XIII International Conference on Computational Plasticity. Fundamentals and Applications
COMPLAS XIII
E. Oñate, D.R.J. Owen, D. Peric & M. Chiumenti (Eds)

HOW TO SIMPLIFY RETURN-MAPPING ALGORITHMS IN COMPUTATIONAL PLASTICITY: PART 2 - IMPLEMENTATION DETAILS AND EXPERIMENTS

MARTIN CERMAK[*,†] AND STANISLAV SYSALA[†]

[*] VSB-TU Ostrava (VSB-TU)
IT4Innovations National Supercomputing Center (IT4I)
Studentska 6231/1b, 708 00 Ostrava, Czech Republic
email: martin.cermak@vsb.cz - web page: http://www.it4i.cz

[†] Institute of Geonics AS CR (IGN)
Department of IT4Innovations (IT4I)
Studentska 1768, 708 00 Ostrava, Czech Republic
e-mail: stanislav.sysala@ugn.cas.cz - web page: http://www.ugn.cas.cz/˜sysala

Key words: Computational Plasticity, Drucker-Prager Criterion, Semismooth Newton Method

Abstract. The paper is devoted to numerical solution of a small-strain quasi-static elastoplastic problem. It is considered an isotropic model containing the Drucker-Prager yield criterion, a non-associative flow rule and a nonlinear hardening law. The problem is discretized by the implicit Euler and finite element methods. It is used an improved return-mapping scheme introduced in "PART 1" and the semismooth Newton method. Algorithmic solution is described and efficiency of the improved scheme is illustrated on numerical examples.

1 INTRODUCTION

The paper is devoted to numerical solution of a small-strain quasi-static elastoplastic problem that contains the Drucker-Prager yield criterion, a non-associative flow rule and a nonlinear hardening law. Such a problem consists of the constitutive initial value problem (CIVP) and the balance equation representing the principle of virtual work. A broadly exploited and universal numerical/computational concept includes the following steps: (a) the implicit Euler discretisation of CIVP leading to an incremental constitutive law; (b) substitution of the law into the balance equation; (c) finite element discretization of this boundary value problem formulated in terms of displacements; (d) solving a related system of nonlinear equations.

The incremental constitutive problem is usually solved by the elastic predictor/plastic corrector method. The plastic correction is called the (implicit) return-mapping scheme. In "PART 1" [3], there was introduced a straightforward simplification of the current return-mapping scheme which can be found e.g. in [6, Chapter 8]. The improved scheme is based on a subdifferential formulation of the plastic flow rule. It leads to a priori information whether the unknown stress tensor lies on the smooth portion or at the apex of the Drucker-Prager yield surface even if the nonlinear isotropic hardening is considered within the model.

Further, the discretized incremental boundary value problem which leads to the system of non-linear equations is usually solved by the Newton method or its modifications. More precisely, one must use a nonsmooth version of the method since the stress-strain operator received from the constitutive problem is nonsmooth. In [7], the semismooth Newton method was introduced. The method can have local quadratic convergence under certain assumptions. In particular, it is necessary to show that the stress-strain operator is strongly semismooth. Such a treatment can be found in [8, 9] for the Drucker-Prager model. From the computational point of view, there is a minimal difference between the standard and semismooth Newton methods. One must only suitably extend the definition of the consistent tangent operator where the derivative $\partial\boldsymbol{\sigma}/\partial\boldsymbol{\varepsilon}$ does not exists. Here, $\boldsymbol{\sigma}, \boldsymbol{\varepsilon}$ denotes stress, and strain, respectively.

In Section 2 and 3, we introduce a few implementation details of the discretized incremental boundary value problem, mainly the improved return-mapping and the semismooth Newton method. We confine ourselves on a plane strain problem, linear simplician elements and algebraic notation. Algorithmic solution to a 3D problem is discussed in [1]. In Section 4, we illustrate the efficiency of the presented algorithm on numerical examples.

