On the Provable Tight Approximation of Optimal Meshing for Non-Convex Regions

Dmytro Chibisov, Victor Ganzha, Ernst W. Mayr[1], Evgenii V. Vorozhtsov[2]

[1] Institute of Informatics, Technical University of Munich, Garching 85748, Boltzmannstr. 3, Germany;
chibisov@in.tum.de,ganzha@in.tum.de,mayr@in.tum.de
[2] Institute of Theoretical and Applied Mechanics, Russian Academy of Sciences, Novosibirsk 630090, Russia; vorozh@itam.nsc.ru

Abstract. Automatic generation of smooth, non-overlapping meshes on arbitrary regions is the well-known problem. Considered as optimization task the problem may be reduced to finding a minimizer of the weighted combination of so-called length, area, and orthogonality functionals. Unfortunately, it has been shown that on the one hand, certain weights of the individual functionals do not admit the unique optimizer on certain geometric domains. On the other hand, some combinations of these functionals lead to the lack of ellipticity of corresponding Euler-Lagrange equations, and finding the optimal grid becomes computationally too expensive for practical applications. Choosing the right functional for the particular geometric domain of interest may improve the grid generation very much, but choosing the functional parameters is usually done in the trial and error way and depends very much on the geometric domain. This makes the automatic and robust grid generation impossible. Thus, in the present paper we consider the way to compute certain approximations of minimizer of grid functionals independently of the particular domain. Namely, we are looking for the approximation of the minimizer of the individual grid functionals in the local sense. This means the functional has to be satisfied on the possible largest parts of the domain. In particular, we shall show that the so called method of envelopes, otherwise called the method of rolling circle, that has been proposed in our previous paper, guarantees the optimality with respect to the area and orthogonality functionals in this local sense. In the global sense, the grids computed with the aid of envelopes, can be considered as approximations of the optimal solution. We will give the comparison of the method of envelopes with well established Winslow generator by presenting computational results on selected domains with different mesh size.

1 Motivation and Introduction

Advanced computer technologies and parallel architectures allow one to solve time dependent problems with 10^9 and more unknowns on rectangular regions in realistic time using hierarchical and adaptive approaches [2, 8]. In order to handle problems of such order of computational complexity on arbitrary regions and,

in particular, with moving boundaries, we are interested to have efficient grid generation techniques, which would support hierarchical approach to computing and provide the possibility of adaptive mesh refinement as well as remeshing, due to the changes of boundaries, with minimal computational costs.

The problem of grid generation on an arbitrary region Ω in the (x, y) plane can be solved by giving a map $x(\xi, \eta), y(\xi, \eta)$ from the unit square in the plane (ξ, η) onto the Ω. By choosing a uniform grid (ξ_i, η_j) in the unit square, the map $x(\xi_i, \eta_j), y(\xi_i, \eta_j)$ would transform the grid (ξ_i, η_j) to the region of interest. The required map may be computed in a number of ways. The variational grid generation is one of the most established approaches for this purpose, due to high quality of resulting grids. It provides the possibility to control the grid properties by choosing appropriate grid functionals to be minimized. The basic functionals are Length (I_L), Area (I_A), and Orthogonality (I_O) functionals, which can be written in the form (see [6]):

$$I_L(x,y) = 1/2 \iint ({x_\xi}^2 + {y_\xi}^2 + {x_\eta}^2 + {y_\eta}^2) d\xi \, d\eta; \tag{1}$$

$$I_A(x,y) = 1/2 \iint ({x_\xi}^2 {y_\eta}^2 + {y_\xi}^2 {x_\eta}^2 - 2\, x_\xi x_\eta y_\xi y_\eta) d\xi \, d\eta; \tag{2}$$

$$I_O(x,y) = 1/2 \iint ({x_\xi}^2 {x_\eta}^2 + 2\, x_\xi x_\eta y_\xi y_\eta + {y_\xi}^2 {y_\eta}^2) d\xi \, d\eta. \tag{3}$$

The map $x(\xi, \eta), y(\xi, \eta)$ minimizing each of above functionals can be found by by solving corresponding Euler–Lagrange equations, which can be written in general form

$$\mathcal{T}_{1,1}\mathbf{x}_{\xi,\xi} + \mathcal{T}_{1,2}\mathbf{x}_{\xi,\eta} + \mathcal{T}_{2,2}\mathbf{x}_{\eta,\eta} + \mathcal{S} = 0,$$

where $\mathcal{T}_{i,j}$ are 2 x 2 matrices and $\mathcal{S}$ is a 2 x 1 vector. The terms in $\mathcal{T}_{i,j}$ and $\mathcal{S}$ depend on the particular functional and are nonlinear in the case of Area and Orthogonality Functionals. In the case of the Length functional I_L, $T_{i,j}$ can be shown to be constant, and the Euler–Lagrange equations reduce to the simplest one:

$$x_{\xi,\xi} + x_{\eta,\eta} = 0, \qquad y_{\xi,\xi} + y_{\eta,\eta} = 0.$$

Minimizing I_L by solving above equations leads to smooth grids. However, the intersections of grid lines may occur (Fig. 1). The folding of resulting grids by using the Length functional is inadmissible for practical applications. The Area functional leads to the following Euler–Lagrange equations, which produce unfolded but, unfortunately, nonsmooth grids:

$$\begin{aligned} &x_{\xi,\xi}{y_\eta}^2 + y_\eta x_\xi y_{\xi,\eta} - y_\eta y_{\xi,\xi} x_\eta - 2\, y_\eta y_\xi x_{\xi,\eta} - y_\xi x_\xi y_{\eta,\eta} + y_\xi y_{\xi,\eta} x_\eta + {y_\xi}^2 x_{\eta,\eta} = 0, \\ &-x_\eta x_{\xi,\xi} y_\eta - 2\, x_\eta x_\xi y_{\xi,\eta} + y_{\xi,\xi}{x_\eta}^2 + x_\eta y_\xi x_{\xi,\eta} + x_\xi x_{\xi,\eta} y_\eta + {x_\xi}^2 y_{\eta,\eta} - x_\xi y_\xi x_{\eta,\eta} = 0. \end{aligned}$$

As described in [6], the further shortcoming of this method is that available numerical procedures for solving the above equations do not converge for certain domains. The Orthogonality functional produces orthogonal and sufficiently

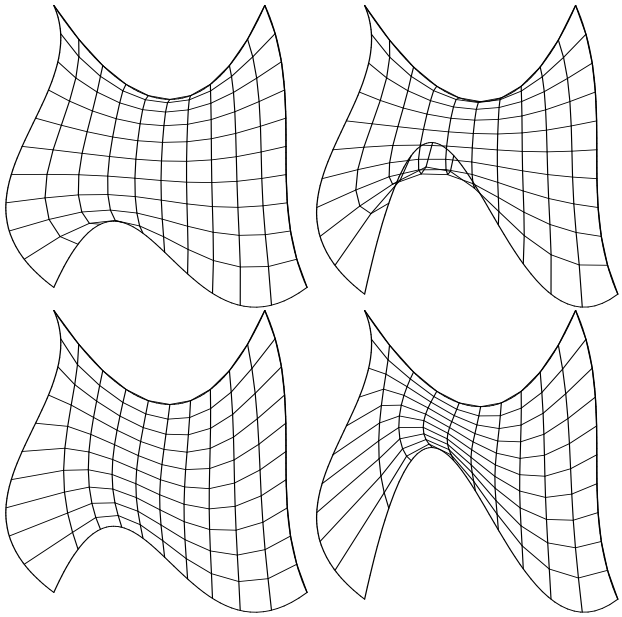

Fig. 1. Grid generation by minimizing the Length functional (top), and by minimizing the Winslow functional (bottom)

