XI International Conference on Computational Plasticity. Fundamentals and Applications
COMPLAS XI
E. Oñate, D.R.J. Owen, D. Peric and B. Suárez (Eds)

AN EFFICIENT FETI BASED SOLVER FOR ELASTO-PLASTIC PROBLEMS OF MECHANICS

M. ČERMAK, T. KOZUBEK AND A. MARKOPOULOS

Department of Applied Mathematics
Faculty of Electrical Engineering and Computer Science
VSB-Technical University of Ostrava
Tr. 17. listopadu 15, CZ 708 33 Ostrava-Poruba, Czech Republic
e-mail: martin.cermak@vsb.cz, http://am.vsb.cz/

Key words: Computational Elasto-Plasticity, TFETI, Domain Decomposition, MatSol

Abstract. This paper illustrates how to implement effectively solvers for elasto-plastic problems. We consider the time step problems formulated by nonlinear variational equations in terms of displacements. To treat nonlinearity and nonsmoothnes we use semismooth Newton method. In each Newton iteration we have to solve linear system of algebraic equations and for the numerical solution of the linear systems we use TFETI algorithm. In our benchmark we compute von Misses plasticity with isotropic hardening and use return mapping concept.

1 INTRODUCTION

The goal of paper is to show how to implement effectively solvers for elasto-plastic problems. Such problems with hardening lead to quasi-static initial-boundary value problems, so the history of loading is taken into account. The problems are often solved by an incremental finite element method, see e.g [1]. For the time-discretisation we can use the explicit or implicit Euler methods or the return mapping concept. Each time-step problem may be formulated in different ways by variational equalities or inequalities described in terms of stress, plastic strain, hardening parameter, and displacements. In this paper, we consider the time-step problems formulated by nonlinear variational equations in terms of displacements. To treat nonlinearity and non-smoothness we use the semismooth Newton method introduced in [2] and used in [3] for elasto-plastic problems.

In each Newton iteration we have to solve an auxiliary (possibly of large size) linear system of algebraic equations. The key idea of our approach is to use for the numerical solution of the linear systems arising in each Newton step the FETI method with optimal convergence properties proposed by Farhat et al. [4] for parallel solution of linear problems. Using this approach, a body is partitioned into non-overlapping subdomains, an

elliptic problem with Neumann boundary conditions is defined for each subdomain, and intersubdomain field continuity is enforced via Lagrange multipliers. The Lagrange multipliers are evaluated by solving a relatively well conditioned dual problem of small size that may be efficiently solved by a suitable variant of the conjugate gradient algorithm. The first practical implementations exploited only the favorable distribution of the spectrum of the matrix of the smaller problem, known also as the dual Schur complement matrix, but such algorithm was efficient only with a small number of subdomains. Later, Farhat, Mandel, and Roux introduced a "natural coarse problem" whose solution was implemented by auxiliary projectors so that the resulting algorithm became in a sense optimal [4]. In our approach, we use the Total-FETI [5] variant of FETI domain decomposition method, where also the Dirichlet boundary conditions are enforced by Lagrange multipliers. Hence all subdomain stiffness matrices are singular with a-priori known kernels which is a great advantage in the numerical solution and also in the theory.

The paper is organized as follows. After introducing a model problem, we briefly review the TFETI methodology that transforms the large primal problem in terms of displacements into the smaller and better conditioned dual one in terms of the Lagrange multipliers whose conditioning is further improved by using the projectors defined by the natural coarse grid. Then we introduce a modification of the conjugate gradient algorithm for the solution of the resulting quadratic programming problem with equality constraints enforced by the orthogonal projector onto the subspace defined by the constraints. Futher we briefly review the elasto-plasticity methodology for von Mises plasticity with isotropic hardening. We illustrate the efficiency of our algorithm on the solution of 3D elasto-plastic model benchmark and give encouraging results of numerical experiments.

2 PROBLEM OF ELASTOSTATICS

Let us consider an isotropic elastic body represented in a reference configuration by a domain Ω in $\mathbb{R}^d, d = 2, 3$, with the sufficiently smooth boundary Γ as in Fig. 1. Suppose that Γ consists of two disjoint parts Γ_U and Γ_F, $\Gamma = \overline{\Gamma}_U \cup \overline{\Gamma}_F$, and that the displacements $\mathbf{U} : \Gamma_U \to \mathbb{R}^d$ and forces $\mathbf{F} : \Gamma_F \to \mathbb{R}^d$ are given. The mechanical properties of Ω are defined by the Young modulus E and the Poisson ratio ν.

Let $c_{ijk\ell} : \Omega \to \mathbb{R}^d$ and $\mathbf{g} : \Omega \to \mathbb{R}^d$ denote the entries of the elasticity tensor and a vector of body forces, respectively. For any sufficiently smooth displacement $\mathbf{u} : \overline{\Omega} \to \mathbb{R}^d$, the total potential energy is defined by

$$J(\mathbf{u}) = \frac{1}{2} a(\mathbf{u}, \mathbf{u}) - \int_\Omega \mathbf{g}^\top \mathbf{u} \, d\Omega - \int_{\Gamma_F} \mathbf{F}^\top \mathbf{u} \, d\Gamma, \tag{1}$$

where

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega c_{ijk\ell} e_{ij}(\mathbf{u}) e_{k\ell}(\mathbf{v}) d\Omega, \quad e_{k\ell}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_\ell} + \frac{\partial u_\ell}{\partial x_k} \right).$$

