ will be a new node, and the triangle $\Delta(z_{min}, \boldsymbol{x}_*)$ will be a new element.

On the basis of the unit z_{min} construct a rectangle Ω, one of which sides is z_{min} and the another is directed normally and its length equals $2H$. Here H is the altitude in $\Delta(z_{min}, \boldsymbol{x}_*)$ dropped on z_{min} :

$$2H = \sqrt{4|\boldsymbol{x}_* - \boldsymbol{x}^1_{min}|^2 - l^2_{min}}.$$

By means of the control domain Ω, let ascertain the criterions of proximity of the new node $\boldsymbol{x}_*$ to the previously constructed nodes and units. If in certain sense $\boldsymbol{x}_*$ is close to nodes or units, then we refuse to construct the new node and choose the best node from the close ones for construction of the new element.

Let define two sets M_0 and M_1 as follows.

M_0 is the set of numbers of nodes of CGB, which got into Ω, with exception of the numbers of nodes $\boldsymbol{x}^1_{min}$ and $\boldsymbol{x}^2_{min}$

$$M_0 = \left\{n : \ \boldsymbol{x}_n \in \Omega, \ \boldsymbol{x}_n \neq \boldsymbol{x}^i_{min}, \quad i = 1, 2\right\}.$$

M_1 is the set of numbers of the units of CGB, which crosses $\partial\Omega$, with exception of the number of the minimal unit z_{min} :

$$M_1 = \{n : \ z_n \cap \partial\Omega \neq \emptyset, z_n \neq z_{min}\}.$$

Introduce two additional points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$:

$$\boldsymbol{z}_i = \boldsymbol{x}^{\,i}_{min} + (-1)^i \Delta l\,\boldsymbol{\tau}, \quad i = 1, 2,$$

where

$$\boldsymbol{\tau} = \frac{1}{l_{min}}(\boldsymbol{x}^{\,2}_{min} - \boldsymbol{x}^{\,1}_{min}),$$

$$\Delta l = \frac{1}{4}(h(\boldsymbol{x}^{\,c}_{min}) - l_{min}).$$

Determine the angle ξ at the vertex $\boldsymbol{z}_\xi$ in the triangle $\Delta(\boldsymbol{x}_*, \boldsymbol{z}_1, \boldsymbol{z}_2)$, where $\boldsymbol{z}_\xi = \boldsymbol{z}_1$, if $z^*_{min} = z^-_{min}$ and $\boldsymbol{z}_\xi = \boldsymbol{z}_2$, if $z^*_{min} = z^+_{min}$ (see step 1).

If one of the sets M_o or M_1 is not empty, then come to step 5.

Determine the angle α_1 at the vertex $\boldsymbol{x}_*$ in the triangle $\Delta(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{x}_*)$. If $\alpha_1 \geq 30^o$ and $\beta_{min} - \xi \geq 20^o$, then declare the point $\boldsymbol{x}_*$ a new node and come to step 2.

If $\alpha_1 < 30^o$, then redetermine the point $\boldsymbol{x}_*$ so that the new node has $\alpha_1 = 30^o$:

$$\boldsymbol{x}_* = \boldsymbol{x}^c_{min} + |\boldsymbol{z}_2 - \boldsymbol{x}^c_{min}| \cdot \mathrm{tg}\, 75^o \cdot \boldsymbol{n}.$$

At that, if $\beta_{min} - 75^o < 20^o$, then come to step 2, else to step 14.

The introduction of the points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ is obliged to the fact that the unit z_{min} at one of the previous steps of construction of the element can be produced by different ways: through connection of two neighbouring units (step 13), through connection of two previously constructed nodes (step 12), through construction of a node (step 14). Therefore the length of z_{min} in the domains of larger gradients of the function of steps $h(x, y)$ can be 2-4 times less than the value $h(\boldsymbol{x}^c_{min})$. Since after construction of the grid an improvement is made which allows to extend z_{min} somewhat in such domains, it is better to estimate the quality of the element being constructed through the points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ which are midpoints between $\boldsymbol{x}^1_{min}, \boldsymbol{x}^c_{min} - \frac{1}{2}h(\boldsymbol{x}^c_{min})\boldsymbol{\tau}$ and $\boldsymbol{x}^2_{min}, \boldsymbol{x}^c_{min} + \frac{1}{2}h(\boldsymbol{x}^c_{min})\boldsymbol{\tau}$, respectively. Thus, in the course of construction of a new node, in such domains we analyse not the elements constructed according to the new node but their possible transformations after inprovement of the grid. The estimation of value of the angle $\beta_{min} - \xi$ is performed in order to avoid acute angles between the units $z^-_{min}, [\boldsymbol{x}^1_{min}, \boldsymbol{x}^2_*]$ (if $z^*_{min} = z^-_{min}$) or $[\boldsymbol{x}_*, \boldsymbol{x}^2_{min}], z^+_{min}$ (if $z^*_{min} = z^+_{min}$).

In Fig. 4 a situation is shown when declaration of the triangle $\Delta(z_{min}, \boldsymbol{x}_*)$ as a new element results further in appearing the triangle $\Delta(z^+_{min}, \boldsymbol{x}_*)$ with acute angle. Therefore in such situations we will refuse to construct new node and (under favourable conditions) take as an element $\Delta(z_{min}, z^*_{min})$, i.e., come to step 2.

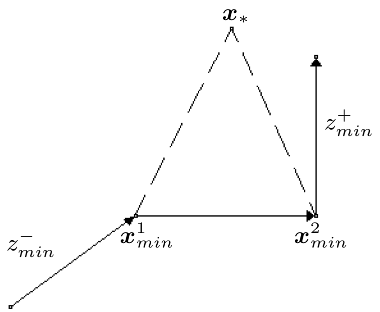
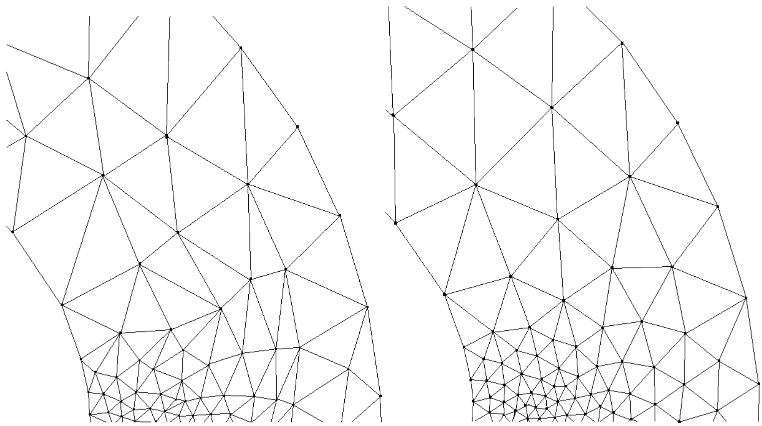


Fig. 4: In this situation there is no new node.