.

smooth grids on many domains, however, fails to converge in certain cases. Euler-Lagrange equations for the Orthogonality functional are:

$$x_{\xi,\xi}{x_\eta}^2 + 4\,x_\eta x_\xi x_{\xi,\eta} + x_\eta y_{\xi,\xi} y_\eta + x_\eta y_\xi y_{\xi,\eta} + 2\,x_{\xi,\eta} y_\xi y_\eta + {x_\xi}^2 x_{\eta,\eta} + x_\xi y_{\xi,\eta} y_\eta + x_\xi y_\xi y_{\eta,\eta} = 0,$$

$$y_\eta x_{\xi,\xi} x_\eta + y_\eta x_\xi x_{\xi,\eta} + y_{\xi,\xi}{y_\eta}^2 + 4\,y_\eta y_\xi y_{\xi,\eta} + 2\,y_{\xi,\eta} x_\xi x_\eta + y_\xi x_{\xi,\eta} x_\eta + y_\xi x_\xi x_{\eta,\eta} + {y_\xi}^2 y_{\eta,\eta} = 0.$$

In order to obtain smooth, orthogonal, and unfolded grids, the weighted combination of Length, Area, and Orthogonality functionals may be used:

$$I(x,y) = \omega_A I_A(x,y) + \omega_L I_L(x,y) + \omega_O I_I(x,y) \tag{4}$$

In particular, Area-Length combination overcomes the limitation of individual functionals because of avoiding grid folding produced by Length functional and producing smooth grids in contrast to the Area functional. However, the corresponding equations do not admit the continuous solution on many practically important domains like airfoil, backstep, and "C"-domains (see [6]). In order to preserve the advantages of the Length functional and avoid the grid foldings the famous Winslow grid generator has been proposed. The Winslow functional

$$I_W(x,y) = \iint \frac{{x_\eta}^2 + {y_\eta}^2}{\left(x_\xi y_\eta - x_\eta y_\xi\right)^2} + \frac{{x_\xi}^2 + {y_\xi}^2}{\left(x_\xi y_\eta - x_\eta y_\xi\right)^2} d\xi\, d\eta$$

leads to equations:

$$\left(x_\xi{}^2 + y_\eta{}^2\right) x_{\xi,\xi} - 2\left(x_\xi x_\eta + y_\xi y_\eta\right) x_{\xi,\eta} + \left(x_\xi{}^2 + x_\eta{}^2\right) x_{\eta,\eta} = 0,$$
$$\left(x_\xi{}^2 + y_\eta{}^2\right) y_{\xi,\xi} - 2\left(x_\xi x_\eta + y_\xi y_\eta\right) y_{\xi,\eta} + \left(x_\xi{}^2 + x_\eta{}^2\right) y_{\eta,\eta} = 0.$$

The Winslow generator inherits the grid smoothness from the Length functional and tends to produce smooth non-folded grids (see Fig. 1). However, the lack of orthogonality may lead, for example, to high truncation errors by using the Winslow grids for numerical solution of PDE's. Further modifications of the presented functionals may be found in the literature (see [6]), which tend to produce good meshes in certain cases and fail to admit the solution in other cases. Choosing the right functional for a certain geometric domain, or, in particular, choosing optimal weights in (4) may improve the resulting grids significantly. The optimal choosing, however, depends on the particular domain very much and is usually performed in the trial-and-error way. All this makes the automatic and robust grid generation impossible. Thus, in the present paper we study the possibility of overcoming this difficulty by considering a domain independent approach for the approximation of the minimizers of grid functional in the local sense. This means, we are interested in satisfying the corresponding Euler–Lagrange equations on the possible large part of the domain. We admit the discontinuities in the resulting mapping $x(\xi,\eta), y(\xi,\eta)$ between certain parts of the geometric domain and study the approximation of the minimizers of functionals (1) – (3) in discrete form, which will be derived in Section 2. In Section 3 we shall describe the method of envelopes, or, otherwise called the method of rolling circle, and show the quality of the approximation of the discrete optimization problem. Section 4 is devoted to the comparison of computational performance of the method of rolling circle with Winslow grid generators.

2 Variational Grid Generation: Discrete Optimization Formulation

In order to admit discontinuous mappings from the unit square onto the geometric region of interest we formulate the Area, Length, and Orthogonality conditions in the discrete sense. We consider the discretized map $x(i,j), y(i,j)$ from uniform grid in the unit square onto the arbitrary region in the (x,y) plane. As can be seen in Fig. 2, the square of the length of two grid segments (horizontal and vertical) intersecting in the common grid vertex (i,j) is given by the following polynomial:

$$L_{i,j,i-1,j-1} = (x_{i,j} - x_{i-1,j})^2 + (y_{i,j} - y_{i-1,j})^2 + (x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2 .$$

Summation over the grid vertices leads to the discrete form of the Length functional:

$$I_L(x_{1,1}, y_{1,1}, ..., x_{N,N}, y_{N,N}) = \sum_{i,j=2}^{N} (x_{i,j} - x_{i-1,j})^2 + (y_{i,j} - y_{i-1,j})^2 + (x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2 .$$

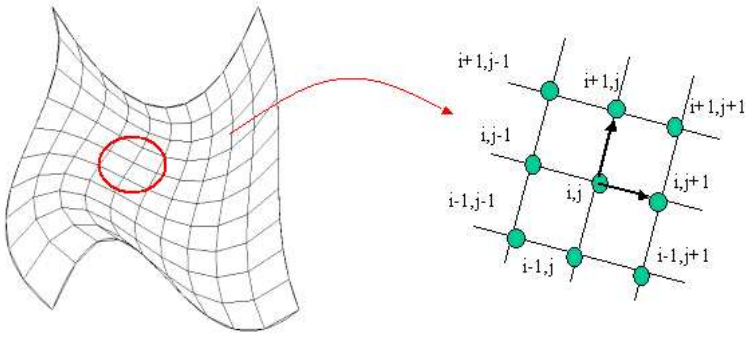

Fig. 2. Grid discrete function

The orthogonality condition between two intersecting line segments can be expressed in a similar way as polynomial by using scalar product of vectors $(x_{i+1,j} - x_{i,j}, y_{i+1,j} - y_{i,j})$ and $(x_{i,j+1} - x_{i,j}, y_{i,j+1} - y_{i,j})$:

$$O_{i,j,i+1,j+1} = ((x_{i+1,j} - x_{i,j})(x_{i,j+1} - x_{i,j}) + (y_{i+1,j} - y_{i,j})(y_{i,j+1} - y_{i,j}))^2.$$

The orthogonality condition for four angles in each cell becomes:

$$O_{i,j}^{cell} = O_{i,j,i+1,j+1} + O_{i+1,j,i-1,j+1} + O_{i+1,j+1,i-1,j-1} + O_{i,j+1,i+1,j-1}.$$

Summation over all cells then leads to the discrete orthogonality functional:

$$I_O(x_{1,1}, y_{1,1}, ..., x_{N,N}, y_{N,N}) = \sum_{i,j=1}^{N-1} O_{i,j}^{cell}.$$

The squared grid cell area can be expressed as follows

$$\begin{aligned} A_{i,j}^{cell} = &(-(x_{i,j} - x_{i,j-1})(y_{i,j-1} - y_{i+1,j-1}) + (x_{i,j-1} - x_{i+1,j-1})(y_{i,j} - y_{i,j-1}) \\ &+ (x_{i+1,j} - x_{i,j})(y_{i+1,j-1} - y_{i+1,j}) - (x_{i+1,j-1} - x_{i+1,j})(y_{i+1,j} - y_{i,j}))^2. \end{aligned}$$

