Numerical Analysis of the Contact Problem. Comparison of Methods for Finding the Approximate Solution

Kestřánek, Zdeněk
1995

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

NUMERICAL ANALYSIS OF THE CONTACT PROBLEM

COMPARISON OF METHODS FOR FINDING THE APPROXIMATE SOLUTION

Zdeněk Kestřánek

Technical report No. 648

September 14, 1995

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: 66053911 fax: (+422) 8585789
e-mail: zdenda@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

NUMERICAL ANALYSIS OF THE CONTACT PROBLEM

COMPARISON OF METHODS FOR FINDING THE APPROXIMATE SOLUTION

Zdeněk Kestřánek

Technical report No. 648
September 14, 1995

Abstract

The simulation of geodynamical and tectonic processes often leads to mathematical models which correspond to the Contact problem in 2-D and 3-D elasticity. In these models a system of several elastic bodies is considered. These bodies are subjected to the fundamental equilibrium laws as well as to the Hooke's law of elasticity. Unlike the classical elastic models, the condition of impenetration must be fulfilled.

The Finite Element Method is very suitable for the numerical solution of this problem. In engineering practice several solutions were suggested on how to solve such a problem. Here we draw on the mathematical formulation of the Contact Problem. In this way, we avoid using the additional contact elements where the estimate of suitable elastic parameters is needed. Mathematical formulation is based on the variational inequality. We are able to study the questions concerning the existence and uniqueness and can also obtain the asymptotic estimate of the error of an approximate solution. Discretization then leads directly to the algorithms of numerical mathematics. This enables us to examine a great variety of methods and select the optimal in view of the speed and memory requirements.

Keywords
Finite Element Method, Contact Problem, Quadratic Programming

The solution of an approximate Contact problem can be divided into several phases. The “Outer” part is the method of succesive approximations. In every iteration it is necessary to find certain saddle point. This is done by another iterative method. Finally, in the “Inner” part we solve the problem of finding the minimum of the Potential energy functional over the set of all admissible displacements. In our case this is equivalent to the Quadratic programming problem. Let us notice, that the phases may be connected and our division of the problem does not need to be observed strictly. If we omit the influence of friction, the problem is only reduced to the “Inner” part. With this contribution we will examine various methods for solving such a problem.

Chapter 1

Formulation of the problem

1.1 Classical formulation of the Contact Problem

Let us suppose, that we have S elastic bodies in the system. Note, that the existence of points to which more than two bodies stick is not necessary. Let these bodies occupy the bounded regions $\Omega^1, \Omega^2, \ldots, \Omega^S \subset R^2$ with Lipschitz boundaries.

We look for the vector field of the displacements $\mathbf{u} = (u_1, u_2)$, the tensor field of small strains $e_{ij} = e_{ij}(\mathbf{u})$ and the stress tensor $\tau_{ij} = \tau_{ij}(\mathbf{u})$, $i, j = 1, 2$, on $\Omega^1 \cup \ldots \cup \Omega^S$.

Let the boundary $\partial\Omega$ be divided into disjunct parts

$$\Gamma_u, \Gamma_\tau, \Gamma_c, \Gamma_0, R, \quad \partial\Omega = \Gamma_u \cup \Gamma_\tau \cup \Gamma_c \cup \Gamma_0 \cup R,$$
$$\Gamma_u = \textstyle\bigcup_{i=1}^S \Gamma_u^i, \quad \Gamma_\tau = \bigcup_{i=1}^S \Gamma_\tau^i, \quad \Gamma_0 = \bigcup_{i=1}^S \Gamma_0^i, \quad \Gamma_c = \bigcup_{k,l} \Gamma_c^{kl},$$
$$\Gamma_c^{kl} = \overline{\Gamma}_c^k \cap \overline{\Gamma}_c^l, \quad k, l \in \{1, \ldots, S\},\ k < l,$$

and the surface measure of R be zero.

Let on $\Gamma_c = \bigcup_{k,l} \Gamma_c^{kl}$

$$u_n = u_i n_i,\ u_t = u_i t_i,\ \tau_n = \tau_i n_i,\ \tau_t = \tau_i t_i \tag{1.1}$$

where n_i are the components of outward normal to $\partial\Omega^k$,
$\mathbf{t} = (-n_2, n_1)$, $\tau_i = \tau_{ij} n_j$.

DEFINITION 1.1. The function $\mathbf{u}$ is a classical solution of the Contact Problem if it fulfills the equilibrium equations

$$\frac{\partial \tau_{ij}}{\partial x_j}(\mathbf{u}) + F_i = 0 \quad i, j = 1, 2 \qquad , \tag{1.2}$$

where F_i are the components of the body forces vector,
the generalized Hooke's law

$$\tau_{ij}(\mathbf{u}) = c_{ijkm} e_{km}(\mathbf{u}) \quad i, j = 1, 2 \tag{1.3}$$

(we use the Einstein's summation convention),
the relation for strain

$$e_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad i, j = 1, 2 \tag{1.4}$$

and the boundary conditions

$$u_i = u_{0i} \quad \text{on}\, \Gamma_u, \tag{1.5}$$

where u_{0i} are the components of a given vector of displacement.

$$\tau_i = P_i \quad \text{on}\, \Gamma_\tau, \tag{1.6}$$

where P_i are the components of surface loads,

$$u_n^k - u_n^l \leq 0, \quad \tau_n^k = -\tau_n^l \leq 0, \quad (u_n^k - u_n^l)\tau_n^k = 0, \quad \tau_t^k = \tau_t^l = 0 \quad \text{on}\, \Gamma_c^{kl} \tag{1.7}$$

(The Signorini conditions on an unilateral contact)

$$u_n = 0, \quad \tau_t = 0 \quad \text{on} \quad \Gamma_0. \tag{1.8}$$

(The conditions on a bilateral contact)

The coefficients in (1.3), $c_{ijkm} \in L^\infty(\Omega)$, have the following types of symmetry

$$c_{ijkm} = c_{jikm} = c_{kmij}. \tag{1.9}$$

Moreover, there exists a constant $c_0 > 0$ such, that

$$c_{ijkm}(x) e_{ij} e_{km} \geq c_0 e_{ij} e_{ij} \tag{1.10}$$

is valid for all sym. matrices e_{ij} and almost everywhere in Ω.

In the case of isotropic bodies and plane strain

$$c_{1112} = \lambda,\, c_{1212} = \mu$$

the same holds for symmetric components (cf.(1.3)), and

$$c_{1111} = c_{2222} = \lambda + 2\mu, \quad c_{ijkm} = 0 \quad \text{otherwise.}$$

1.2 Variational formulation

It is necessary to assume sufficient smoothness for the classical solution. However, in the case when this assumption is not valid, it is possible to define the solution by using the minimum potential energy principle.

First of all, we introduce the space of the functions with finite energy

$$\mathcal{H}^1(\Omega) \equiv \{\mathbf{v} | \mathbf{v} = (\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^S) \in [H^1(\Omega^1)]^2 \times \ldots \times [H^1(\Omega^S)]^2 \}. \tag{1.11}$$

The norm is defined as

$$\|\mathbf{v}\|^2 = \|\mathbf{v}\|^2_{\mathcal{H}^1(\Omega)} = \sum_{l=1}^{S} \|\mathbf{v}^l\|^2_{[H^1(\Omega^l)]^2} = \sum_{l=1}^{S}\sum_{i=1}^{2} \|v_i^l\|_1^2. \tag{1.12}$$

Similarly we define the space $\mathcal{H}^2(\Omega)$

$$\mathcal{H}^2(\Omega) \equiv \{\mathbf{v}|\mathbf{v} = (\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^S) \in [H^2(\Omega^1)]^2 \times \ldots \times [H^2(\Omega^S)]^2\}. \tag{1.13}$$

We will also use the space

$$[W^{1,\infty}(\Gamma)]^2 \equiv \{\mathbf{v}|\frac{\partial v_i}{\partial t} \in L^\infty(\Gamma)\}, \tag{1.14}$$

where $\mathbf{v} = \mathbf{v}(\mathbf{x})$, $\mathbf{x} = \mathbf{x}(t)$ is the parametrisation of the abscissa Γ, $\quad i = 1, 2$

Furthermore, we define the seminorm

$$\|\mathbf{v}\|^2 = \int_\Omega e_{ij}(\mathbf{v})e_{ij}(\mathbf{v})d\mathbf{x} \tag{1.15}$$

We introduce the sets

$$V_{u_0} \equiv \{\mathbf{v} \in \mathcal{H}^1(\Omega)|\, \mathbf{v} = \mathbf{u}_0 \quad \text{on } \Gamma_u, \quad v_n = 0 \quad \text{on } \Gamma_0\}, \tag{1.16}$$

where $\mathbf{u}_0 \in \mathcal{H}^1(\Omega)$, and

$$K_{u_0} \equiv \{\mathbf{v} \in V_{u_0}\,|\, v_n^k - v_n^l \leq 0 \quad \text{on } \Gamma_c^{kl}\} \tag{1.17}$$

(The set of all admissible displacements).

REMARK 2.1. If $\mathbf{u}_0 \equiv 0$ on Γ_u, for simplicity's sake we omit the index $\mathbf{u}_0$ in symbols V and K.