The analysis of the angles α_1 and $\beta_{min} - \xi$ was introduced into the initial variant of the algorithm after consideration of a large number of experimental calculations made in the domains of large gradients of $h(\boldsymbol{x})$. One of examples is shown in Fig. 5: a grid is shown before (Fig. 5a) and after (Fig. 5b) its improvement. From this, one can see that if the analysis of the elements being constructed is performed over the lengths of their sides (see Fig. 5a) but not over average values, then a sharp enough transition is possible from the elements of small sizes to elements of large sizes, what entails poor quality of elements in the domains in such vicinity.



a) **b**)

Fig. 5: The grid before **a**) and after **b**) improvement.

Step 5.

If $M_o \neq \emptyset$ (else go to step 9), then choose from M_0 a number m for which the corresponding node $\boldsymbol{x}_m$ is closest to z_{min}. To do this, determine the distances l_i (see Appendix 2) from the points $\boldsymbol{x}_i, i \in M_o$, to z_{min} and choose

$$l_m = \min_{i \in M_o} l_i \to m. \tag{3.2}$$

In the triangle $\Delta(\boldsymbol{x}_m, \boldsymbol{z}_1, \boldsymbol{z}_2)$ consider the angle α at the vertex $\boldsymbol{x}_m$ ($\boldsymbol{z}_i$ are determined in step 4). If $\alpha \geq 30^o$ (else come to step 8), then test an intersection of the segment $[\boldsymbol{x}_m, \boldsymbol{x}^1_{min}]$ with the units of CGB with numbers from M_1 without the nimbers of units neighbouring the nodes $\boldsymbol{x}_m, \boldsymbol{x}^1_{min}$ (see Appendix 3). For convenience, denote $\boldsymbol{x}_m$ by $\boldsymbol{y}_*$. Introduce an integer parameter IND of switching and set $IND = 0$.

Step 6.

If there are no intersections, then go to step 12 if $IND = 0$ or to step 14 if $IND = 1$.

Step 7.

There are intersections. With use of the nodes of the intersecting unit z_p, construct oriented triangles $\Delta(z_{min};\ \boldsymbol{x}_k), \boldsymbol{x}_k$ are nodes of the unit z_p, $k \in \{k_1, k_2\}$ (see Appendix 4). When constructing these triangles, one should

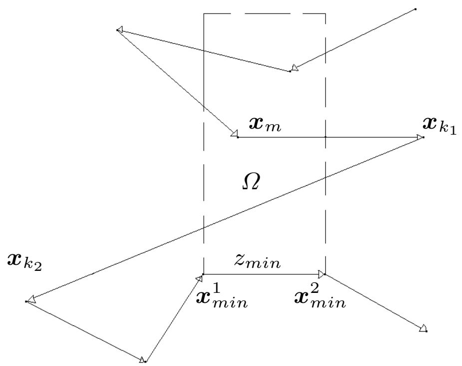


Fig. 6: Testing rectangle Ω.

make sure of their existence (in Fig. 6 oriented triangle $\Delta(z_{min}, \boldsymbol{x}_{k_2})$ does not exist; $\boldsymbol{x}_m$ is the node closest to z_{min}).

From these triangles (if both exist) choose $\Delta(z_{min}, \boldsymbol{x}_{k_i}),\ k_i \in \{k_1, k_2\}$, whose minimal angle is larger (further, we will denote the minimal angle of any triangle $\Delta(z, \boldsymbol{x})$ by $\alpha(z, \boldsymbol{x})$). Having denoted the choosen node $\boldsymbol{x}_{k_i}$ by $\boldsymbol{y}_*$, test an intersection of both lateral sides of $\Delta(z_{min}, \boldsymbol{y}_*)$ with all the units of CGB except the minimal one and those adjoining it and node $\boldsymbol{y}_*$. Then come to step 6.

Step 8.

The closest node $\boldsymbol{x}_m$ is far enough from z_{min} (since $\alpha < 30^o$), therefore displace the constructed point $\boldsymbol{x}_*$ to z_{min} so that the distance between its new location (denote this point by $\boldsymbol{y}_*$) and z_{min} would be equal to $l_m/2$:

$$\boldsymbol{y}_* = \boldsymbol{x}^c_{min} + \frac{1}{2} l_m \boldsymbol{n},$$

and l_m is defined in (2).

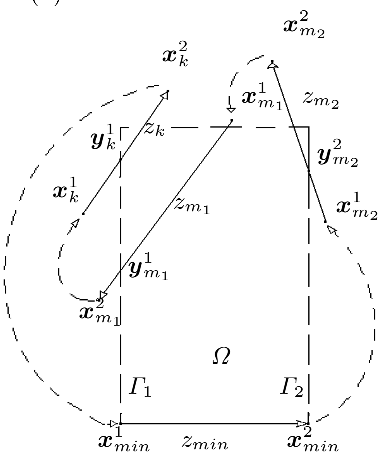


Fig. 7: Test of an intersection rectangle Ω with units.

In the triangle $\Delta(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{y}_*)$ determine the angle ξ at the vertex $\boldsymbol{z}_\xi$ defined in step 4. If $\beta_{min} - \xi < 20^o$, then come to step 2, else test an intersection of one of lateral sides of $\Delta(z_{min}, \boldsymbol{y}_*)$ with the units with numbers from M_1 and come to step 6, setting $IND = 1$.

Step 9.

If $\beta_{min} - \xi < 20^o$, then come to step 2, otherwise choose from all the units intersecting $\partial\Omega$ the units z_{m_1} and z_{m_2} which are "closest" to z_{min}. For this, consider all the points of intersection $\boldsymbol{y}^1_i$ of the units z_{k_i}, $k_i \in M_1$, with the lateral side Γ_1 of rectangle Ω, which comes through $\boldsymbol{x}^1_{min}$, and analogous points $\boldsymbol{y}^2_i$ for Γ_2. Then the numbers m_1 and m_2 are determined as

$$\min_{k_i \in M_1} |\boldsymbol{y}^j_i - \boldsymbol{x}^j_{min}| = m_j, \quad j = 1, \ldots, N_p.$$

If Γ_1 and Γ_2 intersect different "closest units ", then $N_p = 2$ (Fig. 7). If one of the lateral sides Γ_j is not intersected by the units (Fig. 6), or if Γ_1 and Γ_2 are intersected by the same unit (Fig. 7), then $N_p = 1$, and consider only the number m_1 (if necessary, m_1 is specified as m_2, Fig. 8a).

