TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
FUNDAMENTAL SCIENCES | NAUKI PODSTAWOWE
2-NP/2014

JAN KUCWAJ*

# ADAPTIVE UNSTRUCTURED SOLUTION TO THE PROBLEM OF ELASTIC-PLASTIC HARDENING TWIST OF PRISMATIC BARS

# ADAPTACYJNE NIESTRUKTURALNE ROZWIĄZANIE ZAGADNIENIA SKRĘCANIA PRĘTA PRYZMATYCZNEGO W ZAKRESIE SPRĘŻYSTO-PLASTYCZNYM ZE WZMOCNIENIEM

Abstract

This paper presents the appliction of a remeshing algorithm to solution of elastic-plastic torsion of bars with isotropic strain hardening. The remeshing algorithm uses a grid generator with mesh size function [7]. The method of grid generation is based on a coupling of the advancing front method and the Delaunay triangulation. The optimal mesh size for the posed problem is obtained iteratively. For the consequtive steps of the adaptation algorithm error indicators at nodes and in elements are used for mesh size modification. The discretized system of nonlinear algebraic equations is solved by the application of the Newton-Raphson method.

*Keywords: torsion, plasticity, elasticity, grid adaptation, Delaunay triangulation, nonlinearity*

Streszczenie

Praca przedstawia zastosowanie algorytmu typu *remeshing* do rozwiązania zagadnienia sprężysto-plastycznego skręcania prętów pryzmatycznych ze wzmocnieniem. Algorytm typu *remeshing* wykorzystuje generator siatek uwzględniający funkcję rozmiaru siatki [7]. Metoda generowania siatek oparta jest na połączeniu metody postępującego frontu z triangulacją Delaunaya. Optymalny rozmiar siatki dla postawionego problemu otrzymany jest iteracyjnie. W kolejnych krokach adaptacji indykatory błędów w węzłach i elementach są wykorzystane do modyfikacji rozmiaru siatki. Zdyskretyzowany układ nieliniowych równań algebraicznych jest rozwiązywany poprzez zastosowanie metody Newtona-Raphsona.

*Słowa kluczowe: skręcanie, plastyczność, sprężystość, generowanie siatek, adaptacja, triangulacja Delaunaya, nieliniowość*

*Institute of Computer Science, Cracow University of Technology; jkucwaj@usk.pk.edu.pl

## 1. Introduction

The purpose of this paper is to apply the adaptive remeshing [7, 8] to the elastic-plastic torsion of bars with isotropic strain hardening [5]. The problem is posed in a form of the search for the stationary point of the functional defined on function space of infinite dimension satisfying homogenous boundary conditions. In the case of twisted bars, this means that their length is great enough. For the sake of a numerical solution the infinite space is approximated by finite dimensional space spanned by a given set of basis functions [2, 11, 12]. The approximated solution to the problem is equal to a linear combination of the basis functions. The coeffiecients of the linear combination are found from the set of nonlinear algebraic equations. The latter are obtained from the stationarity conditions. These equations are then solved using the Newton-Raphson method.

The grid generation code GRADMESH [6, 7] developed in [7] can be used to generate the FE model. The generator takes into account the mesh size function defined in the whole domain. The mesh size function is iteratively modified in order to give a uniform error distribution. In practice it is rather difficult to obtain uniform error distribution. As a matter of fact the values of mesh size function are defined only at the nodes of the current mesh. Values of the mesh size function are calculated by the linear interpolation in every element. Therefore the modification of the mesh size function leads to the modification of their values at the nodes in the current mesh. These values are multiplied by a value from the interval $[0, 1]$. The values are smaller for nodes with a greater value of error indicator than for nodes with a smaller value of the indicator. Having modified values at the nodes of the current mesh, the new mesh size function is defined in the whole domain. The consecutive mesh is generated next. For the sake of the consecutive mesh generation the whole information about the previous mesh must be stored.

In the current paper, two methods of error evaluation are applied. The first adopts standard methods used in [4, 10, 14] which give values of error estimates in elements. The error estimation at nodes is more involved. A given node is shared by adjacent elements, each in turn, carries a different error. One needs to accommodate these different errors when assigning the final value to the node. The weighted average is used here. The precise equation is given later (eq. 6.5) but the process is as follows. The partial error in a given element is multiplied by its area and is divided by the area of all adjacent elements. Finally, the node error is the sum of partial errors (coming from all elements surrounding the node).