We suppose that the elasticity tensor satisfies natural physical restrictions so that

$$a(\mathbf{u}, \mathbf{v}) = a(\mathbf{v}, \mathbf{u}) \quad \text{and} \quad a(\mathbf{u}, \mathbf{u}) \geq 0. \tag{2}$$

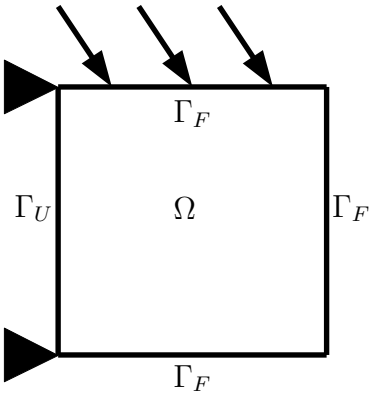

Figure 1: Model problem

Now let us introduce the Sobolev space $V = H^1(\Omega)^d$ and let K denote the set of all kinematically admissible displacements, where $K = \{\mathbf{v} \in V : \ \mathbf{v} = \mathbf{U} \ \text{ on } \ \Gamma_U\}$. The displacement $\mathbf{u} \in K$ of body in equilibrium satisfies

$$J(\mathbf{u}) \leq J(\mathbf{v}) \ \text{ for any } \ \mathbf{v} \in K. \tag{3}$$

Conditions that guarantee existence and uniqueness may be expressed in terms of coercivity of J. More general boundary conditions, such as prescribed normal displacements and periodicity, may be considered without any conceptual difficulties.

3 TFETI DOMAIN DECOMPOSITION

To apply the TFETI domain decomposition, we tear body from the part of the boundary with the Dirichlet boundary condition, decompose body into subdomains, assign each subdomain a unique number, and introduce new "gluing" conditions on the artificial intersubdomain boundaries and on the boundaries with imposed Dirichlet condition.

More specifically, the body Ω is decomposed into a system of s homogeneous isotropic elastic subdomains, each of which occupies, in a reference configuration, a subdomain Ω^p in $\mathbb{R}^d, d = 2, 3$. After decomposition each boundary Γ^p of Ω^p consists of three disjoint parts Γ^p_U, Γ^p_F, and Γ^p_G, $\Gamma^p = \overline{\Gamma}^p_U \cup \overline{\Gamma}^p_F \cup \overline{\Gamma}^p_G$, with the corresponding displacements $\mathbf{U}^p$ and forces $\mathbf{F}^p$ inherited from the originally imposed boundary conditions on Γ. For the artificial intersubdomain boundaries, we use the following notation: Γ^{pq}_G denotes the part of Γ^p that is glued to Ω^q and Γ^p_G denotes the part of Γ^p that is glued to the other subdomains. Obviously $\Gamma^{pq}_G = \Gamma^{qp}_G$. An auxiliary decomposition of the problem of Fig. 1 with renumbered subdomains and artificial intersubdomain boundaries is in Fig. 2. The gluing conditions require continuity of the displacements and of their normal derivatives across the intersubdomain boundaries. The mechanical properties of Ω^p are defined by the Young modulus E^p and the Poisson ratio ν^p.

Let $c^p_{ijk\ell}$ and $\mathbf{g}^p$ denote again the entries of the elasticity tensor and a vector of body forces, respectively. For any sufficiently smooth displacement $\mathbf{u} : \overline{\Omega}^1 \times \ldots \times \overline{\Omega}^s \to \mathbb{R}^d$, the

total potential energy is defined by

$$\mathcal{J}(\mathbf{u}) = \sum_{p=1}^{s} \left\{ \frac{1}{2} a^p(\mathbf{u}^p, \mathbf{u}^p) - \int_{\Omega^p} (\mathbf{g}^p)^\top \mathbf{u}^p d\Omega - \int_{\Gamma_F^p} (\mathbf{F}^p)^\top \mathbf{u}^p d\Gamma \right\}, \tag{4}$$

where

$$a^p(\mathbf{u}^p, \mathbf{v}^p) = \int_{\Omega^p} c_{ijk\ell}^p e_{ij}^p(\mathbf{u}^p) e_{k\ell}^p(\mathbf{v}^p) d\Omega, \qquad e_{k\ell}^p(\mathbf{u}^p) = \frac{1}{2}\left(\frac{\partial u_k^p}{\partial x_\ell^p} + \frac{\partial u_\ell^p}{\partial x_k^p} \right).$$

We suppose that the bilinear forms a^p satisfy (2) and let us introduce the product Sobolev space $\mathcal{V} = H^1(\Omega^1)^d \times \ldots \times H^1(\Omega^s)^d$, and let $\mathcal{K}$ denote the set of all kinematically admissible displacements, where $\mathcal{K} = \{\mathbf{v} \in \mathcal{V} : \ \mathbf{v}^p = \mathbf{U}^p \ \text{ on } \ \Gamma_U^p, \ \ \mathbf{v}^p = \mathbf{v}_{|\overline{\Omega}^p}\}$. The displacement $\mathbf{u} \in \mathcal{K}$ of the system of subdomains in equilibrium satisfies

$$\mathcal{J}(\mathbf{u}) \leq \mathcal{J}(\mathbf{v}) \ \text{ for any } \ \mathbf{v} \in \mathcal{K}. \tag{5}$$