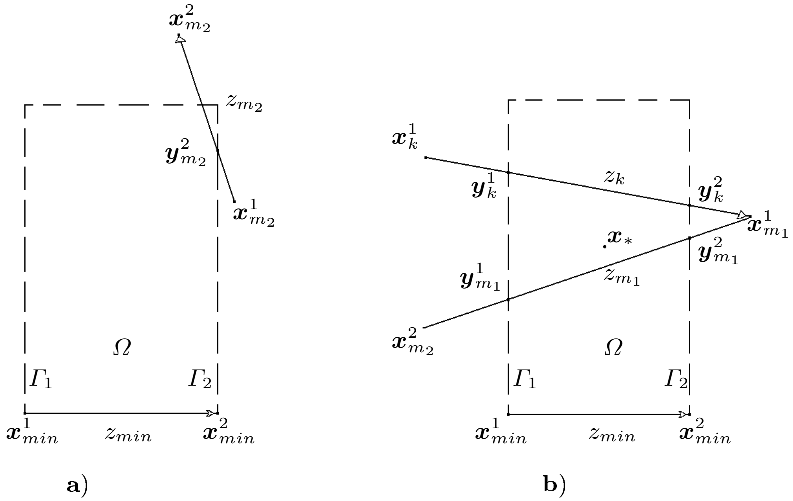


Fig. 8: All possible situations when lateral sides of Ω intersect with units.

Consider oriented $\Delta(z_{m_i}, \boldsymbol{x}_*)$, $\quad i = 1, \ldots, N_p;\quad$ $\boldsymbol{x}_*$ is defined in (3.1). If such oriented triangles do not exist (that is possible only in the case, when the "closest" intersecting unit is unique and comes through Γ_1 and Γ_2 between z_{min} and $\boldsymbol{x}_*$, see Fig. 8b), then denote the unit z_{m_1} by z_p and come to step 7. Under existence of $\Delta(z_{m_i}, \boldsymbol{x}_*)$ consider the angles α_i at the vertex $\boldsymbol{x}_*$ in these triangles, $i = 1, \ldots, N_p$. If $\alpha_i \leq 90^o$, then go to step 14, otherwise choose $\alpha_{i_1} = \max\limits_{1 \leq i \leq N_p} \alpha_i$.

Step 10.

Construct oriented triangles $\Delta(z_{min}, \boldsymbol{x}^j_{m_{i_1}})$, where $\boldsymbol{x}^j_{m_{i_1}}$ are nodes of the unit $z_{m_{i_1}}$. Possible values of the parameter j can be of the following list: $j \in \{1, 2\}$, if both the triangles exist. If only one triangle exists, then $j = 1$ if the node $\boldsymbol{x}^1_{m_{i_1}}$ is used, and $j = 2$ for the second node of the unit $z_{m_{i_1}}$.

Choose from $\Delta(z_{min}, \boldsymbol{x}^j_{m_{i_1}})$ the triangle which has larger minimal angle:

$$\alpha(z_{min}, \boldsymbol{x}^{j_1}_{m_{i_1}}) \geq \alpha(z_{min}, \boldsymbol{x}^j_{m_{i_1}})$$

for all j from the list of values of this index: Then come to step 11.

Step 11.

If

$$\alpha(z_{min}, \boldsymbol{x}^{j_1}_{m_{i_1}}) \leq \alpha(z_{min}, \boldsymbol{x}_*),$$

then come to step 14, otherwise test an intersection of lateral sides of $\Delta(z_{min}, \boldsymbol{x}^{j_1}_{m_{i_1}})$ with all the units of CGB, except the nodes adjoining the node $\boldsymbol{x}^{j_1}_{m_{i_1}}$ and the unit z_{min}. Besides, test a getting the nodes of CGB into this triangle, except the node $\boldsymbol{x}^{j_1}_{m_{i_1}}$ and nodes $\boldsymbol{x}^1_{min}$, $\boldsymbol{x}^2_{min}$.

If there are no intersections and nodes inside the triangle, then redenote the node $\boldsymbol{x}^{j_1}_{m_{i1}}$ by $\boldsymbol{y}_*$ and go to step 12.

If there are intersections or if the triangle contains at least one node of CGB, then choose a new value j_2 from the list of values of parameter j (if it is not exhausted) and come to step 11, preliminary redenoting j_2 by j_1. If the list of parameter j is exhausted, then consider the second intersecting unit $z_{m_{i_2}}$ (under the condition that the list of parameter m_i is not exhausted, $i = 1, \ldots, N_p$), and if $\alpha_{i_2} > 90^o$, come to step 10, preliminarily redenoting i_2 by i_1. Otherwise (either $\alpha_{i_2} < 90^o$, or the list of parameter m_i is exhausted) come to step 14.

Step 12.

Declare the triangle $\Delta(z_{min}, \boldsymbol{y}_*)$ as an element, remove z_{min} from CGB, determine the number of connectivity of the domain, add two new units $[\boldsymbol{x}^1_{min}, \boldsymbol{y}_*], [\boldsymbol{y}_*, \boldsymbol{x}^2_{min}]$, and come to step 1.

The number of connectivity is increased by one, if $\boldsymbol{y}_*$ and $\boldsymbol{x}^1_{min}$ belong to one contour of CGB, and decreased by one, if these nodes belong to different contours (see Appendix 7).

Step 13.

Declare the triangle $\Delta(z_{min}, z^*_{min})$ as an element, remove z_{min}, z^*_{min} from CBN, add one unit $[\boldsymbol{x}^{min}_1, \boldsymbol{x}^{min}_3]$ (see Appendix 5), and come to step 1.

Step 14.

Declare the triangle $\Delta(z_{min}, \boldsymbol{x}_*)$ as a new element, remove z_{min} from CBN, add two new units $[\boldsymbol{x}^1_{min}, \boldsymbol{x}_*], [\boldsymbol{x}_*, \boldsymbol{x}^2_{min}]$ (see Appendix 6), and come to step 1.

4 Conclusion

We illustrate of performance of the algorithms for different domains in figures below.

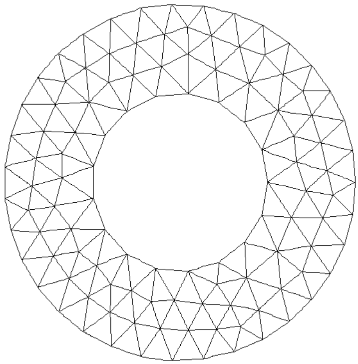


Fig. 9: A grid for a ring.

In Fig. 9 a grid is given for $h(\boldsymbol{x}) = 0.18$ for a ring with internal and external radii 0.5 and 1, respectively (all the values here and below are divided by dimensional unity). The equations of the contours of a ring were given in parametric form:

$$\boldsymbol{x}(t) = (\cos t,\ \sin t), \quad \boldsymbol{x}(t) = 0.5(\cos t,\ -\sin t), \quad 0 \le t \le 2\pi. \qquad (4.1)$$

The value of the parameter ε (see inequality (2.4)) was set as 0.001. The grid has 178 elements and 115 nodes.