Let the potential energy functional have the following form

$$\mathcal{L}(\mathbf{v}) \;=\; \frac{1}{2}A(\mathbf{v},\mathbf{v}) - L(\mathbf{v}) \quad , \tag{1.18}$$

where

$$A(\mathbf{u},\mathbf{v}) = \int_\Omega c_{ijkm}e_{ij}(\mathbf{u})e_{km}(\mathbf{v})d\mathbf{x} \tag{1.19}$$

$$L(\mathbf{v}) = \int_\Omega F_i v_i d\mathbf{x} + \int_{\Gamma_\tau} P_i v_i d\mathbf{x} \quad , \tag{1.20}$$

$$\mathbf{F} \in [L^2(\Omega)]^2,\, \mathbf{P} \in [L^2(\Gamma_\tau)]^2 \;.$$

Regarding (1.9), (1.10) and Schwartz inequality, we have

$$c_0|\mathbf{v}|^2 \leq A(\mathbf{v},\mathbf{v}) \quad , \tag{1.21}$$

$$A(\mathbf{u},\mathbf{v}) \leq C_1|\mathbf{u}||\mathbf{v}| \quad , \tag{1.22}$$

$$|\mathbf{v}|^2 \leq C_2\|\mathbf{v}\|^2 \quad , \tag{1.23}$$

We will now define the variational solution.

DEFINITION 2.1. A function $\mathbf{u} \in K_{u_0}$ is the variational solution of the Contact Problem if it is the minimum of the potential energy functional on the set of all admissible displacements i.e.

$$\mathcal{L}(\mathbf{u}) \leq \mathcal{L}(\mathbf{v}) \quad \forall\, \mathbf{v} \in K_{u_0} \quad . \tag{1.24}$$

We denote this minimization problem by $(\mathcal{P})$.

The following Theorem shows the connection between the classical and variational solutions.

THEOREM 2.1. Every classical solution is also variational. If the variational solution is sufficiently smooth, it is also classical.

Proof. Let $\mathbf{u}$ be the classical solution. Multiplying each of the equations (1.2) by the functions $w_i = v_i - u_i$, $v_i \in K_{u_0}$, adding, integrating by parts and using the symmetries in (1.4)-(1.9) and boundary conditions (1.5), (1.6) and (1.8), we arrive at

$$\begin{aligned} A(\mathbf{u},\mathbf{w}) - L(\mathbf{w}) \;&=\; \int_{\cup\Gamma_c^{kl}} [\tau^k_{n^k}(v^k_{n^k} - u^k_{n^k}) + \tau^k_{t^k}(v^k_{t^k} - u^k_{t^k}) + \\ &\qquad + \tau^l_{n^l}(v^l_{n^l} - u^l_{n^l}) + \tau^l_{t^l}(v^l_{t^l} - u^l_{t^l})]ds. \end{aligned}$$

Here n^k, n^l are the outward normals to $\partial\Omega^k$ and $\partial\Omega^l$; $t^k = -t^l$ tangent directions (1.1); $\tau^l_{n^l} = -\tau^l_{n^k} = \tau^k_{n^k}$, similarly $v^l_{n^l} = -v^l_{n^k}$, $u^l_{n^l} = -u^l_{n^k}$, $\tau^l_{t^l} = \tau^k_{t^k} = 0$.

Thus, we get

$$A(\mathbf{u},\mathbf{u}-\mathbf{v}) - L(\mathbf{v}-\mathbf{u}) = \int_{\cup\Gamma_c^{kl}} [\tau^k_{n^k}((v^k_{n^k} - v^l_{n^k}) - (u^k_{n^k} - u^l_{n^k}))]ds. \tag{1.25}$$

As $\mathbf{v} \in K_{u_0}$, from the first three conditions in (1.7), it finally follows that

$$A(\mathbf{u},\mathbf{v}-\mathbf{u}) - L(\mathbf{v}-\mathbf{u}) \geq 0 \quad \forall \mathbf{v} \in K_{u_0}. \tag{1.26}$$

The solution of this variational inequality is also the solution of the minimization problem (1.24) [5].

On the other hand, let the solution $\mathbf{u} \in K_{u_0}$ be sufficiently smooth.
$\mathbf{u} \in K_{u_0}$ and therefore the conditions in (1.5) and first in (1.7), (1.8) are met. Integrating by parts in (1.26) and choosing suitable trial functions $\mathbf{v}$ we gradually obtain (1.2), (1.6) and the remaining conditions in (1.8) and (1.7). For a 2D problem see [8] in detail. □

1.3 The existence and uniqueness

After forming the variational formulation, we are able to solve the problem by using the variational method. At this point it is natural to ask whether there are existing conditions, which ensure that the solution does exist or whether it is determined uniquely. We will assume the implication $(\Gamma_u \neq \{\emptyset\} \Rightarrow (\mathbf{u}_0 \equiv 0 \quad \text{on}\, \Gamma_u)\ )$.

We have transformed a general case to a homogeneous one, as follows.

Let us consider the decomposition $\mathbf{u} = \mathbf{w} + \mathbf{w}_0$, where $\mathbf{w} \in K$ and $\mathbf{w}_0 \in \mathcal{H}^1(\Omega)$, $\mathbf{w}_0 = \mathbf{u}_0$ on Γ_u, $w^k_{0n} - w^l_{0n} = 0$ on Γ_c^{kl}, $w_{0n} = 0$ on Γ_0. Define the functional

$$\mathcal{L}_{w0}(\mathbf{w}) = \tfrac{1}{2}A(\mathbf{w},\mathbf{w}) - L_{w0}(\mathbf{w}),$$

where $\quad L_{w0}(\mathbf{w}) = L(\mathbf{w}) - A(\mathbf{w},\mathbf{w}_0),$

and consider the problem $(\mathcal{P}_{w0})$

$$\min_{w\in K} \mathcal{L}_{w0}(\mathbf{w}).$$

The following Lemma holds.

LEMMA 3.1. The variational solution of the problem $(\mathcal{P})$ exists and is uniquely determined iff a unique solution of $(\mathcal{P}_{w0})$ exists.

Proof. Choose $\mathbf{h} \in \mathcal{H}^1(\Omega)$ such, that

$$\mathbf{h} \neq \{\emptyset\} \quad \text{and } \mathbf{u}+\mathbf{h} \in K_{u_0} \quad \Leftrightarrow \quad \mathbf{w}+\mathbf{h}+\mathbf{w}_0 \in K_{u_0} \quad \Leftrightarrow \quad \mathbf{w}+\mathbf{h} \in K \quad .$$

The equivalence of the assertions

$$\mathcal{L}(\mathbf{u}) < \mathcal{L}(\mathbf{u}+\mathbf{h}) \quad \text{and } \mathcal{L}_{w0}(\mathbf{w}) < \mathcal{L}_{w0}(\mathbf{w}+\mathbf{h})$$

is now already obvious, as

$$\mathcal{L}_{w0}(\mathbf{w}) - L(\mathbf{w}_0) = \mathcal{L}(\mathbf{u}) < \mathcal{L}(\mathbf{u}+\mathbf{h}) = \mathcal{L}_{w0}(\mathbf{w}+\mathbf{h}) - L(\mathbf{w}_0) \qquad \square$$

Hence, let $\mathbf{u}_0 \equiv 0$ on Γ_u in what follows.

DEFINITION 3.1. Let

$$\mathcal{R}^l = \{\mathbf{z}^l \in [H^1(\Omega^l)]^2 |\ z_1^l = a_1^l - b^l x_2,\ z_2^l = a_2^l + b^l x_1\},$$

where $1 \leq l \leq S, \quad a_1^l, a_2^l, b^l$ are the arbitrary constants

$$\mathcal{R} = \{\mathbf{z} \in \mathcal{H}^1(\Omega) | \quad \forall l \quad 1 \leq l \leq S \ ;\ z^l \in \mathcal{R}^l\}.$$

$\mathcal{R}$ is the set of rigid displacements and small rotations of all bodies of the system.

DEFINITION 3.2. Let $\mathcal{R}^* = \{\mathbf{z} \in V \ \cap\ \mathcal{R} |\ z_n^k - z_n^l = 0 \text{ on } \Gamma_c^{kl}\}$.

LEMMA 3.2. Let $\Gamma_u = \bigcup_{l=1}^S \Gamma_u^l$, Γ_u^l be open, non-empty $\forall l 1 \leq l \leq S$. Then $V \ \cap\ \mathcal{R} = \{\emptyset\}$.

The proof follows from a similar assertion for one elastic body [21].

REMARK 3.1. In the coercive case, when $V \cap \mathcal{R} = \{\emptyset\}$, the Korn inequality is valid on the whole space V :

$$c_1 \|\mathbf{v}\|^2 \leq |\mathbf{v}|^2, \quad c_1 > 0 \qquad (1.27)$$

where c_1 is independent of $\mathbf{v} \in V$.

The remaining cases, when $V \cap \mathcal{R} \neq \{\emptyset\}$, are called semicoercive.

Now, we may proceed to the existence theorems. The first Theorem solves the simplest coercive case.

THEOREM 3.1. Let the assumptions of the Lemma 3.2. be fulfilled.
Then $\mathcal{L}$ is coercive on K and the unique solution of the problem (1.24) exists.

Semicoercive case, which is more general, is considered in Theorem 3.2.