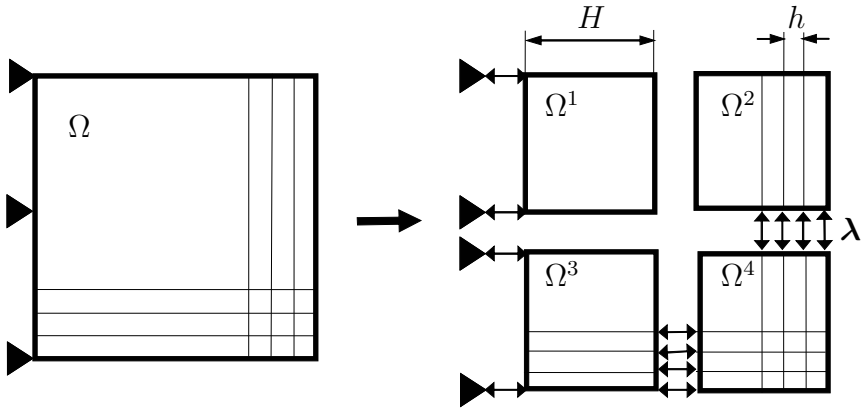

Figure 2: TFETI domain decomposition with subdomain renumbering

The finite element discretization of $\overline{\Omega} = \overline{\Omega}^1 \cup \ldots \cup \overline{\Omega}^s$ with a suitable numbering of nodes results in the quadratic programming (QP) problem

$$\frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} - \mathbf{f}^\top \mathbf{u} \to \min \ \ \text{subject to} \ \ \mathbf{B}\mathbf{u} = \mathbf{c}, \tag{6}$$

where $\mathbf{K} = \operatorname{diag}(\mathbf{K}_1, \ldots, \mathbf{K}_s)$ denotes a symmetric positive semidefinite block-diagonal matrix of order n, $\mathbf{B}$ denotes an $m \times n$ full rank matrix, $\mathbf{f} \in \mathbb{R}^n$, and $\mathbf{c} \in \mathbb{R}^m$.

The diagonal blocks $\mathbf{K}_p$ that correspond to the subdomains Ω^p are positive semidefinite sparse matrices with known kernels, the rigid body modes. The blocks can be effectively decomposed using the Choleski factorization [6]. The vector $\mathbf{f}$ describes the nodal forces arising from the volume forces and/or some other imposed traction.

The matrix $\mathbf{B}$ with the rows $\mathbf{b}_i$ and the vector $\mathbf{c}$ with the entries c_i enforce the prescribed displacements on the part of the boundary with imposed Dirichlet condition and the continuity of the displacements across the auxiliary interfaces. The continuity requires that $\mathbf{b}_i\mathbf{u} = c_i = 0$, where $\mathbf{b}_i$ are vectors of the order n with zero entries except 1 and -1 at appropriate positions. Typically m is much smaller than n.

Even though (6) is a standard convex quadratic programming problem, its formulation is not suitable for numerical solution. The reasons are that $\mathbf{K}$ is typically ill-conditioned, singular, and very large.

The complications mentioned above may be essentially reduced by applying the duality theory of convex programming (see, e.g., Dostál [7]), where all the constraints are enforced by the Lagrange multipliers $\boldsymbol{\lambda}$. The Lagrangian associated with problem (6) is

$$L(\mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} - \mathbf{f}^\top \mathbf{u} + \boldsymbol{\lambda}^\top(\mathbf{B}\mathbf{u} - \mathbf{c}). \tag{7}$$

It is well known [7] that (6) is equivalent to the saddle point problem

$$L(\overline{\mathbf{u}}, \overline{\boldsymbol{\lambda}}) = \sup_{\boldsymbol{\lambda}} \inf_{\mathbf{u}} L(\mathbf{u}, \boldsymbol{\lambda}). \tag{8}$$

4 OPTIMAL SOLVERS TO EQUALITY CONSTRAINED PROBLEMS

The solution of (8) leads to equivalent problem to find $(\bar{\mathbf{u}}, \bar{\boldsymbol{\lambda}}) \in \mathbb{R}^n \times \mathbb{R}^m$ satisfying:

$$\mathcal{A}\begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{c} \end{pmatrix} \tag{9}$$

with the saddle-point matrix

$$\mathcal{A} := \begin{pmatrix} \mathbf{K} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{pmatrix}.$$

We suppose that (9) is uniquely solvable which is guaranteed by the following necessary and sufficient conditions [8]:

$$Ker\,\mathbf{B}^\top = \{0\}, \tag{10}$$

$$Ker\,\mathbf{K} \cap Ker\,\mathbf{B} = \{0\}. \tag{11}$$

Notice that (10) is the condition on the full row-rank of $\mathbf{B}$. Let us mention that an orthonormal basis of $Ker\,\mathbf{K}$ is known à-priori and that its vectors are columns of $\mathbf{R} \in \mathbb{R}^{n\times l}$, $l = n - rank(\mathbf{K})$.