A ring with two circular cuts is shown in Fig. 10a . Two contours have parametrization (4.1), the other two are defined as follows:

$$(x-0.6)^2 + y^2 = (0.05)^2, \quad \boldsymbol{x} = (0.6 + 0.05\cos t,\ -0.05\sin t);$$

$$(4.2)$$

$$(x-0.4)^2 + (y+0.7)^2 = (0.1)^2, \quad \boldsymbol{x} = (0.4 + 0.1\cos t,\ -0.7 - 0.1\sin t),$$

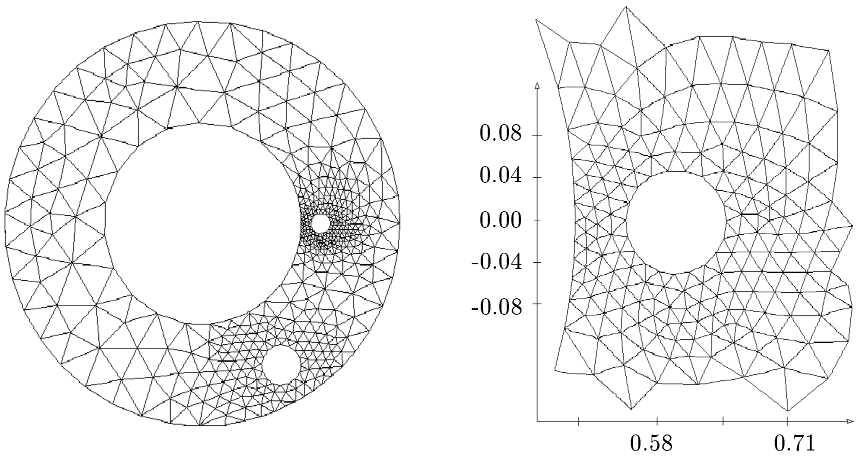
$$-2\pi \le t \le 0.$$

The function of steps has two points of concentration in the centers of the circumferences (4.3):

$$h(x,y) = h_0 + (h_1 - h_0)/A(x,y) + (h_2 - h_0)/B(x,y),$$

$$A(x,y) = 1 + \left[\left(\frac{x-0.6}{0.3}\right)^2 + \left(\frac{y}{0.2}\right)^2\right]^2,$$

$$B(x,y) = 1 + \left[\left(\frac{x-0.4}{0.4}\right)^2 + \left(\frac{y+0.7}{0.3}\right)^2\right]^2,$$

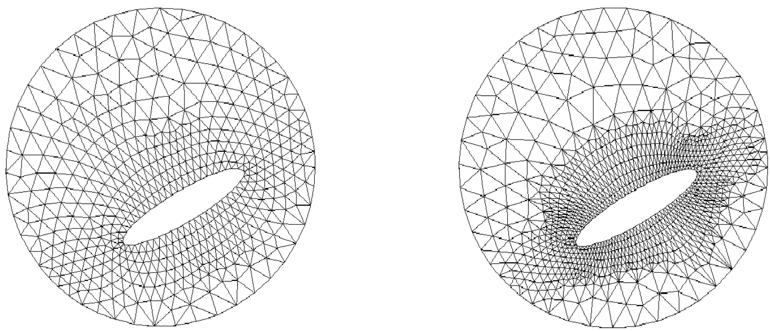


a) zoom, normal size. **b**) zoom, large size.

Fig. 10: A grid for a ring with two circular cuts.

$$h_0 = 0.168, \quad h_1 = 0.02, \quad h_2 = 0.04.$$

The grid has 864 elements and 483 nodes. The vicinity of the circumference with radius 0.05 is shown in Fig. 10b in larger scale.



a) 489 nodes. **b**) 954 nodes.

Fig. 11: Grids for a circular disk with an ellipsoidal cut.

Figures 11a and 11b demonstrate fragmentations of the same domain under diverse parameters of the function of steps. The external boundary of the domain is a circumference with radius 2, and its parametric equation is

$$\boldsymbol{x}(t) = 2(\cos t,\ \sin t), \quad 0 \le t \le 2\pi.$$

The internal boundary is an ellipse with center in the point $\boldsymbol{x}_c = (0.3; -0.5)$ and its major semiaxis is inclined at the angle $\alpha = 30^o$ to the axis Ox. The principal axes are $a = 0.9$ and $b = 0.2$. Parametric equation of the ellipse with the account of clockwise encircling of the boundary has the form

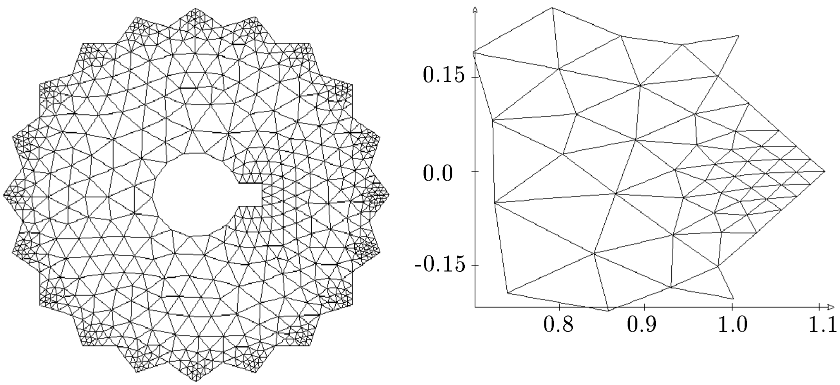
$$x(t) = x_c + a\cos t\cos\alpha + b\sin t\sin\alpha,$$
$$y(t) = y_c + a\cos t\sin\alpha - b\sin t\cos\alpha, \quad 0 \le t \le 2\pi.$$

The center of the domain of concentration is the center of the ellipse. The function of steps for both the triangulations was taken in the form

$$h(x,y) = h_0 + (h_1 - h_0)\left\{1 + \left(\frac{\tilde{x}}{2a}\right)^m + \left(\frac{\tilde{y}}{2b}\right)^m\right\}^{-1},$$

$$\tilde{x} = (x - x_c)\cos\alpha + (y - y_c)\sin\alpha, \quad \tilde{y} = -(x - x_c)\sin\alpha + (y - y_c)\cos\alpha,$$

with $h_0 = 0.3$, $h_1 = 0.03$. In Fig. 11a we take $m = 2$, and in Fig. 11b we set $m = 4$. The grid in Fig. 11a has 880 elements and 489 nodes, and that in Fig. 11b has 1768 and 954, respectively.



a)a pinion: normal size. **b**) one cog: large size.

Fig. 12: Grids for a pinion with 20 cogs.