THEOREM 3.2. Let $\mathcal{R}^* = \{\emptyset\}$, $L(\mathbf{y}) \neq 0 \quad \forall \mathbf{y} \in V \ \cap \ \mathcal{R} - \{\emptyset\}$.
Let either $\qquad K \ \cap \ \mathcal{R} = \{\emptyset\}$
or $\qquad K \ \cap \ \mathcal{R} \neq \{\emptyset\}, \quad L(\mathbf{y}) < 0 \quad \forall \mathbf{y} \in K \ \cap \ \mathcal{R} - \{\emptyset\}$.
Then $\mathcal{L}$ is coercive on K and the unique solution of (1.24) exists.
Proof. See [8]

Let us emphasize that fulfilling the assumptions of the previous Theorem does not always need to be easy, especially when more than two bodies in contact are considered.

1.4 Finite element approximation of the problem

The problem $(\mathcal{P})$ in the form of (1.24) cannot be solved generally. It is necessary to replace it by the sequence of problems for which we can find a solution. We will construct the finite dimensional approximation of the set of admissible displacements. This set will be used for the definition of the approximate solution of $(\mathcal{P})$.

Consider the regular, consistent triangulation T_h of the regions Ω^s $1 \leq s \leq S$ with nodes a_i. Ω^s have a polygonal boundary and h designates the longest side of the triangles (cf. e.g. [8]). As the boundary is polygonal, it holds $\Gamma_c^{kl} = \bigcup_{j=1}^{J} \Gamma_{cj}^{kl}$, $\Gamma_0 = \bigcup_{j=1}^{J'} \Gamma_{0j}$, where Γ_{cj}^{kl}, Γ_{0j} are the abscissae, whose endpoints are the vertices of the region Ω. $J = J(k,l)$ is the number of straight lines on the unilateral contact boundary between the bodies k and l, and J' is the number of straight lines on the bilateral contact boundary. For every node a_i of the triangulation on Γ_c^{kl}, and on Γ_0, define the set of indices $\mathcal{N}_i^{kl} = \{j \in \{1,\ldots,J\} | \, a_i \in \Gamma_{cj}^{kl}\}$ and $\mathcal{N}_i = \{j \in \{1,\ldots,J'\} | \, a_i \in \Gamma_{0j}\}$, respectively. (In plane problems $\mathcal{N}_i$ has 1 or 2 members. In the latter case the node a_i is the vertex of the region laying inside Γ_c^{kl} or Γ_0). Let, on the abscissae Γ_{cj}^{kl} $\mathbf{n}_j$ denote the outward normal to the boundary $\partial\Omega^k$. Let us define the finite dimensional approximations of V_{u_0} and K_{u_0}.

$$\begin{aligned}
(V_{u_0})_h \quad &= \quad \{\mathbf{v}_h \in [C(\overline{\Omega}^1)]^2 \times \ldots \times [C(\overline{\Omega}^S)]^2 | \, \mathbf{v}_{|T} \in [P_1(T)]^2 \, \forall T \in T_h \, ; \\
&\qquad \mathbf{v}_h(a_i)\mathbf{n}_j = 0, \, j \in \mathcal{N}_i, \, a_i \in \Gamma_0; \\
&\qquad \mathbf{v}_h(a_i) = \mathbf{u}_0(a_i), \, a_i \in \Gamma_u \, \}, \qquad (1.28) \\
(K_{u_0})_h \quad &= \quad \{\mathbf{v}_h \in (V_{u_0})_h | (\mathbf{v}_h^k - \mathbf{v}_h^l)(a_i)\mathbf{n}_j \leq 0, \\
&\qquad j \in \mathcal{N}_i^{kl}, \, a_i \in \Gamma_c^{kl}, \, 1 \leq k \leq l \leq S \, \}. \qquad (1.29)
\end{aligned}$$

REMARK 4.1. Similarly as Remark 2.1., for $\mathbf{u}_0 \equiv 0$ we omit the index $\mathbf{u}_0$ in symbols V_h and K_h.

REMARK 4.2. It holds $K_h \subset K$.

REMARK 4.3. If we consider the term $v_n = \mathbf{v} \cdot \mathbf{n}$ (or $v_{hn} = \mathbf{v}_h \cdot \mathbf{n}$) on a certain edge Γ_m (e.g. the interpolation $r_h v_n$ on the element or the integration on Γ_m $\int_{\Gamma_m} v_n ds$ - see below), then the construction of $(K_{u_0})_h$ is convenient in this way. The definition of the interpolation will still be understood in this manner. However, we do not have a "bothsided" value of an outward normal in the vertices of the region. Hence, we define a unique value of the normal in the vertices of Ω and use the modifications of $(V_{u_0})_h$ and $(K_{u_0})_h$. The sets V and K which belong to the continuous problem remain unchanged.

$$\begin{aligned}
(V_{u_0})_h &= \{\mathbf{v}_h \in [C(\overline{\Omega}^1)]^2 \times \ldots \times [C(\overline{\Omega}^S)]^2 | \mathbf{v}_{|T} \in [P_1(T)]^2 \,\forall\, T \in T_h; \\
&\quad \mathbf{v}_h(a_i)\mathbf{n}(a_i) = 0,\ a_i \in \Gamma_0\,; \\
&\quad \mathbf{v}_h(a_i) = \mathbf{u}_0(a_i),\ a_i \in \Gamma_u\,\}\,, \qquad (1.30) \\
(K_{u_0})_h &= \{\mathbf{v}_h \in (V_{u_0})_h | (\mathbf{v}_h^k - \mathbf{v}_h^l)(a_i) \cdot \mathbf{n}(a_i) \le 0,\ a_i \in \Gamma_c^{kl}, \\
&\quad 1 \le k < l \le S\ \}, \qquad (1.31)
\end{aligned}$$

where $\mathbf{n}(a_i) = \|(\sum_{j \in \mathcal{N}_i} \mathbf{n}_j)/\overline{p}_i\|^{-1} \cdot (\sum_{j \in \mathcal{N}_i} \mathbf{n}_j)/\overline{p}_i$, and $\overline{p}_i$ is the cardinality of $\mathcal{N}_i^{kl}$ (or $\mathcal{N}_i$).

In the case when $\mathbf{u} \equiv 0$ on Γ_u, it holds $K_h \subset K$ again, as the projections of newly defined normals on the original normals are positive. This inclusion is also valid when the components of $\mathbf{u}_0$ are piecewise linear and continuous on Γ_u or constant on every Γ_u^l.

REMARK 4.4. This modified formulation does not create the almost linearly dependent rows in a constraint matrix which can cause certain difficulties in some methods. (See e.g. Lemma 2.6.1.). Rows that are numerically almost dependent rows may occur. For example when one approximates a curved boundary by a polygon [14] and especially in 3-D where more than 3 planes may stick in one point. In the developed preprocessor code it is possible to consider both definitions of $(V_{u_0})_h$, $(K_{u_0})_h$ and change them interactively for the particular problem (The difference is in few lines of source code).

DEFINITION 4.1. A function $\mathbf{u}_h \in (K_{u_0})_h$ is the solution of the approximate problem $(\mathcal{P}_h)$, if it is the minimum of the potential energy functional on the set of all admissible displacements, i.e.

$$\mathcal{L}(\mathbf{u}_h) \le \mathcal{L}(\mathbf{v}_h) \quad \forall\, \mathbf{v}_h \in (K_{u_0})_h. \qquad (1.32)$$

The problem (1.32) is equivalent to [5].
Find $\mathbf{u}_h \in (K_{u_0})_h$, such that

$$A\ (\mathbf{u}_h, \mathbf{v}_h - \mathbf{u}_h) \ge L(\mathbf{v}_h - \mathbf{u}_h) \quad \forall\, \mathbf{v}_h \in (K_{u_0})_h. \qquad (1.33)$$

Suppose that in the case when $(K_{u_0})_h \not\subset K_{u_0}$ (i.e. $\mathbf{u}$ is general function) at least it holds that $\mathbf{w}_0 \in [\mathcal{H}^2(\Omega)]$. By the decomposition $\mathbf{u}_h = \mathbf{w}_h + \mathbf{r}_h\mathbf{w}_0$ we transform this case into the problem with zero Dirichlet boundary condition. By using the symbol $\mathbf{r}_h\mathbf{w}_0$ we designate the linear interpolation of the vector function $\mathbf{w}_0$ on the triangulation, i.e. $\mathbf{r}_h\mathbf{w}_0 = (r_h w_{01},\, r_h w_{02})$.

The following equivalence holds.
$\mathbf{u}_h$ is the solution of (1.33) iff $\mathbf{w}_h$ is the solution of

$$\begin{aligned} A(\mathbf{w}_h, \mathbf{t}_h - \mathbf{w}_h) &\geq L(\mathbf{t}_h - \mathbf{w}_h) - A(\mathbf{r}_h\mathbf{w}_0, \mathbf{t}_h - \mathbf{w}_h) \\ &= L_0(\mathbf{t}_h - \mathbf{w}_h) \quad \forall\, \mathbf{t}_h \in K_{0h}. \end{aligned} \tag{1.34}$$

If we know the behaviour of $\|\mathbf{w} - \mathbf{w}_h\|$, we have

$$\|\mathbf{u} - \mathbf{u}_h\| \leq \|\mathbf{w} - \mathbf{w}_h\| + \|\mathbf{w}_0 - \mathbf{r}_h\mathbf{w}_0\| \leq \|\mathbf{w} - \mathbf{w}_h\| + O(h). \tag{1.35}$$

Hence, we consider $\mathbf{u}_0 \equiv 0$ in what follows.