The first equation in (9) is satisfied iff

$$\mathbf{f} - \mathbf{B}^\top \bar{\boldsymbol{\lambda}} \in Im\,\mathbf{K} \tag{12}$$

and

$$\bar{\mathbf{u}} = \mathbf{K}^\dagger(\mathbf{f} - \mathbf{B}^\top \bar{\boldsymbol{\lambda}}) + \mathbf{R}\bar{\boldsymbol{\alpha}} \tag{13}$$

for an appropriate $\bar{\boldsymbol{\alpha}} \in \mathbb{R}^l$ and arbitrary generalized inverse $\mathbf{K}^\dagger$ satisfying $\mathbf{K}\mathbf{K}^\dagger\mathbf{K} = \mathbf{K}$. Moreover, (12) can be equivalently written as

$$\mathbf{R}^\top(\mathbf{f} - \mathbf{B}^\top\bar{\boldsymbol{\lambda}}) = 0. \tag{14}$$

Further substituting (13) into the second equation in (9) we arrive at

$$-\mathbf{B}\mathbf{K}^\dagger\mathbf{B}^\top\bar{\boldsymbol{\lambda}} + \mathbf{B}\mathbf{R}\bar{\boldsymbol{\alpha}} = \mathbf{c} - \mathbf{B}\mathbf{K}^\dagger\mathbf{f}. \tag{15}$$

Summarizing (15) and (14) we find that the pair $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\alpha}}) \in \mathbb{R}^m \times \mathbb{R}^l$ satisfies:

$$\mathcal{S}\begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}, \tag{16}$$

where

$$\mathcal{S} := \begin{pmatrix} \mathbf{B}\mathbf{K}^\dagger\mathbf{B}^\top & -\mathbf{B}\mathbf{R} \\ -\mathbf{R}^\top\mathbf{B}^\top & \mathbf{0} \end{pmatrix}$$

is the (negative) *Schur complement* of $\mathbf{K}$ in $\mathcal{A}$, $\mathbf{d} := \mathbf{B}\mathbf{K}^\dagger\mathbf{f} - \mathbf{c}$, and $\mathbf{e} := -\mathbf{R}^\top\mathbf{f}$. As both $\mathcal{S}$ and $\mathcal{A}$ are simultaneously invertible [8], we can compute first $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\alpha}})$ by solving (16) and then we obtain $\bar{\mathbf{u}}$ from (13). Let us note that (16) has formally the same saddle-point structure as that of (9), however, its size is considerably smaller.

Before discussing the solution method for (16) we introduce new notation

$$\mathbf{F} := \mathbf{B}\mathbf{K}^\dagger\mathbf{B}^\top, \quad \mathbf{G} := -\mathbf{R}^\top\mathbf{B}^\top$$

which changes (16) into

$$\begin{pmatrix} \mathbf{F} & \mathbf{G}^\top \\ \mathbf{G} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}. \tag{17}$$

Now we shall split (17) using the orthogonal projector $\mathbf{P}_\mathbf{G}$ onto $Ker\,\mathbf{G}$. As (11) implies that $\mathbf{G}$ is of full row-rank, we can identify $\mathbf{P}_\mathbf{G}$ with the following matrix:

$$\mathbf{P}_\mathbf{G} := \mathbf{I} - \mathbf{G}^\top(\mathbf{G}\mathbf{G}^\top)^{-1}\mathbf{G}.$$

Applying $\mathbf{P}_\mathbf{G}$ on the first equation in (17) we obtain that $\bar{\boldsymbol{\lambda}}$ satisfies:

$$\mathbf{P}_\mathbf{G}\mathbf{F}\boldsymbol{\lambda} = \mathbf{P}_\mathbf{G}\mathbf{d}, \quad \mathbf{G}\boldsymbol{\lambda} = \mathbf{e}. \tag{18}$$

In order to arrange (18) as one equation on the vector space $Ker\,\mathbf{G}$ we decompose the solution $\bar{\boldsymbol{\lambda}}$ into $\bar{\boldsymbol{\lambda}}_{Im} \in Im\,\mathbf{G}^\top$ and $\bar{\boldsymbol{\lambda}}_{Ker} \in Ker\,\mathbf{G}$ as

$$\bar{\boldsymbol{\lambda}} = \bar{\boldsymbol{\lambda}}_{Im} + \bar{\boldsymbol{\lambda}}_{Ker}. \tag{19}$$

Since $\bar{\boldsymbol{\lambda}}_{Im}$ is easily available via

$$\bar{\boldsymbol{\lambda}}_{Im} = \mathbf{G}^\top(\mathbf{G}\mathbf{G}^\top)^{-1}\mathbf{e},$$

it remains to show how to get $\bar{\boldsymbol{\lambda}}_{Ker}$. Substituting (19) into (18) we can see that $\bar{\boldsymbol{\lambda}}_{Ker}$ satisfies:

$$\mathbf{P_G F}\lambda_{Ker} = \mathbf{P_G}(\mathbf{d} - \mathbf{F}\bar{\boldsymbol{\lambda}}_{Im}), \quad \boldsymbol{\lambda}_{Ker} \in Ker\mathbf{G}. \tag{20}$$

Let us note that this equation is uniquely solvable, as $\mathbf{P_G F} : Ker\mathbf{G} \mapsto Ker\mathbf{G}$ is invertible if $\mathcal{A}$ is invertible [8]. Finally note that, if $\bar{\boldsymbol{\lambda}}$ is known, the solution component $\bar{\boldsymbol{\alpha}}$ is given by

$$\bar{\boldsymbol{\alpha}} = (\mathbf{G}\mathbf{G}^\top)^{-1}\mathbf{G}(\mathbf{d} - \mathbf{F}\bar{\boldsymbol{\lambda}}). \tag{21}$$

Let us algorithmically summarize the previous results. It turns out to be reasonable to form and store the $l \times m$ matrix $\mathbf{G}$ and the $l \times l$ matrix $\mathbf{H} := (\mathbf{G}\mathbf{G}^\top)^{-1}$ because l is usually small (the Cholesky factor of $\mathbf{G}\mathbf{G}^\top$ may be used instead of $\mathbf{H}$). On the other hand, the $m \times m$ matrices $\mathbf{F}$ and $\mathbf{P_G}$ are not assembled explicitly, since only their matrix-vector products are needed. Finally note that the actions of $\mathbf{B}$ are inexpensive in our problems due to sparsity of $\mathbf{B}$ and the actions of $\mathbf{K}^\dagger$ are computed effectively by the Cholesky factorization of $\mathbf{K}^p$, $p = 1, ..., s$ ([6]). All the above steps are summarized in the following algorithmic scheme.