The error indicator at a given node can be evaluated in a different way, for example, by considering first derivatives with respect to all variables, these are taken at the node. In this approach the error indicator is formed summing squares all derivatives. The final value is obtained by taking the square root of the sum. Detailed expression is given later (eq. 6.4). Yet another approach which is frequently and widely known is given by equation [2].

Once the magnitude of the final error indicator at a given node is established, it serves as a size indicator for the new, remeshed model.

The above information is used for remeshing, i.e., for the generation of a new grid of nodes. The grid in the FE model utilizes the advancing front technique (AFT) [9] in conjunction with the Delaunay approach [6, 7, 9].

The adaptation method based on points enrichment [1, 10, 11] leads to unnecessary mesh enrichment or bad quality elements [15]. Remeshing on the other hand [8, 14] seems to be one of the best adaptation techniques, especially in the case of the elastic-plastic twisting of bars. It almost exactly densifies the mesh only at the yield points. In short, the method focuses on adaptation (remeshing) of the grid at places where it is necessary as dictated by local error indicators. It does not alter mesh size where it is not necessary. It is clear that this approach is effiecient and superior to other approaches.

## 2. Problem formulation

In this section, the elastic-plastic torsion of bars with isotropic strain hardening is formulated. According to [5] the problem leads to the search for the extremum of the following functional:

$$I(u) = \iint_{\Omega} [\int_0^T sg(s)ds - 2\omega u] d\Omega, \tag{2.1}$$

where T is the stress intensity

$$T = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2}, \qquad \tau_{13} = \frac{\partial u}{\partial y}, \qquad \tau_{23} = -\frac{\partial u}{\partial x}$$

.

The function $g$ defines the dependence between the effective stress and the effective strain: $T = g(\Gamma)\Gamma$ (Fig. 1), where $\Gamma = \sqrt{\epsilon_{ij}\epsilon_{ij}}$, $\epsilon_{ij}$ is the strain tensor and $\omega$ is the torsion angle (see Ref. [5]).

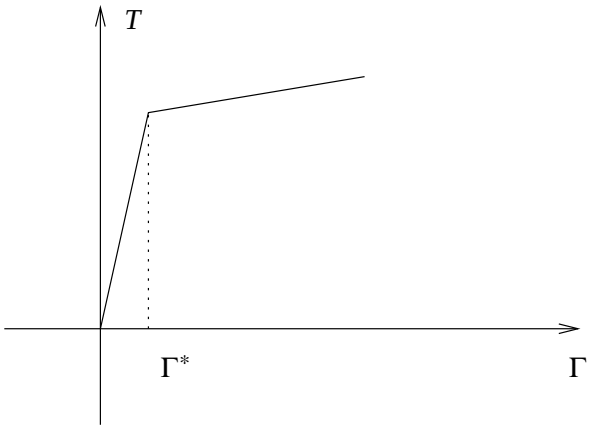


Fig. 1: The dependence between the strain intesity of strain, F and stress intesity ,T

After the substitution $s = \sqrt{r}$, the equation (2.1) is:

$$I(u) = \iint_{\Omega} \left[ \int_0^{T^2} g(\sqrt{r}) \frac{1}{2} dr - 2\omega u \right] d\Omega. \tag{2.2}$$

### 3. Discretization approach and tangent matrix

In this section, a class of problems leading to the search for a stationary point of a functional of the following form is addressed:

$$I(u) = \iint_{\Omega} F(u, u_x, u_y) d\Omega, \text{ where} \tag{3.1}$$

$\Omega$ - is a two-dimensional domain,

$F$ - is a double diffentiable function of several variables $F : \mathbb{R}^M \mapsto \mathbb{R}$.

Let us introduce finite element basis $\{U_i\}_{i=1}^N$ in the space $V^0$ of functions satisfying homogenous boundary conditions. Then, the discretization of the problem leads to the solution of the following system of nonlinear algebraic equations:

$$g_k(\lambda_1, \lambda_2, \ldots, \lambda_N) = \frac{\partial I(\sum_{i=1}^N \lambda_i U_i)}{\partial \lambda_k} = 0, \quad \text{for } k = 1, \ldots, N, \tag{3.2}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_N$ are coefficients of the linear combination of the approximate solution.