2 ALGEBRAIC FORMULATION OF THE PROBLEM

In this section, we sketch the incremental boundary value elastoplastic problem in an algebraic form and summarize the semismooth Newton method. For more details, we refer e.g. [2].

Assume a deformable body from an elastoplastic homogenoues material. The material model contains the Drucker-Prager yield criterion, the non-associative flow rule and the nonlinear hardening law as in [3, 6]. For the sake of simplicity, consider a plane strain problem. Let Ω be a polygonal 2D domain represented the cross-section of the investigated body and $\mathcal{T}_h$ denote its triangulation. Further, consider linear and conforming elements. So displacement fields are approximated by continuous and piecewise linear functions and strain, stress and the isotropic hardening fields are approximated by piecewise constant functions. Assume that the history of loading is prescribed in the interval $[t_0, t_{max}]$. Consider the partition

$$t_0 < t_1 < \ldots < t_k < \ldots < t_{\max}$$

of this interval and the implicit Euler discretization of the initial value constitutive prob-

lem introduced in "PART 1" [3].

Let $\mathcal{N}$ denote a number of nodal points of $\mathcal{T}_h$ and set $n = 2\mathcal{N}$. Define the space

$$\mathcal{V} := \{\boldsymbol{v} \in \mathbb{R}^n |\ \boldsymbol{B}_D \boldsymbol{v} = \boldsymbol{o}\},$$

where $\boldsymbol{B}_D \in \mathbb{R}^{m\times n}$ is the restriction matrix represented prescribed Dirichlet boundary conditions. Then admissible displacement vectors at the k-th step belong to the set $\boldsymbol{u}_{k,N} + \mathcal{V}$ where the vector $\boldsymbol{u}_{k,N} := \boldsymbol{u}_N(t_k)$ includes eventual nonhomogeneous Dirichlet conditions.

Strain, plastic strain and stress tensors are represented by the following vectors

$$\begin{array}{ll} \boldsymbol{\varepsilon} = (\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12})^T, & \tilde{\boldsymbol{\varepsilon}} = (\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}, 0)^T, \\ \boldsymbol{\varepsilon}^p = (\varepsilon^p_{11}, \varepsilon^p_{22}, 2\varepsilon_{12})^T, & \tilde{\boldsymbol{\varepsilon}}^p = (\varepsilon^p_{11}, \varepsilon^p_{22}, 2\varepsilon^p_{12}, \varepsilon^p_{33})^T, \\ \boldsymbol{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{12})^T, & \tilde{\boldsymbol{\sigma}} = (\sigma_{11}, \sigma_{22}, \sigma_{12}, \sigma_{33})^T, \end{array}$$

respectively. Let $\bar{\varepsilon}^p$ denote an isotropic hardening variable. The nonlinear and implicit stress-strain operator is represented by a function T. In particular, we write

$$\boldsymbol{\sigma}_k = T(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}), \quad k = 1, 2 \ldots, m,$$

at the k-th step. For purposes of the semismooth Newton method, we also need a function $T^o := T^o(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) \in \mathbb{R}^{3\times 3}$, representing a generalized derivative $\partial\boldsymbol{\sigma}_k/\partial\boldsymbol{\varepsilon}_k$. The algebraic forms of the operators T and T^o are introduced in the next section.

Further, we introduce a matrix $\boldsymbol{R}_\triangle \in \mathbb{R}^{6\times n}$ that restricts a displacement vector $\boldsymbol{v} \in \mathbb{R}^n$ on an element $\triangle \in \mathcal{T}_h$, i.e.

$$\boldsymbol{v}_\triangle = \boldsymbol{R}_\triangle \boldsymbol{v}.$$

The stress-displacement relation on an element $\triangle \in \mathcal{T}_h$ is represented by the matrix $\boldsymbol{G}_\triangle \in \mathbb{R}^{3\times 6}$, i.e.

$$\boldsymbol{\varepsilon}_\triangle = (\varepsilon_{11,\triangle}, \varepsilon_{22,\triangle}, 2\varepsilon_{12,\triangle})^T = \boldsymbol{G}_\triangle \boldsymbol{R}_\triangle \boldsymbol{v}.$$