In Fig. 12a the domain is a pinion with $N = 20$ cogs. A grid has 1219 elements and 702 nodes. The parameters of k−th cog are given in Fig. 13a, where

$$\alpha = 360^o/N, \quad \alpha_k = (k - 1.5)\alpha, \quad \alpha_k^* = \alpha_k + 0.5\alpha\ , 1 \le k \le N.$$

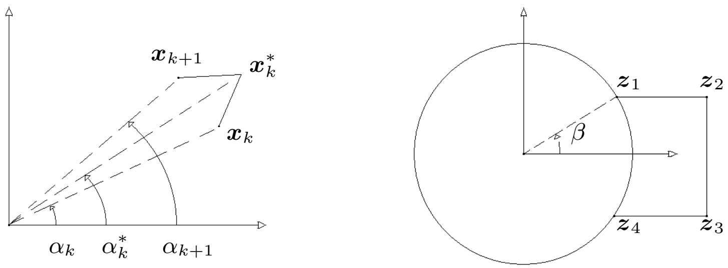
Then the points of bases and tops of the cogs are calculated as

$$\boldsymbol{x}_k = r(\cos\alpha_k; \sin\alpha_k), \quad \boldsymbol{x}_k^* = R(\cos\alpha_k^*; \sin\alpha_k^*), \quad k = 1,..,N; \ \boldsymbol{x}_{N+1} = \boldsymbol{x}_1 .$$

Parametrization of the external boundary is fulfilled for each side of a cog:

$$\boldsymbol{x} = \boldsymbol{x}_k + t(\boldsymbol{x}_k - \boldsymbol{x}_k), \quad \boldsymbol{x} = \boldsymbol{x}_k + t(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k), \quad 0 \le t \le 1, \quad 1 \le k \le N.$$

Parametrization of the internal boundary is demonstrated in Fig. 13b:



a) the external boundary. **b**) the internal boundary.

Fig. 13: The boundary parametrization for a pinion.

$$\boldsymbol{x}(t) = r_1(\cos t\ ; -\sin t)\ , \quad r_1 = 0.25, \ \ \beta \le t \le 2\pi - \beta,$$
$$\boldsymbol{x}(t) = \boldsymbol{z}_{i-1} + t(\boldsymbol{z}_i - \boldsymbol{z}_{i-1})\ , \quad 0 \le t \le 1, \ \ i = 2,3,4$$

Here

$$\boldsymbol{z}_1 = r_1(\cos\beta;\ \sin\beta), \quad \boldsymbol{z}_2 = r_1(1.5;\ \sin\beta),$$
$$\boldsymbol{z}_3 = r_1(1.5;\ -\sin\beta), \quad \boldsymbol{z}_4 = r_1(\cos\beta;\ -\sin\beta).$$

The centers of the concentration domains for the cogs are located in the vertices $\boldsymbol{x}_k^*$; the axes of the domains lies on the rays $\beta = \alpha_k^*$ and in orthogonal directions. The value of step is the same and equals $h_1 = 0.25|\boldsymbol{x}_k - \boldsymbol{x}_k|$. The center of the concentration domain for internal cut is located in the point $(r_1; 0)$; the value of step is $h_2 = 0.05$; the concentration domain is stretched along the axis Ox.

So, the final form of the function of steps is

$$h(x,y) = h_0 + (h_1 - h_0)\sum_{i=1}^{N}\left\{1 + \left(\frac{\tilde{x}_k}{a}\right)^4 + \left(\frac{\tilde{y}_k}{b}\right)^4\right\}^{-1} +$$
$$(h_2 - h_0)\left\{1 + \left(\frac{x - r1}{cr_1}\right)^4 + \left(\frac{y}{dr_1}\right)^4\right\}^{-1},$$

where

$$a = 1.5(R - r), \quad b = 0.7\pi R/N, \quad c = 1.3, \quad d = 0.7,$$
$$\tilde{x}_k = (x - x_k{}^*)\cos\alpha_k^* + (y - y_k{}^*)\sin\alpha_k^*,$$
$$\tilde{y}_k = -(x - x_k{}^*)\sin\alpha_k^* + (y - y_k{}^*)\cos\alpha_k^*.$$

In order to demonstrate details of the grid on a cog, one of the cogs was cut out, and the enlarged grid is shown in Fig. 12b.

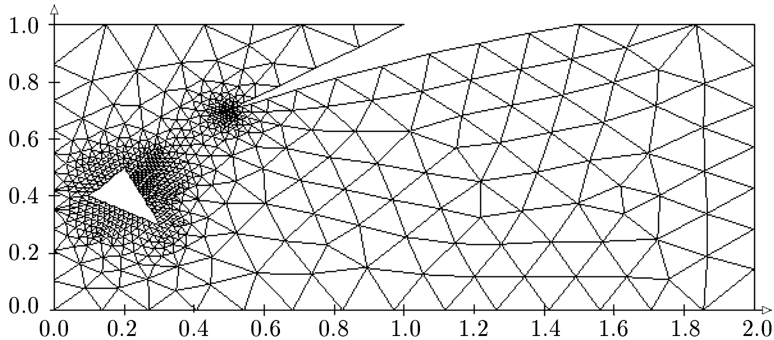


Fig. 14: A grid for the rectangle with wedge-shaped and triangle cuts.

In Fig. 14 a grid for a rectangle with wedge-shaped and triangle cuts is shown. The grid has 1348 elements and 726 nodes.

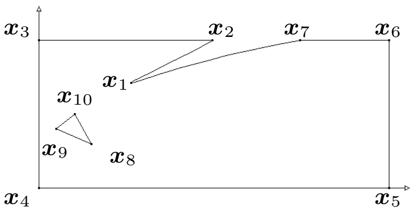


Fig. 15: The rectangle with wedge-shaped and triangle cuts.

In Fig. 15 the vertices $\boldsymbol{x}_i$, $i = 1, ..., 10$, are shown which determine a domain where

$$\boldsymbol{x}_1 = (0.5;\ 0.7),\ \boldsymbol{x}_2 = (1;\ 1),\ \boldsymbol{x}_3 = (0;\ 1),\ \boldsymbol{x}_4 = (0;\ 0),$$
$$\boldsymbol{x}_5 = (2;\ 0),\ \boldsymbol{x}_6 = (2;\ 1),\ \boldsymbol{x}_7 = (1.5;\ 1),\ \boldsymbol{x}_8 = (0.3;\ 0.3),$$
$$\boldsymbol{x}_9 = (0.1;\ 0.4),\ \boldsymbol{x}_{10} = (0.2;\ 0.5).$$

All the lines are straight except of the line from $\boldsymbol{x}_7$ into $\boldsymbol{x}_1$. T heir parametrization is

$$\boldsymbol{x}(t) = \boldsymbol{x}_* + t(\boldsymbol{x}_{**} - \boldsymbol{x}_*), \quad 0 \le t \le 1;$$

$\boldsymbol{x}_*$ is the beginning point of segment, $\boldsymbol{x}_{**}$ is the end point of segment. The line from $\boldsymbol{x}_7$ into $\boldsymbol{x}_1$ is the parabola $y = ax^2+bx+c$, where a, b, c are selected according to the conditions that the parabola comes through the points $\boldsymbol{x}_7$ and $\boldsymbol{x}_1$ and the value of derivative under $x = x_1$ is 0.4. Parametrization of this line with account of counterclockwise encircling of the external contour is

$$\boldsymbol{x}(t) = (-t;\ at^2 - bt + c), \quad -x_7 \le t \le -x_1.$$

The grid obtained has two concentration domains; the center of the first domain is the point $\boldsymbol{x}_c = \frac{1}{3}(\boldsymbol{x}_8 + \boldsymbol{x}_9 + \boldsymbol{x}_{10})$, the center of the second one is the point $\boldsymbol{x}_1$. Sizes of steps were chosen as

$$\begin{aligned} h_1 &= 0.1\ \min(|\boldsymbol{x}_8 - \boldsymbol{x}_{10}|,\ |\boldsymbol{x}_9 - \boldsymbol{x}_{10}|,\ |\boldsymbol{x}_8 - \boldsymbol{x}_9|) \simeq 0.022, \\ h_2 &= 0.04\,|\boldsymbol{x}_1 - \boldsymbol{x}_7| \simeq 0.023. \end{aligned}$$