Finally, the Area functional is given as

$$I_A(x_{1,1}, y_{1,1}, \ldots, x_{N,N}, y_{N,N}) = \sum_{i,j=1}^{N-1} A_{i,j}^{cell}.$$

Similarly to the Euler–Lagrange equations for the continuous Area, Length, and Orthogonality functionals, we obtain the system of $2N^2$ algebraic equations necessary for the function $I(x_{1,1}, y_{1,1}, \ldots, x_{N,N}, y_{N,N})$ to reach a minimum in $(x_{1,1}, y_{1,1}, ..., x_{N,N}, y_{N,N})$:

$$\frac{\partial I}{\partial x_{i,j}} = 0, \qquad \frac{\partial I}{\partial y_{i,j}} = 0. \tag{5}$$

Similarly to the continuous case, equations (5) for the Length functional I_L are linear

$$8\,x_{i,j} - 2\,x_{i-1,j} - 2\,x_{i,j-1} - 2\,x_{i+1,j} - 2\,x_{i,j+1} = 0,$$
$$8\,y_{i,j} - 2\,y_{i-1,j} - 2\,y_{i,j-1} - 2\,y_{i+1,j} - 2\,y_{i,j+1} = 0$$

and equations (5) for I_A and I_O are cubic and are similar to the corresponding discretized Euler–Lagrange equations.
Please note, in our approach we do not optimize the meshing with respect to conditions (5) directly. It can be shown that existing global optimization approaches like branch and bound strategy, would become computationally too expensive, especially in the case when mesh size decreases. Instead of applying computationally expensive direct optimization techniques, we use the method of envelopes, which has been introduced in [4] and will be described in Section 3. We will, namely, show that this method leads to satisfaction of (5) on the most part of the region for the Length, Area, and Orthogonality functionals.

3 Approximation of Grid Functionals by the Method of Envelopes

In the present section, we describe the so-called method of rolling circle that has been introduced in [4]. We shall show that this method produces nearly optimal grids in the sense that equations (5) are satisfied locally. Let the region $\Omega \subset R^2$ be bounded by the roots of polynomials $f_i(x, y)$. The so-called Tarski formula describing the set of points, which belong to this region can be written as follows:

$$\Omega(x, y) \equiv \bigwedge_i f_i(x, y) \geq 0$$

We propose to calculate the lines of the curvilinear grid in the following way. We contact a circle $C(x, y) = x^2 + y^2 - r^2 = 0$ with Ω and move C along the boundary of Ω keeping them in contact. The motion of a circle can be produced by shifting it by x_0, y_0 units:

$$C(x - x_0, y - y_0) = 0.$$

The circle moving along some boundary curve $f_i(x, y) = 0$ describes a curve $g_i(x, y) = 0$ called *envelope* (Fig. 3). More precisely, the envelope in our case is a curve, whose tangent at each point coincides with the tangent of a moving circle at each time of its motion. In our grid generation approach the envelopes correspond to grid lines parallel to the boundary (Fig. 5, left at the top). As will be shown in Section 3.2, connecting the intersection points of the circle C and envelope g_i on the one hand and the circle C and boundary f_i on the other hand produces the line segment which is orthogonal to both curves (Fig. 5, right at the top) and satisfies (5) for Length, Area and Orthogonality Functionals.

The contact of C and f_i can be expressed in terms of common roots of bounding polynomials. The envelope g_i corresponds also to such shifts x_0, y_0 of

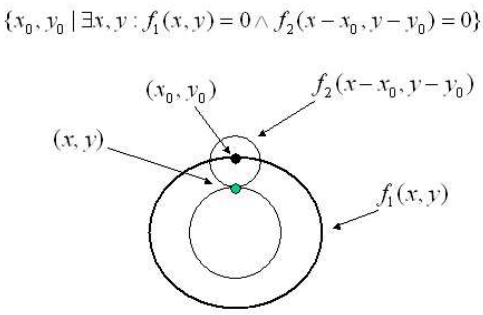

Fig. 3. Calculating of envelopes by quantifier elimination

C, where polynomials f_i and C have common roots and coinciding tangents. This can be formalized using polynomial equations as follows:

$$h : \{(x_0, y_0) | \exists x, y : f(x, y) = 0 \wedge C(x - x_0, y - y_0) = 0 \wedge \\ -\frac{\partial f(x,y)}{\partial x} \frac{\partial C(x-x_0, y_y 0)}{\partial y} + \frac{\partial f(x,y)}{\partial y} \frac{\partial C(x-x_0, y_y 0)}{\partial x} = 0\}.$$

Alternatively, if the boundary curve $f(x, y) = 0$ is given parametrically ($x = x(t), y = y(t)$), the envelope may be defined by (see [1]):

$$h : \{(x_0, y_0) | \exists t : C(x(t) - x_0, y(t) - y_0, r) = 0 \wedge \frac{\partial}{\partial t} C(x(t) - x_0, y(t) - y_0, r) = 0.$$

Eliminating $\exists$-quantifiers with existing methods described below produces the point set, which corresponds to the envelope g (Fig. 3). After $h(x, y)$ is calculated for different values of radius r of the circle, the grid points distributed along them should be connected with those of $f(x, y)$ in such a way the resulting grid satisfies equations (5) for the Length, Area, and Orthogonality functionals on the possible large part of the region. This construction will be presented in the next Section after introducing the method to calculate $h(x, y)$.

3.1 Elimination of Variables Using Resultants

In this section we shall describe how the well known approach to the elimination of variables from the following first-order formulas (so-called existential first-order theory over the reals) with the aid of resultants can be used to calculate the grid lines parallel to the boundaries. Assume the geometric region is bounded by N parametric curves $[x_{(j)}(t), y_{(j)}(t)], t \in [0, 1]$ of degree $deg \leq M$:

$$\begin{array}{ll} x_{(1)}(t) = \sum_{i=1}^{M} a_i^{(1)} t^i, & y_{(1)}(t) = \sum_{i=1}^{M} b_i^{(1)} t^i, \\ \ldots & \\ x_{(N)}(t) = \sum_{i=1}^{M} a_i^{(N)} t^i & y_{(N)}(t) = \sum_{i=1}^{M} b_i^{(N)} t^i. \end{array} \tag{6}$$

The envelope described by a circle rolling along parametric curves (6) can be described with the following formula:

$$\exists t : \bigvee_{i=1}^{N} C(x_{(i)}(t)-x_c, y_{(i)}(t)-y_c, r) = 0 \wedge \frac{\partial}{\partial t} C(x_{(i)}(t)-x_c, y_{(i)}(t)-y_c, r) = 0, \quad (7)$$

where C is the circle equation with indeterminate radius r and center position x_c, y_c on the curve $(x^{(i)}(t), y^{(i)}(t))$. Given a polynomial $f(x)$ of degree n with roots α_i and a polynomial $g(x)$ of degree m with roots β_j, the resultant is defined by

$$\rho(f,g) = \prod_{i,j} (\alpha_i - \beta_j).$$

$\rho(f,g)$ vanishes iff $\exists a : f(a) = 0 \wedge g(a) = 0$. The resultant can be computed as the determinant of the so-called Sylvester Matrix [3]. In the multivariate case, the computation of resultant can be reduced to the univariate one by considering the polynomials $f, g \in \mathbb{K}[x_1, ..., x_N]$ as univariate polynomials in $\mathbb{K}[x_1]$ with unknown coefficients in $\mathbb{K}[x_2, ..., x_N]$ (denoted by $\mathbb{K}(x_2, ..., x_N)[x_1]$). In the following we call the resultant of $f, g \in \mathbb{K}(x_1, ..., x_{i-1}, x_{i+1}, ..., x_N)[x_i]$ as $res_{x_i}(f,g)$.