Finally, denote $\boldsymbol{f}_k \in \mathbb{R}^n$ as the load vector at the time t_k and define the operators $F_k : \mathbb{R}^n \to \mathbb{R}^n$, $K_k : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ representing the system of nonlinear equations and the tangential stiffness matrix, respectively:

$$\begin{aligned} F_k(\boldsymbol{v}) &= \sum_{\triangle\in\mathcal{T}_h} |\triangle| \left(T(\boldsymbol{G}_\triangle \boldsymbol{R}_\triangle \boldsymbol{v}; \tilde{\boldsymbol{\varepsilon}}^p_{k-1,\triangle}, \bar{\varepsilon}^p_{k-1,\triangle})\right)^T \boldsymbol{G}_\triangle \boldsymbol{R}_\triangle, \quad \boldsymbol{v} \in \mathbb{R}^n, \\ K_k(\boldsymbol{v}) &= \sum_{\triangle\in\mathcal{T}_h} |\triangle| \left(T^o(\boldsymbol{G}_\triangle \boldsymbol{R}_\triangle \boldsymbol{v}; \tilde{\boldsymbol{\varepsilon}}^p_{k-1,\triangle}, \bar{\varepsilon}^p_{k-1,\triangle}) \boldsymbol{G}_\triangle \boldsymbol{R}_\triangle\right)^T \boldsymbol{G}_\triangle \boldsymbol{R}_\triangle, \quad \boldsymbol{v} \in \mathbb{R}^n. \end{aligned}$$

Then the incremental elastoplastic problem reads as follows:

$$\text{find } \boldsymbol{u}_k \in \boldsymbol{u}_{k,N} + \mathcal{V}: \qquad \boldsymbol{v}^T \left[F_k(\boldsymbol{u}_k) - \boldsymbol{f}_k\right] = 0 \quad \forall \boldsymbol{v} \in \mathcal{V}.$$

This leads to the system of nonlinear equations after elimination the rows corresponding to the Dirichlet boundary conditions. The system can be solved by the semismooth Newton method.

Algorithm 1 (Semismooth Newton method).

1: initialization: $\boldsymbol{u}_k^0 = \boldsymbol{u}_{k-1}$
2: **for** $i = 0, 1, 2, \dots$ **do**
3: find $\delta\boldsymbol{u}^i \in \mathcal{V}$: $\boldsymbol{v}^T K_k(\boldsymbol{u}_k^i)\delta\boldsymbol{u}^i = \boldsymbol{v}^T\left[\boldsymbol{f}_k - F_k(\boldsymbol{u}_k^i)\right] \quad \forall \boldsymbol{v} \in \mathcal{V}$
4: compute $\boldsymbol{u}_k^{i+1} = \boldsymbol{u}_k^i + \delta\boldsymbol{u}^i$
5: **if** $\|\boldsymbol{u}_k^{i+1} - \boldsymbol{u}_k^i\|/(\|\boldsymbol{u}_k^{i+1}\| + \|\boldsymbol{u}_k^i\|) \le \epsilon_{Newton}$ **then stop**
6: **end for**
7: set $\boldsymbol{u}_k = \boldsymbol{u}_k^{i+1}$