Each concentration domain was symmetric with respect to its center and the axes of a local coordinate system obtained by parallel transfer of the initial system into the center of concenteration

$$\begin{aligned} r_1 &= 1.6\ \max(|\boldsymbol{x}_8 - \boldsymbol{x}_{10}|,\ |\boldsymbol{x}_9 - \boldsymbol{x}_{10}|,\ |\boldsymbol{x}_8 - \boldsymbol{x}_9|) \simeq 0.22, \\ r_2 &= 0.2\,|\boldsymbol{x}_1 - \boldsymbol{x}_7| \simeq 0.12\ . \end{aligned}$$

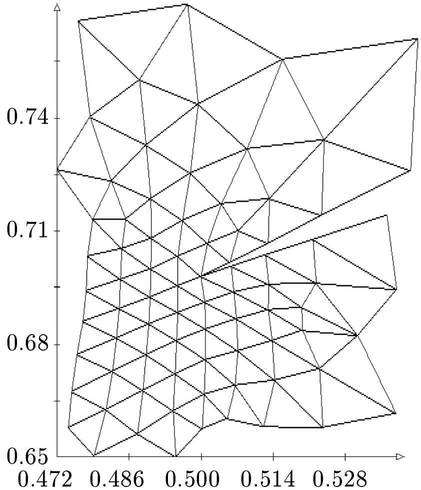
Major step of the grid is $h_0 = 0.15$. Then, with account of the form of the concentration domains and their centers, the function of steps was taken as

$$\begin{aligned} h(x,y) = h_0 + (h_1 - h_0)\left\{1 + \left(\frac{x - x_c}{r_1}\right)^4 + \left(\frac{y - y_c}{r_1}\right)^4\right\}^{-1} + \\ + (h_2 - h_0)\left\{1 + \left(\frac{x - x_7}{r_2}\right)^2 + \left(\frac{y - y_7}{r_2}\right)^2\right\}^{-1}. \end{aligned}$$

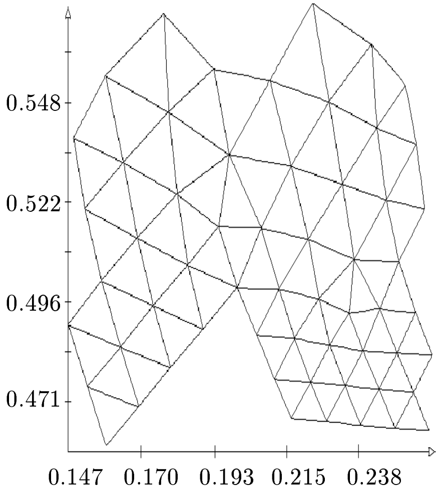
In figures 16a, 16b enlarged vicinities of the vertex $\boldsymbol{x}_1$ of wedge-shaped cut and of the vertex $\boldsymbol{x}_{10}$ of triangle cut are shown.

From explanations it is clear that for the construction of a grid it is necessary to input only the sufficient information: equations of contours and function of steps.

There are two disadvantages of the proposed algorithms. First, it is impossible to determine beforehand the length of the arrays storing the information about the grid. Second, numeration of nodes is not optimal. Since



a) the wedge-shaped cut.



b) the triangle cut.

Fig. 16: Zoom of cut areas , large size.

the length of arrays is not known beforehand, then in programs it is necessary to check the border of the array. If the ordered length of some array is less than it is necessary, then the programs halts and a message is displayed. Since the programs of fragmentation are used within the frames of more extensive computations, then for determination of lengths of the arrays it is recommended at first to run these programs for the given domain without the complementary programs.

More essential disadvantage is the non-optimality of numeration of nodes, what results in sparse stiffness matrix when using the finite element method. Storage of the whole stiffness matrix considerably increases the required resources of memory, therefore in the present case it is necessary either to use specific methods of storage and solution of large sparsed systems [6-18], or to avoid constructing global matrix and use some iterative methods. The reason for use of iterative methods is that they presuppose only calculation of products of matrix by vector, what can be done if we known the local siffness matrices. The array which stores the numbers of elements adjoining $\boldsymbol{x}_k$ can be filled immediately in the process of triangulation of the domain.

Thus, the existence of practically effective algorithms for sparse matrices and a possibility to solve a system of equations without formation of global matrix by some iterative method allow to eliminate the second shortcoming.

As a conclusion let note that these algorithms extremely convenient for use due to the possibility of elimination of the direct and indirect shortcomings of the algorithms of fragmentation together with the simplicity of handling, high degree of complexity of triangulated domains and a good quality of grid.

5 Appendix 1

Let the points $(x_1,\ y_1), (x_2,\ y_2), (x_3,\ y_3)$ be vertices of a triangle which are written down in the order of counterclockwise encircling. The point (x_*, y_*) lies outside the triangle if at least one of the following inequalities is valid:

$$v_i > 0, \qquad i = 1, 2, 3,$$

where

$$v_1 = (x_* - x_1)(y_2 - y_1) - (x_2 - x_1)(y_* - y_1),$$
$$v_2 = (x_* - x_2)(y_3 - y_2) - (x_3 - x_2)(y_* - y_2),$$
$$v_3 = (x_* - x_3)(y_1 - y_3) - (x_1 - x_3)(y_* - y_3).$$

6 Appendix 2

Let in the triangle $\Delta(\boldsymbol{x}, \boldsymbol{x}_1, \boldsymbol{x}_2)$ the angles at the vertices $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are acute. Only such situations arise in the algorithm of triangulation. The distance l from the point $\boldsymbol{x}$ to the segment $[\boldsymbol{x}_1, \boldsymbol{x}_2]$ under the condition is

$$l = |\boldsymbol{x} - t(\boldsymbol{x}_2 - \boldsymbol{x}_1)|, \quad t = \frac{(\boldsymbol{x},\ \boldsymbol{x}_2 - \boldsymbol{x}_1)}{|\boldsymbol{x}_2 - \boldsymbol{x}_1|^2},$$

where $(\ ,\ )$ is the Euclidian scalar product; $|\ \cdot\ |$ is length of vector.

