

The impact of the Dirichlet boundary conditions on the convergence of the discretized system of nonlinear equations for potential problems

Jan Kucwaj

Institute of Computer Science

Faculty of Physics, Mathematics and Computer Science

Cracow University of Technology

Warszawska 24, 31-155 Kraków, Poland

e-mail: jkucwaj@pk.edu.pl

The purpose of this paper is the analysis of numerical approaches obtained by describing the Dirichlet boundary conditions on different connected components of the computational domain boundary for potential flow, provided that the domain is a rectangle. The considered problem is a potential flow around an airfoil profile. It is shown that in the case of a rectangular computational domain with two sides perpendicular to the speed direction, the potential function is constant on the connected components of these sides. This allows to state the Dirichlet conditions on the considered parts of the boundary instead of the potential jump on the slice connecting the trail edge with the external boundary. Furthermore, the adaptive remeshing method is applied to the solution of the considered problem.

Keywords: adaptation, rate of convergence, remeshing, Delaunay triangulation, finite element method, potential flow, Kutta-Joukovsky condition, Dirichlet condition.

1. INTRODUCTION

The main goal of this paper is to compare the convergence speed of the method to the solution of the potential flow problem by stating formally different, but physically equivalent mathematical formulations. The difference lies in stating the Dirichlet conditions [3, 5] on different, connected components of a part of the rectangular computational domain boundary, which is perpendicular to the speed direction. Assuming that the speed at infinity has a horizontal direction, the vertical component of the speed vanishes, which means that the speed potential is constant on the vertical component of the boundary. The value of the potential is thus chosen to satisfy the Kutta-Joukovsky condition. The choice of the connected component acts on the speed of the convergence of the secant method used to solve the nonlinear algebraic equation representing the Kutta-Joukovsky condition and the speed of the Newton-Raphson method for the nonlinear algebraic system of equations representing the finite element approximation of the considered problem. It can be mentioned that the loop over secant method is external to the loop of the solution of nonlinear system of algebraic equations representing a discretized form of the physical problem. The whole problem is led to the solution of a system of two equations, i.e., a nonlinear elliptic equation of the second order and nonlinear algebraic equation of the so-called the Kutta-Joukovsky condition [1, 15]. The parameters defining the nonlinear algebraic equation of the Kutta-Joukovsky condition depend on the solution of the differential equation. The algebraic system of equations is solved for every step of iteration of the secant method used in the solution of Kutta-Joukovsky condition. An adaptive method based on numerical grid generator with a mesh size function [6, 7] is applied. The secant method is applied to satisfy Kutta-Joukovsky condition. Six–ten iterations of the secant method gave an error of order 10^{-9} .

2. THE CLASSICAL FORMULATION OF THE PROBLEM

It is assumed that the flow is stationary, irrotational, compressible, and inviscid in domain Ω around the profile P (Fig. 1). The following notations are used:

Γ_P – the boundary of profile P ,

Σ – the slit from \mathbf{A} to \mathbf{B} ,

Γ_∞ – external component of the boundary Ω .

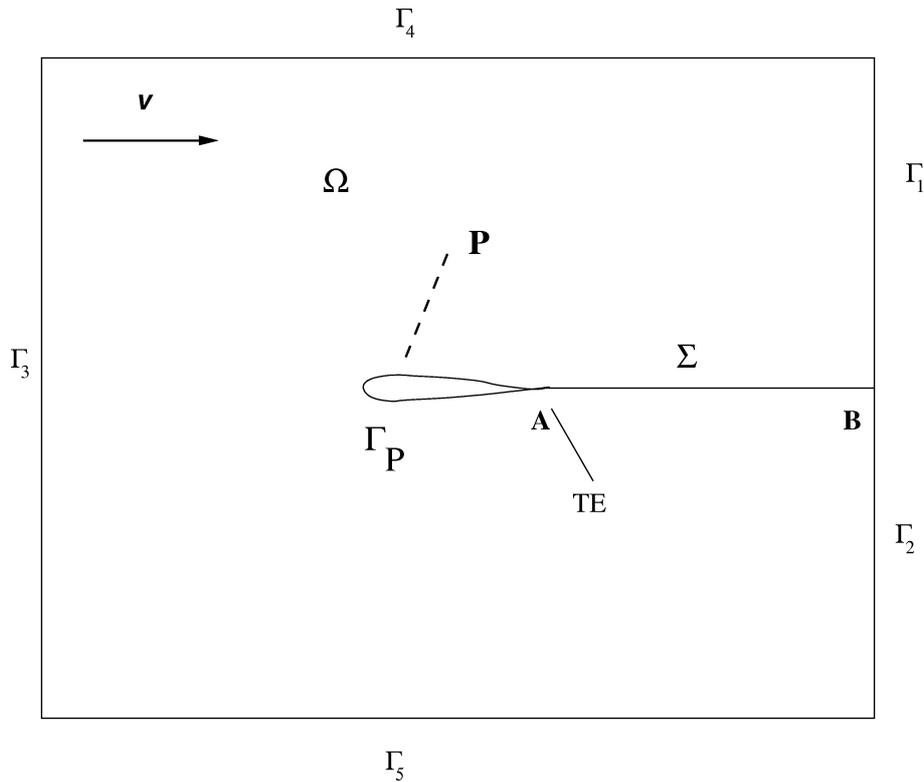


Fig. 1. Computational domain.