3 CONSTRUCTION OF T AND T^o

As follows from Algorithm 1, it is necessary to compute values of $T(\boldsymbol{\varepsilon}_{k,\triangle}^i; \tilde{\boldsymbol{\varepsilon}}_{k-1,\triangle}^p, \bar{\varepsilon}_{k-1,\triangle}^p)$ and $T^o(\boldsymbol{\varepsilon}_{k,\triangle}^i; \tilde{\boldsymbol{\varepsilon}}_{k-1,\triangle}^p, \bar{\varepsilon}_{k-1,\triangle}^p)$ for any $\triangle \in \mathcal{T}_h$, where $\boldsymbol{\varepsilon}_{k,\triangle}^i = \boldsymbol{G}_\triangle \boldsymbol{R}_\triangle \boldsymbol{u}_k^i$. To simplify the notation, we omit the indices i and $\triangle$. So, let the vectors $\boldsymbol{\varepsilon}_k$, $\tilde{\boldsymbol{\varepsilon}}_{k-1}^p$ and the scalar $\bar{\varepsilon}_{k-1}^p$ be given and denote $\boldsymbol{\varepsilon}_k^{e,tr} := \boldsymbol{\varepsilon}_k - \boldsymbol{\varepsilon}_{k-1}^p$, $\tilde{\boldsymbol{\varepsilon}}_k^{e,tr} := \tilde{\boldsymbol{\varepsilon}}_k - \tilde{\boldsymbol{\varepsilon}}_{k-1}^p$ and $\bar{\varepsilon}_k^{p,tr} = \bar{\varepsilon}_{k-1}^p$.

We use the results introduced in "PART 1" [3, Section 3]. Recall that the investigated Drucker-Prager model is given by the parameters $E, \nu, \eta, \bar{\eta}, \xi, c_0$ and by the function H. We standardly set $K = \frac{E}{3(1-2\nu)}$ and $G = \frac{E}{2(1+\nu)}$.

Let $\boldsymbol{\iota} = (1,1,0)^T$, $\tilde{\boldsymbol{\iota}} = (1,1,0,1)^T$,

$$\boldsymbol{I}_{dev} = \begin{pmatrix} 2/3 & -1/3 & 0 \\ -1/3 & 2/3 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}, \quad \tilde{\boldsymbol{I}}_{dev} = \begin{pmatrix} 2/3 & -1/3 & 0 & -1/3 \\ -1/3 & 2/3 & 0 & -1/3 \\ 0 & 0 & 1/2 & 0 \\ -1/3 & -1/3 & 0 & 2/3 \end{pmatrix}$$

and $\boldsymbol{D}_e = K\boldsymbol{\iota}\boldsymbol{\iota}^T + 2G\boldsymbol{I}_{dev}$, $\tilde{\boldsymbol{D}}_e = K\tilde{\boldsymbol{\iota}}\tilde{\boldsymbol{\iota}}^T + 2G\tilde{\boldsymbol{I}}_{dev}$. Further, we compute the following trial vectors and scalars:

$$\begin{aligned}
&\tilde{\boldsymbol{\sigma}}_k^{tr} = \tilde{\boldsymbol{D}}_e \tilde{\boldsymbol{\varepsilon}}_k^{e,tr}, && \tilde{\boldsymbol{\sigma}}_k^{tr} = \left((\boldsymbol{\sigma}_k^{tr})^T, \sigma_{k,33}^{tr}\right)^T, \\
&\tilde{\boldsymbol{s}}_k^{tr} = 2G\tilde{\boldsymbol{I}}_{dev}\tilde{\boldsymbol{\varepsilon}}_k^{e,tr}, && \tilde{\boldsymbol{s}}_k^{tr} = \left((\boldsymbol{s}_k^{tr})^T, s_{k,33}^{tr}\right)^T, \\
&p_k^{tr} = K\tilde{\boldsymbol{\iota}}^T\tilde{\boldsymbol{\varepsilon}}_k^{e,tr}, && \varrho_k^{tr} = 2G\sqrt{(\tilde{\boldsymbol{\varepsilon}}_k^{e,tr})^T\tilde{\boldsymbol{I}}_{dev}\tilde{\boldsymbol{\varepsilon}}_k^{e,tr}}, \\
&\boldsymbol{n}_k^{tr} = \frac{\boldsymbol{s}_k^{tr}}{\varrho_k^{tr}}, && \tilde{\boldsymbol{n}}_k^{tr} = \frac{\tilde{\boldsymbol{s}}_k^{tr}}{\varrho_k^{tr}}, \quad \text{if } \varrho_k^{tr} \neq 0.
\end{aligned}$$