7 Appendix 3

Let two segments be determined by the points $\boldsymbol{x}_1 = (x_1^1,\ x_2^1), \boldsymbol{x}_2 = (x_1^2,\ x_2^2)$ and $\boldsymbol{y}_1 = (y_1^1, y_2^1)$, $\boldsymbol{y}_2 = (y_1^2,\ y_2^2)$. The test of intersection of two these segments can be performed as follows: points of the first segment are

$$\boldsymbol{x} = \boldsymbol{x}_1 + t_1(\boldsymbol{x}_2 - \boldsymbol{x}_1), \quad t_1 \in [0, 1],$$

and points of the second segment are

$$\boldsymbol{y} = \boldsymbol{y}_1 + t_2(\boldsymbol{y}_2 - \boldsymbol{y}_1), \quad t_2 \in [0, 1].$$

These segments do not intersect, if the system of two equations with respect to t_1 and t_2

$$\boldsymbol{x}_1 + t_1(\boldsymbol{x}_2 - \boldsymbol{x}_1) = \boldsymbol{y}_1 + t_2(\boldsymbol{y}_2 - \boldsymbol{y}_1)$$

does not have solution (the segments are parallel) or one of the solutions does not belong to the interval [0, 1].

More effective algorithm of testing is as follows: if at least one of the inequalities

$$v_i > 0 \qquad i = 1, 2,$$

is satisfied, then the segments do not intersect. Here

$$v_1 = [(y_1^1 - x_1^1)(x_2^2 - x_2^1) - (x_1^2 - x_1^1)(y_2^1 - x_2^1)]$$
$$\times[(y_1^2 - x_1^1)(x_1^2 - x_1^1) - (x_1^2 - x_1^1)(y_2^2 - x_2^1)],$$
$$v_2 = [(x_1^1 - y_1^1)(y_2^2 - y_2^1) - (y_1^2 - y_1^1)(x_2^1 - y_2^1)]$$
$$\times[(x_1^2 - y_1^1)(y_2^2 - y_2^1) - (y_1^2 - y_1^1)(x_2^2 - y_2^1)].$$

8 Appendix 4

Oriented triangle $\Delta(z_{min}, \boldsymbol{x}_k)$ is understood as a triangle with ordered list of vertices $\boldsymbol{x}_{min}^1 = (x_1^1,\ x_2^1)$, $\boldsymbol{x}_{min}^2 = (x_1^2,\ x_2^2)$, $\boldsymbol{x}_k = (x_1^k,\ x_2^k)$, its direction of encircling is determined counterclockwise and this encircling does not contradict to the list of vertices. Existence criterion of oriented triangle is the inequality

$$(x_1^2 - x_1^1)(x_2^k - x_2^1) - (x_1^k - x_1^1)(x_2^2 - x_2^1) > 0.$$

9 Appendix 5

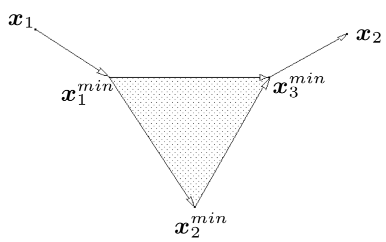


Fig. 17: Connection of two adjacent units creates the new triangular element.

When constructing triangular element by connection of two adjacent units on l-th contour of CGB, the value $K(l)$ of the array K is decreased by one, and the transformation of the array M is performed as shown in the figure 17. I.e. from the list of units of l-th contour the units

$$[\boldsymbol{x}_1^{min}, \boldsymbol{x}_2^{min}],\ [\boldsymbol{x}_2^{min}, \boldsymbol{x}_3^{min}]$$

are removed, and a new unit $[\boldsymbol{x}_1^{min}, \boldsymbol{x}_3^{min}]$ is included. In other words, the contour $\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_2^{min},\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1$ is transformed into the contour $\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1$.

10 Appendix 6

When constructing triangle element (on the basis of unit z_{min} belonging to l-th contour of CGB) by construction of a new node $\boldsymbol{x}_*$, the value $K(l)$ of the array K is increased by one, and transformation of the array M is performed as shown in the figure 18.

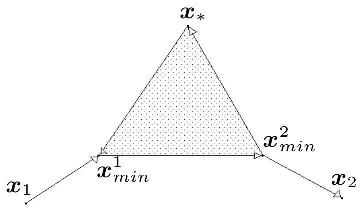


Fig. 18: A new node $\boldsymbol{x}_*$ creates the new triangular element.

I.e., from the list of units of l-th contour the unit $[\boldsymbol{x}_{min}^1, \boldsymbol{x}_{min}^2]$ is removed, and the units $[\boldsymbol{x}_{min}^1, \boldsymbol{x}_*]$, $[\boldsymbol{x}_*, \boldsymbol{x}_{min}^2]$ are included. In other words, the contour $\boldsymbol{x}_1,\ \boldsymbol{x}_{min}^1,\ \boldsymbol{x}_{min}^2,\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1$ is transformed into the contour $\boldsymbol{x}_1,\ \boldsymbol{x}_{min}^1,\ \boldsymbol{x}_*,\ \boldsymbol{x}_{min}^2,\ \boldsymbol{x}_2 \ldots,\ \boldsymbol{x}_1$.

11 Appendix 7

When constructing a triangular element (on the basis of the unit z_{min} belonging to l_1 -th contour of CGB) through the previously constructed node $\boldsymbol{y}_*$ belonging to l_2-th contour, the number of connectivity of the domain is increased by one if $l_1 = l_2$, and two new contours of CGB are introduced due to fragmentation of the previous one; if $l_1 \neq l_2$, then the connectivity is decreased by one, to $K(l_1)$ the value $K(l_1) + K(l_2) + 1$ is assigned to $K(l_1)$ and $K(l_2)$ is set to be zero.

Transformation of the array M is performed as shown in the figures 19 and 20.

Under condition $l_1 = l_2$

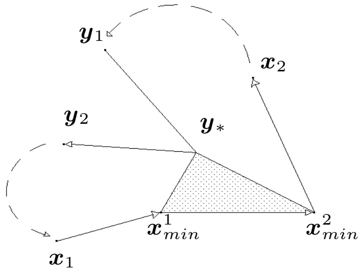


Fig. 19: The connectivity is increased.

the contour $\boldsymbol{x}_1,\ \boldsymbol{x}^1_{min},\ \boldsymbol{x}^2_{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{y}_1,\ \boldsymbol{y}_*,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{x}_1$ has divided into two contours $\boldsymbol{x}_1,\ \boldsymbol{x}^1_{min},\ \boldsymbol{y}_*,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{x}_1$ and $\boldsymbol{x}^2_{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{y}_1,\ \boldsymbol{y}_*,\ \boldsymbol{x}^2_{min}$. If $l_1 \neq l_2$

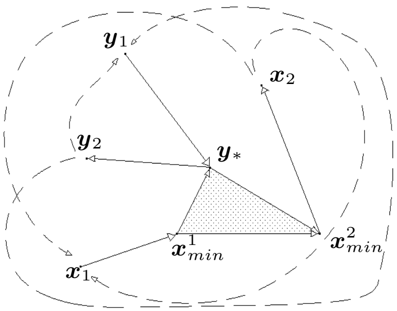


Fig. 20: The connectivity is decreased.

the contours $\boldsymbol{y}_1,\ \boldsymbol{y}_*,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{y}_1$ and $\boldsymbol{x}_1,\ \boldsymbol{x}^1_{min},\ \boldsymbol{x}^2_{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1$ have combined into one contour

$$\boldsymbol{y}_1,\ \boldsymbol{y}_*,\ \boldsymbol{x}^2_{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1,\ \boldsymbol{x}^1_{min},\ \boldsymbol{y}_*,\ \boldsymbol{y}_2,\ \ldots \boldsymbol{y}_1.$$

The possible variants of closure of l_1-th and l_2-th contours are shown with dotted lines, with exception of the case when the lines intersect.

