TECHNICAL TRANSACTIONS | **CZASOPISMO TECHNICZNE**
FUNDAMENTAL SCIENCES | NAUKI PODSTAWOWE
1-NP/2016

JAN KUCWAJ*

COMPUTATIONAL EXPERIMENTS OF A REMESHING ALGORITHM BASED ON MESH GENERATOR

NUMERYCZNA EFEKTYWNOŚĆ ALGORYTMU OPARTEGO NA GENERATORZE SIATEK

A b s t r a c t

The main goal of the presented paper are numerical experiments of the convergence of the adaptation algorithm [4] developed by the author based on remeshing, which form the proof of the concept for the presented algorithm. The main feature of the considered algorithm is an application of the mesh generator to the adaptation with a mesh size function [5]. The proposed method uses a sequence of meshes obtained by successive modification of the mesh size function. The rate of the convergence is obtained numerically considering a known solution. The analysis of the unknown solution was restricted to the assessment of some properties of the strict solution.

Keywords: adaptivity, *mesh generation*, *error estimation*, *finite element method*, *finite difference method*, *nolinearity*

S t r e s z c z e n i e

Głównym celem artykułu są numeryczne eksperymenty zbieżności algorytmu adaptacji [4] rozwijanego przez autora opartego na „remeshingu", które potwierdzają koncepcję prezentowanego algorytmu. Główną cechą analizowanego algorytmu jest zastosowanie generatora siatek [5] z zadaną funkcją rozmiaru siatki. Rozwijana metoda wykorzystuje ciąg siatek pokrywających obszar otrzymanych z odpowiednio modyfikowaną funkcją rozmiaru siatki. Otrzymane tempo numerycznej zbieżności zostało uzyskane na znanym rozwiązaniu. Analiza nieznanego rozwiązania została sprowadzona do oceny znanych własności danego rozwiązania, które mogą być obserwowane na rozwiązaniach przybliżonych

Słowa kluczowe: adaptacja, *generowanie siatek*, *wskaźnik błędu*, *metoda elementów skończonych*, *metoda różnic skończonych*, *zagadnienia nieliniowe*

DOI: 10.4467/2353737XCT.16.148.5759

* Jan Kucwaj (jkucwaj@pk.edu.pl), Institute of Computer Science, Cracow University of Technology.

1. Introduction

The paper concerns numerical speed of the convergence of the adaptive algorithm based on a grid generator with a mesh size function [6, 7]. The rate of convergence will be calculated by a description of the dependence between the number of degrees of freedom and norm of error defined as the difference between a strict solution and an approximate solution for a given mesh, provided that the strict solution is known. In case of an unknown solution some properties of the solution are known and their fulfilment can be assessed.

For the sake of the numerical solution the infinite space is approximated by a finite dimensional space spanned by a given set of basis functions [7, 11] of the finite element method [10] generated by linear shape functions [10], the approximated solution to the problem is equal to a linear combination of the basis functions. The coefficients of the linear combination are found from the nonlinear algebraic system of equations. The system is led out from stationarity conditions. The system of nonlinear algebraic equations is solved by using the Newton-Raphson method. In consecutive remeshing (this means separate finite element problems) steps of the adaptation algorithm the values of the mesh size function taken at the nodes are so modified that at the points with greatest values of an error indicator [2, 5] the values of mesh size function are the most diminished. Having the values of the mesh size function at nodes the new mesh size function is defined by the linear interpolation. The process is performed till the error indicator attains the assumed value. The error indicator is found at every node as an approximated residual by the finite difference method for the appropriate local formulation.

The presented numerical analysis of the convergence suggests better than linear dependence between number of degrees of freedom and error norm for derivatives. In further development it is planned to generalize the method to apply anisotropic meshes. The proposed method was applied to both problems, in which the solution is known and unknown. The obtained results were consistent with physical interpretations [4].