LEMMA 4.1. (Falk's lemma, [8, 19, 20])

$$\begin{aligned} c_0|\mathbf{u} - \mathbf{u}_h|^2 \;\leq\; A(\mathbf{u} - \mathbf{u}_h, \mathbf{u} - \mathbf{u}_h) \;\leq\; & A(\mathbf{u}_h - \mathbf{u}, \mathbf{v}_h - \mathbf{u}) + A(\mathbf{u}, \mathbf{v}_h - \mathbf{u}) \\ & - L(\mathbf{v}_h - \mathbf{u}) + A(\mathbf{u}, \mathbf{v} - \mathbf{u}_h) \\ & - L(\mathbf{v} - \mathbf{u}_h) \end{aligned} \tag{1.36}$$

$$\forall \mathbf{v} \in K \;,\; \forall \mathbf{v}_h \in K_h, \quad h \in (0,1),\ c_0 > 0 \text{ is indep. of } \mathbf{u}.$$

It is obvious that for the existence and uniqueness of $(\mathcal{P}_h)$ it is sufficient to fulfill the conditions ensuring the existence and uniqueness in a continuous case. Indeed, the coercivity $\mathcal{L}$ on K ensures the coercivity on $K_h \subset K$. The following Theorem shows the relation between $(\mathcal{P})$ and $(\mathcal{P}_h)$ when $h \to 0$. The basic assumption is the sufficient smoothness of the solution. The uniqueness is not required.

THEOREM 4.1. Let $\mathbf{u}$ and $\mathbf{u}_h$ be the solutions of the problems $(\mathcal{P})$ and $(\mathcal{P}_h)$, respectively. Let $\mathbf{u} \in \mathcal{H}^2(\Omega) \cap K$, $\mathbf{u}^k, \mathbf{u}^l \in [W^{1,\infty}(\Gamma^{kl}_{cj})]^2$, $\tau^k, \tau^l \in [L^\infty(\Gamma_c)]^2$. Let the number of points, where the change $u^k_n - u^l_n < 0$ to $u^k_n - u^l_n = 0$ appears, be finite. Then

$$|\mathbf{u} - \mathbf{u}_h| = O(h).$$

Proof. As $K_h \subset K$, we can take $\mathbf{v} = \mathbf{u}_h$ and the last two terms in (1.36) vanish. Furthermore, due to (1.25)

$$A(\mathbf{u}, \mathbf{v}_h - \mathbf{u}) - L(\mathbf{v}_h - \mathbf{u}) = \int_{\cup\Gamma^{kl}_c} \tau^k_n((v^k_{hn} - v^l_{hn}) - (u^k_n - u^l_n))ds\,,$$

and

$$A(\mathbf{u}_h - \mathbf{u}, \mathbf{v}_h - \mathbf{u}) \;\leq\; \frac{1}{2}[A(\mathbf{u}_h - \mathbf{u}, \mathbf{u}_h - \mathbf{u}) + A(\mathbf{v}_h - \mathbf{u}, \mathbf{v}_h - \mathbf{u})].$$

By virtue of (1.21)-(1.23) and both of the inequalities in (1.36), we get

$$\frac{1}{2}c_0|\mathbf{u}-\mathbf{u}_h|^2 \leq \frac{1}{2}C_1C_2\|\mathbf{v}_h-\mathbf{u}\|^2 + \int_{\cup\Gamma_c^{kl}} \tau_n^k((v_{hn}^k - v_{hn}^l) - (u_n^k - u_n^l))ds\,, \tag{1.37}$$

Let $\mathbf{v}_h = \mathbf{r}_h\mathbf{u}$. Then $\|\mathbf{v}_h - \mathbf{u}_h\|^2 = O(h^2)$.
It holds on Γ_c:

$$v_{hn}^k - v_{hn}^l = (\mathbf{v}_h^k - \mathbf{v}_h^l)\cdot\mathbf{n}_j = (\mathbf{r}_h\mathbf{v}^k - \mathbf{r}_h\mathbf{v}^l)\cdot\mathbf{n} = r_h(v_n^k - v_n^l)\,,$$

where $\mathbf{n}$ is the outward normal to $\Gamma_{cj}^{kl} \subset \Gamma_c^{kl}$.
Now, if $u_n^k - u_n^l \equiv 0$ on Γ_{cj}^{kl}, then

$$\int_{\cup\Gamma_{cj}^{kl}} \tau_n^k(r_h(u_n^k - u_n^l) - (u_n^k - u_n^l))ds = 0.$$

If $u_n^k - u_n^l < 0$ on Γ_{cj}^{kl}, then $\tau_n^k = 0$ and this integral is zero again. Thus,

$$\begin{aligned}\int_{\Gamma_c} \tau_n^k(r_h(u_n^k - u_n^l) - (u_n^k - u_n^l))ds \;\;&=\;\; \sum_{j'}\int_{\cup\Gamma_{cj'}^{kl}} \tau_n^k(r_h(u_n^k - u_n^l) - \\ &\qquad (u_n^k - u_n^l))ds,\end{aligned} \tag{1.38}$$

where $\Gamma_{cj'}^{kl}$ are such abscissae, on which both $u_n^k - u_n^l = 0$ and $u_n^k - u_n^l < 0$. By the assumption, their number is finite.

$$\begin{aligned}\int_{\cup\Gamma_{cj'}^{kl}} \tau_n^k(r_h(u_n^k - u_n^l) - (u_n^k - u_n^l))ds \;\;&\leq \\ \|\tau_n^k\|_{\infty,\Gamma_{cj'}^{kl}} \cdot \|r_h(u_n^k - u_n^l) - (u_n^k - u_n^l)\|_{\infty,\Gamma_{cj'}^{kl}} \cdot h \;\;&\leq\;\; C_3h^2.\end{aligned} \tag{1.39}$$

Combining (1.37-1.39) we get the assertion. □

COROLLARY. In the coercive case, (1.27) and Theorem 4.1 gives

$$\|\mathbf{u} - \mathbf{u}_h\| = O(h).$$

Chapter 2

Numerical methods for the contact problem

2.1 Introduction of degrees of freedom and the constraint matrix

Study now how to solve the problem $(\mathcal{P}_h)$. If we do not consider the constraints on Γ_0 and Γ_u, we may write for $\mathbf{v}_h \in V_h$,

$$\mathbf{v}_h = (\mathbf{v}_h^1, \mathbf{v}_h^2, \ldots, \mathbf{v}_h^S),\ \mathbf{v}_h^l = (v_{h1}^l, v_{h2}^l),\ 1 \le l \le S,$$

$$v_{hi}^l(\mathbf{x}) = \sum_{j=1}^{M(l)} v_i^l(a_j^l)\varphi_j^l(\mathbf{x}) = \sum_{j=1}^{M(l)} x_{ij}^l \varphi_j^l(\mathbf{x})\,, i = 1, 2;\ l = 1, \ldots, S, \tag{2.1}$$

where a_j^l are the nodes of the triangulation, x_{ij}^l the degrees of freedom, $\varphi_j^l(\mathbf{x})$ the basis functions on V_h such, that

$$\varphi_i^l(a_j^l) = \delta_{ij} \quad i, j = 1, \ldots, M(l),\ l = 1, \ldots, S, \tag{2.2}$$

and $M(l)$ is the number of nodes in the l-th body.

In regard to (2.1),(2.2), the constraints on Γ_0 and Γ_u always bind degrees of freedom x_{ij}^l which belong to one node of the triangulation. The constraints on $\Gamma_c = \cup\Gamma_c^{kl}$ (see Sec. 1.4.) express the relation between the displacements $\mathbf{u}_h^k$ and $\mathbf{u}_h^l$ of the two nodes, which form the contact pair, and each of them belongs to different body $(1 \le k < l \le S)$ of the model. Therefore, one constraint binds two pairs of degrees of freedom. For simplicity's sake we denote the nodes in a contact pair by the same symbol.

All constraints can be written as

$$\begin{array}{llll} x_{i1} & = & \mathbf{u}_{01}(a_i) & a_i \in \Gamma_u, \\ \quad x_{i2} & = & \mathbf{u}_{02}(a_i) & a_i \in \Gamma_u, \\ x_{i1}n_1(a_i) + x_{i2}n_2(a_i) & = & 0 & a_i \in \Gamma_0, \\ x_{i1}^k n_1(a_i) + x_{i2}^k n_2(a_i) - x_{i1}^l n_1(a_i) - x_{i2}^l n_2(a_i) & \le & 0 & a_i \in \Gamma_c, \end{array} \tag{2.3}$$

where the normal $\mathbf{n}(a_i) = (n_1(a_i), n_2(a_i))$ was defined in Sec. 1.4.

The conditions on Γ_u will be satisfied during the assembling of the stiffness matrix and the right hand side vector, i.e. during the assembling of the functional $\mathcal{L}$ in (1.30). The corresponding degrees of freedom are constant, i.e. they are not dependent. In the conditions on Γ_0 one parameter of x_{i1}, x_{i2} can be also expressed by the second one. (We choose that one with greater value of $|n_s(a_i)|$ as the dependent one).