12 Appendix 8

When constructing two triangle elements by division of a quadrangle into two triangles so that the minimal angle of the triangles would be maximal, two situations, as in App. 7, are possible.

1. The units z_{min} and z^*_{min} belong to l_1-th contour of CGB, and the node $\boldsymbol{x}_m$ which completes these units to a quadrangle belongs to l_2-th contour ($l_1 \neq l_2$). In this case the number of connectivity is decreased by one, because a junction of two contours into one takes place. The contours

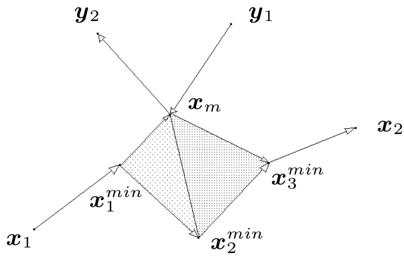


Fig. 21: The number of connectivity is decreased by 1.

$\boldsymbol{y}_1,\ \boldsymbol{x}_m,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{y}_1$ and $\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_2^{min},\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1$ have formed the contour

$$\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_m,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{y}_1,\ \boldsymbol{x}_m,\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_1.$$

2. The units z_{min} and z^*_{min} and the node $\boldsymbol{x}_m$ belong to the same contour of CGB. In this case the number of connectivity can increase by one or remain the same.

Increase of connectivity by one takes place if the node $\boldsymbol{x}_m$ does not form an unit of CGB with one of the nodes $\boldsymbol{x}_1^{min}$ or $\boldsymbol{x}_3^{min}$. Transformation of the array M is performed according to the figure 21 given above, but in this case the contour

$$\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_2^{min},\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{y}_1,\ \boldsymbol{x}_m,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{x}_1$$

is divided into two contours $\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_m,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{x}_1$ and $\boldsymbol{x}_m,\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{y}_1,\ \boldsymbol{x}_m$.

The number of connectivity remains the same if the node $\boldsymbol{x}_m$ forms an unit with one of the nodes $\boldsymbol{x}_1^{min}$ or $\boldsymbol{x}_3^{min}$ (in the figure 22 it is $\boldsymbol{x}_3^{min}$). The contour

$$\boldsymbol{x}_1,\ \boldsymbol{x}_1^{min},\ \boldsymbol{x}_2^{min},\ \boldsymbol{x}_3^{min},\ \boldsymbol{x}_m,\ \boldsymbol{y}_1,\ \ldots,\ \boldsymbol{x}_1$$

turns into a new contour $\boldsymbol{x}_1$, $\boldsymbol{x}_1^{min}$, $\boldsymbol{x}_m$, $\boldsymbol{y}_1$, $\ldots$, $\boldsymbol{x}_1$.

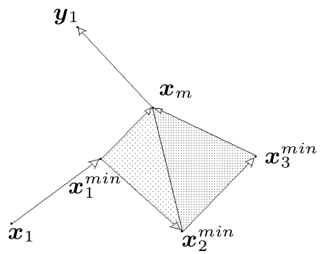


Fig. 22: The number of connectivity either increase by 1 or remain the same.

Note that in the previous Appendix similar case was not taken into consideration (when $\boldsymbol{y}_*$ forms an unit with $\boldsymbol{x}^1_{min}$ or $\boldsymbol{x}^2_{min}$ under $l_1 = l_2$), since it is eliminated by the algorithm.

References

1. Kamel Kh.A., Eisenshtein G.K.: *Automated construction of grid in two- and three- dimensional composite domains.* In: Calculation of elastic constructions by means of computer, Leningrad, 1974, pp. 21–35 (In Russian).
2. Kvitka A.L., Voroshko P.P., Bobritskaya S.D.: *Strained and deformed state of rotational bodies.* Kiev, 1977 (In Russian).
3. Umansky S.E.: *An algorithm and a program for triangulation of two-dimensional domain of arbitrary shape.* Problems of durability, 1978, № 6, pp. 83–87 (In Russian).
4. Milkova N.I.: *Peculiaries of discretization of a domain for solution of the problems of stress concentration by the method of finite elements.* Mashinovedenie, 1979, № 2, pp. 67–71 (In Russian).
5. Sakalo V.I., Shkurin A.A.: *An universal program for triangulation of dwo-dimensional domain of arbitrary shape with concentrations of the grid.* Problems of durability, 1985, № 1, pp. 106–108(In Russian).
6. George A., Lu G.: *Numerical solution of large sparse systems of equations.* Moscow, 1984 (In Russian).
7. Zlatnev Z., Esterbu O.: *Direct methods for sparse matrices.* Moscow, 1987 (In Russian).
8. Pissanetsky S.: *Technology of sparse matrices.* Moscow, 1988 (In Russian).
9. Tewarson R.P.: *Sparse Matrices.* Academic Press, New York, 1973.
10. Stewart G.W.: *Introduction to Matrix Computations.* Academic Press, New York, 1973.
11. Bunch J.R., Hopcroft J.E.: *Triangular factorization and inversion by fast matrix multiplication.* Math. Comput., 1974, 28, pp. 231–236.

12. Bunch J.R., Rose D.J.: *Sparse Matrix Computations.* Academic Press, New York, 1976.
13. Duff I.S.: *Sparse Matrices and their Uses.* Proceedings of the IMA Conference, University of Reading, 9-11 July, 1 980. Academic Press, London.
14. Gustavson F.G.: *An efficient algorithm to perform sparse matrix multiplication.* Argonne Conference on Sparse Matrix Computations, Argonne National Laboratory, 1976, Research Report RC-6176, IBM T.J. Watson Research Center, New York.
15. Munksgaard N.: *Fortran subroutines for direct solution of sets of sparse and symmetric linear equations.* Report 77.05, 1977, Numerical Institute Lyngby, Denmark.
16. Willoughby R.A.: *Proceedings of the symposium on sparse matrices and their applications.* Yorktown Heights, NY, IBM Report RAI, № 11707, 1969.
17. Willoughby R.A.: *A survey of sparse matrix technology.* IBM Research Report RC 3872, 1972.
18. Duff I.S.: *The solution of nearly symmetric sparse linear equations.* In: Computing Methods in Applied Sciences and Engineering VI, Proc. 6th Int. Symp. (Versailles, 1983), Amsterdam, North Holland, 1984, pp. 57–74.