The chain formula for the calculations of partial derivatives leads to:

$$g_k(\lambda_1, \lambda_2, \ldots, \lambda_N) = \tag{3.3}$$

$$= \frac{\partial}{\partial \lambda_k} \iint_\Omega F(x, y, \sum_{i=1}^{N} \lambda_i U_i, \sum_{i=1}^{N} \lambda_i U_{i,x}, \sum_{i=1}^{N} \lambda_i U_{i,y}) d\Omega = \tag{3.4}$$

$$= \iint_\Omega D_F^T \Psi_k d\Omega, \quad \text{and} \tag{3.5}$$

$$\frac{\partial g_k}{\partial \lambda_j} = \iint_\Omega \Psi_k^T D_{FF} \Psi_j d\Omega. \tag{3.6}$$

In the expressions (3.4), (3.6)

$$\mathbf{D}_F = \left[ \frac{\partial F}{\partial u}, \frac{\partial F}{\partial u_x}, \frac{\partial F}{\partial u_y} \right]^T, \tag{3.7}$$

$$\mathbf{D}_{FF} = \begin{bmatrix} \frac{\partial^2 F}{\partial u \partial u}, & \frac{\partial^2 F}{\partial u \partial u_x}, & \frac{\partial^2 F}{\partial u \partial u_y} \\ \frac{\partial^2 F}{\partial u_x \partial u}, & \frac{\partial^2 F}{\partial u_x \partial u_x}, & \frac{\partial^2 F}{\partial u_x \partial u_y} \\ \frac{\partial^2 F}{\partial u_y \partial u}, & \frac{\partial^2 F}{\partial u_y \partial u_x}, & \frac{\partial^2 F}{\partial u_y \partial u_y} \end{bmatrix}, \tag{3.8}$$

$$\text{and} \qquad \mathbf{U} = \begin{bmatrix} U_1, & \ldots, & U_N \\ U_{1,x}, & \ldots, & U_{N,x} \\ U_{1,y}, & \ldots, & U_{N,y} \end{bmatrix}, \tag{3.9}$$

The above formulas are valid for arbitrary basis functions (for example, polynomial) $\{U_1, U_2, \ldots, U_N\}$ defined in $\Omega$.

In the case of FEM one has the following equations:

$$g_j = \iint_\Omega \mathbf{D}_F^T \psi_j \, dx \, dy = \sum_{e=1}^{N_T} \iint_{T_e} \mathbf{D}_F^T \psi_j \, dx \, dy, \tag{3.10}$$

$$\frac{D(g_1, g_2, \ldots, g_N)}{D(\lambda_1, \lambda_2, \ldots, \lambda_N)} = \left[ \frac{\partial g_i}{\partial \lambda_j} \right] = \left[ \sum_{e=1}^{N_T} \iint_{T_e} \psi_i^T \mathbf{D}_{FF} \psi_j \, dx \, dy \right] \tag{3.11}$$

where $\psi_k$ is the *k-th* column of matrix $\mathbf{U}$.

Introduce the following vector:

$$\mathbf{G}(\mathbf{\Lambda}) = [g_1(\mathbf{\Lambda}), \ldots, g_N(\mathbf{\Lambda})]^T, \text{ and matrix} \tag{3.12}$$

$$\mathbf{J_G} = \left[ \begin{array}{c} \frac{\partial g_i}{\partial \lambda_j} \end{array} \right]. \tag{3.13}$$

Let the set $\{u_1^e, u_2^e, \ldots, u_{n_e}^e\}$ be a set of shape functions of the element with number e, for $e = 1, \ldots, N_T$. The matrix $\mathbf{u}_e$ is then defined as follows:

$$\mathbf{u}_e = \left[ \begin{array}{ccc} u_1, & \ldots, & u_{n_e} \\ u_{1,x}, & \ldots, & u_{n_e,x} \\ u_{1,y}, & \ldots, & u_{n_e,y} \end{array} \right], \tag{3.14}$$

where $n_e$ is number of the shape functions of *e-th* element. Introduce the vector $\mathbf{g}_e$ and the matrix $\mathbf{A}_e$

$$\mathbf{g}_e := \mathbf{D}_F^T \mathbf{u}_e, \quad \mathbf{A}_e := \mathbf{u}_e^T \mathbf{D}_{FF} \mathbf{u}_e. \tag{3.15}$$

The solution of equation (3.6) will normally determine a numerical approach. The latter can be applied to a different engineering problem of the same mathematical class. The only replacement routines would be routines calculating vector $\mathbf{D}_F$, matrix $\mathbf{D}_{FF}$ and starting vector $\mathbf{\Lambda}$ for the Newton-Raphson iteration.