The boundary of Ω consists of the following parts (Fig. 1):

$$\partial\Omega = \Gamma_\infty \cup \Gamma_P, \quad \Gamma_\infty = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \cup \Gamma_5, \quad (1)$$

whose interiors do not intersect. The curve Γ_∞ is an artificially taken boundary to obtain bounded computational domain. The boundary Γ_P is the border of the contact with an obstacle, which has vertex modelling of the trail edge [1]. The unbounded domain besides Γ_∞ is introduced as

$$\Omega^c = \mathbb{R}^2 \setminus (\overline{\Omega \cup P}). \quad (2)$$

From the continuity equations the nonlinear differential equation is obtained [1]

$$\operatorname{div}[\rho(|\nabla\phi|^2)\nabla\phi] = 0 \quad \text{in } \Omega, \quad (3)$$

where ϕ is the potential of the speed

$$\mathbf{v} = \nabla\phi \quad \text{and} \quad \rho(|\nabla\phi|^2) = \rho_0 \left(1 - \frac{\kappa - 1}{2a_0^2} |\nabla\phi|^2 \right)^{\frac{1}{\kappa - 1}}$$

is the gas density.

Here $\kappa > 1$ is the adiabatic gas constant, e.g., $\kappa = 1.4$ for dry air. The constants ρ_0 , a_0 are the density and the local speed of sound, respectively, for the motionless gas.

Classical boundary conditions

The boundary conditions are as follows [1]:

$$\frac{\partial \phi^+}{\partial \mathbf{n}} - \frac{\partial \phi^-}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \Sigma \quad (4)$$

where ϕ^+ and ϕ^- are the values of ϕ over upper and lower part on the slit respectively and

$$\phi^+ - \phi^- = \beta \quad (5)$$

for some jump β on Σ (Fig. 2),

$$\rho(|\nabla \phi|^2) \frac{\partial \phi}{\partial \mathbf{n}} = \rho(|\mathbf{v}_\infty|^2) \mathbf{v}_\infty \mathbf{n}_\infty \quad \text{on} \quad \Gamma_\infty, \quad (6)$$

where \mathbf{n}_∞ is the external normal to Γ_∞ , and $\rho_\infty = \rho(|\mathbf{v}_\infty|)$.

To determine the jump β , we need an additional condition the so-called Kutta-Joukovsky condition:

$$K(\beta) = |\nabla \phi^+|_{\mathbf{A}}^2 - |\nabla \phi^-|_{\mathbf{A}}^2 = 0, \quad (7)$$

at the trail edge.

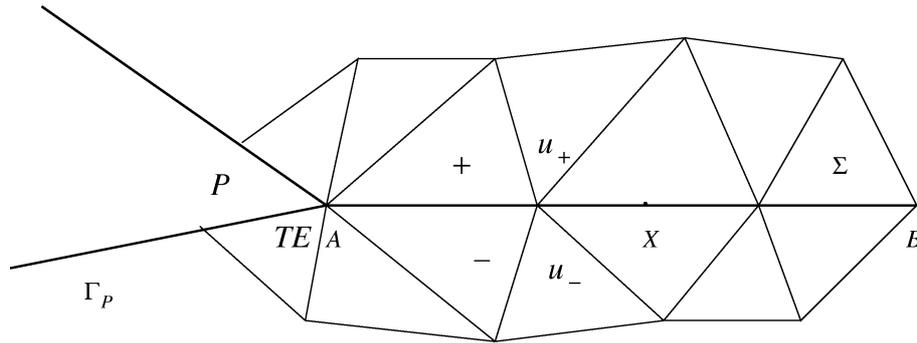


Fig. 2. The slit from trail edge to the domain boundary.

Weak formulation

For the weak formulation, the following functional spaces are introduced [1, 8]:

$$W^{1,p}(\Omega) = \left\{ v \in L^p(\Omega), \frac{\partial v}{\partial x_i} \in L^p(\Omega), \quad i = 1, 2 \right\}, \quad (8)$$

$$W^{1,p}(\dot{\Omega}) = \left\{ v \in L^p(\dot{\Omega}), \frac{\partial v}{\partial x_i} \in L^p(\dot{\Omega}), \quad i = 1, 2 \right\}, \quad (9)$$

$$W^{1,\infty}(\dot{\Omega}) = \left\{ v \in L^\infty(\dot{\Omega}), \frac{\partial v}{\partial x_i} \in L^\infty(\dot{\Omega}), \quad i = 1, 2 \right\}. \quad (10)$$

Multiplying (3) by an arbitrary trial function ψ and, integrating by parts and taking into account the Neumann boundary conditions the following weak formulation is obtained [1, 8]:

$$\int_{\Omega} \int \rho(|\nabla \phi|) \nabla \phi \nabla \psi dx dy = \int_{\Gamma} h v d\Gamma, \quad \psi \in W^{1,\infty}(\dot{\Omega}), \quad (11)$$

$$\phi^+ - \phi^- = \beta \quad \text{for some jump } \beta \quad \text{on } \Sigma, \quad (12)$$

$$|\nabla\phi^+|_{\mathbf{A}}^2 - |\nabla\phi^-|_{\mathbf{A}}^2 = 0, \quad (13)$$

where

$$h = \rho_\infty \phi_\infty \mathbf{n}_\infty \quad \text{on } \Gamma_\infty. \quad (14)$$

Here, it can be mentioned that in the weak formulation only the conditions (12) and (13) must be taken into account. The conditions (4) and (6) are taken into account in the integration by part, and in a numerical solution they are satisfied in an approximate way [14].