For these reasons we may consider only the conditions on Γ_c in what follows. These can be written in a matrix form as

$$\begin{array}{ll} Ax \le 0, & A \text{ is of the type } M \times N, \\ & M \text{ is the number of constraints,} \\ & N \text{ is the number of degrees of freedom in the whole model.} \end{array}$$

2.2 The assembling of the functional $\mathcal{L}$

At first, we will form $\mathcal{L}$ on particular triangles and edges of the triangulation. Let us introduce the vector 3×1, $\overline{e}_{ij}$, $1 \le i \le j \le 2$, by the relations

$$\begin{aligned} \overline{e}_{ii} &= e_{ii} \\ \overline{e}_{12} &= 2e_{12}\ , \end{aligned} \tag{2.4}$$

and $f(x) = \mathcal{L}(x_s\varphi_s) = \mathcal{L}(\mathbf{v}_h), \quad x \in R^N$.

It holds that

$$\sum_{i,j,k,l=1}^{2} c_{ijkl}e_{kl}e_{ij} = \sum_{\substack{i\le j\ k\le l\\ i,j,k,l=1}}^{2} c_{ijkl}\overline{e}_{kl}\overline{e}_{ij},$$

which can be written in the matrix form as $\overline{\mathbf{e}}^T D\overline{\mathbf{e}}$, where the matrix D is 3×3, symmetric.

In regard to the choice of V_h, we seek the vector $\mathbf{u}_h = (u_{h1}, u_{h2})$ in the form of linear polynomial on every triangle T_k and edge B_l of the triangulation. Similarly as in [14], we will obtain $f_k(x_k)$ on a given element in the form $f_k(x) = \frac{1}{2}x_k^T C_k x_k - x_k^T d_k$, C_k is 6×6, $x_k = (6 \times 1)$, $d_k = (6 \times 1)$. We will also obtain the contributions from the edges on Γ_τ, $x_l^T h_l$, $x_l = (4 \times 1)$, $h_l = (4 \times 1)$ which will be added to the linear term of $\mathcal{L}$.

Then, we eliminate the contingent degrees of freedom on Γ_u or Γ_0. During the assembling of $\mathcal{L}$ in the whole model, we follow the global numbering of nodes and the numbering of degrees of freedom (i.e. the numbering of the variables in the functional).

The problem (1.32) then leads to the problem $(\mathcal{P}_d)$:

$$f(x) = \tfrac{1}{2}x^T Cx - x^T d \to \min$$

with constraints

$$Ax \le 0.$$

REMARK 2.1. The global stiffness matrix C is of the type $N \times N$, block diagonal, every block is sparse, symmetric, positive semidefinite matrix and corresponds to just one body in the model. In the coercive case (Rem. 1.3.1) C is positive definite. This property of the stiffness matrix is the fundamental assumption for some tested methods.

The constraint matrix A is of the type $M \times N$, $M \ll N$; we assume its rows to be linearly independent.

REMARK 2.2. We denote $K_d = \{x \in R^N | Ax \leq 0\}$.

REMARK 2.3. During the assembling we order the degrees of freedom in the following manner

$$x^1_{11}, x^1_{21}, x^1_{12}, x^1_{22}, \dots, x^1_{1,M(1)}, x^1_{2,M(1)}, \\ x^2_{1,M(1)+1}, x^2_{2,M(1)+1}, \dots,$$

which can be simplified as:

$$\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4, \dots, \overline{x}_{2M-1}, \overline{x}_{2M}, \overline{x}_{2M+1}, \overline{x}_{2M+2}, \dots$$

where we put $M = M(1)$.
By the contact equations we then mean the equations with such indices n; $\overline{x}_n = x^l_{ij} = v^l_i(a^l_j)$, for which $a^l_i \in \Gamma_c$.

2.3 Storage of the matrices

It is obvious that C and A have a great number of zero entries. As a result, it is necessary to devote some attention to the modes of their storage in computer memory.

Stiffness matrix C was stored in two formats. We use its symmetry in both of them. At first we tried to test SKY-LINE (profile) format (e.g. [4]), where we store only the active length from each column j, i.e. the entries $c_{jj}, c_{j-1j}, \dots, c_{i_0 j}$, $i_0 = i_0(j) = \min\{i | c_{ij} \neq 0\}$. The stored entries of all columns form the Sky-line. It is convenient, after the mesh generation, to renumber the nodes in the whole model in order to reduce the active length of the columns (bandwidth). (It turned out to be more convenient to pass through particular regions Ω^l of the model).

In the second format - SPARSE (e.g. [1]), we store only non-zero entries, which lie above the diagonal, from each column.

Some of the methods for solving $(\mathcal{P}_d)$ were tested in both formats. The results show that if we do not use the renumbering, the SPARSE format is under the same conditions faster and has smaller memory requirements than SKY-LINE. These differences in two dimensional problems almost vanish when using renumbering.

It is reasonable to use only SKY-LINE in the methods in which fill-in occurs (elimination, complete factorization). It turns out that the solution of $(\mathcal{P}_d)$ will accelerate by using the "preconditioning" that is based on the complete decomposition surprisingly. On the whole we obtain the fastest tested method which, at least in the 2-D using renumbering, has not very great memory requirements.

SKY-LINE (C):

N - number of degrees of freedom (size of C)

NWK - number of stored entries

C(NWK) - real∗8 array of entries C

MAXC(N+1) - $MAXC(J)$ is the address to the array C
where c_{jj} is stored ;
$MAXC(J+1)$ is the address to the array C
where $c_{i_0 j}$ is stored.

SPARSE (C):

N - size of C

LJC=N+1 - length of the array JC

LIC - number of stored entries

C(LIC) - real∗8 array of entries C

IC(LIC) - row indices of corresponding entries in C array

JC(LJC) - addresses of the first non-zero entries
of particular columns in C and IC arrays, i.e.
$I = JC(J) \Rightarrow IC(I) = I_0$ and $C(I) = c_{i_0 j}$.

SPARSE (A):

M,N - dimensions of A

LIA=M+1 - length of the array JC

LJA - number of stored entries

A(LJA) - real∗8 array of entries A

JA(LJA) - column indices of corresponding entries in A array

IA(LIA) - addresses of the first non-zero entries
of particular rows in A and JA arrays, i.e.
$J = IA(I) \Rightarrow JA(J) = J_0$ and $A(J) = a_{IJ_0}$,
where $J_0 = \min\{K | a_{IK} \neq 0\}$.

2.4 The elementary operations with the matrices

The most essential operations are: the matrix product, the elimination and the decomposition.

The matrix product occurs very often in iterative methods. Here we deal with the following types:

Cx C symmetric, in SKY-LINE or in SPARSE

Ax, A^Tx A stored in SPARSE, we often multiply only by a certain subset of the rows of A. Therefore, the elementary operation is $(Ax)_i$ (multiplication by i-th row)

$E^{-1}y$, $E^{-T}y$ The factor E^{-T} is stored (SKY-LINE or SPARSE, unlike C, however, is not symmetric); the multipication of inverse matrices by the vector is transformed into the solution of corresponding triangular systems.

The multiplication Cx is carried out by the columns. We will find the adresses and row indices for a given column from the corresponding arrays. For $y = Cx$, we have

$$y_i = \sum_{j=1}^{N} c_{ij}x_j = \sum_{j=1}^{i} c_{ji}x_j + \sum_{j=i+1}^{N} c_{ij}x_j \quad ,$$

i.e. for $1 \leq i \leq N$:

$$\begin{array}{rll} y_i = 0 & & \text{after passing through the columns } 1,\ldots,i-1, \\ y_i^{(i)} = \sum_{j=1}^{i} c_{ji}x_j & & \text{after passing through col. } i, \\ y_i^{(j)} = y_i^{(j-1)} + c_{ij}x_j & & \text{after pass. thr. col. } j,\ i < j \leq N. \end{array}$$

By the partial Gaussian elimination on the system $Cx = d$ to the row L, we will call its transformation to the form

$$\left(\begin{array}{cc|c} I & B & \overline{d}_1 \\ \emptyset & \overline{C} & \overline{d}_2 \end{array} \right)$$

where $I = L \times L$, $\overline{C} = (N-L)\times(N-L)$, $B = L\times(N-L)$, $\overline{d}_1 = L\times 1$, $\overline{d}_2 = (N-L)\times 1$.

For the elimination, we assume C to be positive definite and therefore we do not consider the permutations of rows and columns. A more general version does not assume the position of contact equations on the last $N-L$ places (see Sec.9.).

At first we perform the forward elimination of the first L unknowns thus obtaining the triangular form. Then, for the same unknowns we perform the backward elimination (similar to Gauss-Jordan elimination). It is obvious that if C is symmetric, then i-th derived system is also symmetric.