In order to eliminate t from (7) and find the envelope h of the circle and the curve $(x(t), y(t))$ we may calculate

$$h(x_0, y_0, r) = res_t \left(C(x(t) - x_0, y(t) - y_0, r), \frac{\partial}{\partial t} C(x(t) - x_0, y(t) - y_0, r) \right). \quad (8)$$

In this way the first family of grid lines, namely parallel to the boundary, can be calculated in analytic form. A Maple calculation shows that the resultant for the envelope to the curve $(x(t), y(t))$ with indeterminate coefficients has already 2599 terms even if the degree of the original curve is equal to 2. This makes direct computation with resulting envelopes impossible. Bellow we shall describe the way how to avoid such computations for by the generation of grid lines perpendicular to the envelopes.

In order to generate the second family of grid lines, which are perpendicular to the first family, we discretize each of the curves (6) with step size $\Delta t = \frac{1}{M}$ and obtain a number of points $p_j = (x_j, y_j), j = 1..M$. This can be done with symbolic M by substitution of $t = \frac{j}{M}, j = 1, \ldots, N$ in (1). We place the circle center in each p_j and compute the intersection of $C(x-x_j, y-y_j)$ with $h(x,y,r)$. Let us denote the common roots of the both polynomials as $\mathbf{V}(C(x - x_j, y - y_j), h(x,y,r))$. Thus, the second family of grid lines $v(j,r)$ perpendicular to $h(x,y,r) = 0$ can be obtained by computing

$$v(j,r) = \mathbf{V}(C(x - x_j, y - y_j), h(x,y,r)). \quad (9)$$

Since $h(x,y,r)$ is a large symbolic expression, as mentioned previously, computing (9) in a direct way by elimination of variables using resultants becomes a very expensive task.

GenVertex(x_i, y_i, j, r)
(* The procedure computes grid vertex corresponding to the boundary curve $[x_i(j), y_i(j)]$ using the circle of radius r*)

(*generate new grid vertex*)
$x_i^{j,r} \leftarrow x_i(j/N) - \frac{r\frac{dy_i}{dt}|_{x_i(j/N)}}{\sqrt{\frac{dx_i}{dt}|^2_{x_i(j/N)} + \frac{dx_i}{dt}|^2_{x_i(j/N)}}}$;
$y_i^{j,r} \leftarrow y_i(j/N) + \frac{r\frac{dx_i}{dt}|_{y_i(j/N)}}{\sqrt{\frac{dx_i}{dt}|^2_{y_i(j/N)} + \frac{y_i(j/N)}{dt}|^2_{y_i(j/N)}}}$;
return $x_i^{j,r}, y_i^{j,r}$;

Fig. 4. Calculating the grid vertex corresponding to the given boundary curve and radius of the circle according to the Proposition 1

Therefore, we use the following simple result, which gives the intersection of a circle with middle point $(x_j, y_j) \in [x(t), y(t)]$ and radius r and $h(x, y, r)$ given by (8):

Proposition 1. *Let $h(x, y, r)$ be envelope of a family of circles $C(x - x(t), y - y(t), r)$ with radius r given by (8). Then for any $t_j \in \mathbb{R}$ the following is satisfied:*

$$\mathbf{V}(C(x - x(t_j), y - y(t_j), r), h(x, y, r)) =$$
$$\left(x(t_j) \pm \frac{r\frac{dy}{dt}|_{t_j}}{\sqrt{\left(\frac{dy}{dt}|_{t_j}\right)^2 + \left(\frac{dx}{dt}|_{t_j}\right)^2}}, \quad y(t_j) \mp \frac{r\frac{dx}{dt}|_{t_j}}{\sqrt{\left(\frac{dx}{dt}|_{t_j}\right)^2 + \left(\frac{dx}{dt}|_{t_j}\right)^2}} \right).$$

Proof. According to (8) $\mathbf{V}(C(x-x(t_j), y-y(t_j)), h(x, y, r)) = \mathbf{V}(C(x-x(t), y-y(t), r), \frac{\partial}{\partial t}C(x - x(t), y - y(t), r))$ for some t. Note that

$$\frac{\partial}{\partial t}C(x - x(t), y - y(t), r) = \frac{\partial C}{\partial x}\frac{dx}{dt} + \frac{\partial C}{\partial y}\frac{dy}{dt} = -2(x - x(t))\frac{dx}{dt} - 2(y - y(t))\frac{dy}{dt}.$$

This means that all solutions of $\mathbf{V}(C(x - x(t_j), y - y(t_j), r), h(x, y, r))$ lie on a line $x\frac{dx}{dt}|_{t_j} + y\frac{dy}{dt}|_{t_j} - x(t_j)\frac{dx}{dt}|_{t_j} - y(t_j)\frac{dy}{dt}|_{t_j} = 0$ independently of r. Thus, we are interested to find the intersections of circle $C(x - x_j, y - y_j, r)$ and this line going through the middle point of C. Using a bit of elementary mathematics we obtain the statement of this proposition. $\diamondsuit$

Now we are able to find the points (x_h, y_h) on envelope $h(x, y, r)$, which correspond to the particular position (x_b, y_b) of a circle on the boundary of the

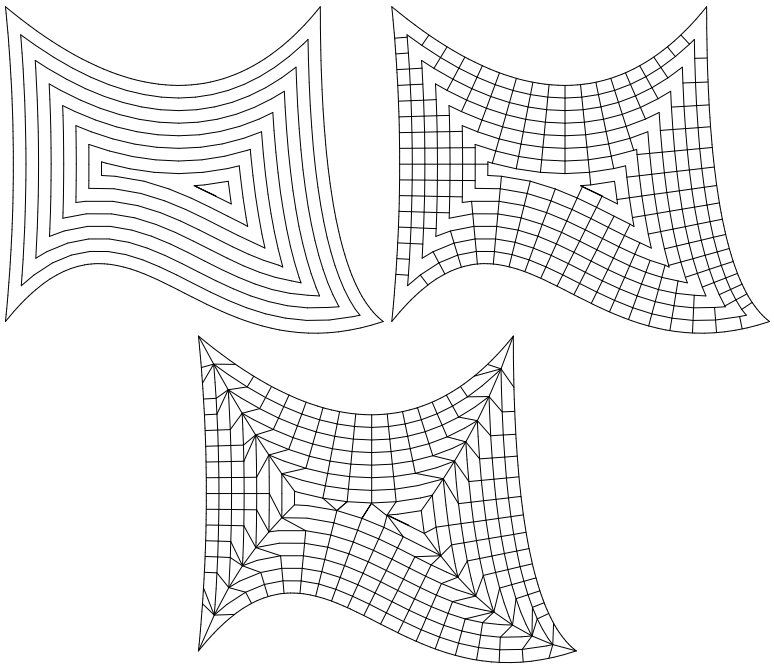

Fig. 5. Grid generation by the method of rolling circle: 1) calculating the first family of grid lines parallel to the boundary; 2) calculating grid lines orthogonal to the first family; 3) connecting "hanging" grid vertices

region. For example, when the bounding curve (1) is of degree 3 with unknown coefficients $a_1, ..., a_4, b_1, ..., b_4$ we obtain using Proposition 1:

$$\begin{aligned}
x_b &= a_1 + a_2 t_j + a_3 {t_j}^2 + a_4 {t_j}^3, \\
y_b &= b_1 + b_2 t_j + b_3 {t_j}^2 + b_4 {t_j}^3, \\
x_h &= a_1 + a_2 t_j + a_3 {t_j}^2 + a_4 {t_j}^3 + \frac{r\left(b_2 + 2\, b_3 t_j + 3\, b_4 t_j^2\right)}{\sqrt{{b_2}^2 + 4\, b_2 b_3 t_j + 4\, {b_3}^2 {t_j}^2 + {a_2}^2 + 4\, a_2 a_3 t_j + 4\, a_3{}^2 {t_j}^2}}, \\
y_h &= b_1 + b_2 t_j + b_3 {t_j}^2 + b_4 {t_j}^3 - \frac{r\left(a_2 + 2\, a_3 t_j + 3\, a_4 t_j^2\right)}{\sqrt{{b_2}^2 + 4\, b_2 b_3 t_j + 4\, {b_3}^2 {t_j}^2 + {a_2}^2 + 4\, a_2 a_3 t_j + 4\, a_3{}^2 {t_j}^2}}.
\end{aligned} \tag{10}$$