The weak formulation (11)–(14) can be equivalently formulated as a search for the extremum of the following functional:

$$I(\phi) = \frac{1}{2} \int_{\Omega} \int R(|\nabla\phi|) dx dy - \int_{\Gamma} hvd\Gamma, \quad (15)$$

where

$$R(s) = \int_0^s \rho(s) ds. \quad (16)$$

Equation (3) is directly obtained from Euler's equations of variational calculus applied to (15). By the substitution $F(x, y, \phi, \phi_x, \phi_y) = \frac{1}{2}R(|\nabla\phi|^2)$, the following is obtained:

$$I(\phi) = \int_{\Omega} F(x, y, \phi, \phi_x, \phi_y) d\Omega - \int_{\Gamma} hvd\Gamma, \quad (17)$$

where $\Omega \subset \mathbb{R}^2$ is a domain of the variational problem.

3. EQUIVALENT FORMULATIONS

It is possible to formulate three equivalent problems of the potential flow besides the formulation presented in the previous section (15). All of them differ from (15) by replacing the boundary condition (5). Namely, the speed \mathbf{v} at infinity is perpendicular to the Γ_1 , Γ_2 , Γ_3 vertical parts of the computational domain. This means that the vertical components of \mathbf{v} are equal to 0, then (Fig. 1)

$$\mathbf{v}_y = \frac{\partial\phi}{\partial y} = 0 \quad \text{on } \Gamma_1 \cup \Gamma_2 \cup \Gamma_3. \quad (18)$$

From this it follows:

$$\phi = \text{constant} \quad \text{on } \Gamma_1,$$

$$\phi = \text{constant} \quad \text{on } \Gamma_2,$$

$$\phi = \text{constant} \quad \text{on } \Gamma_3.$$

Formulation I

Search for stationary point of the functional:

$$I(\phi) = \frac{1}{2} \int_{\dot{\Omega}} \int R(|\nabla\phi|) dx dy - \int_{\Gamma} hvd\Gamma, \quad (19)$$

where

$$R(s) = \int_0^s \rho(s) ds. \quad (20)$$

With the boundary condition and the Kutta-Joukovsky condition:

$$\phi = \beta \quad \text{for some } \beta \quad \text{on } \Gamma_1, \quad (21)$$

$$|\nabla\phi^+|_{\mathbf{A}}^2 - |\nabla\phi^-|_{\mathbf{A}}^2 = 0, \quad (22)$$

where

$$h = \rho_{\infty} \mathbf{v}_{\infty} \mathbf{n}_{\infty} \quad \text{on } \Gamma_{\infty}. \quad (23)$$

Formulation II

Search for stationary point of the functional:

$$I(\phi) = \frac{1}{2} \int_{\dot{\Omega}} \int R(|\nabla\phi|) dx dy - \int_{\Gamma} hvd\Gamma, \quad (24)$$

where

$$R(s) = \int_0^s \rho(s) ds. \quad (25)$$

With the boundary condition and Kutta-Joukovsky condition:

$$\phi = \beta \quad \text{for some } \beta \quad \text{on } \Gamma_2, \quad (26)$$

$$|\nabla\phi^+|_{\mathbf{A}}^2 - |\nabla\phi^-|_{\mathbf{A}}^2 = 0, \quad (27)$$

where

$$h = \rho_{\infty} \mathbf{v}_{\infty} \mathbf{n}_{\infty} \quad \text{on } \Gamma_{\infty}. \quad (28)$$

Formulation III

Search for stationary point of the functional:

$$I(\phi) = \frac{1}{2} \int_{\dot{\Omega}} \int R(|\nabla\phi|) dx dy - \int_{\Gamma} hvd\Gamma, \quad (29)$$

where

$$R(s) = \int_0^s \rho(s) ds. \quad (30)$$

With the boundary condition and the Kutta-Joukovsky condition:

$$\phi = \beta \quad \text{for some } \beta \quad \text{on } \Gamma_3, \quad (31)$$

$$|\nabla\phi^+|_{\mathbf{A}}^2 - |\nabla\phi^-|_{\mathbf{A}}^2 = 0, \quad (32)$$

where

$$h = \rho_\infty \mathbf{v}_\infty \mathbf{n}_\infty \quad \text{on } \Gamma_\infty. \quad (33)$$

All three formulations are equivalent to the classical formulation presented in Sec. 2 such that both partial derivatives are that same for all cases. It follows from that all physical conditions are satisfied in all formulations including the Kutta-Joukovsky condition (32), and in all cases the solution is unique. The potential functions usually differ from each other only by a constant.

Formal pure mathematical proof is as follows.