By using the common notation (we put $s = 1$ at the begining of the process; $c_{ij}^{(1)} = c_{ij}$), we have

$$c_{ij}^{(s)} = c_{ji}^{(s)} \qquad i,j = s,\ldots,N\, s = 1,\ldots,L+1. \tag{2.5}$$

We adjust the well-known formula

$$c_{ij}^{(k+1)} = c_{ij}^{(k)} - \left(\frac{c_{ik}^{(k)}}{c_{kk}^{(k)}} \right) c_{kj}^{(k)} \qquad i,j = k+1, \ldots, N; \quad k = 1, \ldots, L$$

so that we could pass through the columns and perform the elimination for each entry at one time.

$$\begin{aligned} c_{ij}^{(i_0)} &= c_{ij}^{(1)} - \sum_{m=1}^{i_0-1} \left(\frac{c_{im}^{(m)}}{c_{mm}^{(m)}} \right) c_{mj}^{(m)} = \\ &= c_{ij}^{(1)} - \sum_{m=1}^{i_0-1} \left(\frac{c_{mi}^{(m)}}{c_{mm}^{(m)}} \right) c_{mj}^{(m)} = \\ &= c_{ij}^{(1)} - \sum_{m=1}^{(i_0-1)} \overline{c}_{mi}^{(m)} c_{mj}^{(m)} \\ & \quad i = 1, \ldots, N; \quad i \leq j; \qquad i_0 = \min(i, L+1). \end{aligned} \tag{2.6}$$

Suppose that we already have $j-1$ columns ($j \leq L$) after the elimination, i.e.

$$\begin{pmatrix} c_{11}^{(1)} & \overline{c}_{12}^{(1)} & \ldots & \ldots & \overline{c}_{1j-1}^{(1)} & c_{1j}^{(1)} & \ldots & \ldots & c_{1N}^{(1)} \\ & c_{22}^{(2)} & \ldots & \ldots & \overline{c}_{2j-1}^{(2)} & c_{2j}^{(1)} & \ldots & \ldots & c_{2N}^{(1)} \\ & & & & \ldots & & & & \\ & & & & c_{j-1j-1}^{(j-1)} & c_{j-1j}^{(1)} & \ldots & \ldots & c_{j-1N}^{(1)} \\ & & & & & & & \ldots & \\ & & & & & & & & c_{NN}^{(1)} \end{pmatrix}$$

(A unit diagonal is created during the elimination and we store here the corresponding coefficients for the final adjustment of the j-th column).

It can be seen now that we do not need to perform the elimination for c_{1j}. To eliminate c_{2j} we only need the entries from the second column and $c_{1j}^{(1)}$. Generally, to eliminate c_{ij} $(i \leq j)$ we only need the entries from the i-th column and the already created entries in the j-th column. To eliminate c_{jj}, we only need the entries $c_{mj}^{(m)}$ and $\overline{c}_{mj}^{(m)}$, $1 \leq m \leq \min(j-1, L)$.

When using SKY-LINE, we do not perform the elimination on entries outside the Sky-line (the role of entry in the first row has now a non-zero entry with the lowest row index). Furthermore, we do not need to calculate $\overline{c}_{mi}^{(m)} c_{mj}^{(m)}$ in (2.6) when at least one of these entries is outside the Sky-line.

The forward elimination for the right hand side is done in the same way.

During the backward elimination we zero the rows $1 \leq l \leq L$ which are above the diagonal to the L-th column. We succesively obtain the values

$$c_{L-1,j_1}^{(L)}, c_{L-2,j_2}^{(L-1)}, c_{L-2,j_2}^{(L)}, \ldots \qquad L - i < j_i \leq N$$

At the same time we have for $i = 1, \ldots, L-1$, $j = L-i+1, \ldots, L$

$$\left. \begin{array}{rl} c_{L-i,j}^{(m)} = c_{L-i,j}^{(L-i)} & L-i \leq m < j, \\ c_{L-i,j}^{(m)} = 0 & L-i < j \leq m \leq L, \\ c_{L-i,L-i}^{(m)} = 1 & L-i \leq m \leq L \end{array} \right\} \tag{2.7}$$

Thus, we use the elimination formula

$$\begin{aligned} c^{(L)}_{L-i,j} &= c^{(L-i)}_{L-i,j} - \sum_{l=0}^{i-1} \left(\frac{c^{(L-l-1)}_{L-i,L-l}}{c^{(L)}_{L-l,L-l}} \right) c^{(L)}_{L-l,j} = \\ &= \sum_{l=0}^{i-1} c^{(L-i)}_{L-i,L-l} c^{(L)}_{L-l,j} \\ &\qquad i = 1, \ldots, L-1 \quad j = L+1, \ldots, N \end{aligned}$$

The entries $c^{(L-i)}_{L-i,j}$, $c^{(L-i)}_{L-i,L-l}$ are known from the forward elimination and $c^{(L)}_{L-l,j}$ from the already performed backward elimination. Consequently, the backward elimination can also be performed through the columns. We may consider only the right hand side and the columns for which $j > L$. Obviously, fill-in occurs for such columns in the upper part of C. It is necessary to store the full length of these columns. If $L \ll N$, we would lose the advantages of the SKY-LINE format, but this is not our case, since L is the number of the non-contact degrees of freedom. For the columns $1, \ldots, L$ the SKY-LINE is very efficient.

The variants of Choleski decomposition (incomplete, incomplete with adding to the diagonal, complete) are performed similarly as the elimination. By doing this, we proceed from the formula

$$\begin{aligned} l_{ij} &= c_{ij} - \sum_{m=1}^{i-1} l_{mi} l_{mj} & & 1 \le i < j, \quad j = 1, \ldots, N, \\ l_{jj} &= \sqrt{c_{jj} - \sum_{m=1}^{i-1} l^2_{mj}} & & j = 1, \ldots, N \end{aligned}$$

We again pass through the columns and consider only the entries in the Sky-line (for SKY-LINE format) or only the non-zero entries (for SPARSE format). Therefore, in the SPARSE we are selecting the entries between the addresses $JC(J)$ and $JC(J+1)-1$. However, in the SPARSE format it is necessary for the variant with adding to the diagonal to pass through each entry in the Sky-line . This can be accomplished by a small modification of the algorithm. The calculation of $l_{mi} l_{mj}$ is similar for SKY-LINE and for elimination. For SPARSE we must succesively search in columns i and j for the pairs with the same row indices (The array IC). The complete decomposition is created only for SKY-LINE format.

2.5 The termination

In the following paragraphs we desribe and test several numerical algorithms for the problem $(\mathcal{P}_d)$. To stop the process, we use the usual termination criterion:
stop, if $ERR < \epsilon$, where
$ERR = \|x^{k+1} - x^k\| / \max(1.0, \|x^k\|)$, x^k is the solution in k-th iteration, $k \le MAXIT$, and ϵ is the prescribed tolerance (mostly $\epsilon = 10^{-6}$). $MAXIT$ is the maximum number of iterations. For the overflow test we use the value $MAXVAL = 10^{15} - 10^{20}$.

2.6 The conjugate gradient method with constraints

This method belongs to the gradient projection methods, and generally solves the problem

$$\begin{aligned} f(x) = \tfrac{1}{2}x^T C x - x^T d &\rightarrow \min \\ x^T a_i - b_i \le 0 \quad & i \in I^- \\ x^T a_i - b_i = 0 \quad & i \in I^0 \end{aligned}$$

where $x, a_i \in R^N$, $d \in R^M$, $\quad I^- \cup I^0 = \{1, \dots, M\}$, C symmetric, positive semidefinite matrix $N \times N$, $b_i \in R$.

In our case, if we include the conditions on Γ_0 into $\mathcal{L}(\mathbf{v}_h)$, we will have $I^0 = \{\emptyset\}$, i.e. the problem $(\mathcal{P}_d)$.

The principal idea of the algorithm [23] lies in the succesive minimization of $f(x)$ on the facets created by constraints, for which the equality is satisfied. We solve minimization problem on each of such facets by using the conjugate gradient method (CGM). As CGM has finite number of steps and the number of facets is also finite (sometimes very great, however), it is obvious that the algorithm converges after a finite number of steps.

Denote by A_I the matrix whose rows have the indices $i \in I \subseteq (I^- \cup I^0)$.

LEMMA 6.1. Let the vectors a_i, $i \in I \subseteq (I^- \cup I^0)$. Then the matrix $A_I A_I^T$ is regular.
Proof. See [23].