To decide whether the unknown stress $\tilde{\boldsymbol{\sigma}}_k = (\sigma_{11,k}, \sigma_{22,k}, \sigma_{12,k}, \sigma_{33,k})^T$ lies in the elastic region, the smooth portion or at the apex of the yield surface, we define the function

$$q(\gamma) = \sqrt{\frac{1}{2}}\left(\varrho_k^{tr} - \gamma G\sqrt{2}\right)^+ + \eta(p_k^{tr} - \gamma K\bar{\eta}) - \xi\left(c_0 + H(\bar{\varepsilon}_k^{p,tr} + \gamma\xi)\right), \quad \gamma \ge 0.$$

The elastic response. It happens if and only if $q(0) \leq 0$. Then we set

$$T(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) = \boldsymbol{\sigma}^{tr}, \quad T^o(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) = \boldsymbol{D}_e, \quad \tilde{\boldsymbol{\varepsilon}}^p_k = \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \quad \bar{\varepsilon}^p_k = \bar{\varepsilon}^p_{k-1}.$$

The return to the smooth portion. It happens if and only if $q(0) > 0$ and $q\left(\varrho^{tr}_k / G\sqrt{2}\right) < 0$. Then $\triangle\lambda \in (0, \varrho^{tr}_k / G\sqrt{2})$ and satisfies

$$0 = q(\triangle\lambda) = \sqrt{\frac{1}{2}}\left(\varrho^{tr}_k - \triangle\lambda G\sqrt{2}\right) + \eta(p^{tr}_k - \triangle\lambda K\bar{\eta}) - \xi\left(c_0 + H(\bar{\varepsilon}^{p,tr}_k + \triangle\lambda\xi)\right).$$

After finding $\triangle\lambda$ we set $H_1 = H'(\bar{\varepsilon}^{p,tr}_k + \triangle\lambda\xi)$ and

$$\begin{aligned}
T(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) &= \boldsymbol{\sigma}^{tr}_k - \triangle\lambda\left(G\sqrt{2}\boldsymbol{n}^{tr}_k + K\bar{\eta}\boldsymbol{\iota}\right), \\
T^o(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) &= \boldsymbol{D}_e - \triangle\lambda\frac{2G^2\sqrt{2}}{\varrho^{tr}_k}\left(\boldsymbol{I}_{dev} - \boldsymbol{n}^{tr}_k(\boldsymbol{n}^{tr}_k)^T\right) - \\
&\qquad -(G\sqrt{2}\boldsymbol{n}^{tr}_k + K\bar{\eta}\boldsymbol{\iota})\left(\frac{G\sqrt{2}\boldsymbol{n}^{tr}_k + \eta K\boldsymbol{\iota}}{G + K\eta\bar{\eta} + \xi^2 H_1}\right)^T, \\
\tilde{\boldsymbol{\varepsilon}}^p_k &= \tilde{\boldsymbol{\varepsilon}}^p_{k-1} + \triangle\lambda\left(\sqrt{\frac{1}{2}}\tilde{\boldsymbol{n}}^{tr}_k + \frac{\bar{\eta}}{3}\tilde{\boldsymbol{\iota}}\right), \\
\bar{\varepsilon}^p_k &= \bar{\varepsilon}^p_{k-1} + \triangle\lambda\xi.
\end{aligned}$$

The return to the apex. It happens if and only if $q\left(\varrho^{tr}_k / G\sqrt{2}\right) \geq 0$. Then $\triangle\lambda \geq \varrho^{tr}_k / G\sqrt{2}$ and satisfies

$$0 = q(\triangle\lambda) = \eta(p^{tr}_k - \triangle\lambda K\bar{\eta}) - \xi\left(c_0 + H(\bar{\varepsilon}^{p,tr}_k + \triangle\lambda\xi)\right).$$