It is known that the solution of the classical problem exists and is unique [2]. Let C_1 be the restriction of the solution ϕ to the Γ_1 of the classical formulation (it is a constant) e.q., $\phi|_{\Gamma_1=C_1}$. Additionally, we assume that $\phi = \phi_I, \phi_{II}, \phi_{III}$ are the solutions appropriately for problems I, II, III. The restriction of ϕ_I to the Γ_1 is a constant too, e.q., $\phi_I|_{\Gamma_1=C_2}$. It can be mentioned that adding a constant to any of those four functions gives a function satisfying Eq. (3) and the Kutta-Joukovsky condition (7), because both equations depend only on partial derivatives of the solution. If this is posed for the problem I with $\phi_I|_{\Gamma_1=C_1}$, then the obtained solution $\phi^* = \phi$ for the whole domain. Additionally, $\phi^* = \phi_I + C_1 - C_2$. This follows from the uniqueness of the considered problem with the Dirichlet boundary conditions. A very similar reasoning can be provided for the solutions $\phi = \phi_I, \phi_{II}, \phi_{III}$.

4. FINITE ELEMENT METHOD DISCRETISATION

For the finite element solution to the problem, the grid \mathcal{T}_0 is generated with the given, positively defined mesh size function [9]: $\gamma_0 : \Omega \mapsto \mathbb{R}$. Then, the approximation space is defined as

$$V^0 = \left\{ v : \bigcup_{i=1}^{n_0} \overline{T}_i^0 \mapsto \mathbb{R}, \quad v \text{ continuous, } v|_{\overline{T}_i^0} \text{ is a polynomial of first order } \forall i \right\}, \quad (34)$$

where $\mathcal{T}_0 = \{T_i : i = 1, \dots, n_0\}$ is the set of non-intersecting triangles covering the domain.

Introducing the finite element basis $\{U_i\}_{i=1}^N$ in to the space V^0 the potential can be expresses as $\phi = \sum_{i=1}^N \lambda_i U_i$. The coefficients $\lambda_1, \lambda_2, \dots, \lambda_N$ are found from the extremum conditions of a function of N -variables:

$$g_k(\lambda_1, \lambda_2, \dots, \lambda_N) = \frac{\partial I \left(\sum_{i=1}^N \lambda_i U_i \right)}{\partial \lambda_k} = 0, \quad \text{for } k = 1, \dots, N. \quad (35)$$

The following vector and matrix are respectively introduced:

$$\mathbf{D}_F = \left[\frac{\partial F}{\partial \phi}, \frac{\partial F}{\partial \phi_x}, \frac{\partial F}{\partial \phi_y} \right]^T, \quad (36)$$

$$\mathbf{D}_{FF} = \begin{bmatrix} \frac{\partial^2 F}{\partial \phi \partial \phi} & \frac{\partial^2 F}{\partial \phi \partial \phi_x} & \frac{\partial^2 F}{\partial \phi \partial \phi_y} \\ \frac{\partial^2 F}{\partial \phi_x \partial \phi} & \frac{\partial^2 F}{\partial \phi_x \partial \phi_x} & \frac{\partial^2 F}{\partial \phi_x \partial \phi_y} \\ \frac{\partial^2 F}{\partial \phi_y \partial \phi} & \frac{\partial^2 F}{\partial \phi_y \partial \phi_x} & \frac{\partial^2 F}{\partial \phi_y \partial \phi_y} \end{bmatrix}, \quad (37)$$

and

$$\mathbf{U} = \begin{bmatrix} U_1, & \dots, & U_N \\ U_{1,x}, & \dots, & U_{N,x} \\ U_{1,y}, & \dots, & U_{N,y} \end{bmatrix}. \quad (38)$$

The following formulas can be carried out [8]

$$g_j = \int_{\Omega} \mathbf{D}_F^T \psi_j dx dy = \sum_{e=1}^{N_T} \int_{T_e} \mathbf{D}_F^T \psi_j dx dy, \quad (39)$$

$$\frac{D(g_1, g_2, \dots, g_N)}{D(\lambda_1, \lambda_2, \dots, \lambda_N)} = \left[\frac{\partial g_i}{\partial \lambda_j} \right] = \left[\sum_{e=1}^{N_T} \int_{T_e} \psi_i^T \mathbf{D}_{FF} \psi_j dx dy \right], \quad (40)$$

where ψ_k is the k -th column of the matrix \mathbf{U} .

The following vector and matrix are respectively introduced:

$$\mathbf{G}(\boldsymbol{\Lambda}) = [g_1(\boldsymbol{\Lambda}), \dots, g_N(\boldsymbol{\Lambda})]^T, \quad \mathbf{J}_G = \left[\frac{\partial g_i}{\partial \lambda_j} \right]. \quad (41)$$

Provided that the set $\{u_1^e, u_2^e, \dots, u_{n_e}^e\}$ is a set of shape functions of the element numbered by e for $e = 1, \dots, N_T$, then the matrix \mathbf{u}_e is introduced:

$$\mathbf{u}_e = \begin{bmatrix} u_1^e, & \dots, & u_{n_e}^e \\ u_{1,x}^e, & \dots, & u_{n_e,x}^e \\ u_{1,y}^e, & \dots, & u_{n_e,y}^e \end{bmatrix}, \quad (42)$$

where n_e is the number of the shape function of e -th element. By these notations the e -th element components of the assembled matrix \mathbf{A}_e and right-hand side vector \mathbf{g}_e are appropriately equal:

$$\mathbf{A}_e := \mathbf{u}_e^T \mathbf{D}_{FF} \mathbf{u}_e, \quad \mathbf{g}_e := \mathbf{D}_F^T \mathbf{u}_e. \quad (43)$$

The considered problem in the form of a search for a extremum of the functional (17) gives a possibility to perform a general computer code giving the solution possibility of class of problems, which can lead to such a formulation. The only replacement routines would be the routines calculating vector \mathbf{D}_F , matrix \mathbf{D}_{FF} and starting vector $\boldsymbol{\Lambda}$ for the Newton-Raphson method.