For the elastic-plastic torsion of bars, after substitution of the function:

$$h(r) = \frac{1}{2} g(\sqrt(r)), \tag{3.16}$$

into equation (2.2) matrices $\mathbf{D}_F$ and $\mathbf{D}_{FF}$ can be evaluated using equations (3.4) and (3.6). For the current case they are:

$$\mathbf{D}_F = \left[ -2\omega, 2h(T)\frac{\partial u}{\partial x}, 2h(T)\frac{\partial u}{\partial y} \right]^T, \tag{3.17}$$

$$\mathbf{D}_{FF} = \left[ \begin{array}{ccc} -2, & 0, & 0 \\ 0, & 2h(T) + 4\frac{dh}{dT}(\frac{\partial u}{\partial x})^2, & 4\frac{dh}{dT}\frac{\partial u}{\partial x}\frac{\partial u}{\partial y} \\ 0, & 4\frac{dh}{dT}\frac{\partial u}{\partial x}\frac{\partial u}{\partial y}, & 2h(T) + 4\frac{dh}{dT}(\frac{\partial u}{\partial y})^2 \end{array} \right]. \tag{3.18}$$

## 4. Application of the Newton-Raphson method to the solution of nonlinear algebraic equations

For the solution of the system of nonlinear equations, the Newton-Raphson method is applied. The vector $\mathbf{G}$ and matrix $\mathbf{J_G}$ depend upon $\mathbf{\Lambda}$. The algorithm of Newton-Raphson can be divided into the following steps:

1. Set initial vector $\mathbf{\Lambda}_0$, set i=0;

2. Repeat points 3, 4, 5 until $||\mathbf{G}(\mathbf{\Lambda_i})|| < \epsilon\, ||\mathbf{\Lambda_i}||$;

3. Solve the following system of linear equations:

$$\mathbf{J_G}(\mathbf{\Lambda}_i)\mathbf{\Delta\Lambda}_{i+1} = -\mathbf{G}(\mathbf{\Lambda}_i); \tag{4.1}$$

4. $\mathbf{\Lambda}_{i+1} = \mathbf{\Lambda}_i + \mathbf{\Delta\Lambda}_{i+1}$;

5. $i := i + 1$.

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

$$||\mathbf{x}|| = \max_{i=1,\ldots,N} |x_i|, \text{ where } \mathbf{x} = (x_1, \ldots, x_N)^T \in \mathbb{R}^N. \tag{4.2}$$

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

## 5. Unstructured grid generation with mesh size function in arbitrary domains

The generation of a grid with arbitrary size is performed by 2D generator [6, 7]. The main idea of grid generation is based upon the algorithm of the advancing front technique and generalization of Delaunay triangulation for a wide class of 2D domains. 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 a circle,
- a B-spline curve.

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

1. Generation of points on the boundary components of the boundary of the domain,
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.

The algorithm for generation of boundary points depends upon the type of boundary segment.

## 6. Algorithm for remeshing

The whole adaptation algorithm consists of the successive generation of meshes $\{\mathbf{T}_\nu\}$, where $\nu = 0, 1, 2, \ldots$, which are based on a fresh mesh size function. By using every mesh of the sequence the problem is solved and, next appropriate error indicators in every element are calculated. The values of the error indicator are reduced to the nodes by an averaging method. Having the values of errors at nodes a continuous error function in the whole domain is constructed by using a piecewise linear interpolation. Clearly this is a simple plane for each element. Each error function spans three nodes. When this 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 the newly generated elements are.

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

### 6.1. Remeshing scheme

The algorithm for remeshing 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. Fixing an initial mesh size function,
3. mMesh generation with the mesh size function,
4. Solution of the problem given by equation 3.2 on the generated mesh,
5. Evaluation of error indicator in every element,
6. Calculation of nodal error indicator by using the averaging method,
7. Definition of the new mesh size function by using the errors found at every point,
8. If the error is not small enough, go to the point 3,
9. end of computations.

In the examples given later, it was sufficient to make 3 to 7 steps of adaptation.