Using (10) we are able to calculate the spatial positions of the individual grid nodes lying on envelopes dependent on the distance r from the boundary (r is a radius of the circle, which produces corresponding envelope). Because of Proposition 2 (see below) the line segments induced by the calculated nodes are orthogonal. For example, the vertices of the grid shown in Figs. 5 and 6 have been generated in this way.

So far we have considered successive generation of grid cells starting from an individual boundary curve by computing two families of grid lines: perpendicular and parallel to this curve. Since the given region is bounded by several trimmed curves, it is convenient to provide a method guaranteeing that the edges of grid

GenGrid$(x_1, ..., x_n, y_1, ..., y_n)$
(* The procedure computes grid vertices for the region with a boundary given by parametric polynomials $[x_i(t), y_i(t)]$*)

(*preprocessing: calculating envelopes *)
for i **from** 1 **to** n **do**
(*calculating the envelopes with distance r from boundary*)
$h_i(x, y, r) := res_t(C(x - x_i(t), y - y_i(t), r), \frac{\partial}{\partial(t)} C(x - x_i(t), y - y_i(t), r))$
od:

(*calculating grid vertices*)
$nodes \leftarrow Empty$;
for r **from** 1 **to** m; i **from** 1 **to** n; j **from** 1 **to** N **do**
(*generate new grid vertex*)
$(x_i^{j,r}, y_i^{j,r}) \leftarrow GenVertex(x_i, y_i, j, r)$

(* check the intersection with already generated grid edges using equation of envelopes and append the new vertex to the list *nodes**)
if $h_k(x_i^{j,r}, y_i^{j,r}) \leq 0$ **for all** $k \neq i$ **then** $nodes \leftarrow (x_i^{j,r}, y_i^{j,r})$;
od:
return *nodes*;

Fig. 6. Grid generation algorithm: Calculating the grid vertices. The generated vertices may use the algorithm shown in Fig. 7

cells generated for individual curves do not intersect or even coincide in their nodes. As can easily be seen by considering, for example, Fig. 5, a certain initial distribution of points (x_b, y_b) on boundary curves could produce the coincidence of the grid nodes generated by (10) separately for each curve. However, the calculation of the boundary point distribution is computationally very expensive because of involving the solution of nonlinear equations like (10). With regard to needed CPU time this computational task can be compared with solving equations (5) themselves, which we are looking to approximate. Thus, instead of calculating the initial distribution of boundary points, we start with any given distribution. Because of possible intersection of grid edges generated by (10) in the case of arbitrary boundary points distribution, we check at each step of our algorithm the intersection. Please note, we do not need to check intersection with already calculated grid edges. Instead of it we may use the equations of the envelopes by substituting the coordinates of grid nodes (x_h, y_h) calculated using (10) in the equations $h_i(x, y) = 0$. Depending on the sign of $h_i(x_h, y_h)$ the new generated grid edge intersects already generated edges or does not intersect. In this way the grid vertices shown in Fig. 5 on the top, right are produced. The description of this part of the algorithm is given in Figs. 4 and 6 The arising "hanging" nodes are eliminated by connecting them to the closest nodes using

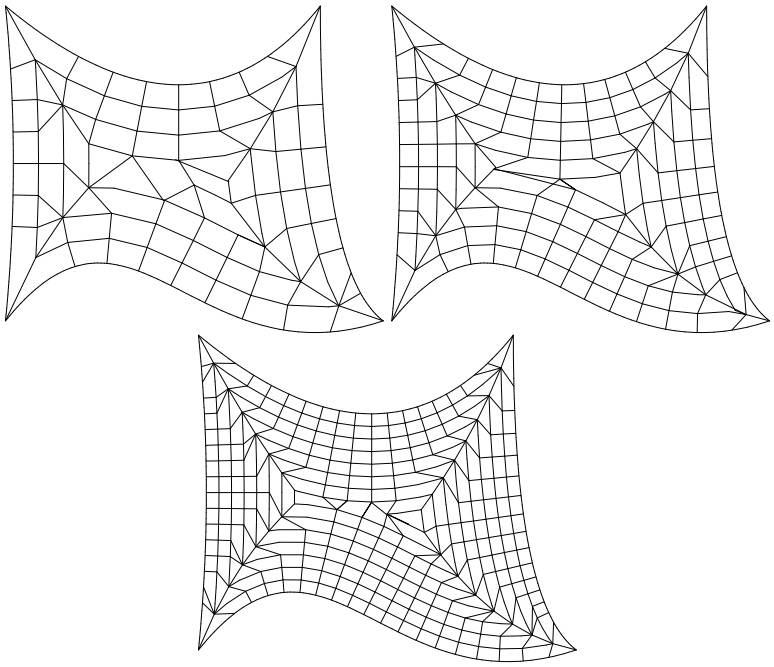

Fig. 7. Grids with different mesh size

the algorithm shown in Fig. 9. The resulting grid is depicted in Fig. 5, at the bottom.
As can be seen the grid edges induced by grid vertices generated by the algorithm shown in Figs. 4, 6, and 9 are orthogonal to the boundary curves since the grid vertices lye on the normals to the boundary curves (Proposition 1). In the same way it can easily be shown that most grid edges induced by (10) are orthogonal to all the envelopes:

Proposition 2. *Let $h(x, y, r)$ be envelope of a family of circles $C(x - x(t), y - y(t), r)$ with radius r given by (8). Then for any $t_j \in \mathbb{R}$ the line segment given by P_1, $P_2 \in \mathbb{R}^2$, where $P_1 = (x(t_j), y(t_j))$ and P_2 lies on the envelope $h(x, y, r)$ and is given by*

$$P_2 = \left(x(t_j) \pm \frac{r\frac{dy}{dt}|_{t_j}}{\sqrt{\left(\frac{dy}{dt}|_{t_j}\right)^2 + \left(\frac{dx}{dt}|_{t_j}\right)^2}}, \quad y(t_j) \mp \frac{r\frac{dx}{dt}|_{t_j}}{\sqrt{\left(\frac{dx}{dt}|_{t_j}\right)^2 + \left(\frac{dx}{dt}|_{t_j}\right)^2}} \right)$$

and intersects $h(x, y, r)$ orthogonally.

