The application of adaptive algorithm to potential problem

Jan Kucwaj[*]

*Institute of Computer Modelling, Cracow University of Technology,
Warszawska 24, 31-155 Kraków, Poland*

Abstract

The paper presents an algorithm of adaptation by successive mesh regeneration and its application to the potential nonlinear flow problems. The Newton-Raphson method is applied to the solution of nonlinear problem. The Kutta-Zhukovsky condition is implemented by the penalty function method. The adaptation algorithm is based on the modification of mesh size function depending on the error indicator calculated at every node. Numerical tests are presented.

1. Introduction

In the case of flows when the Mach number is less than 1 it is still sensible [1] to solve the nonlinear potential model instead of *Navier-Stokes* equations. It is assumed that the flow is stationary, irrotational, subsonic, comprehensible, inviscid in domain Ω around profile P [1,2].

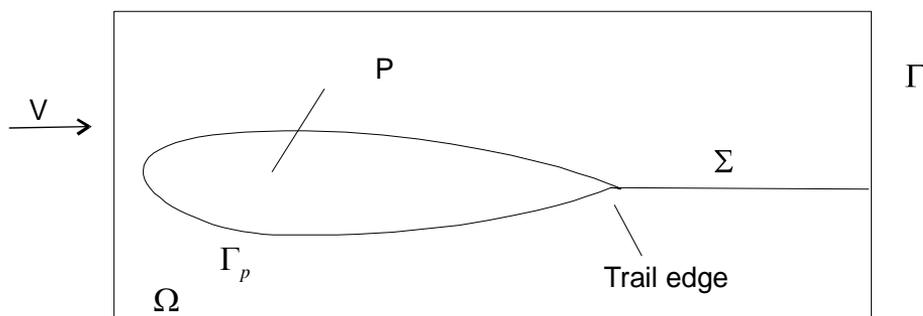The boundary of Ω consists of three parts: $\partial \Omega = \Gamma \cup \Gamma_P \cup \Sigma$.

Fig. 1. The computational domain

---

[*] *E-mail address*: jkucwaj@.pk.edu.pl

The following constants are introduced:

$\overline{V}_\infty$ – travelling velocity at infinity, ρ_0, c_0 – density and sound speed of motionless air,

$\kappa > 1$ – adiabatic gas constant ($\kappa = 1.4$ for dry air).

As the flow is irrotational there exists the potential u of the speed then from the continuity equation it is obtained:

$$div \left[\rho \left(|\nabla u|^2 \right) \nabla u \right] = 0 \text{ in } \Omega, \tag{1}$$

where ρ is the density of air and $\rho \quad (s) = \rho_0 \left(1 - \frac{\kappa - 1}{2c_0^2} s \right)^{\frac{1}{\kappa - 1}}$.

On the slit Ξ the following conditions [3] should be satisfied:

$$u^+ - u^- = \beta, \quad \frac{\partial u^+}{\partial n^+} - \frac{\partial u^-}{\partial n^+} = 0 \quad \text{on} \quad \Xi, \qquad \rho \frac{\partial u}{\partial n} = \rho_\infty u n_\infty \quad \text{on} \quad \Gamma_\infty,$$

where:

n_∞ normal to Γ_∞, $\rho_\infty = \rho \left(|v_\infty| \right)$.

β is found by the Kutta-Zhukovsky condition [2]:

$$K(\beta) = \left|\nabla u^+\right|_{TE}^2 - \left|\nabla u^-\right|_{TE}^2 = 0. \tag{2}$$

2. Full potential problem formulation

Weak formulation is followed by [3]:

$$div \left[\rho \left(|\nabla u|^2 \right) \nabla u \right] = 0 \quad \text{in} \quad \Omega,$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on} \quad \Gamma_P, \quad u^+ - u^- = \beta, \quad \frac{\partial u^+}{\partial n^+} - \frac{\partial u^-}{\partial n^+} = 0 \quad \text{on} \quad \Xi,$$

$$K(\beta) = \left|\nabla u^+\right|_{TE}^2 - \left|\nabla u^-\right|_{TE}^2 = 0.$$

3. Weak formulation and the derivation of principal vectors and matrices for discretisation

For weak formulation of the problem the following Sobolew spaces [2] spaces are defined.