### 6.2. Error indicators

The applied indicators are calculated for every element or directly at the nodes [7, 14]. Let $e_i$ for $i = 1, \ldots, n_0$ be an error indicator at *i-th* apex of the grid $\mathcal{T}_0$, and $\mathcal{P}_0$={ $P_i$,

$i = 1, \ldots, n_P\}$ − set of nodes. We define a patch of elements incidental for a given node $P_i$ as:

$$L_i = \{k : P_i \in \overline{T}_k\} \text{ for } i = 1, \ldots, n_P. \tag{6.1}$$

1. Let $N_i$ be a set of neighbours of *i-th* element:

$$N_i = \{k : \ T_k \ \textit{has a common edge with} \ T_i\}, \tag{6.2}$$

$$\text{then} \qquad \tilde{e}_i = \sqrt{\sum_{k \in N_i} \left( \frac{\partial u_i}{\partial n_k} - \frac{\partial u_k}{\partial n_k} \right)^2}, \tag{6.3}$$

   where $u_i$ is the restriction of the solution to *i-th* element and $n_k$ is unit normal to the the edge common of the *k-th* and the *i-th* element.

2. In this case, it is suggested to directly introduce the values of error indicator at every node of the mesh. The following error indicator is being adopted in the current program. From the numerical analyses it follows that the usage of this error indicator generates similar meshes to the one firstly defined.

$$e_i = \sqrt{\sum_{k \in L_i, \ l \in L_i, \ l \neq k} \left( \frac{\partial u_i}{\partial x} - \frac{\partial u_k}{\partial x} \right)^2 + \left( \frac{\partial u_i}{\partial y} - \frac{\partial u_k}{\partial y} \right)^2}, \tag{6.4}$$

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

### 6.3. 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 the multiplication 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 maximum and the greatest where the value of the error is minimum. The value increases when the error decreases. To describe the algorithm of the mesh size function modification, it is necessary to reduce the values of the error indicators to nodes. For every node $P_i$ the weighted averaged value of the indicator is defined as follows:

$$\tilde{e}_i = \frac{\sum_{k \in L_i} area(T_k) e_k}{\sum_{k \in L_i} area(T_k)}, \tag{6.5}$$

where

$$L_i = \{k : \ P_i \in T_k\} \ \text{ and } T_k \text{ is the } k\textit{-th} \text{ element.} \tag{6.6}$$

In such a manner, a set of values of the error at every nodal point is obtained.

$$\alpha = \min_{k=1,2,\ldots,N_{NOD}} \tilde{e}_k, \qquad \beta = \max_{k=1,2,\ldots,N_{NOD}} \tilde{e}_k, \tag{6.7}$$

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

The following new values are introduced:

$\lambda$ – a value indicating the greatest mesh size function reduction,

$\mu$ – a value indicating the smallest mesh size function reduction.

Usually $\lambda$ and $\mu$ have positive values and usually are one smaller than 1, additionally $\mu < \lambda$. The following transformation is defined

$$l : [\alpha, \beta] \mapsto [\mu, \lambda] \tag{6.8}$$

which satisfies the conditions: $l(\alpha) = \lambda$ and $l(\beta) = \mu$. By these assumptions it can be observed that $\mu \leq l(x) \leq \lambda$.

Provided that

$$Q_i = l(\tilde{e}_i) \ for \ i = 1, \ldots, N_{NOD}, \tag{6.9}$$

then one has: $\min_{i=1,2,\ldots,N_{NOD}} Q_i = \mu, \quad \max_{i=1,2,\ldots,N_{NOD}} Q_i = \lambda$.

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

$$\Pi(P_i) = Q_i \ for \ i = 1, 2, 3, \tag{6.10}$$

where $P_1$, $P_2$, $P_3$ are the vertices of the triangle $T_s$ of the triangulation of $\Omega$, and appropriately $Q_1$, $Q_2$, $Q_3$ are the values defined by the formula (6.9). The function $r(\overline{x})$ is defined in the whole domain because the triangles $\{\overline{T}_s\}_{s=1}^{n_e}$ cover it. The new mesh size function is defined as follows:

$$\gamma_{i+1}(\overline{x}) = \gamma_i(\overline{x}) r(\overline{x}). \tag{6.11}$$

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 can be checked that:

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

It can be shown that

$$||\gamma_{i+1} - \gamma_i||_{\overline{\Omega},max} \leq ||\gamma_i||_{\Omega,max} \max\{|1-\mu|, |1-\lambda|\} \tag{6.12}$$

$$\text{where} \quad ||\gamma||_{\Omega,max} := \max_{\overline{x} \in \Omega}\{|\gamma(\overline{x})|\}. \tag{6.13}$$

# 7. Numerical Examples

## 7.1. Numerical test for squared domain

The size function modification depends on an error indicator and on the coefficients $\lambda$, $\mu$, which determine the value of mesh size reduction. If the values of the coefficients $\lambda$, $\mu$ are small then a smaller number of adaptation steps is necessary. How quickly an adapted grid will be close enough to an optimal mesh, next to the error indicator, it depends on an initial mesh too. In the problems solved here, it was arbitrarily assumed that $\lambda = 0.6$ and $\mu = 0.9$. The proposed algorithm was benchmarked against a known problem of elastic-plastic torsion of a bar with a square cross-section, Figures 2 and 3 provides results. The distribution of elements is seen in Fig. 2. The isolines of the Prandtl-Reuss function are given in Fig. 3. The results given in Figures 2 - 4 were obtained for the angle of twist, $\omega = 0.015 \ \ rad/m^2$. The dependence between stress and strain intensities was (see Fig. 1):

$$T = \begin{cases} 8 \times 10^5 \Gamma & \text{if} = \Gamma \leq 0,0025, \\ 1940 + 24 \times 10^3 \Gamma & \text{otherwise.} \end{cases} \tag{7.1}$$