Proof. By the result of Proposition 1 the envelope $h(x, y, r) = 0$ may be parameterized by

$$\left(x - \frac{ry_t}{\sqrt{{x_t}^2 + {y_t}^2}}, y + \frac{rx_t}{\sqrt{{x_t}^2 + {y_t}^2}} \right).$$

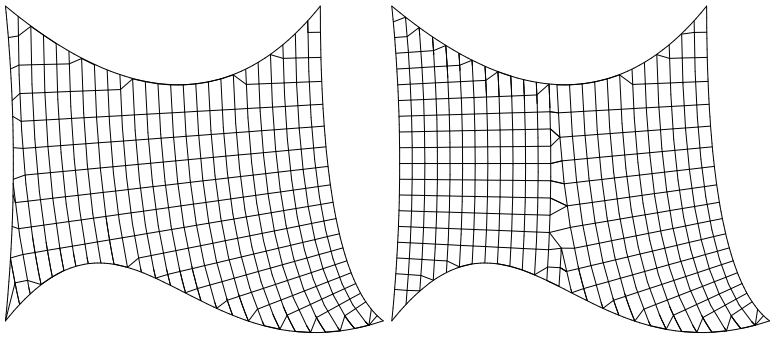

Fig. 8. Grids generated by rolling a circle along one boundary curve (left) and along two boundary curves (right)

Differentiation yields the tangent vector $\mathbf{t}_h$ to the envelope:

$$\mathbf{t}_h = \left(x_t - \frac{r y_{t,t}}{\sqrt{{x_t}^2 + {y_t}^2}} + 1/2 \, \frac{r y_t \left(2 \, x_t x_{t,t} + 2 \, y_t y_{t,t}\right)}{\left({x_t}^2 + {y_t}^2\right)^{3/2}}, \right.$$
$$\left. y_t + \frac{r x_{t,t}}{\sqrt{{x_t}^2 + {y_t}^2}} - 1/2 \, \frac{r x_t \left(2 \, x_t x_{t,t} + 2 \, y_t y_{t,t}\right)}{\left({x_t}^2 + {y_t}^2\right)^{3/2}} \right) .$$

Then the inner product of vectors $P_2 - P_1$ and $\mathbf{t}_h$ is given by the following expression. After reducing the expression to the common denominator, we obtain the inner product to be equal to 0:

$$- \left(x_t - \frac{r y_{t,t}}{\sqrt{{x_t}^2 + {y_t}^2}} + 1/2 \, \frac{r y_t \left(2 \, x_t x_{t,t} + 2 \, y_t y_{t,t}\right)}{\left({x_t}^2 + {y_t}^2\right)^{3/2}} \right) r y_t \frac{1}{\sqrt{{x_t}^2 + {y_t}^2}}$$
$$+ \left(y_t + \frac{r x_{t,t}}{\sqrt{{x_t}^2 + {y_t}^2}} - 1/2 \, \frac{r x_t \left(2 \, x_t x_{t,t} + 2 \, y_t y_{t,t}\right)}{\left({x_t}^2 + {y_t}^2\right)^{3/2}} \right) r x_t \frac{1}{\sqrt{{x_t}^2 + {y_t}^2}}$$
$$= \frac{r^2 y_t y_{t,t}}{{x_t}^2 + {y_t}^2} - \frac{r^2 {y_t}^2 x_t x_{t,t}}{\left({x_t}^2 + {y_t}^2\right)^2} - \frac{r^2 {y_t}^3 y_{t,t}}{\left({x_t}^2 + {y_t}^2\right)^2} + \frac{r^2 x_t x_{t,t}}{{x_t}^2 + {y_t}^2}$$
$$- \frac{r^2 {x_t}^3 x_{t,t}}{\left({x_t}^2 + {y_t}^2\right)^2} - \frac{r^2 {x_t}^2 y_t y_{t,t}}{\left({x_t}^2 + {y_t}^2\right)^2} = 0.$$

In this way, we have shown that grid edges generated by our algorithm are orthogonal to boundary as well as to envelopes. $\diamondsuit$

3.2 Satisfaction of the Local Optimality Conditions

In the previous sections we have described the calculation of the grid nodes starting from each boundary curve. The algorithm shown in Fig. 6 generates grid nodes iteratively till no more grid nodes can be generated because of intersections. As can be seen in Fig. 5, on the top, right, the hanging nodes are produced. At the next step we connect the hanging nodes in such a way as

ConnectVertices$(x_1, ..., x_n, y_1, ..., y_n)$
(* The procedure connects grid vertices computed by *GenGrid* *)

(*calculating intersections of envelopes, which correspond to the adjacent boundary curves*)
for all i,j,r if $adjacent(h_i, h_j)$ **do**
$intersections[i, j, r] \leftarrow solve(h_i(x, y, r) = 0, h_j(x, y, r) = 0)$
od:

$line\_segments \leftarrow Empty$;
(*calculating grid line segments*)
(*connect grid vertices corresponding to the particular position on the boundary curve and the radius of the circle*)
for all r, i, j **do**

(*calculating grid line segments parallel to the boundary: if both vertices could be generated, then connect them*)
if $(x_i^{j-1,r}, y_i^{j-1,r})$, $(x_i^{j,r}, y_i^{j,r}) \in nodes$ **then**
$line\_segments \leftarrow [(x_i^{j-1,r}, y_i^{j-1,r}), (x_i^{j,r}, y_i^{j,r})]$

(*if the left vertex could not be generated, then connect the right vertex to the intersection of envelopes on the left hand side*)
if $(x_i^{j-1,r}, y_i^{j-1,r}) \notin nodes$ **then**
$line\_segments \leftarrow [intersections[i, i-1, r+1], (x_i^{j,r}, y_i^{j,r})]$

(*if the right vertex could not be generated, then connect the left vertex to the intersction of envelopes on the right hand side*)
if $(x_i^{j,r}, y_i^{j,r}) \notin nodes$ **then**
$line\_segments \leftarrow [intersections[i, i+1, r+1], (x_i^{j-1,r}, y_i^{j-1,r})]$

(*calculating grid line segments perpendicular to the boundary: if both vertices could be generated, then connect them*)
if $(x_i^{j,r}, y_i^{j,r})$, $(x_i^{j,r+1}, y_i^{j,r+1}) \in nodes$ **then**
$line\_segments \leftarrow [(x_i^{j,r}, y_i^{j,r}), (x_i^{j,r+1}, y_i^{j,r+1})]$
od:

(*if the top vertex could not be generated, then connect the bottom vertex to the closest intersection point of envelopes*)
if $(x_i^{j,r+1}, y_i^{j,r+1}) \notin nodes$ **then**
$line\_segments \leftarrow [find\_closest\_intersection(), (x_i^{j,r}, y_i^{j,r})]$

return $line\_segments$;

Fig. 9. Grid generation algorithm: Connecting the grid vertices

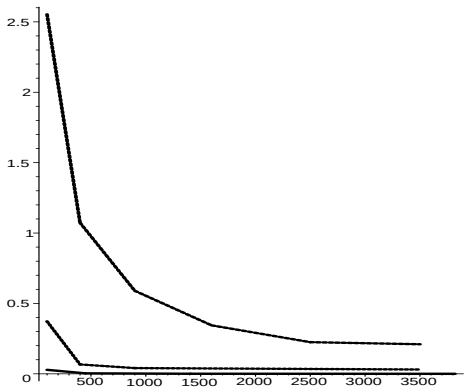

Fig. 10. Values of the Orthogonality Functional I_O in dependence on the number of mesh nodes for the Winslow Generator (dash-dot), the Length Generator (dot), and, the method of envelopes (solid)

.

to obtain the valid meshing. In this section we shall consider our method with respect to local minimization of Orthogonality and Length functionals. As has been mentioned in the Introduction, we are interested in satisfying the corresponding Euler–Lagrange equations locally. This means the following. Let us fix the hanging nodes and consider above functionals for all the nodes in between. First, consider the Orthogonality functional:

Proposition 3. *The grid generated by*

$$\begin{aligned} x(\xi,\eta) &= x(\xi) \pm \frac{\eta \frac{dy}{d\xi}}{\sqrt{\left(\frac{dx}{d\xi}\right)^2+\left(\frac{dy}{d\xi}\right)^2}};\\ y(\xi,\eta) &= y(\xi) \mp \frac{\eta \frac{dx}{d\xi}}{\sqrt{\left(\frac{dx}{d\xi}\right)^2+\left(\frac{dy}{d\xi}\right)^2}}. \end{aligned} \tag{11}$$

minimizes I_O.