Let $\partial \overset{\circ}{\Omega} = \Gamma \cup \Gamma_P \cup \Sigma$ where $\overset{\circ}{\Omega} = \Omega - \Sigma$, then

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

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

The weak formulation takes the form: find u which satisfies the following variational principle

$$\iint\limits_{\overset{\circ}{\Omega}} \rho\left(|\nabla u|\right)\nabla u \nabla v \, dxdy = \int\limits_{\overset{\circ}{\Omega}} gv \, dxdy \qquad v \in W^{1,\infty}\left(\overset{\circ}{\Omega}\right),$$

$$u^+ - u^- = \beta, \text{ on } \Sigma, \quad g = \rho_\infty \overline{u}\overline{n} \qquad \text{on} \quad \Gamma.$$

Equivalent formulation is in the form of extremum for the following functional:

$$I(u) = \frac{1}{2}\iint\limits_{\overset{\circ}{\Omega}} R\left(|\nabla u|^2\right) dxdy - \int\limits_{\Gamma_\infty} gu d\Gamma,$$

where $R(s) = \int\limits_0^s \rho(s)ds$.

By substitution $H\left(x,y,u,u_x,u_y\right) = \frac{1}{2}R\left(|\nabla u|^2\right)$, it is obtained:

$$I(u) = \iint\limits_{\overset{\circ}{\Omega}} H\left(x,y,u,u_x,u_y\right)dxdy - \int\limits_{\Gamma} gu d\Gamma, \quad \Omega \subset \Re^2.$$

3. The finite element method

Let $\{u_1, u_{2,}, \ldots, u_N\}$ be a set of basis functions for FEM then:

$$u = \sum_{i=1}^N \lambda_i u_i.$$

Let $f\left(\lambda_1, \ldots, \lambda_N\right) := I\left(\sum_{i=1}^N \lambda_i u_i\right)$.

The necessary conditions for the extremum of function f lead to the following solution for the system of algebraic nonlinear equations:

$$g_k\left(\lambda_1, \ldots, \lambda_N\right) = \frac{\partial f}{\partial \lambda_k} = 0.$$

It can be derived that $g_k\left(\lambda_1, \ldots, \lambda_N\right) = \iint\limits_{\overset{\circ}{\Omega}} D_F^T \varphi_k \, dxdy$ thus:

$$\frac{\partial g_k}{\partial \lambda_j} = \iint \varphi_k^T D_{FF} \varphi_k \, dxdy, \quad k, j = 1, 2, \ldots, N,$$

where:

$$D_F = \begin{bmatrix} \dfrac{\partial H}{\partial u} & \dfrac{\partial H}{\partial u_x} & \dfrac{\partial H}{\partial u_y} \end{bmatrix}^T ,$$

the Hessian of function H is introduced:

$$D_{FF} = \begin{vmatrix} \dfrac{\partial^2 H}{\partial u \partial u} & \dfrac{\partial^2 H}{\partial u \partial u_x} & \dfrac{\partial^2 H}{\partial u \partial u_y} \\[2mm] \dfrac{\partial^2 H}{\partial u_x \partial u} & \dfrac{\partial^2 H}{\partial u_x \partial u_x} & \dfrac{\partial^2 H}{\partial u_x \partial u_y} \\[2mm] \dfrac{\partial^2 H}{\partial u_y \partial u} & \dfrac{\partial^2 H}{\partial u_y \partial u_y} & \dfrac{\partial^2 H}{\partial u_y \partial u_y} \end{vmatrix} ,$$

and

$$U = \begin{bmatrix} u_1 & .,\ldots, & u_N \\ u_{1,x} & ,\ldots, & u_{N,x} \\ u_{1,y} & ,\ldots, & u_{N,y} \end{bmatrix} .$$

Then φ_k is the k-th column of matrix U.