The adapted mesh for an example problem, where the strict solution is known, is presented. It can be observed that the rapid change of the size function corresponds to the great gradient of the solution. Additionally, it can be said that the final t mesh depends on both the solution and the assumed error indicator. As an example problem the Poisson equation was taken with known solution and elastic-plastic problem of twisting of bars with hardening, where some physical properties of the solution to the problem are known.

2. Example problem

2.1. The Poisson equation

The boundary value problem for the Poisson equation is formulated as follows:

$$\Delta u = f(x, y), \quad \text{in} \quad \Omega, \tag{1}$$

$$u = 0, \quad \text{in} \quad \partial\Omega. \tag{2}$$

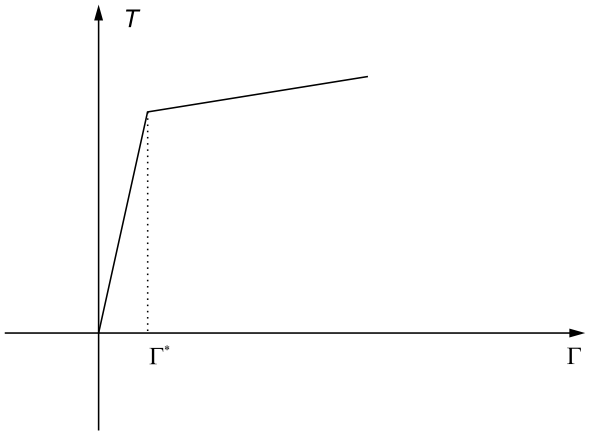


Fig. 1. The dependence between the strain and stress intensity

This equation is used for error indicator calculation:

$$e_i = \tilde{\Delta} u_h(P_i) - f(x,y) \quad \text{at } i\text{-th node} \tag{3}$$

where $\tilde{\Delta}$ is the finite difference approximation of Δ. In this case of the Poisson equation. This problem is equivalent to search for the stationary point of the following functional:

$$I(u) = \int_{\Omega} u_x^2(x,y) + u_y^2(x,y))\, d\Omega. \tag{4}$$

2.2. The elastic-plastic twisting of bars with hardening

In this section the elastic-plastic problem of twisting of bars with hardening is formulated. According to [3] the problem can be led to search for the extremum of the following functional:

$$I(u) = \iint_{\Omega} \left[\int_0^T sg(s)ds - 2\omega u \right] d\Omega, \tag{5}$$

where T is the stress intensity:

$$T = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}, \quad \tau_{13} = \frac{\partial u}{\partial x}, \quad \tau_{13} = -\frac{\partial u}{\partial y}.$$

The function g defines the dependence between the effective stress and the effective strain: $T = g(\Gamma)\Gamma$ (Fig. 1), where $\Gamma = \sqrt{\varepsilon_{ij}\varepsilon_{ij}}$, ε_{ij} is the strain tensor and ω is the angle of the torsion.

After the substitution $s = \sqrt{r}$, it is obtained:

$$I(u) = \iint_{\Omega} \left[\int_0^{T^2} sg(\sqrt{s}) \frac{1}{2} ds - 2\omega u \right] d\Omega. \tag{6}$$

In both problems the current function u varies in the Sobolev space

$$H^1(\Omega) = \left\{ v \in L^2(\Omega), \frac{\partial v}{\partial x_i} \in L^2(\Omega), i = 1,2 \right\}.$$

In both problems the current function u varies in the Sobolev [3] space.

For the sake of the approximation the finite dimensional space of functions is defined:

$$V^0 = \left\{ v : \bigcup_{i=0}^{N_T} \bar{T}_i \to \Re,\ v \ is\ continuos,\ v|_{\bar{T}_i} \in P_1 \right\},$$

where $T = \{T_i : i = 1,\ldots,N_T\}$ is a set of non-intersecting triangles covering the domain. For the finite element approximation the approximate solution is defined as finite linear combination of basis functions [10] of the space V^0. The unknown coefficients of the linear combination are found by solving the nonlinear system of algebraic equations, obtained from the stationarity condition [6].