After finding $\triangle\lambda$ we set $H_1 = H'(\bar{\varepsilon}^{p,tr}_k + \triangle\lambda\xi)$ and

$$\begin{aligned}
T(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) &= \left(p^{tr}_k - \triangle\lambda K\bar{\eta}\right)\boldsymbol{\iota}, \\
T^o(\boldsymbol{\varepsilon}_k; \tilde{\boldsymbol{\varepsilon}}^p_{k-1}, \bar{\varepsilon}^p_{k-1}) &= \frac{\xi^2 K H_1}{K\eta\bar{\eta} + \xi^2 H_1}\boldsymbol{\iota}\boldsymbol{\iota}^T, \\
\tilde{\boldsymbol{\varepsilon}}^p_k &= \tilde{\boldsymbol{\varepsilon}}_k - \frac{p^{tr}_k - \triangle\lambda K\bar{\eta}}{3K}\tilde{\boldsymbol{\iota}}, \\
\bar{\varepsilon}^p_k &= \bar{\varepsilon}^p_{k-1} + \triangle\lambda\xi.
\end{aligned}$$

Notice that the function T^o is defined everywhere. If the derivative $\partial T / \partial\boldsymbol{\varepsilon}_k$ exists then $T^o = \partial T / \partial\boldsymbol{\varepsilon}_k$.

4 NUMERICAL EXPERIMENTS

To illustrate the efficiency of the presented algorithmic solution we choose the strip footing benchmark introduced in [6], which is solved as a plane strain problem. We use the same geometry and material parameters as in [6] for comparison. A symmetric half of the cross section is depicted in Figure 1. It is a square with the length 5m. The homogeneous Dirichlet boundary conditions in the normal direction are prescribed on the left, right, and bottom sides. The footing length is $B = 1$ m, i.e., the length $B/2$ is depicted in Figure 1. The loading is controlled by the vertical displacement, u, which is assumed to be uniform under the footing. A total displacement $u = 20.15$ mm is applied in 29 increments. The material parameters are defined as follows:

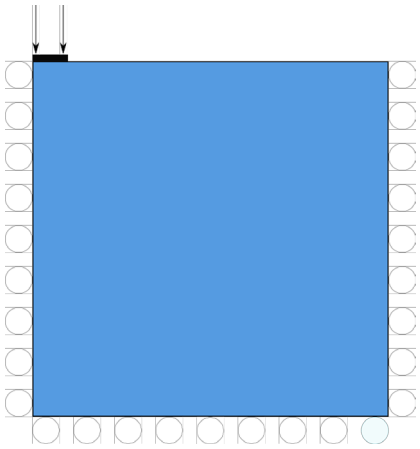

Figure 1: Geometry

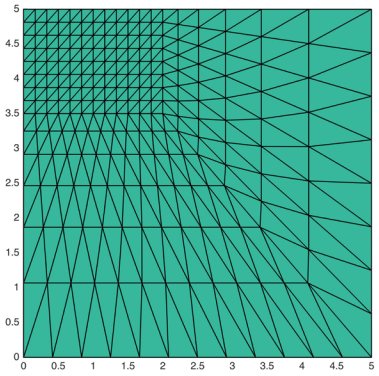

Figure 2: Mesh

$$E = 10^7\,\text{kPa},\ \nu = 0.48,\ c_0 = 490\,\text{kPa},\ \phi = 20^\circ,\ \text{and}\ \psi = 20^\circ,$$

$$\eta = \frac{3\tan(\phi)}{\sqrt{9+12(\tan(\phi))^2}},\quad \overline{\eta} = \frac{3\tan(\psi)}{\sqrt{9+12(\tan(\psi))^2}},\quad \xi = \frac{3}{\sqrt{9+12(\tan(\phi))^2}}.$$

Further, we compare results for two constant functions H: $H = 0$ kPa (perfect plasticity) and $H = 1000$ kPa (linear hardening).

We use three different triangular meshes with 511, 1561, and 4497 nodes. Their scheme is depicted in Figure 2. One can see that the meshes are regular and finer in vicinity of the footing. The tolerance of the Newton method is $\epsilon_{Newton} = 10^{-5}$. The problem has been implemented in `MatSol` library [5, 4, 2] developed in Matlab.