5. APPLICATION OF THE NEWTON-RAPHSON METHOD TO THE SOLUTION OF NONLINEAR ALGEBRAIC SYSTEM OF EQUATIONS

To solve the system of nonlinear algebraic equations the Newton-Raphson [15] method is applied. The vector \mathbf{G} and matrix \mathbf{J}_G depend on $\boldsymbol{\Lambda}$. The Newton-Raphson method consists of the following steps:

1. Fix the initial vector $\boldsymbol{\Lambda}_0$, set $i = 0$;
2. Repeat points (a), (b), (c) until $\|\mathbf{G}(\boldsymbol{\Lambda}_i)\| < \epsilon \|\boldsymbol{\Lambda}_i\|$;
 - (a) solve the following system of linear equations: $\mathbf{J}_G(\boldsymbol{\Lambda}_i) \boldsymbol{\Delta} \boldsymbol{\Lambda}_{i+1} = -\mathbf{G}(\boldsymbol{\Lambda}_i)$;
 - (b) $\boldsymbol{\Lambda}_{i+1} = \boldsymbol{\Lambda}_i + \boldsymbol{\Delta} \boldsymbol{\Lambda}_{i+1}$;
 - (c) $i := i + 1$.

It is assumed, that the norm in \mathbb{R}^N is defined as

$$\|\mathbf{x}\| = \max_{i=1,\dots,N} |x_i|, \quad \text{where} \quad \mathbf{x} = (x_1, \dots, x_N)^T \in \mathbb{R}^N. \quad (44)$$

The sequence of vectors $\mathbf{\Lambda}_0, \mathbf{\Lambda}_1, \dots$ is convergent. At every iteration step the Jacobi matrix must be assembled. In the presented examples, usually 8–15 iterations were sufficient to obtain the value $\|\mathbf{G}(\mathbf{\Lambda}_i)\|$ of residuum norm of order 10^{-9} .

6. UNSTRUCTURED GRID GENERATION WITH MESH SIZE FUNCTION IN ARBITRARY DOMAINS

The generation of a grid with arbitrary size is performed by the 2D generator [6, 9]. The main idea of grid generation is based on the algorithm of the advancing front technique and the generalization of the Delaunay triangulation for wide class of 2D domains including curved boundaries and multiconnectivity. It is assumed that the domain is multiconnected with arbitrary numbers of internal loops. The boundary of the domain may be composed of the following curves:

- a straight line segment,
- an arc of circle,
- a B-spline curve.

In the case of the advancing front technique combined with the Delaunay triangulation the point insertion and triangulation can be divided into the following steps:

1. Generation of points on the boundary components of the domain boundary.
2. Generation of internal points by the advancing front technique.
3. Delaunay triangulation of the previously obtained set of points.
4. Laplacian smoothing of the obtained mesh.

An algorithm for point generation on a particular curve depends on the type of the boundary curve [6].

7. ALGORITHM FOR REMESHING

The whole adaptation algorithm consists in successive generation of meshes $\{\mathbf{T}_\nu\}$ covering the computational domain, where $\nu = 0, 1, 2, \dots$, with an updated mesh size function. By using every mesh of the sequence the problem is solved, and, next the appropriate error indicators at each point of the mesh are calculated. Having the values of errors at nodes, a continuous error function for the whole domain is constructed by using a piecewise linear interpolation. Clearly this is a simple plane for each element. In each element, the error function spans three nodes. When it is extended to all nodes, one obtains the error function for the whole domain. The error function is appropriately transformed to obtain a multiplier for the mesh size function. The mesh size function decides how big are the newly generated elements.

The proposed approach gives the possibility to solve the considered problem for well-conditioned meshes and to obtain the optimally graded meshes.

7.1. Remeshing scheme and the whole algorithm

The algorithm for remeshing can be divided into the following steps:

1. Preparation of the information about the geometry [16] and boundary conditions of the problem to be solved.
2. Arrangement of an initial mesh size function.
3. Mesh generation with the mesh size function.
4. Solution to the problem given by equation (35) for the generated mesh.
5. Calculation of nodal error indicator.
6. Definition of the new mesh size function by using the errors found at every point.
7. If the error is not small enough go to point 3.
8. End of computations.

In the performed numerical simulations it was sufficient to make 3 to 7 steps of adaptation.

The overall algorithm is presented in the flow chart (Fig. 3). It can be observed that three appropriately nested iterations are executed.