Algorithmic scheme

Step 1.a: Compute $\mathbf{G} := -\mathbf{R}^\top\mathbf{B}^\top$, $\mathbf{H} := (\mathbf{G}\mathbf{G}^\top)^{-1}$, $\mathbf{d} := \mathbf{B}\mathbf{K}^\dagger\mathbf{f} - \mathbf{c}$, and $\mathbf{e} := -\mathbf{R}^\top\mathbf{f}$.
Step 1.b: Compute $\bar{\boldsymbol{\lambda}}_{Im} := \mathbf{G}^\top\mathbf{H}\mathbf{e}$.
Step 1.c: Compute $\tilde{\mathbf{d}} := \mathbf{d} - \mathbf{F}\bar{\boldsymbol{\lambda}}_{Im}$.
Step 1.d: Compute $\bar{\boldsymbol{\lambda}}_{Ker}$ by solving $\mathbf{P_G F}\boldsymbol{\lambda}_{Ker} = \mathbf{P_G}\tilde{\mathbf{d}}$ on $Ker\mathbf{G}$.
Step 1.e: Compute $\bar{\boldsymbol{\lambda}} := \bar{\boldsymbol{\lambda}}_{Im} + \bar{\boldsymbol{\lambda}}_{Ker}$.
Step 2: Compute $\bar{\boldsymbol{\alpha}} := \mathbf{H}\mathbf{G}(\mathbf{d} - \mathbf{F}\bar{\boldsymbol{\lambda}})$.
Step 3: Compute $\bar{\mathbf{u}} := \mathbf{K}^\dagger(\mathbf{f} - \mathbf{B}^\top\bar{\boldsymbol{\lambda}}) + \mathbf{R}\bar{\boldsymbol{\alpha}}$.

Finally, we introduce the projected conjugate gradient method with preconditioning (ProjCGM) [4] that we use for computing $\bar{\boldsymbol{\lambda}}_{Ker}$ in Step 1.d of Algorithmic scheme. Thus we want to compute $\bar{\boldsymbol{\lambda}}_{Ker}$ by solving the system $\mathbf{P_G F}\boldsymbol{\lambda}_{Ker} = \mathbf{P_G}\tilde{\mathbf{d}}$ on $Ker\mathbf{G}$ with the lumped preconditioner $\overline{\mathbf{F}^{-1}}$ [4] to $\mathbf{F}$.

Algorithm ProjCGM

1. Initialize
   $\mathbf{r}^0 = \tilde{\mathbf{d}}$, $\boldsymbol{\lambda}^0_{Ker} = \mathbf{o}$.
2. Iterate k =1, 2, ..., until convergence
   Project $\mathbf{w}^{k-1} = \mathbf{P_G}\mathbf{r}^{k-1}$.
   Precondition $\mathbf{z}^{k-1} = \overline{\mathbf{F}^{-1}}\mathbf{w}^{k-1}$.
   Project $\mathbf{y}^{k-1} = \mathbf{P_G}\mathbf{z}^{k-1}$.

$\boldsymbol{\beta}^k = (\mathbf{y}^{k-1})^\top \mathbf{w}^{k-1} / (\mathbf{y}^{k-2})^\top \mathbf{w}^{k-2}; \qquad (\boldsymbol{\beta}^1 = 0).$
$\mathbf{p}^k = \mathbf{y}^{k-1} + \boldsymbol{\beta}^k \mathbf{p}^{k-1}; \qquad (\mathbf{p}^1 = \mathbf{y}^0).$
$\boldsymbol{\alpha}^k = (\mathbf{y}^{k-1})^\top \mathbf{w}^{k-1} / (\mathbf{p}^k)^\top \mathbf{F}\mathbf{p}_k.$
$\boldsymbol{\lambda}_{Ker}^k = \boldsymbol{\lambda}_{Ker}^{k-1} + \boldsymbol{\alpha}^k \mathbf{p}^k.$
$\mathbf{r}^k = \mathbf{r}^{k-1} - \boldsymbol{\alpha}^k \mathbf{F}\mathbf{p}_k.$

3. $\bar{\boldsymbol{\lambda}}_{Ker} = \boldsymbol{\lambda}_{Ker}^k.$

Using TFETI in combination with ProjCGM algorithm we are able to find the solution of the original elasto-plastic problem in O(1) matrix-vector multiplications independently of the problem size provided the ratio between the decomposition step H and the discretization step h is kept bounded. For more details about optimality see [4].

5 ELASTO-PLASTICITY

Elasto-plastic problems are the so-called quasi-static problems where the history of loading is taken into account. We consider the von Mises elasto-plasticity with the strain isotropic hardening and incremental finite element method with the return mapping concept [1].