Comparison of loading paths for the investigated meshes and the hardening functions is depicted in Figure 3 and 4. Here, P denote computed average pressure supported by the footing. We observe dependence on a mesh parameter. The dependence is not so significant for the two finer meshes. The curves are comparable with ones introduced in

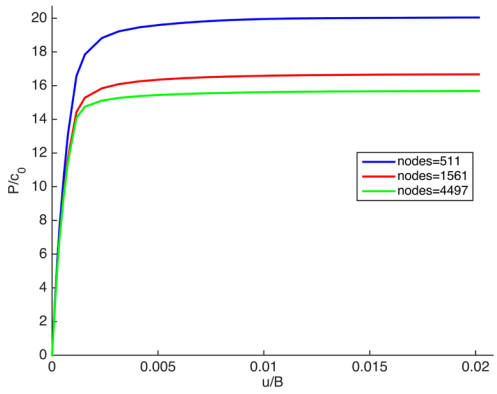

Figure 3: Loading paths for perfect plasticity.

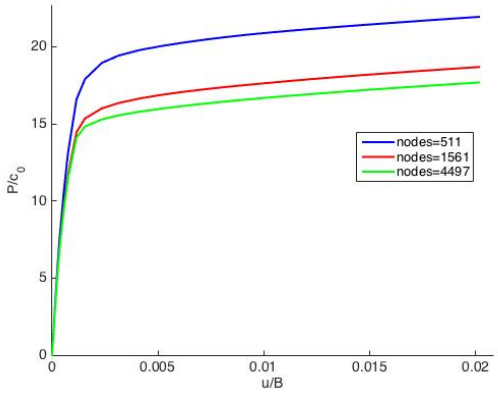

Figure 4: Loading paths for linear hardening.

[6] where eight-noded quadrilaterals with fourth-point quadrature were used to reduce a possible locking effect.

For illustration, we add Figure 5 and 6 depicted displacements in the vertical direction and the plastic strain increments $\triangle\varepsilon_k^p = \|\boldsymbol{\varepsilon}_k^p\| - \|\boldsymbol{\varepsilon}_{k-1}^p\|$, respectively. The figures corresponds to the end of the loading process, the perfect plasticity and the finest mesh.

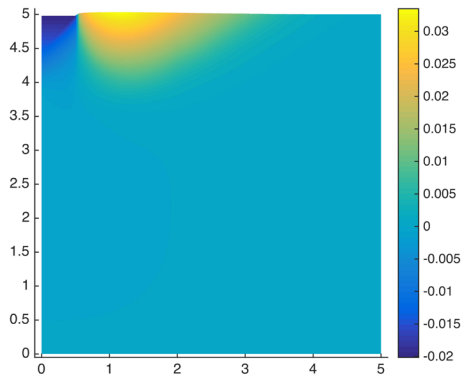

Figure 5: Vertical displacements for the perfect plasticity and the finest mesh.

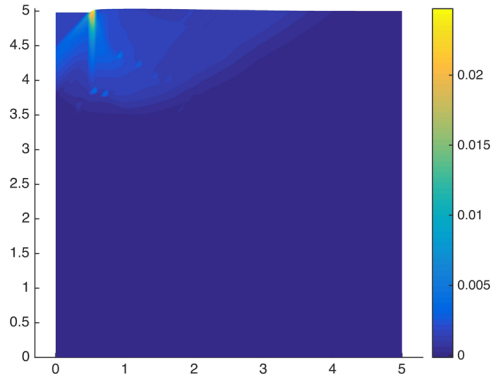

Figure 6: Plastic strain increments for perfect plasticity and the finest mesh.

In Table 1, iteration numbers in selected loading steps are summarized. The numbers are small in each time step. The convergence is slowest around the loading step $k = 13$ where $u = 4.15$ mm and the loading curves are strongly nonlinear. In Table 2, values of the criterion introduced in Algorithm 1 are summarized for the perfect plasticity and the finest mesh. It is readily seen that the convergence of the semismooth Newton method is at least superlinear.