Proof. Substituting (11) into each of Euler–Lagrange Equations for the Orthogonality functional I_O given by

$$\begin{aligned} x_{\xi,\xi}{x_\eta}^2 + 4\,x_\eta x_\xi x_{\xi,\eta} + x_\eta y_{\xi,\xi} y_\eta + x_\eta y_\xi y_{\xi,\eta} + 2\,x_{\xi,\eta} y_\xi y_\eta + {x_\xi}^2 x_{\eta,\eta} + x_\xi y_{\xi,\eta} y_\eta \\ + x_\xi y_\xi y_{\eta,\eta} = 0, \\ y_\eta x_{\xi,\xi} x_\eta + y_\eta x_\xi x_{\xi,\eta} + y_{\xi,\xi}{y_\eta}^2 + 4\,y_\eta y_\xi y_{\xi,\eta} + 2\,y_{\xi,\eta} x_\xi x_\eta + y_\xi x_{\xi,\eta} x_\eta + y_\xi x_\xi x_{\eta,\eta} \\ + {y_\xi}^2 y_{\eta,\eta} = 0, \end{aligned}$$

we obtain with the aid of Maple a large differential expression containing 27 terms. After reducing this expression to the common denominator we obtain the expression equal to 0:

$$-8\,\frac{\eta\,{y_{\xi,\xi}}^2 y_\xi}{({x_\xi}^2+{y_\xi}^2)^{3/2}} - 2\,\frac{{x_\xi}^4 x_{\xi,\xi}}{({x_\xi}^2+{y_\xi}^2)^2} + 2\,\frac{\eta\,{y_\xi}^3 x_\xi x_{\xi,\xi,\xi}}{({x_\xi}^2+{y_\xi}^2)^{5/2}} + \cdots$$

$$2\,\frac{\eta\, y_{\xi,\xi,\xi} {x_\xi}^2 {y_\xi}^2}{({x_\xi}^2+{y_\xi}^2)^{5/2}} + 6\,\frac{\eta\, y_{\xi,\xi} {x_\xi}^3 x_{\xi,\xi}}{({x_\xi}^2+{y_\xi}^2)^{5/2}} + 2\,\frac{{x_\xi}^2 x_{\xi,\xi}}{{x_\xi}^2+{y_\xi}^2} = 0.$$

The Euler–Lagrange equations for the Orthogonality functional are also satisfied. $\diamondsuit$

In this way we have proved that grid calculated using our algorithm is locally optimal with respect to the Orthogonality functional. In the global sense the computational comparison for the Winslow generator and our method has been performed and is shown in Fig. 10.

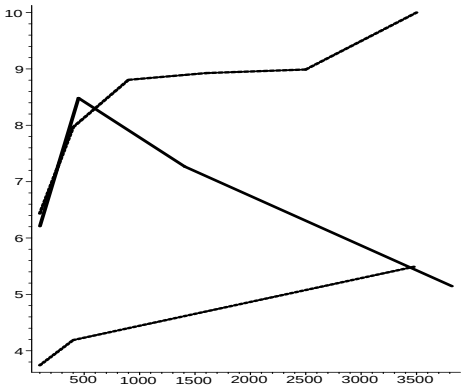

Fig. 11. Values of the Length Functional I_L^v in dependence on the number of mesh nodes for the Winslow generator (dash-dot), the Length generator (dot), and, the method of envelopes (solid)

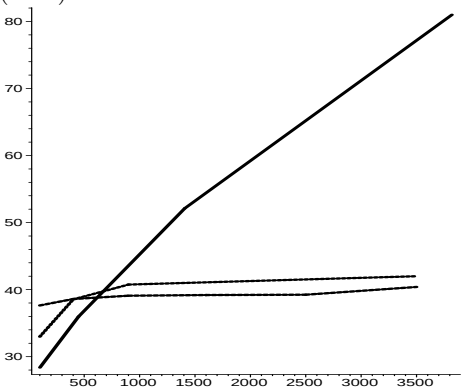

Fig. 12. Values of the Length Functional I_L^h in dependence on the number of mesh nodes for the Winslow generator (dash-dot), the Length generator (dot), and, the method of envelopes (solid)

.

Furthermore, by the construction, the method does not produce grid foldings. Since the meshing is required to cover the whole area of the domain, the method

is optimal with respect to the area. Let us consider the presented method with respect to the Length functional. Let us write the Length functional as a sum of two functionals $I_L = I_L^v + I_L^h$ corresponding to grid edges, which are perpendicular and parallel to the boundary:

$$I_L^h(x,y) = 1/2 \iint (x_\xi^2 + y_\xi^2) d\xi\, d\eta; \qquad I_L^v(x,y) = 1/2 \iint (x_\eta^2 + y_\eta^2) d\xi\, d\eta\,.$$

The following Proposition shows that our method optimizes I_L^v.

Proposition 4. *The grid generated by (11) minimizes I_L^v.*

Proof. The proof is similar to the proof of Proposition 3. Substituting (11) into the Euler–Lagrange equations corresponding to I_L^v yields the statement of the proposition. $\diamondsuit$

From the geometric point of view, the statement of Proposition 4 holds because of the following reasons. Consider the fixed boundary point $(x(t_j), y(t_j))$. By our construction (as shown in Proposition 1) increasing r produces grid nodes $(x_h(j,r), y_h(j,r))$, which lie on the straight line perpendicular to the boundary in $(x(t_j), y(t_j))$. Of course, a straight line produces the minimal length among all the curves, which may connect the points $(x_h(j,r), y_h(j,r))$. On the other hand, the statement does not hold for I_L^h, because the points $(x_h(j,r), y_h(j,r))$ by increasing j lie on the envelope, which is not the curve of minimal length between these grid vertices, in contrast to the straight line. Our method allows to minimize I_L^h as well, namely, by appropriate changing of the radius of the rolling circle in such a way as the points $(x_h(j,r), y_h(j,r))$ lie on the straight line. However, as can be easily seen from the presented considerations, minimizing I_L^h would destroy the orthogonality of grid edges. The computational comparsion for our method, the Winslow generator, and the Length generator with respect to I_L^h and I_L^v is shown in Figs. 11 and 12.

4 Computational Experiments

In order to compare the presented method of envelopes with well-established grid generation methods, we describe the numerical procedure implementing a finite-difference method for solving the Euler–Lagrange equations corresponding to the individual functionals. For this purpose we use the *Alternating Direction Implicit* (ADI) method introduced in [7].