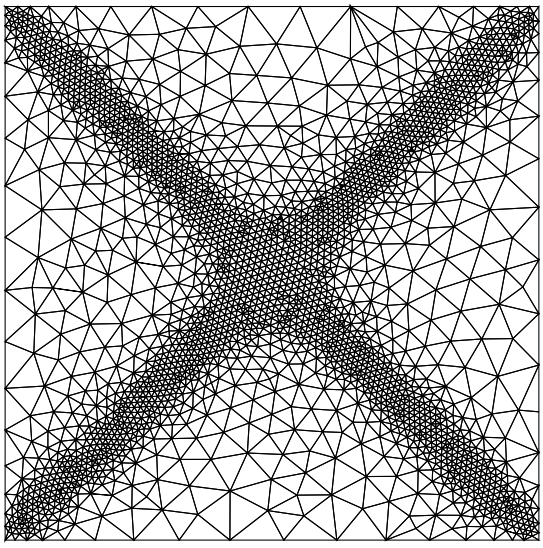

Fig. 2: Mesh after 7 adaptation steps

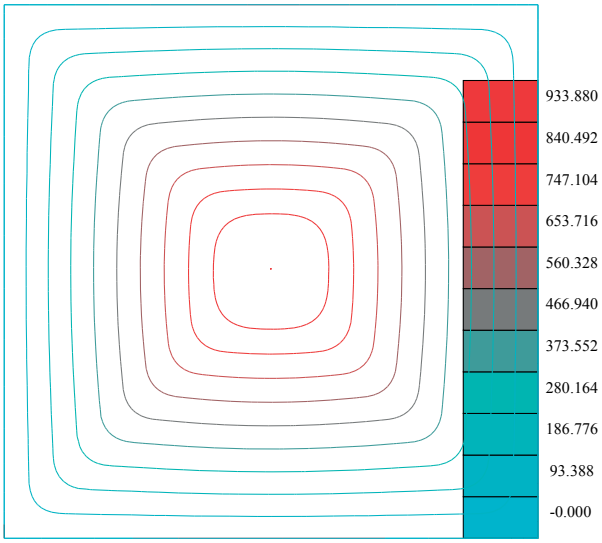


Fig. 3: Prandtl-Reuss function

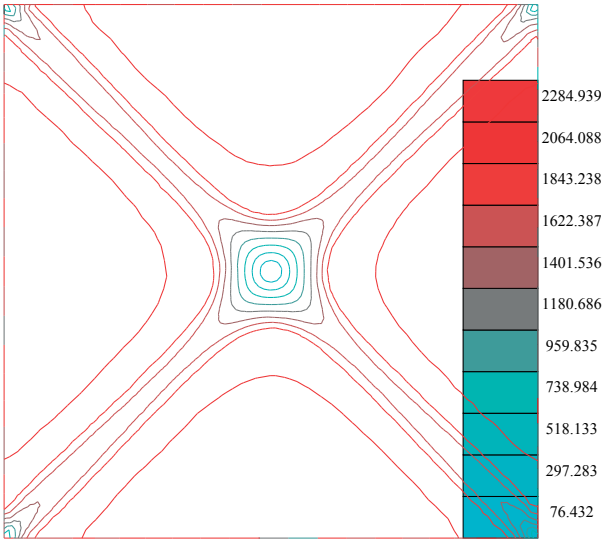


Fig. 4: Stress intensity, T

In nonlinear problems, it is sometimes useful to perform a greater number of adaptation steps. The success of the method is best illustrated in Figures 5 and 6, where a similar problem is solved for an L-shaped domain. It is seen here that when starting from a coarse mesh (Fig. 5) the algorithm quickly converges to a denser mesh (see Fig. 6) where large variations in stress take place. At the the same time the region where plastic staining took place there is no need for a dense grid (outer edges). Fig. 7 shows the Prandtl-Reuss function, and Fig. 4 depicts the stress contours.

It would be rather impossible to obtain the same effect by the methods based on mesh enrichment [1, 3, 10].