Table 1: Numbers of Newton iterations in selected loading steps.

k	2	4	9	13	20	29
u [mm]	0.05	0.15	1.15	4.15	11.15	20.15
	Perfect plasticity					
511 nodes	2	3	6	8	4	3
1 561 nodes	2	4	6	12	4	3
4 497 nodes	3	4	6	12	4	3
	Linear hardening					
511 nodes	2	3	6	6	3	3
1 561 nodes	2	4	6	7	3	3
4 497 nodes	3	4	6	8	4	3

Table 2: Convergence in selected loading steps for the perfect plasticity and the finest mesh.

$i \mid k$	2	4	9	13	20	29
1	3.33e-01	1.98e-01	2.31e-01	1.75e-01	5.15e-02	2.71e-02
2	1.69e-04	1.35e-03	4.48e-02	2.42e-03	2.11e-04	3.90e-05
3	5.83e-07	8.74e-05	1.21e-02	1.31e-03	4.90e-05	7.38e-07
4		2.79e-07	1.29e-03	9.45e-04	6.57e-06	
5			5.73e-05	3.34e-04		
6			5.30e-07	2.15e-04		
7				1.81e-04		
8				8.85e-05		
9				1.54e-04		
10				3.27e-05		
11				3.42e-05		
12				5.52e-06		

Finally, we also considered the following nonlinear isotropic hardening:

$$c_0 = 450\text{kPa}, \quad H(\bar{\varepsilon}^p) = \min\{1000\bar{\varepsilon}^p; 40\}\,\text{kPa}.$$

The computed loading curves visually coincide with the curves for perfect plasticity. Moreover, the superlinear convergence is also observed.

5 CONCLUSIONS

In this paper, the improved solution algorithm for the Drucker-Prager elastoplastic problem was introduced. Its efficiency was checked on the well-known strip footing benchmark. The results are comparable with [6] and the quadratic convergence was observed.

Acknowledgements

This work has been supported by the project 13-18652S (GA CR) and the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

REFERENCES

[1] Cermak, M., Kruis J., Koudelka, T., Sysala, S. and Zeman, J. Simplified numerical realization of elastoplastic constitutive problems: PART I - criteria given by Haigh-Westergaard coordinates. *arXiv:1503.03605*, (2015).

[2] Cermak, M., Kozubek, T., Sysala, S. ,Valdman, J. A TFETI Domain Decomposition Solver for Elastoplastic Problems. *Applied Mathematics and Computation* (2014) **231**: 634–653.

[3] Cermak, M. and Sysala, S. How to Simplify Return-Mapping Algorithms in Computational Plasticity: PART 1 - Main Idea. *To appear in: XIII International Conference on Computational Plasticity. Fundamentals and Applications - COMPLAS XIII*. E. Oñate, D.R.J. Owen, D. Peric & M. Chiumenti (Eds), (2015).

[4] Dostál, Z., Horák, D., Kučera, R., *Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE*, Commun. Numer. Methods Eng. 22 (12), 1155–1162, 2006.

[5] Kozubek, T., Markopoulos, A., Brzobohatý, T., Kučera, R., Vondrák, V., Dostál, Z. *MatSol - MATLAB efficient solvers for problems in engineering*, http://matsol.vsb.cz/.

[6] de Souza Neto, E.A., Perić, D. and Owen, D. R. J. *Computational methods for plasticity: theory and application.* Wiley, (2008).

[7] Qi, L., Sun, J. A nonsmooth version of Newton's method. *Math. Program.* (1993) **58**: 353–367.

[8] Sauter, M., Wieners, C. On the superlinear convergence in computational elastoplasticity. *Comp. Meth. Eng. Mech.* (2011) **200**:3646–3658.

[9] Sysala, S. Properties and simplifications of constitutive time-discretized elastoplastic operators. *ZAMM - Z. Angew. Math. Mech.* (2014) **94**:233-255 .