In the case of FEM:

$$g_k \rightleftharpoons (\lambda_1,\ldots,\lambda_N) = \iint_\Omega D_F^T \varphi_k \, dxdy = \sum_{e=1}^{T_e} \iint_{T_e} D_F^T \varphi_j \, dxdy .$$

Let $\Lambda = [\lambda_1,\ldots,\lambda_N]$, then

$$J_g(\Lambda) = \frac{D(u_1,u_2,\ldots,u_N)}{D(\lambda_1,\lambda_2,\ldots,\lambda_N)} = \left[\frac{\partial g_i}{\partial \lambda_j} \right] = \left[\sum_{e=1}^{N_T} \iint_{T_e} \psi_i^{\ T} D_{FF} \psi_j \right]. \tag{3}$$

The implementation of Kutta-Zhukovsky condition is fulfilled by the peanalty method:

$$I^*(u) = I(u) + \sum_{i=1}^{l} M \left(\lambda_{k_i} - \lambda_{m_i} \right)^2 . \tag{4}$$

In formula (3) M is a big number in comparison to the elements of matrix (3).

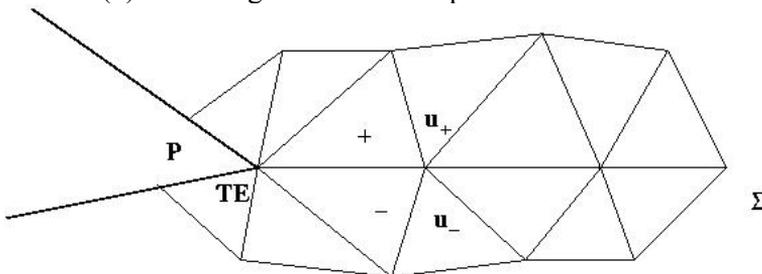Fig. 2. The Kutta-Zhukovsky condition on slit Σ

4. The application of Newton-Raphson method

1. Fix the initial vector Λ_0 satisfying the Kutta-Zhukovsky condition,
2. Repeat points 3, 4, 5 until $\| g(\Lambda_i) \| < \varepsilon \| \Lambda_i \|$,
3. Solve the following system of linear algebraic equations with the implementation of the Kutta-Zhukovsky condition $J_G(\Lambda_i)\Delta_{i+1} = -G(\Lambda_i)$;
4. $\Lambda_{i+1} = \Lambda_i + \Delta\Lambda_{i+1}$,
5. $i := i+1$,
6. End of computations.

The number ε is an error tolerance, thus the norm in $\mathfrak{R}^n$ is defined:

$$\| x \| = \| x \|_\infty = \max_{i=1,...,N} |x_i|, \quad where \quad x = (x_1,...,x_N)^T \in \mathfrak{R}^N .$$

5. Unstructured grid generator with the mesh density function

It is assumed, that the boundary of the 2-D domain consists of the following curves topologically equivalent to the straight line segment [4,5]: straight line segment, arch of the circle and B-spline curve.

Main steps of the grid generation:
1. Points generation on the elementary curves with a given mesh density function by an algorithm dependent on the curve type,
2. Starting with boundary segments right oriented with respect to the domain points insertion by the advancing front technique,
3. Delaunay triangulation on the obtained set of points.
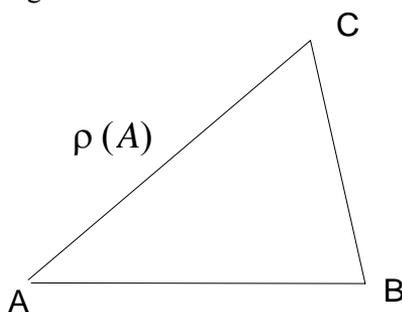4. Laplacian smoothing.