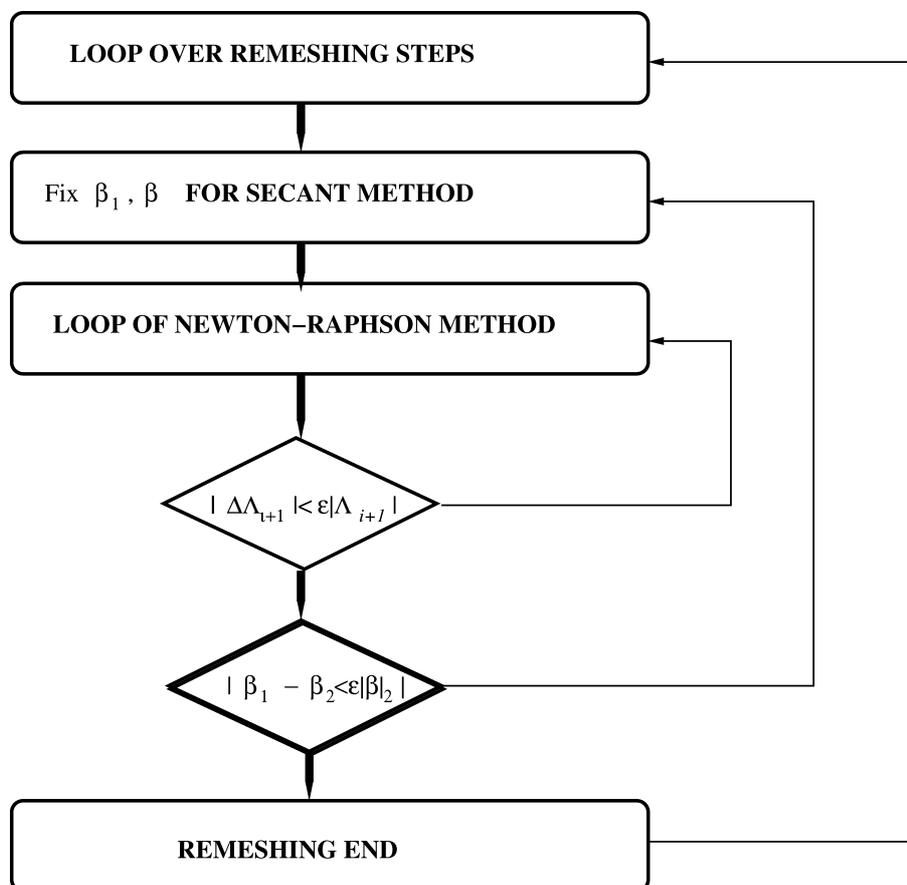


Fig. 3. The flow chart of the whole algorithm.

7.2. Error indicators

The applied indicators are calculated directly for every node, and not for elements like in [6, 10]: Let e_i for $i = 1, \dots, n_\nu$ be an error indicator at i -th apex of the mesh \mathcal{T}_ν , and $\mathcal{P}_\nu = \{P_i, i = 1, \dots, n_P\}$ – set of nodes. We define a patch of elements for every node P_i as

$$L_i = \{k : P_i \in \overline{T}_k\} \quad \text{for } i = 1, \dots, n_P. \quad (45)$$

1. The first proposed error indicator is based on the discretized form of Eq. (11). At every node partial derivatives are found $\frac{\partial u_h}{\partial x}, \frac{\partial u_h}{\partial y}, \frac{\partial^2 u_h}{\partial x^2}, \dots$ according to the following recipe:

Having found $u_h(\mathbf{P}_i)$ for $i = 1, \dots, N_P$, the recurrent formula is applied:

$$\frac{\partial u_h}{\partial x}(\mathbf{P}_i) = \frac{\sum_{k \in L_i} \frac{\partial u_h^k}{\partial x}(\mathbf{P}_i) \text{area}(T_k)}{\sum_{k \in L_i} \text{area}(T_k)}, \quad (46)$$

where u_h^k is the restriction of the approximate solution to the k -th element. As the restriction of u_h^k of the solution u_h to the k -th element is a linear combination of shape functions of the k -th element, then:

$$u_h^k = \sum_{j=0}^{n_e} \lambda_j N_j^k, \quad \text{what gives } \frac{\partial u_h^k}{\partial x} = \sum_{j=0}^{n_e} \lambda_j \frac{\partial (N_j^k)}{\partial x}, \quad (47)$$

where N_j^k is a shape function of the k -th element. Formula (47) is applied at nodal points. The derivatives found in that way $\frac{\partial u_h}{\partial x}(\mathbf{P}_i)$ $i = 1, \dots, N_P$ are used for calculation of second order derivatives at the nodes in the similar way by using the recurrent formulas:

$$\frac{\partial^2 u_h}{\partial x^2}(\mathbf{P}_i) = \frac{\partial}{\partial x} \left(\frac{\partial u_h}{\partial x} \right) (\mathbf{P}_i). \quad (48)$$

In the similar way, it is possible to calculate derivatives of arbitrary order and apply them to the formula (11) to obtain the value of the error indicator at i -th node.

2. In this case, it is suggested to evaluate directly the values of derivatives of error indicator at every node of the mesh in the following way:

$$e_i = \sqrt{\sum_{k \in L_i, l \in L_i, l \neq k} \left(\frac{\partial u_i}{\partial x} - \frac{\partial u_k}{\partial x} \right)^2 + \left(\frac{\partial u_i}{\partial y} - \frac{\partial u_k}{\partial y} \right)^2}, \quad (49)$$

where L_i is the set of numbers of elements meeting at i -th node.

From the numerical analyses it follows that the application of both indicators causes the generation of similar meshes for both cases.