3. The unstructured grid generation with mesh size function in arbitrary domains

Grid generation with arbitrary mesh size function is performed using a 2-D generator [5, 6]. The main idea of grid generation is based upon the algorithm of the advancing front technique and a generalization of the Delaunay triangulation [5, 8] for wide class of $2 - D$ domains. It is assumed that the domain is multiconnected with an arbitrary number 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 case of the advancing front technique combined with the Delaunay triangulation the point insertion and triangulation can be divided into the following steps:

1. Point generation on the boundary,
2. Internal point generation by the advancing front technique,
3. Delaunay triangulation of the previously obtained set of points,
4. The Laplacian smoothing of the obtained mesh.

The algorithm for boundary point generation depends upon the type of boundary segment: [5].

4. Algorithm of remeshing

The whole algorithm of the adaptation is realized in the successive generation of a sequence of meshes $\{T_\nu\}$, where $\nu = 0,1,2,\ldots$ with a modified mesh size function. By using every mesh of the sequence the approximate solution to the problem is obtained and then appropriate error indicators at each node are calculated. Having values of errors at nodes a continuous error function in the whole domain is constructed by using piecewise

linear interpolation at all elements. The error function is appropriately transformed to obtain a multiplier for mesh size function.

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

4.1. Remeshing scheme

The algorithm of remeshing [4, 13] can be divided into the following steps:

1. Preparation of the information about the geometry and boundary conditions of the problem to be solved,
2. Arrangement of an initial mesh size function,
3. Mesh generation with mesh size function,
4. Solution to the considered on the generated mesh,
5. Evaluation of error indicator at each node,
6. Definition of the new mesh size function by using the errors found at every point of the computational grid,
7. If the error not small enough go to the point 3,
8. End of computations.

In the examples solved by the author it was sufficient to make from 5 to 9 steps of adaptation.

4.2. Error indicators

The applied error indicators are calculated directly for every node, not in elements like in [6, 9].

Let e_i for $i = 1,\ldots,n_P$ be an error indicator at i-the apex of the mesh $\tilde{T}_\nu$, and $\tilde{P}_\nu = \{P_i,\ i = 1,\ldots,n_P\}$ set of nodes. We define a patch of elements for every node P_i as:

$$L_i = \{k,\ P_i \in \bar{T}\} \quad \text{for} \quad i = 1,\ldots,n_P \tag{7}$$

where T_P is the k-th element of the mesh.

1. The first proposed error indicator is biased on the discretized form of the equation (1). At every node partial derivatives are found according to the following recipe:

Having found

$$\frac{\partial u_h}{\partial x}(P_i) = \frac{\sum_{k\in L_i} \frac{\partial u}{\partial x}(P_i)\, area(T_k)}{\sum_{k\in L_i} area(T_k)}. \tag{8}$$

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

$$u_h^k = \sum_{j=0}^{n_e} \lambda_j N_j^k, \quad \text{which gives} \quad \frac{\partial u_h}{\partial x} = \sum_{j=0}^{n_e} \lambda_j \frac{\partial (N_j^k)}{\partial x}, \tag{9}$$

where $u_h = N_j^k$ is a shape function of the *k*-th element. Formula 9 is applied at nodal points. The derivatives $\frac{\partial u_h}{\partial x}(P_i)$, $\frac{\partial u_h}{\partial y}(P_i)$, $i = 1,\ldots,N_P$ found in that way are used for calculation of second order derivatives at nodes in the similar way by using the recurrent formula:

$$\frac{\partial^2 u_h}{\partial x^2}(P_i) = \frac{\partial}{\partial x}\left(\frac{\partial (u_h)}{\partial x}\right)(P_i). \tag{10}$$

In the similar way it is possible to calculate the derivatives of arbitrary order and put them into formula 2 to obtain the value of the error indicator at *i*-th node.