Fig. 3. Satisfaction of the mesh density functions requirements

Let AB be the current boundary segment, point C should satisfy the following conditions:

$$\frac{\rho(A)+\rho(C)}{2}=|AC|,$$

$$\frac{\rho(B)+\rho(C)}{2}=|BC|.$$

In the practical code implementation the simplified condition is used:

$$|AC|=\rho(C),$$
$$|BC|=\rho(B).$$

6. The algorithm of adaptation

The algorithm of adaptation consists of the following steps [6]:
1. Division of the domain into subdomains in such a way that slit Σ is a part of the common boundary for the two neighboring subdomains.
2. Taking the initial mesh density function.
3. Grid generation in subdomains with the taken mesh density.
4. Solution to potential problem on the generated mesh.
5. Evaluation of nodal values of error indicators.
6. Modification of nodal values of mesh density function
7. Description of a new mesh density function by the interpolation of the nodal values.
8. If an error is not satisfactory go to point 3.
9. End of computations.

7. Modification of the mesh density function

Explanation needs point 6 of the adaptation algorithm, which realized during every adaptation step. The main idea of this part of the algorithm consists in the reduction of the values of mesh density function by the appropriately chosen function.

For every node P_i, $i=1,\ldots,N_{NOD}$, a weighted value of the error indicators is defined as follows:

$$\tilde{e}_i=\frac{\sum_{k\in L_k}area(T_i)e_k}{\sum_{k\in L_k}area(T_k)},$$

where T_i is a triangle of the triangulation, and: $L_i=\{j:P_i\in T_j\}$.

Let $\alpha=\min_{k=1,\ldots,N_{NOD}}\tilde{e}_k$, $\beta=\max_{k=1,\ldots,N_{NOD}}\tilde{e}_k$

Obviously $\alpha\le\tilde{e}_k\le\beta$ *for* $k=1,\ldots,N_T$.

The following values are introduced:

λ – a value indicating the smallest mesh density reduction,

μ – a value indicating the greatest mesh density reduction.

It is assumed, that: $0 \le \mu \le \lambda$, and additionally $\lambda \le 1$.

Let $l:[\alpha,\beta] \mapsto [\mu,\lambda]$ be an affine mapping satisfying: $l(\alpha) = \lambda$ and $l(\beta) = \mu$.

Provided, that: $Q_i = l(\tilde{e}_i)$, then we have: $\min\limits_{i=1,\ldots,N_{NOD}} Q_i = \mu$, $\max\limits_{i=1,\ldots,N_{NOD}} Q_i = \lambda$.

Introducing the function $r :: \bar{D} \mapsto \Re$

$$r(x) = \Pi(x), \; if \; x \in T_k \; for \; some \; 1 \le k \le N_T,$$

where Π is an affine mapping of two variables satisfying the following conditions:

$\Pi(P_i) = Q_i \; for \; i = 1,2,3$, where P_1, P_2, P_3 are the vertices of the triangle T_k of the triangulation of Ω.

The function $r(x)$ is defined in the whole domain Ω because the triangles $\left\{ \bar{T}_j \right\}_{j=1}^{N_T}$ cover it. The new mesh density function is defined by the recurrent formula:

$$\rho_{i+1}(x) = \rho_i(x) r(x) \; \forall x \in \bar{\Omega}.$$

It can be shown that:

$$\exists x, y \in \bar{\Omega} \; such \; that \; \mu\rho_i(x) = \rho_{i+1}(x) \; and \; \rho_{i+1}(y) = \lambda\rho_i(y).$$