Define the projection

$$\begin{aligned} P_I = A_I^T \cdot (A_I A_I^T)^{-1} \cdot A_I \quad & \text{if } I \neq \{\emptyset\} \\ P_I = 0 \quad & \text{if } I = \{\emptyset\} \end{aligned}$$

$$\begin{aligned} \text{Let } \; & J = \{i \in I^0 \cup I^-, \, (x^0)^T a_i - b_i = 0\} \\ \text{and } \; & u^k = -(A_J A_J^T)^{-1} \cdot A_J f'(x^k) \qquad k = 0, 1, \dots \end{aligned}$$

$$\begin{aligned} \text{It holds} \quad & f'(x^k) = Cx^k - d, \text{ and} \\ & (I - P_J) f'(x^k) = f'(x^k) + A_J^T u^k. \end{aligned}$$

We may now express the scheme of the algorithm as follows

```
$x^0$ ...the initial guess, which satisfies the constraints
$IT = 0$
$f'(x^0) = Cx^0 - d$
DO WHILE ( $IT < MAXIT$ )
   Set J
     CALL PROJECT($J, f'(x^0), u^0, (I - P_J) f'(x^0)$)

     IF ($\|(I - P_J) f'(x^0)\| \approx 0$) THEN
```

```
            IF $(u_i^0 \geq 0 \; \forall i \in J \cap I^-)$ THEN
                    $x^* = x^0$   { solution }
                    GOTO 2
            ELSE
                    $j := \{ \, i \in J \cap I^- \; | u_i^0 < 0 \, \}$
                    $J' = J - \{j\}$
            ENDIF
        ELSE
          $J' = J$
        ENDIF

        CALL CG$(J', x^0, f'(x^0))$
    IT = IT + 1
    ENDDO
    {  maximum number of iterations reached  }

    2 END


       SUBROUTINE CG$(J', x, f')$
    { Conjugate gradients - unlike the standard CGM, we use the projection $(I - P_{J'})f'(x^k)$
    instead of the gradient $f'(x^k)$. We also have to check the non-active constraints and
    correct, in every iteration, the new step length $\alpha^{k+1} := \min(\alpha^{k+1}, \overline{\alpha}^{k+1})$, where
    $\overline{\alpha}^{k+1} = \min_{\mathcal{M}} \frac{-(a_i, x^k)}{(a_i, p^{k+1})}$   and   $\mathcal{M} := \{i | i \notin J' \wedge (a_i, p^{k+1}) > 0\}$. }

    Input: $J', x$
    Output: $x, f'$

    $k = 0$
    $x^0 = x$
    $f'(x^0) = f'$    { from previous iteration }
    DO WHILE $(k \; < \; MAXIT2)$

    CALL PROJECT$(J', f'(x^k), u, (I - P_{J'})f'(x^k))$
    $g = -(I - P_{J'})f'(x^k)$
    $r^{k+1} = \|g\|^2$
    IF $(r^{k+1} \; < \epsilon)$ THEN
         $x = x^k$
         $f' = f'(x^k)$
         RETURN
    ENDIF

    IF $(k = 0)$ THEN $p^1 = g$
    ELSE   $\beta^{k+1} = r^{k+1}/r^k$
```

```
           $\underline{p^{k+1} = g + \beta^{k+1} p^k}$
ENDIF

$\alpha 1 = r^{k+1}$
$\alpha 2 = (p^{k+1}, Cp^{k+1})$        { scal. product in $R^N$ }

IF $(\alpha 1 < \min(1.0, |\alpha 2|) * MAXVAL)$ THEN
   $\underline{\alpha^{k+1} = \alpha 1 / \alpha 2}$
ELSE
   $\alpha^{k+1} = MAXVAL$
ENDIF

$\mathcal{M} := \{i | i \notin J' \wedge (a_i, p^{k+1}) > 0\}$
IF $\mathcal{M} \neq \{\emptyset\}$ THEN
   $\underline{\overline{\alpha}^{k+1} = \min_{\mathcal{M}} \frac{b_i - (a_i, x^k)}{(a_i, p^{k+1})}}$   $\{b_i = 0$ in our case $\}$        (FF)
ELSE $\overline{\alpha}^{k+1} = MAXVAL$
ENDIF

IF $(\overline{\alpha}^{k+1} < \alpha^{k+1})$ THEN
             $x = x^k + \overline{\alpha}^{k+1} p^{k+1}$
             $f' = f'(x^k) + \overline{\alpha}^{k+1} C p^{k+1}$
             RETURN
ELSEIF $(\alpha^{k+1} = MAXVAL)$ THEN
                        STOP
ELSE
   $\underline{x^{k+1} = x^k + \alpha^{k+1} p^{k+1}}$
   $\underline{f'(x^{k+1}) = f'(x^k) + \alpha^{k+1} C p^{k+1}}$
ENDIF

$dd = \|x^{k+1} - x^k\| / (\max(1, \|x^k\|))$
IF $(dd < \epsilon)$ THEN
      $x = x^{k+1}$
      $f' = f'(x^{k+1})$
      RETURN
ENDIF

$k = k + 1$

ENDDO

   $x = x^k$    { point obtained after max. num. of iterations }
   $f' = f'(x^k)$

RETURN
```

```
$SUBROUTINE\ PROJECT(J, f'(x), u, (I - P_J)f'(x))$
{ The calculation of $u = -(A_J A_J^T)^{-1} \cdot A_J f'(x)$ and $(I - P_J)f'(x) = f'(x) + A_J^T u$ by the CG Method }

Input: $J, f'(x)$
Output: $u, (I - P_J)f'(x)$

$RETURN$
```

REM. 6.1. We set $x^0 = (0, \ldots, 0)$ for the initial guess. As A_I- has a special structure, we may also choose x^0 so that the inequalities are satisfied strictly ("inner point"). For the models, having only two bodies stuck in one point, degree of freedom x_r appears at most in one constraint a_s; we choose $x_r =$ - sign$(a_{sr}) \cdot k$, $k > 0$ suitable const. not exceeding the dimension of the model. We may also choose non-constrained degrees as proportional to k. When more than two bodies stick, the restricted number of degrees of freedom may appear in more constraints. We arrive at a contradiction to the previous choice if the corresponding coefficients for x_r have the opposite signs. Here we choose $x_r = 0$ again.

REM. 6.2. Denote the value of $\|(I - P_J)f'(x^0)\|$ in IT-th iteration $(0 \leq IT < MAXIT)$ by pg^{IT}. Then $pg^{IT} \approx 0$ numerically represents the comparison $[pg^{IT+1}/\max(1.0, pg^{IT})] < \epsilon$. Similarly, we use the test $u_i^0/u > (-\epsilon)$, where $u = \max(1.0, u_l^0)$ and $u_l^0 = \max\limits_{m \in J}(0.0, u_m^0)$ for the multipliers u_i^0. It is also necessary to test the magnitudes of x^k and p^k in a semicoercive case.

REM. 6.3. The value $MAXIT1$ depends on N. $MAXIT2$ is the number of iterations in CG. We should choose $N - m'$ where m' is the number of active constraints (see [23]). However, the result will be more accurate if we choose the value slightly greater than N (e.g. $\approx 2N$).

REM. 6.4. For some models, it is convenient to use the following strategy which is similar to [10]. We choose less strict tolerance for subproblems (subr. CG) in the first several iterations within the CGC subroutine. The tolerance is set to more strict value after a limited number of these iterations. We can get remarkable acceleration of the process.

REM. 6.5. If C is positive definite (cf. Rem. 2.1.), it can occur

$$(f'(x^k), p^{k+1}) \neq 0 \text{ and } (p^{k+1}, Cp^{k+1}) = 0.$$

In this case $f(x^k + \alpha p^{k+1})$ decreases when α is increased. If $\overline{\alpha}^{k+1} = MAXVAL$, then f on K_d is not bounded from below.

REM. 6.6. We may use the diagonal form of $(A_J A_J^T)$ in the case of "two bodies contact" (cf. Rem. 6.1.) for the calculation of the vector u in subroutine PROJECT. A more general case (when more than two bodies stick in one point or the preconditioning) can be solved as follows:
u solves the system $(A_J A_J^T)u = -A_J f'(x)$, where $A_J A_J^T$ is symmetric and positive definite. This property is due to definition and lin. independence of rows A_J. The minimization is carried out by the conjugate gradient method again. In this case, the dimension of the problem is far more lower (contact pairs), the matrix A_J is sparse and there are no constraints.

Matrix $(A_J A_J^T)$ is not stored, the multiplication $w = (A_J A_J^T)u$ is gradually transformed to $v = A_J^T u$, $w = A_J v$.

On the basis of the fact that $\overline{\alpha}^1 > 0$ (see Subroutine CG), we can prove that the CG algorithm makes a non-zero step (i.e. does not cycle) in the same way as in [23].

If the implication

$$j \in J \Rightarrow (\, j \in J' \vee (a_j, p^1) \leq 0\,).$$

is valid then it follows from the formula (FF) in the subroutine CG that $\overline{\alpha}^1 > 0$. Therefore, it is sufficient to focus the case $\|(I - P_J)f'(x^0)\| \approx 0$ and the removed index $j \in J - J'$.

LEMMA 6.2.([23]) Let $\|(I - P_J)f'(x^0)\| = 0$. Let $A_{J'}$ be created from A_J by removing the row with index $j|u_j^0 < 0$.
Then $(a_j, p^1) < 0$, $j \in J - J'$.

If the condition for removing more indices fulfill then, similarly as in [6], we choose the one with the greatest absolute value.

However, the condition $(a_j, p^1) < 0 \quad j \in J - J'$ may be fulfilled even in the case where more indices $\{j|u_j^0 < 0\}$ are removed (e.g. all with $j|u_j^0 < (-\epsilon)$ cf. Rem. 6.2.). The following Lemma shows this. In some cases we can accelerate the algorithm very much through these means.