7.3. Modification of the mesh size function

The modification of the mesh size function is performed at every adaptation step for the realization of the next one. The main idea of this part of the algorithm relies on reduction of the values of the mesh size function by an appropriately chosen function. The chosen function is continuous, linear and has the smallest value at the node where the value of the error indicator is maximal and the

greatest where the value of the error is minimal. It increases when the error decreases. To describe the algorithm of the mesh size function modification it is necessary to use the values of the error indicators at nodes:

$$\alpha = \min_{k=1,2,\dots,N_{\text{NOD}}} \tilde{e}_k, \quad \beta = \max_{k=1,2,\dots,N_{\text{NOD}}} \tilde{e}_k, \quad (50)$$

where N_{NOD} is the number of nodes. Obviously, $\alpha \leq \tilde{e}_k \leq \beta$ for $k = 1, \dots, N_{\text{NOD}}$.

The following new values are introduced:

λ – a value indicating the greatest mesh size function reduction,

μ – a value indicating the smallest mesh size function reduction.

Usually λ and μ have positive values of less than 1, and additionally $\mu < \lambda$. Let define the affine transformation:

$$l: [\alpha, \beta] \mapsto [\mu, \lambda] \quad (51)$$

satisfying conditions: $l(\alpha) = \lambda$ and $l(\beta) = \mu$, where $l(x) = \frac{\lambda - \mu}{\alpha - \beta}(x - \alpha) + \lambda$. By these assumptions $\mu \leq l(x) \leq \lambda$. Provided that

$$Q_i = l(\tilde{e}_i) \quad \text{for} \quad i = 1, \dots, N_{\text{NOD}}, \quad (52)$$

then we have: $\mu = \min_{i=1,2,\dots,N_{\text{NOD}}} Q_i$, $\lambda = \max_{i=1,2,\dots,N_{\text{NOD}}} Q_i$.

Let us introduce the function $r: \overline{D} \mapsto \mathbb{R}$ as follows: $r(\overline{x}) = \Pi(\overline{x})$, if $\overline{x} \in \overline{T}_s$, where Π is the affine mapping of two variables satisfying the following conditions:

$$\Pi(P_i) = Q_i \quad \text{for} \quad i = 1, 2, 3, \quad (53)$$

where P_1, P_2, P_3 are the vertices of the triangle T_s of the triangulation of Ω , and appropriately Q_1, Q_2, Q_3 values defined by the formula (52). The function $r(\overline{x})$ is defined for the whole domain because the triangles $\{\overline{T}_s\}_{s=1}^{n_e}$ cover it. The new mesh size function is defined as follows:

$$\gamma_{i+1}(\overline{x}) = \gamma_i(\overline{x})r(\overline{x}). \quad (54)$$

As $\mu \leq r(\overline{x}) \leq \lambda$ then $\mu\gamma_i(\overline{x}) \leq \gamma_{i+1}(\overline{x}) \leq \lambda\gamma_i(\overline{x})$.

It is easy to show that: $\exists \overline{x}, \overline{y} \in \overline{\Omega}$ such that: $\mu\gamma_i(\overline{x}) = \gamma_{i+1}(\overline{x})$, and $\gamma_{i+1}(\overline{y}) = \lambda\gamma_i(\overline{y})$.

It can be shown that

$$\|\gamma_{i+1} - \gamma_i\|_{\overline{\Omega}, \max} \leq \|\gamma_i\|_{\Omega, \max} \max\{|1 - \mu|, |1 - \lambda|\}, \quad (55)$$

where

$$\|\gamma\|_{\Omega, \max} := \max_{\overline{x} \in \overline{\Omega}} \{|\gamma(\overline{x})|\}. \quad (56)$$

8. NUMERICAL RESULTS

The numerical simulations were focused on stating the Dirichlet boundary conditions on the connected boundary conditions for different parts of the boundary $\Gamma_1 \cup \Gamma_2 \cup \Gamma_4$ of the computational domain (Fig. 1). The simulations were performed for a subsonic inviscid potential flow for profile NACA0012 with angle of attack 5° with 0.1 Mach for the speed at infinity. It is assumed that speed has horizontal direction from left to right. In all cases, the initial meshes contained 657 elements, but final meshes had therefore about 4000–6000 triangles.

In Fig. 4 the adapted mesh after three steps of remeshing is presented. In Table 1 the number of iterations for the Newton-Raphson method and the number of iterations for the secant method to the solution of Kutta-Joukovsky condition for all the three formulations are presented.