$$\|\rho_{i+1} - \rho_i\|_{\Omega,\max} \le \|\rho_i\|_{\Omega,\max} \max\left\{|1-\mu|,|1-\lambda|\right\}$$
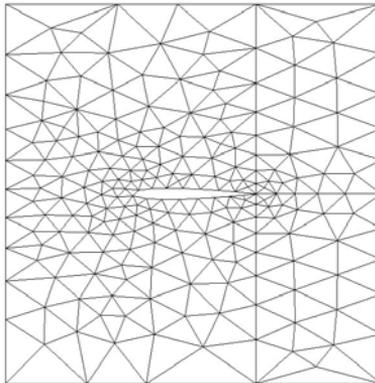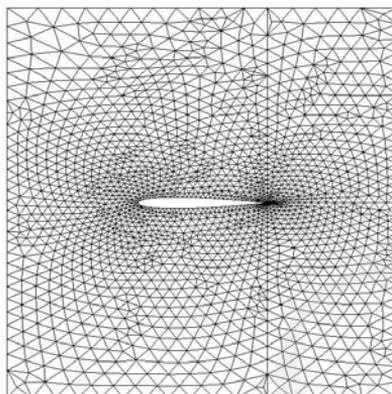

Fig. 4. Initial mesh

Fig 5. Mesh after 6 adaptation steps

8. Numerical tests

Numerical tests were performed for the NACA0012 profile with an angle of attack equal to zero and the traveling velocity equal to 0.6 Mach. Figs. 4 and 5 represent initial and final mesh, and Figs. 6 and 7 isolines for potential function and velocity. Further calculations will be performed with the variety of angles of attacks and traveling velocities.
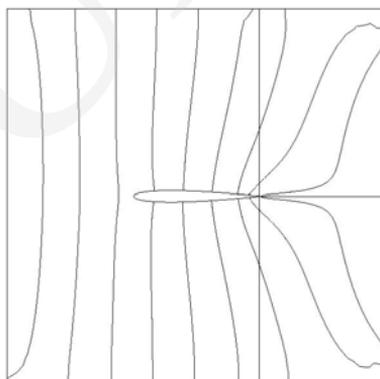

Fig. 6. Isopotential lines

Standard error indicator [6] is applied based on the calculating differences of discontinuity of derivatives at the edges of neighboring elements of the obtained solution.
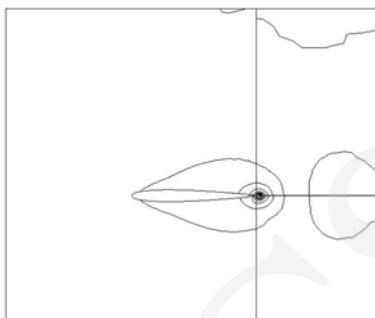
Fig. 7. Speed isolines

9. Conclusions

This paper presents an algorithm of adaptation based upon grid generation with the mesh density function. The algorithm is applied to subsonic potential flow, and numerical results are presented.

Further development will be connected with solution to transonic potential flow and consideration of variety of angles of attacks over known profiles. To find the coeffiecient β in formula (2) the secant method is applied. The whole numerical procedure has three iterations:

- loop over successively generated meshes,
- loop of secant method,
- loop of Newton-Raphson method.

References

[1] Berger H., Warnecke G., Wendland W.L., *Analysis of a FEM/BEM subsonic and transonic flow*, Computations, Univ. Stuttgart, Mathematisches Institute A, (1993).

[2] Glowiński R., Pan T.W., Hesla T.I., Joseph D.D., Periaux J., *A distributed Lagrange Multipiers/frictions Domain Method for the simulation of flow around moving rigid body: application to particular flows*, Comp. Meth. Appl. Mech. Engrg., 184 (2000) 241.

[3] Bristeaux M.O., Pironneau O., Glowiński R., Periaux J., Perrier P., Poirier G., *On the numerical solution of nonlinear problems in fluid dynamics by the least squares and finite element method; application to flow simulation*, Comp. Meth. Appl. Mech. Engrg., 51 (1985) 363.

[4] Kucwaj J., Karafiat A., *Surface grid generation with mesh density function*, Annales Informatica UMCS, (2003).

[5] Kucwaj J., *The modelling and triangulation of geometrically complicated plane domains and some applications to fluid mechanics*, Journal of Theoretical and Applied Mechanics, 35(2) (1997) 369.

[6] Kucwaj J., *The algorithm of adaptation by using graded meshes generator,* Comp. Assist. Meth. in Eng., 7 (2000) 615.