The elasto-plastic deformation of an body Ω after loading is descibed by the Cauchy stress tensor $\boldsymbol{\sigma}$, the small strain tensor $\boldsymbol{\varepsilon}$, the displacement $\mathbf{u}$, and the nonnegative hardening parameter $\boldsymbol{\kappa}$. Symmetric tensor is represented by the vector and its deviatoric part is denoted by the symbol *dev*.

Let us denote the space of continuous and piecewise linear functions constructed over a regular triangulation of Ω with the discretization norm h by $V_h \subset V$, where $V = \left\{\mathbf{v} \in H^1(\Omega)^d : \quad \mathbf{v} = 0 \text{ on } \Gamma_U\right\}$. Let

$$0 = t_0 < t_1 < \ldots t_k < \ldots < t_N = t^* \tag{22}$$

be a partition of the time interval $[0, t^*]$. Then the solution algorithm after time and space discretizations has the form:

Algorithm 3.

1. Initial step: $\mathbf{u}_h^0 = 0,\ \boldsymbol{\sigma}_h^0 = 0,\ \boldsymbol{\kappa}_h^0 = 0,$
2. **for** $k = 0,\ \ldots, N-1$ **do** (load step)
3. From previous step we know: $\mathbf{u}_h^k,\ \boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k$ and compute $\triangle\mathbf{u}_h,\ \triangle\boldsymbol{\sigma}_h,\ \triangle\boldsymbol{\kappa}_h$

$$\triangle\boldsymbol{\varepsilon}_h = \varepsilon(\triangle\mathbf{u}_h), \quad \triangle\mathbf{u}_h \in V_h, \tag{23}$$

$$\triangle\boldsymbol{\sigma}_h = T_\sigma(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k,\ \triangle\boldsymbol{\varepsilon}_h), \tag{24}$$

$$\triangle\boldsymbol{\kappa}_h = T_\kappa(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k,\ \triangle\boldsymbol{\varepsilon}_h). \tag{25}$$

4. Solution $\triangle\boldsymbol{\sigma}_h(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k, \varepsilon(\triangle\mathbf{u}_h))$ is substituted into equation of equilibrium:

$$\int_\Omega \triangle\boldsymbol{\sigma}_h^T(\boldsymbol{\sigma}_h^k,\ \boldsymbol{\kappa}_h^k, \varepsilon(\triangle\mathbf{u}_h))\varepsilon(\mathbf{v}_h)dx = \langle\triangle\mathbf{f}_h^k,\ \mathbf{v}_h\rangle, \quad \forall\mathbf{v}_h \in V_h. \tag{26}$$

This leads to a nonlinear system of equations with unknown $\triangle\mathbf{u}_h$ which is solved using the Newton method. The linearized problem arising in each Newton step is solved by TFETI algorithmic scheme proposed above.

5. Then we compute values for the next step: $\mathbf{u}_h^{k+1} = \mathbf{u}_h^k + \triangle\mathbf{u}_h$, $\boldsymbol{\sigma}_h^{k+1} = \boldsymbol{\sigma}_h^k + \triangle\boldsymbol{\sigma}_h$, $\boldsymbol{\kappa}_h^{k+1} = \boldsymbol{\kappa}_h^k + \triangle\boldsymbol{\kappa}_h$.

6. **enddo**

Above we consider the following notation. Let $\mathbf{C}$ denote the Hook's matrix, $\mathbf{E}$ represent linear operator *dev*, μ, λ be the Lamé coefficients, $\triangle\mathbf{f}_h^k$ be the increment of the right hand side and $\boldsymbol{\sigma}_h^t = \boldsymbol{\sigma}_h^k + \mathbf{C}\triangle\boldsymbol{\varepsilon}_h$. For return mapping concept we define

$$\begin{aligned} \triangle\boldsymbol{\sigma}_h &= T_\sigma(\boldsymbol{\sigma}_h^k, \boldsymbol{\kappa}_h^k, \triangle\boldsymbol{\varepsilon}_h) = T_\sigma^{RM}(\boldsymbol{\sigma}_h^k, \boldsymbol{\kappa}_h^k, \triangle\boldsymbol{\varepsilon}_h) = \\ &= \begin{cases} \mathbf{C}\triangle\boldsymbol{\varepsilon}_h & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) \le 0, \\ \mathbf{C}\triangle\boldsymbol{\varepsilon}_h - \gamma_R\widehat{\mathbf{n}} & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) > 0, \end{cases} \end{aligned} \tag{27}$$

$$\begin{aligned} \triangle\boldsymbol{\kappa}_h &= T_\kappa(\boldsymbol{\sigma}_h^k, \boldsymbol{\kappa}_h^k, \triangle\boldsymbol{\varepsilon}_h) = T_\kappa^{RM}(\boldsymbol{\sigma}_h^k, \boldsymbol{\kappa}_h^k, \triangle\boldsymbol{\varepsilon}_h) = \\ &= \begin{cases} 0 & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) \le 0, \\ \gamma z = \gamma_R\|\mathbf{Cp}\|^{-1}z & \text{if } P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) > 0, \end{cases} \end{aligned} \tag{28}$$

where

$$\begin{aligned} \gamma_R &= \tfrac{3\mu}{3\mu+H_m}\sqrt{\tfrac{2}{3}}\left(\sqrt{\tfrac{3}{2}}\|dev(\boldsymbol{\sigma}_h^t)\| - (Y_0 + H_m\boldsymbol{\kappa}_h^k)\right) = \\ &= \tfrac{3\mu}{3\mu+H_m}\sqrt{\tfrac{2}{3}}P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k), \end{aligned} \tag{29}$$