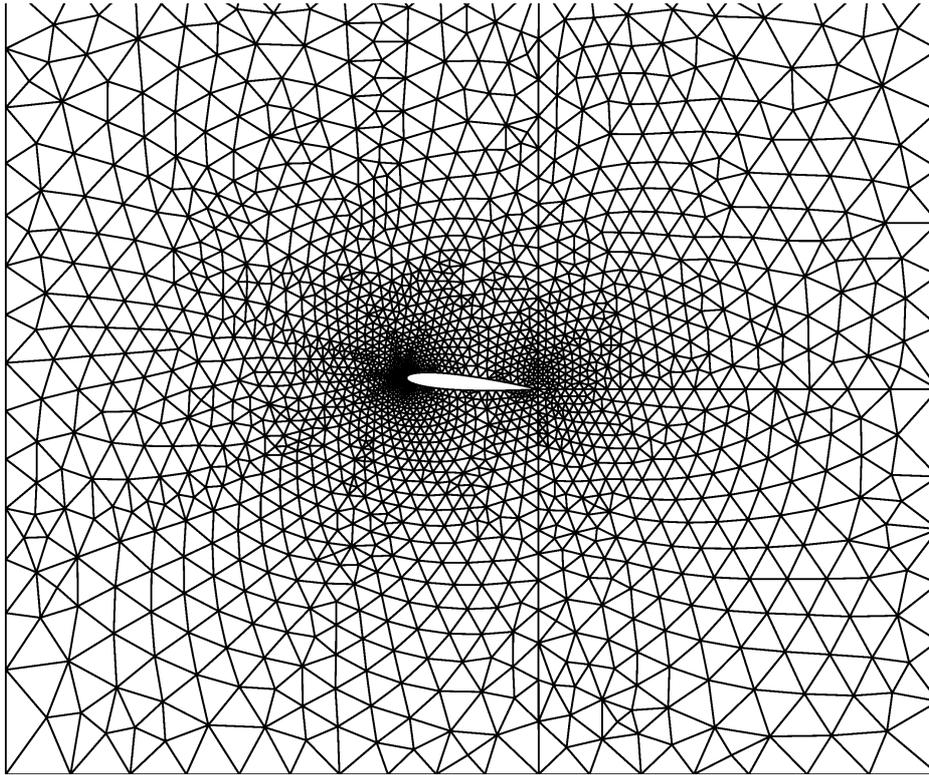


Fig. 4. Final mesh for NACA0012 with angle of attack 5° .

Table 1. Number of iterations.

	Number of N-R iterations	Number of secant method iterations	Number of points on the Dirichlet boundary
Formulation I	15	9	123
Formulation II	13	8	147
Formulation III	8	6	312

9. CONCLUSIONS

In this paper, the new equivalent formulations are proposed. The presented numerical experiments prove that the method is most efficient in the case of stating the Dirichlet conditions for boundary Γ_3 . In all cases, all physical conditions are satisfied. It is proved that there exists a unique solution for each of these three equivalent formulations. It is observed that the number of iterations for both the Newton-Raphson and the secant method is smaller when the number of points with the stated Dirichlet conditions is greater.

Further studies will be connected with:

- performance of more numerical simulations,
- taking into account conditioning of stiffness matrix,
- improvement of mesh generation methods especially taking into account curve generation techniques,
- evolving a method for anisotropic adaptation by using the generator with mesh size function.

REFERENCES

- [1] H. Berger, G. Warnecke, W.L. Wendland. Analysis of a FEM/BEM coupling method for transonic flow computations. *Mathematics of Computation*, **66**(220): 1407–1440, 1997.
- [2] B. Bojarski, Subsonic flow of compressible fluid, *Arch. Mech.*, **18**(4): 497–520, 1966.
- [3] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland Publishing Company, Amsterdam, New York-Oxford, Studies in Mathematics and its Application, 1978.
- [4] L. Demkowicz, J.T. Oden, W. Rachowicz, O. Hardy. Towards a universal h-p adaptive finite element strategy. Part 1: Constrained approximation and data structure. *Comp. Meth. Appl. Mech. Engrg.*, **77**: 79–112, 1989.
- [5] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux. A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow. *Comput. Methods Appl. Mech. Engrg.*, **184**: 241–267, 2000.
- [6] J. Kucwaj. The algorithm of adaptation by using graded meshes generator. *Computer Assisted Mechanics and Engineering Sciences*, **7**: 615–624, 2000.
- [7] J. Kucwaj. Adaptive unstructured solution to the problem of elastic-plastic hardening twist of prismatic bars. *Technical Transactions. Fundamental Sciences* [in Polish: *Czasopismo Techniczne. Nauki Podstawowe*], **2-NP**: 63–79, 2014.
- [8] J. Kucwaj. Numerical investigations of the convergence of a remeshing algorithm on an example of subsonic flow. *Computer Assisted Mechanics and Engineering Sciences*, **17**: 147–160, 2010.
- [9] J.F. Thompson, B.K. Soni, N.P. Weatherill. *Handbook of Grid Generation*. CRC Press, Boca Raton, 1999.
- [10] S.H. Lo. Finite element mesh generation and adaptive meshing. *Progress in Structural Engineering and Materials*, **4**(4): 381–399, 2002.
- [11] J.T. Oden. h-p adaptive finite element methods for compressible and incompressible flows. *Computing Systems in Engineering*, **1**(2–4): 523–534, 1990.
- [12] A. Zdunek, W. Rachowicz. hp-Adaptive CEM in practical applications, *Lecture Notes in Computational Science and Engineering*, **76**: 339–346, 2011.
- [13] A. Safjan, L. Demkowicz, D.P. Young. Compressible flow hp-adaptivity and Kutta-Joukovsky condition. The Boeing Company Project, 2006.
- [14] D. Wang, O. Hassan, K. Morgan, N. Weatherill. Enhanced remeshing from STL files with applications to surface grid generation. *Comm. in Num. Meth. in Engrg.*, **65**: 734–751, 2006.
- [15] O.C. Zienkiewicz, J.Z. Zhu. Adaptivity and mesh generation, *Int. J. Num. Meth. Engng.*, **32**: 783–810, 1991.
- [16] MAdLib: an open source Mesh Adaptation Library, <http://sites.uclouvain.be/madlib/>, 2010.