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

$$T = \sqrt{\sum_{k\in L_i, l\in L_i, l\neq k}\left(\frac{\partial u_i}{\partial x} - \frac{\partial u_k}{\partial x}\right)^2 + \left(\frac{\partial u_i}{\partial y} - \frac{\partial u_k}{\partial y}\right)^2}. \tag{11}$$

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

4.3. Error indicators

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 a multiplication of the values of the mesh size function by an appropriately chosen function. The chosen function should be continuous, linear and should have 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 maximal. It should increase when the error decreases.

The error indicators $\tilde{e}_k$ are calculated at each node of the current mesh, then the minimal and maximal values of the error are found:

$$\alpha = \min_{k=1,2,..,N_{NOD}} \tilde{e}_k, \quad \alpha = \max_{k=1,2,..,N_{NOD}} \tilde{e}_k, \tag{12}$$

where N_{NOD} is the number of nodes. Certainly, $\alpha \le \tilde{e}_k \le \beta$ for $k = 1,2,\ldots,N_{NOD}$.

The following values are introduced:

λ – a value indicating the greatest mesh size reduction.

μ – a value indicating the greatest mesh size reduction.

The values of λ and μ usually should be greater than 0.5, which means that the mesh size does not change too rapidly, which would have an influence on mesh quality in the vicinity, where there are big errors. Usually it is assumed that λ varies from 0.5 to 0.6 and μ from 0.8 to 1.0.

The following affine transformation is defined:

$$l : [\alpha,\beta] \to [\mu,\lambda], \tag{13}$$

which satisfies the conditions $l(\alpha) = \lambda$ and $l(\alpha) = \mu$. By these assumptions it can be observed that $\mu \le l(x) \le \lambda, \ \forall x \in [\alpha, \beta]$.

Provided that

$$Q_i = l(\tilde{e}_i) \le \lambda, \ \forall i = 1, \ldots, N_{NOD}, \tag{14}$$

then we have

$$\min_{i=1,2,\ldots,N_{NOD}} Q_i = \mu, \qquad \max_{i=1,2,\ldots,N_{NOD}} Q_i = \lambda.$$

Introducing the function $r : \bar{D} \to \Re$, as follows: $r(x) = \Pi(x)$, if $x \in \bar{T}_s$, where Π is an affine mapping of two variables satisfying the following equalities:

$$\Pi(P_i) = Q_i, \quad \text{for} \quad i = 1, 2, 3, \tag{15}$$

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 are the values defined by the formula (14). The function $r(x)$ is defined in the whole domain because the triangles $\{\bar{T}_s\}_{s=1}^{N_T}$ cover it. The new mesh size function is defined as follows:

$$\gamma_{i+1}(x) = \gamma_i(x) r(x). \tag{16}$$

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

It can be checked that: $\exists x, \ y \in \bar{\Omega}$ such that:

$$\mu\gamma_i(x) = \gamma_{i+1}(x) \quad \text{and} \quad \gamma_{i+1}(x) = \lambda\gamma_i(x). \tag{17}$$

It can be shown, that $\|\gamma_{i+1} - \gamma_i\|_{\bar{\Omega},\max} \le \|\gamma_i\|_{\bar{\Omega},\max} \max\{|1-\lambda|, |1-\mu|\}$, where

$$\|\gamma\|_{\bar{\Omega},\max} = \max_{x \in \bar{\Omega}} \{|\gamma(x)|\}. \tag{18}$$

5. Numerical examples

The manner of size function modification depends on the error indicator and on the coefficients λ, μ, which determine the details of the mesh size function modification. If the values of the coefficients λ, μ, are small then fewer adaptation steps is necessary. How quickly an adapted grid will be close enough to an optimal mesh, besides of error indicator function, depends on the initial mesh too. For the solved problems it was assumed that $\lambda = 0.6$ and $\lambda = 0.8$, which caused performing greater number of adaptation steps, what may lead to a better solution. In the plasticity theory problems it can observed (figures 5, 6), that the adapted mesh densities at the border between elastic and plastic zones and the adapted mesh (Fig. 5) coincide with the sand heap analogy [3]. It would be rather impossible to obtain the effect by the methods based on mesh enrichment [1, 9].