$$\widehat{\mathbf{n}} = \frac{dev(\boldsymbol{\sigma}_h^t)}{\|dev(\boldsymbol{\sigma}_h^t)\|}, \quad \|\mathbf{Cp}\| = 2\mu\sqrt{\frac{3}{2}}, \quad z = 1, \tag{30}$$

and plasticity function

$$P(\boldsymbol{\sigma}_h^t, \boldsymbol{\kappa}_h^k) = \sqrt{\frac{3}{2}}\|dev(\boldsymbol{\sigma}_h^t)\| - (Y + H_m\boldsymbol{\kappa}_h^k), \qquad Y, H_m > 0. \tag{31}$$

The function $\gamma_R\widehat{\mathbf{n}}$ is semismooth and potential. The derivative of T_σ^{RM} is

$$\begin{aligned} (T_\sigma^{RM})'(\triangle\boldsymbol{\varepsilon}) &= \mathbf{C} - 2\mu\tfrac{3\mu}{3\mu+H_m}\left[\mathbf{E} + \right. \\ &+ \left. \sqrt{\tfrac{2}{3}}\tfrac{Y_0+H_m\boldsymbol{\kappa}_h^k}{\|dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon})\|}\left(\tfrac{dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon})(dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon}))^T}{\|dev(\boldsymbol{\sigma}_h^k+\mathbf{C}\triangle\boldsymbol{\varepsilon})\|^2} - \mathbf{E}\right)\right]. \end{aligned} \tag{32}$$

If we represent a function $\mathbf{v}_h \in V_h$ by the vector $\mathbf{v} \in \mathbb{R}^n$ and omit index k then (26) can be rewritten as the system of nonlinear equations

$$F(\triangle \mathbf{u}) = \triangle \mathbf{f}, \tag{33}$$

where

$$\begin{array}{rcll} \langle F(\mathbf{v}), \mathbf{w} \rangle & = & \int_\Omega \langle T_\sigma^{RM}(\varepsilon(\mathbf{v}_h)), \varepsilon(\mathbf{w}_h) \rangle dx, & \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^n \\ \langle \triangle \mathbf{f}, \mathbf{w} \rangle & = & \triangle \mathbf{f}_h(v_h), \quad \forall \mathbf{w} \in \mathbb{R}^n. & \end{array} \tag{34}$$

6 NUMERICAL EXPERIMENTS

Described algorithms were implemented in `MatSol` library [9] developed in Matlab environment and tested on solution of 3D problems.

Let us consider a 3D plate with a hole in the center (due to symmetry only a quatre of the whole structure is used) with the geometry depicted in Fig. 3. Boundary conditions are specified in Fig. 4. Symmetry conditions are prescribed on the left and lower sides of Ω. The surface load $g(t) = 450 \sin(2\pi t)$ [MPa], $t \in [0, \frac{1}{4}]$ [sec], is applied to the upper side of Ω. The elasto-plastic material parameters are $E = 206900$ [MPa], $\nu = 0.29$, $Y = 450$, $H_m = 100$ and the time interval $[0, \frac{1}{4}]$ [sec] is divided into 50 steps. We consider a mesh with 5489 nodes and 19008 tetrahedrons. The body Ω is decomposed into 20 subdomains.

In the nth Newton iteration we compute an approximation $\triangle \mathbf{u}^n$ by solving the constrained linear problem of the form

$$\min_{\mathbf{B}\triangle \mathbf{u}^n = \mathbf{o}} \frac{1}{2} (\triangle \mathbf{u}^n)^\top \mathbf{K}^n \triangle \mathbf{u}^n - (\triangle \mathbf{u}^n)^\top \triangle \mathbf{f}^n$$

using the TFETI algorithmic scheme proposed above. We stop the Newton method in every time step if $\|\triangle \mathbf{u}^{n+1} - \triangle \mathbf{u}^n\| / \left(\|\triangle \mathbf{u}^{n+1}\| + \|\triangle \mathbf{u}^n\|\right)$ is less than 10^{-6}.

Notice that the maximum number of the Newton iterations is small for all time steps (less than 7), therefore the method is suitable for the problem. In the following figures, we depict plastic and elastic elements, graph of maximum value of hardenining at each time step and von Mises stress in the xy plane cross-section with the z coordinate 0 [mm] corresponding to the surface of Ω. In Figs. 5, 6, we see which elements are plastic (gray color) and which are elastic (white color) in chosen time steps. Particularly, in time steps 1-12 we observe only elastic behavior, and in time steps 13-50 plastic behavior of some elements. The maximum value of hardenining at each time step is depicted in Fig. 7. The von Mises stress distribution on deformed mesh is showed in Fig. 8.