For instance, consider the Winslow grid generator, which is based on the solution of the following system of nonlinear coupled PDE's:

$$\begin{aligned}
\left({x_\xi}^2 + {y_\eta}^2\right) x_{\xi,\xi} - 2\,\left(x_\xi x_\eta + y_\xi y_\eta\right) x_{\xi,\eta} + \left({x_\xi}^2 + {x_\eta}^2\right) x_{\eta,\eta} = 0,\\
\left({x_\xi}^2 + {y_\eta}^2\right) y_{\xi,\xi} - 2\,\left(x_\xi x_\eta + y_\xi y_\eta\right) y_{\xi,\eta} + \left({x_\xi}^2 + {x_\eta}^2\right) y_{\eta,\eta} = 0.
\end{aligned}$$

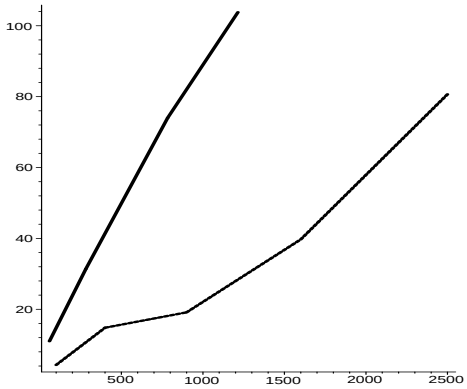

Fig. 13. CPU time versus the number of mesh nodes for the Length generator (dot) and the method of envelopes (solid)

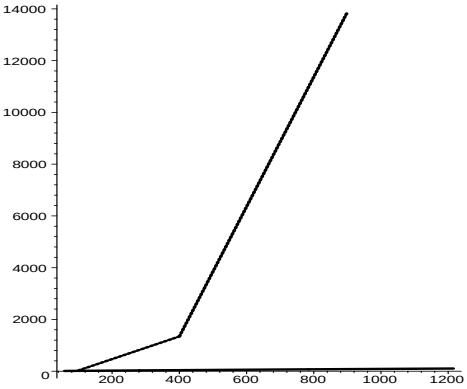

Fig. 14. CPU time versus the number of mesh nodes for the Winslow generator (dash-dot) and the method of envelopes (solid)

As described in [5], we use the following second-order approximation for the partial derivatives of the function $f(\xi, \eta)$:

$$
\begin{aligned}
(f_\xi)_{i,j} &= 1/2(f_{i+1,j} - f_{i-1,j}, \\
(f_\eta)_{i,j} &= 1/2(f_{i,j+1} - f_{i,j-1}), \quad (f_{\xi,\xi})_{i,j} = (f_{i+1,j} - 2f_{i,j} + f_{i-1,j}), \\
(f_{\eta,\eta})_{i,j} &= (f_{i,j+1} - 2f_{i,j} + f_{i,j-1}), \\
(f_{\eta,\xi})_{i,j} &= 1/4(f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1})\,.
\end{aligned}
$$

Let us introduce the following difference operators:

$$
\begin{aligned}
\Lambda^n_\xi f_{i,j} &= \left[(x^2_\eta)^n_{i,j} + (y^2_\eta)^n_{i,j}\right](f_{i+1,j} - 2f_{i,j} + f_{i-1,j})\,, \\
\Lambda^n_\eta f_{i,j} &= \left[(x^2_\xi)^n_{i,j} + (y^2_\xi)^n_{i,j}\right](f_{i,j+1} - 2f_{i,j} + f_{i,j-1})\,, \\
\Lambda^n_{\eta,\xi} f_{i,j} &= -1/2\left[(x_\xi x_\eta)^n_{i,j} + (y_\xi y_\eta)^n_{i,j}\right](f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1})\,.
\end{aligned}
$$

The superscript denotes the number of iterations. Then the ADI difference scheme, which converges to the solution of Winslow equations using pseudo-time steps τ, may be written as follows:

$$\frac{\tilde{x}_{i,j} - x^n_{i,j}}{0.5\tau} = \Lambda^n_\xi \tilde{x}_{i,j} + \Lambda^n_{\xi,\eta} x^n_{i,j} + \Lambda^n_\eta x^n_{i,j}\,,$$

$$\frac{x^{n+1}_{i,j} - \tilde{x}_{i,j}}{0.5\tau} = \Lambda^n_\xi \tilde{x}_{i,j} + \Lambda^n_{\xi,\eta} \tilde{x}_{i,j} + \Lambda^n_\eta x^{n+1}_{i,j}\,,$$

$$\frac{\tilde{y}_{i,j} - y^n_{i,j}}{0.5\tau} = \Lambda^n_\xi \tilde{y}_{i,j} + \Lambda^n_{\xi,\eta} y^n_{i,j} + \Lambda^n_\eta y^n_{i,j}\,,$$

$$\frac{y^{n+1}_{i,j} - \tilde{y}_{i,j}}{0.5\tau} = \Lambda^n_\xi \tilde{y}_{i,j} + \Lambda^n_{\xi,\eta} \tilde{y}_{i,j} + \Lambda^n_\eta y^{n+1}_{i,j}\,.$$

Using this scheme, for example, the grid in Fig. 1 has been obtained. In Figs. 10, 11, and 12 the values of the discrete Length and Orthogonality functionals, which have been introduced in Section 2, are compared for the Winslow Generator and our method of envelopes for different sizes of the mesh. We have used the region shown in Fig. 5. The proposed method of envelopes proves to be efficient in terms of the CPU time needed for its computer implementation. As shown in Figs. 13 and 14, increasing mesh size produces almost linear time growth in the case of our method, whereas the CPU time for the Winslow generator tends to grow much faster.

5 Conclusion

In the present paper we have presented a method of envelopes, called otherwise the method of rolling circle, which allows us to obtain the tight approximation of optimal meshing on nonconvex regions. As has been shown, Area and Orthogonality functionals, which have to be minimized in order to obtain optimal unfolded meshing are minimized by the method of envelopes in the local sense. This means, the corresponding Euler–Lagrange equations are satisfied locally. For the split Length functional (in the sense of Proposition 4) one part is completely minimized, and the value of another part depends on the curvature of the boundary. Increasing curvature leads to the worse approximation of the minimizer. Minimizing the second part of Length functional is also possible using our method. However, as can easily be derived from the presented consideration, minimizing the second part of the split Length functional would destroy the orthogonality. The obtained results show the tight approximation of the minimizers of individual mesh functionals by the method of envelopes. The main advantage of the method is its algorithmic simplicity and efficiency. Furthermore, the method is domain independent and can be applied to domains for which many classical iterative procedures do not converge, or require the manual choosing of weights of individual functionals.

References

1. Bruce, J.W., Giblin, P.J.: Curves and Singularities, Cambridge University Press, 1984
2. Bungartz, H.-J.: Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. Dissertation, Institut für Informatik, Technische Universitait München, 1992
3. Cox, D.A., Little, J.B., O'Shea, D.: Ideals, Varieties, and Algorithms, Springer-Verlag, Berlin, 1996
4. Chibisov, D., Ganzha, V. G., Mayr, E. W., Vorozhtsov, E.V.: Generation of orthogonal grids on curvilinear trimmed regions in constant time. In: Proc. CASC'2005, LNCS 3718, Springer-Verlag, Berlin, Heidelberg, 2005, 105–114
5. Ganzha, V. G., Vorozhtsov, E.V.: Numerical Solution for Partial Differential Equations: Problem Solving Using *Mathematica*, CRC Press, Boca Raton, Ann Arbor, 1996
6. Knupp, P., Steinberg, S.: Fundamentals of Grid Generation. CRC Press, Boca Raton, Ann Arbor, 1994
7. Peaceman, D. W. and Rachford, H.H., jr.: The numerical solution of parabolic and elliptic differential equations, J. of SIAM **3**(1955) 28–41
8. Zenger, C.: Sparse grids. In: Parallel Algorithms for Partial Differential Equations, Proc. Sixth GAMM-Seminar, Kiel, 1990, Hackbusch, W. (ed.), Vol. 31 of Notes on Num. Fluid Mech. Vieweg-Verlag, Braunschweig/ Wiesbaden (1991) 241–251