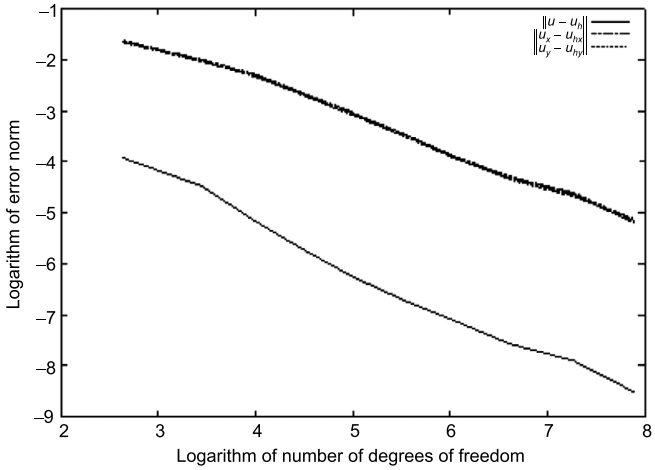


Fig. 2. The convergence curves for u, u_x, u_y for problem 1 with respect to the norms

For the sake of numerical rate of the convergence of the proposed method for the problem defined in 4 the function f was defined in the way that the solution to the problem is the function [13]: $u(x,y) = x(1-x)y(1-y)\arctan\left(a\left(\frac{x+y}{\sqrt{2}} - \xi\right)\right)$, where a = 20 and ξ = 0.8. The figure 2 presents the dependence between number of nodes and norms $\|u-u_h\|$, $\left\|\frac{\partial u}{\partial x} - \frac{\partial u_h}{\partial x}\right\|$ and $\left\|\frac{\partial u}{\partial y} - \frac{\partial u_h}{\partial y}\right\|$. The graphs of the norms $\left\|\frac{\partial u}{\partial x} - \frac{\partial u_h}{\partial x}\right\|$ and $\left\|\frac{\partial u}{\partial y} - \frac{\partial u_h}{\partial y}\right\|$ almost cover each other.

The figure 3 presents the adaptive mesh.

It can seen that the mesh for the example problem 1 and its strict solution 5 coincide.