7 CONCLUSIONS AND GOALS

We have presented an efficient algorithm for the numerical solution of elasto-plastic problems. These problems lead to the quasi-static problems, where each nonlinear and nonsmooth time step problem is solved by the semismooth Newton method. In each Newton iteration we solve an auxiliary (possibly of large size) linear system of algebraic

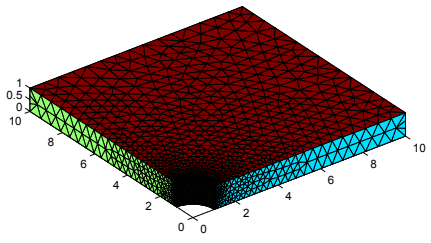

Figure 3: 3D plate geometry in [mm]

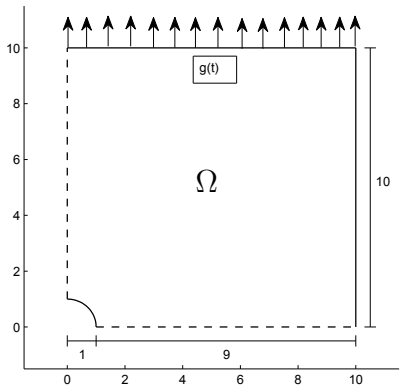

Figure 4: 2D plate geometry in [mm] and boundary conditions

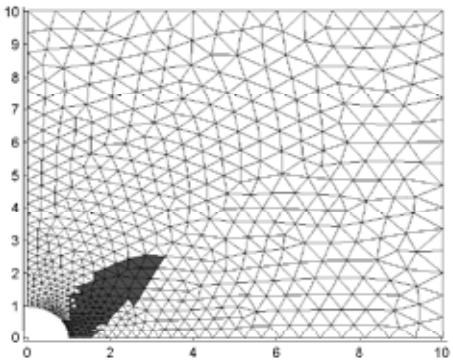

Figure 5: Plastic and elastic elements after 35 time steps

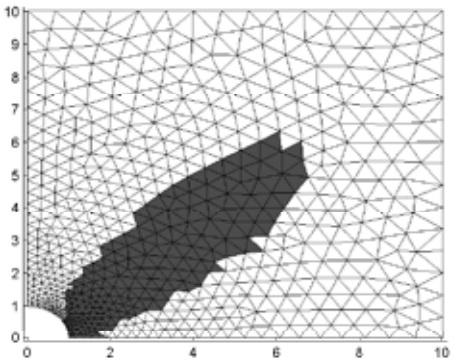

Figure 6: Plastic and elastic elements after 50 time steps

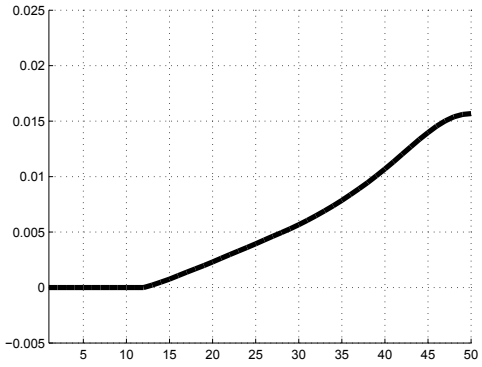

Figure 7: Maximum values of hardening in time iterations

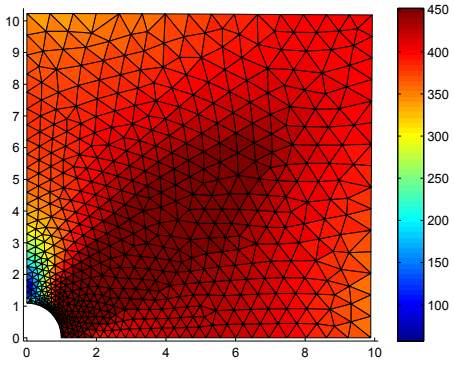

Figure 8: Von Mises stress distribution on the deformed mesh (scaled 10x)

equations using in a sense optimal algorithm based on our Total-FETI variant of FETI domain decomposition method. We illustrated the efficiency of our algorithm on the solution of 3D elasto-plastic model benchmark and gave results of numerical experiments. The results indicate that the algorithm may be efficient. In the future we would like to adapt this approach to the solution of contact problems.

REFERENCES

[1] Blaheta, R. *Numerical methods in elasto-plasticity*, Documenta Geonica 1998, PERES Publishers, Prague, (1999).

[2] Gruber, P. G. and Valdman, J. *Solution of One-Time Step Problems in Elastoplasticity by a Slant Newton Method*, SIAM J. Sci. Comp. **31** (2009) 1558-1580.

[3] Qi, L. and Sun, J. *A nonsmooth version of Newtons method*, Mathematical Programming **58** (1993) 353-367.

[4] Farhat, C., Mandel, J. and Roux, F-X. *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Eng. **115**, (1994); 365–385.

[5] Dostál, Z., Horák, D. and Kučera, R. *Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE*, Communications in Numerical Methods in Engineering, 22(2006), 12, pp. 1155-1162

[6] Brzobohatý, T., Dostál, Z., Kovář, P., Kozubek, T. and Markopoulos, A. *Cholesky decomposition with fixing nodes to stable evaluation of a generalized inverse of the stiffness matrix of a floating structure*, IJNME, DOI: 10.1002/nme.3187

[7] Dostál, Z. *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*, 1st edition, SOIA 23, Springer US, New York, (2009)

[8] Haslinger, J., Kozubek, T., Kučera, R. and Peichl, G. *Projected Schur complement method for solving non-symmetric saddle-point systems arising from fictitious domain approach*, Numerical Linear Algebra with Applications (2007); **14**(9):713–739.

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

[10] Dostl, Z., Kozubek, T., Markopoulos, A., Brzobohat, T., Vondrk, V., Horyl, P *.: Theoretically supported scalable TFETI algorithm for the solution of multibody 3D contact problems with fiction*, (2011) doi:10.1016/j.cma.2011.02.015