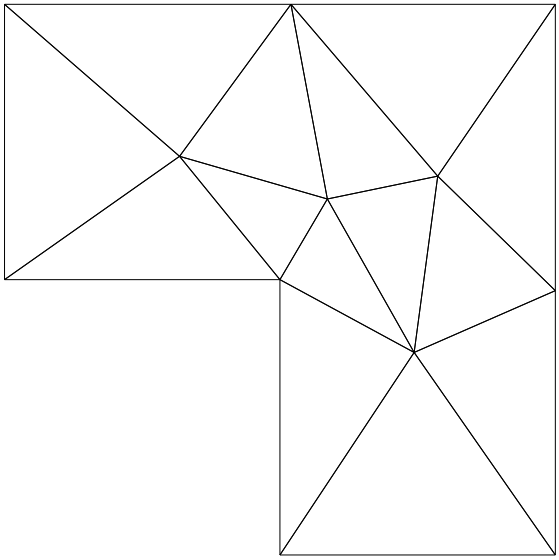

Fig. 5: Initial mesh

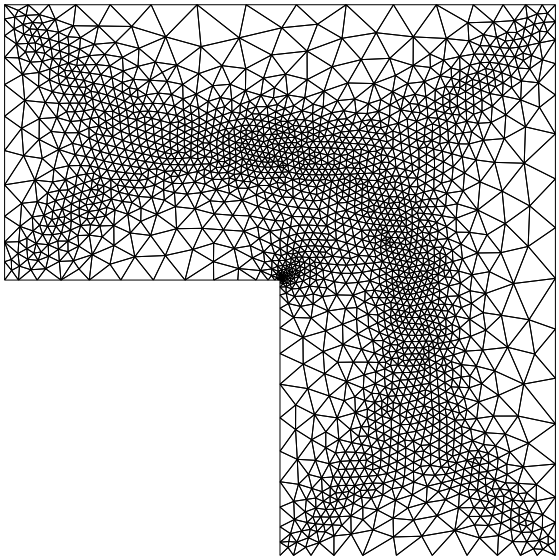

Fig. 6: Final mesh after 7 adaptation steps

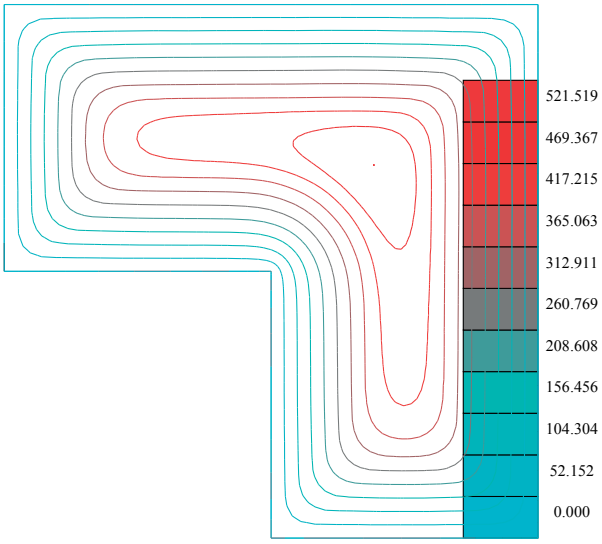


Fig. 7: Prandtl-Reuss function

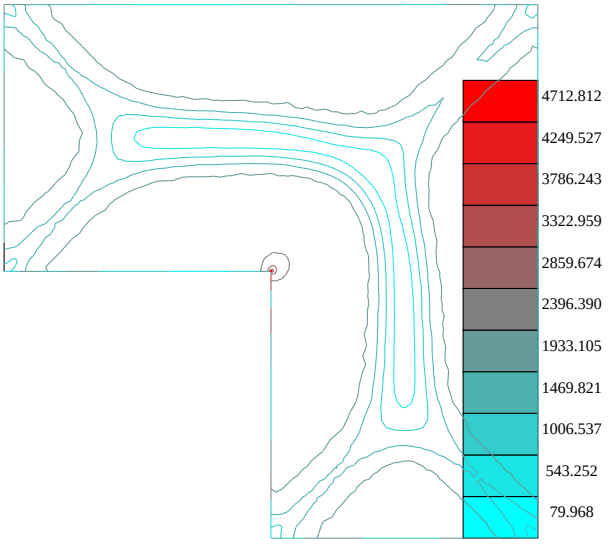


Fig. 8: Stress intensity, T

## 8. Summary and closure

- This paper presents the application of the grid generator with mesh size function of the adapted solution to the elastic-plastic twisting of bars with strain hardening.
- The mesh generator based on the Delaunay condition and the advancing front technique seems very suitable to the class of problems where zones of different in the domain are to be represented.
- After the discretization, the problem led to the solution of nonlinear systems of algebraic equations. For the considered problems, the applied Newton-Raphson method was always convergent giving the residual error of an order $10^{-10}$ in $10 - 14$ iterations.
- The whole algorithm has two loops, the external over adaptation steps and the internal over the Newton-Raphson iterations.
- The effiecient mesh size function is obtained iteratively and it is linked to the values of error indicators at nodes.
- The presented method, with its superiority above other known approaches, makes it a suitable tool for nonlinear model computations.
- An investigation of the anisotropic mesh generation would be interesting.

## References


[1] Bank R., Sherman A. and Weiser A., *Some Refinement Algorithms And Data Structures for Regular Local Mesh Refinement. Scientific Computing IMACS,* 1983.

[2] Bieterman M.B., Busseletti J.E., Hilmes C.L., Johnson F.T., Melvin R.G., Young D.P., *An adaptive grid method for analysis of 3D aircraft configurations, The Boeing Company Seattle*, Technical Report, Washington 1991.

[3] Borouchaki H., Hecht F., Frey P.J., *Mesh gradation control,* Int. J. Num. Meth. in Engng, **43**, 1998 1143—1165.

[4] Huerta A., Diez P., *Error estimation including pollution assessment for nonlinear finite element analysis,* Comp. Meth. Appl. Mech. Engng, **181**, 2000 21—41,.

[5] Kachanov L. M., *Fundamentals of plasticity theory of plasticity, Dover Publications Inc.,* 2004, 479p. (ISBN 0-486-43583-0), Mineola, NY, USA. Moscow 1968.

[6] Thompson J.F., Soni B. K., Weatherwill N.P., *Handbook of Grid Generation,* CRC Press, Boca Raton, 1999.

[7] Kucwaj J., *The Algorithm of Adaptation by Using Graded Meshes Generator,* Computer Assisted Mechanics and Engineering Sciences, **7**, 2000, 615—624.

[8] Kucwaj J., *Numerical Investigations of the Covergence of a Remeshing Algorithm on an Example of Subsonic Flow,* Computer Assisted Mechanics and Engineering Sciences, **17**, 2010, 147—160.

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

[10] 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. Engng., **77**, 1989, 113—180.

[11] Rachowicz W., *An anisotropic h-type mesh-refinement strategy,* Comp. Meth. Appl. Mech. Engng, **109** 1993, 169—181.

[12] 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.

[13] Zienkiewicz O.C.,*Achievements and some unsolved problems of the finite element method,* Int. J. Num. Meth. Engng, **47**, 2000, 9—28.

[14] Zienkiewicz O.C., Zhu J.Z., *Adaptivity and mesh generation,* Int. J. Num. Meth. Engng., **32**, 1991, 783—810.

[15] MAdLib: an open source Mesh Adaptation Library, http://sites.uclouvain.be/madlib/ 2010.