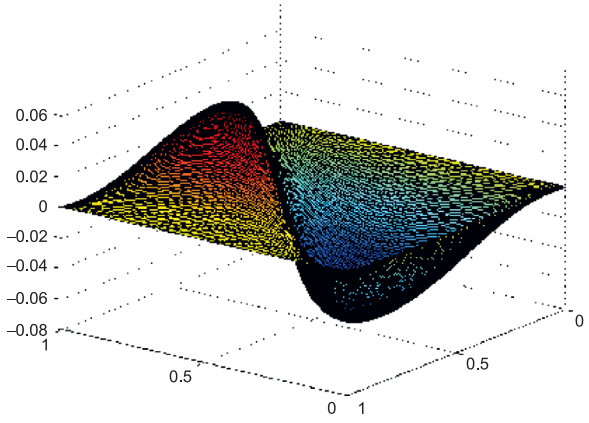


Fig. 3. Strict solution for the problem 1

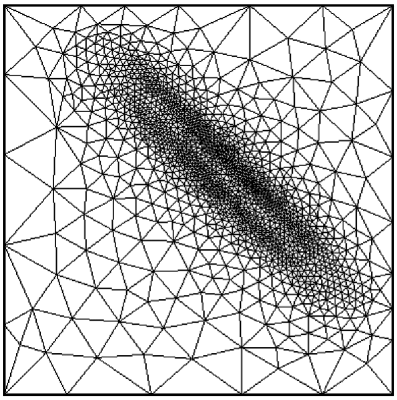


Fig. 4. Adapted mesh for the problem 1

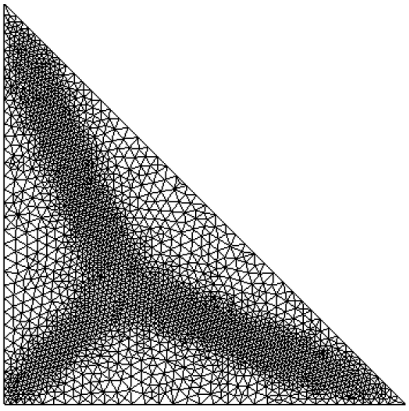


Fig. 5. Final mesh after 7 adaptation steps for problems 6

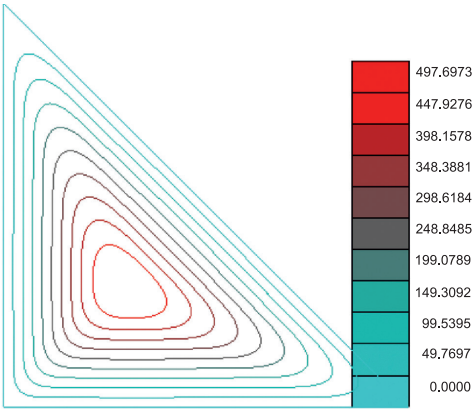


Fig. 6. Adapted mesh for the problem 6

6. Summary

- New error indicators based on generalized finite difference method were introduced applied to the proposed adaptive remeshing.
- The numerical rate of the convergence was calculated by using the known strict solution.
- The optimal mesh size function is obtained iteratively and depends on values of error indicators at nodes.
- The generator based on Delaunay condition and advancing front technique seems very suitable to the class of problems where different zones of the domain are to be appointed.
- For further investigations the anisotropic mesh generation algorithm will be developed an appropriate anisotropic adaptation algorithms as well too.

References

[1] Bank R., Sherman A., Wieser A., *Some Refonement Algorithms and Data Structures for regular local mesh refinement*, Scientific Computing, IMACS, 1983.

[2] Huerta A., Diez P., *Error estimation including pollution assessement for nonlinear finite element analysis*, Comp. Meth. Appl. Mech. Engn., vol. 181, 2000, 21-24.

[3] Kachanov L.M., *Fundamentals of plasticity theory, Dover Publications Inc*., 2004, 479 p. (ISBN 0-486-43483-0), Mineola, NY, USA, Moscow, 1968.

[4] Kucwaj J., *The Algorithm of Adaptation by using graded meshes generator*, Computer Assisted Mechanics and Engineering Sciences, vol. 7, 2000, 615-624.

[5] Thompson J.F., Soni B.K., Weatherwill N.P., *Handbook of grid generation*, CRC Press, Boca Raton, 1999.

[6] Kucwaj J., *Numerical Investigation of the convergence of remeshing algorithm na an axample of subsonic flow*, Computer Assisted Mechanics and Engineering Sciences, vol. 17, 2010, 147-160.

[7] Kucwaj J., *Adaptive unstructured solution to the problem of elastic-plastic hardening of prismatic bars*, Technical Transactions, vol. 111, 2014, 63-79.

[8] Lo S.H., *Finite element mesh generation and adaptive meshing*, Progress in Structural Engineering and Materials, vol. 4, 2002, 381-399.

[9] Oden J.T., Demkowicz L., Rachowicz W., Westermann T.A., *Towards a universal h-p adaptive finite element strategy, part 2, aposteriori error estimation*, Comp. Meth. Appl. Mech. Engn., vol. 77, 1989, 113-180.

[10] Zienkiewicz O.C., Taylor R.L., *The finite element method*, 4-th edition, vol. 1, basic formulation and linear problems, McGraw-Hill Book Company, London, Washington, 1989.

[11] Zienkiewicz O.C., *Achievements and some unsolved problems of the finite element method*, Int. J. Num. Meth. Engn., vol. 47, 2000, 9-28.

[12] Zienkiewicz O.C., Zhu J.Z., *Adaptavity and mesh generation*, Int. J. Num. Meth. Engn., vol. 32, 1991, 783-810.

[13] Madlib: An open source mesh adaptation library, http://sites.uclouvain.be/madlib/, 2010.