LEMMA 6.3. Let $\|(I - P_J)f'(x^0)\| = 0$. Let $A_{J'}$ be created from A_J by removing the rows with indices $j|u_j^0 < 0$. Furthermore, let the rows of A_J satisfy $(a_i, a_j) = 0$, $i \neq j$, $i, j \in J$.
Then $(a_j, p_1) < 0$, $j \in J - J'$.
Proof.

$$\begin{aligned} 0 = (I - P_J)f'(x^0) &= f'(x^0) + A_J^T u^0 &&= f'(x^0) + A_{J'}^T u_{\prime}^0 + A_{J-J'}^T u_{\prime\prime}^0 \\ -p^1 = (I - P_{J'})f'(x^0) &= f'(x^0) + A_{J'}^T v_{\prime} &&, \\ & && \text{where } v_{\prime} = -(A_{J'} A_{J'}^T)^{-1} A_{J'} f'(x^0) \end{aligned}$$

Subtracting and multiplying by the vector a_j, $j \in J - J'$, we obtain $(a_j, p^1) = c \cdot u_{j\prime\prime}^0$ where $c = (a_j, a_j) > 0$ and from the assumption $u_{\prime\prime}^0 < 0$.
Thus, $(a_j, p^1) < 0$. □

COROLLARY. Let the assumptions of the previous Lemma be fulfilled.
Then $\overline{\alpha}^1 > 0$, and as a result the algorithm CGC does not cycle. □

The condition for the rows of A_J is fulfilled in “two bodies contact” (cf. Rem. 6.1., again). It may be slightly violated in a general case and also when the preconditioning is used. Nevertheless for such cases we often have an acceleration as well.

2.7 The preconditioning

Consider again the problem $(\mathcal{P}_d)$, i.e.

$$\begin{array}{c} f(x) = \frac{1}{2}x^T Cx - x^T d \rightarrow \min \\ Ax \leq 0\,. \end{array}$$

Now we assume C to be positive definite. Let W be a positive definite matrix $N \times N$ in the form $W = EE^T$. Introduce the transformation $y = E^T x$ and express $(\mathcal{P}_d)$ in terms of a new variable y.

$$\begin{array}{c} \overline{f}(y) = \frac{1}{2}y^t \overline{C} y - y^T \overline{d} \rightarrow \min \\ \overline{A} y \leq 0 \end{array}$$

where

$$\overline{C} = E^{-1} C E^{-T},\, \overline{d} = E^{-1} d \overline{A} = AE^{-T}$$

As $E^{-T}\overline{C}E^T = W^{-1}C$, the matrices $\overline{C}$ and $W^{-1}C$ have the same eigenvalues. The convergence of CGM depends on the condition number $(\lambda_{max}/\lambda_{min})$ of the matrix in the functional, in our case these are the matrices $C, \overline{C}$. The speed of the convergence increases when the condition number [1] is decreased. The lowest cond. number has a unity matrix. Therefore, we try to find W which is an easy invertible approximation of C or for which we can show that $W^{-1}C$ has lower condition number.

The preconditioning will be used when solving the problem on particular facets, i.e. in the subroutine CG. Let us write its steps for the transformed problem (without supplementary commands and tests).

```
$SUBROUTINE\ PCG(J', x\{= E^{-T}y\}, E^T, \overline{f}')$
$y^0 = y = E^T x$
$\overline{f}'(y^0) = \overline{C} y^0 - \overline{d}$

For $k = 0, 1, \ldots$

$\overline{g} = (I - \overline{P}_{J'})\overline{f}'(y^k)$
$\overline{r}^{k+1} = \|\overline{g}\|^2$
$IF\ (k = 0)\ THEN$
        $\overline{p}^1 = \overline{g}$
$ELSE$
```

```
        $\beta^{k+1} = \overline{r}^{k+1}/\overline{r}^{k}$
        $\overline{p}^{k+1} = \overline{g} + \beta^{k+1}\overline{p}^{k}$
ENDIF
$\alpha^{k+1} = \overline{r}^{k+1}/(\overline{p}^{k+1}, \overline{C}\overline{p}^{k+1})$
$\overline{\alpha}^{k+1} = \min_{\overline{\mathcal{M}}} \frac{-(\overline{a}_i, y^k)}{(\overline{a}_i, \overline{p}^{k+1})}$
IF $(\overline{\alpha}^{k+1} < \alpha^{k+1})$ THEN
        $y = y^k + \overline{\alpha}^{k+1}\overline{p}^{k+1}$
        $\overline{f}' = \overline{f}'(y^k) + \overline{\alpha}^{k+1}\overline{C}\overline{p}^{k+1}$     { and return to CGC }
ELSE
        $y^{k+1} = y^k + \alpha^{k+1}\overline{p}^{k+1}$
        $\overline{f}'(y^{k+1}) = \overline{f}'(y^k) + \alpha^{k+1}\overline{C}\overline{p}^{k+1}$
ENDIF
```

At the same time $\overline{P}_{J'} = \overline{A}_{J'}^T(\overline{A}_{J'}\overline{A}_{J'}^T)^{-1}\overline{A}_{J'}$
and $\overline{\mathcal{M}}$ is connected with $\mathcal{M}$ by the transformation $y = E^T x$.

Introducing a vector v^{k+1} by $v^{k+1} = E^{-T}\overline{p}^{k+1}$ and using

$$\begin{gathered} h^k := \overline{f}'(y^k) = E^{-1}f'(x^k), \\ (\overline{p}^{k+1}, \overline{C}\overline{p}^{k+1}) = (v^{k+1}, Cv^{k+1}) \quad \text{and} \\ (\overline{a}_i, \overline{p}^{k+1}) = (a_i, v^{k+1}) \ , \end{gathered}$$

we can write PCG in x variable.

```
SUBROUTINE PCG$(J', x, E^T, f')$
$f'(x^0) = f'$     { from previous iteration }

For $k = 0, 1, \ldots$

$h^k = E^{-1}f'(x^k)$
$\overline{g} = -(I - \overline{P}_{J'})h^k$
$\overline{r}^{k+1} = \|\overline{g}\|^2$
IF $(k = 0)$ THEN
        $v^1 = E^{-T}\overline{g}$
ELSE
        $\beta^{k+1} = \overline{r}^{k+1}/\overline{r}^{k}$
        $v^{k+1} = E^{-T}\overline{g} + \beta^{k+1}v^k$
ENDIF
$\alpha^{k+1} = \overline{r}^{k+1}/(v^{k+1}, Cv^{k+1})$
$\overline{\alpha}^{k+1} = \min_{\overline{\mathcal{M}}} \frac{-(a_i, x^k)}{(a_i, v^{k+1})}$

IF $(\overline{\alpha}^{k+1} < \alpha^{k+1})$ THEN
```

```
        $x = x^k + \overline{\alpha}^{k+1} v^{k+1}$
        $f' = f'(x^k) + \overline{\alpha}^{k+1} C v^{k+1}$    { and return to CGC }
$ELSE$
        $x^{k+1} = x^k + \alpha^{k+1} v^{k+1}$
        $f'(x^{k+1}) = f'(x^k) + \alpha^{k+1} C v^{k+1}$
$ENDIF$
```

In subroutine PROJECT, if it is called from PCG (the calculation of $\overline{g}$), the multiplications $A_{J'}x$, $A_{J'}^T x$ are replaced by $\overline{A}_{J'}y$, $\overline{A}_{J'}^T y$, i.e. $A_{J'}E^{-T}y$, $E^{-1}A_{J'}^T y$. As E^{-T} is regular, $\overline{A}_{J'}$ also has linearly independent rows.

The matrix $\overline{C}$ does not occur in the transformed problem.

2.8 The choice of the preconditioning matrix

The simplest choice is $W = D$ where D is the diagonal of C. In this case $E^T = D^{\frac{1}{2}}$ and it is sufficient to store only the vector.

Another possibility is the $SSOR$ decomposition [1] . Let $C = D + L + L^T$. The preconditioning matrix is of the form

$$W = \frac{1}{2-\omega}\left(\frac{1}{\omega}D + L\right)\left(\frac{1}{\omega}D\right)^{-1}\left(\frac{1}{\omega}D + L\right)^T \quad , \qquad 0 < \omega < 2 \tag{2.8}$$

factor $\frac{1}{2-\omega}$ may be omitted, thus

$$E^T = \left(\frac{1}{\omega}D\right)^{-\frac{1}{2}}\left(\frac{1}{\omega}D + L^T\right).$$

The condition number $\overline{C} = W^{-1}C$, $\kappa(\overline{C})$, may be under the certain assumptions smaller than $\kappa(C)$, as the following assertion shows [1].

THEOREM 8.1. Let C be positive definite and W be determined by (2.8). Let

$$\|D^{-\frac{1}{2}}LD^{-\frac{1}{2}}\|_\infty \le \frac{1}{2},\ \|D^{-\frac{1}{2}}L^TD^{-\frac{1}{2}}\|_\infty \le \frac{1}{2}.$$

Then

$$\min_{0<\omega<2} \kappa(\overline{C}) \le \sqrt{\frac{1}{2}\kappa(C)} + \frac{1}{2}$$

The optimal value of ω can be determined [1], if we can estimate the numbers

$$\mu = \max_{x\neq 0}\left(x^T Dx / x^T Hx\right) ,$$
$$\delta = \max_{x\neq 0} \frac{x^T(LD^{-1}L^T - \frac{1}{4}D)x}{x^T Hx}.$$

However, in our case (the presence of the constraints) the numerical experiments have shown that by choosing $\omega \neq 1$, the speed of the process does not change very much.

The incomplete factorization is more effective. Consider factorization $C = LL^T$ where L is a lower triangular. The incomplete factorization in the simplest form lies on determining only the entries of L where the original matrix C has non-zeros. We will obtain certain "approximation" of C.

Define $S_C = \{(i,j),\, c_{ij} \neq 0\}$. Proceeding from the Gaussian elimination, the steps of incomplete factorization can be written as follows:

for $r = 1, \dots, N-1$

$$l_{ir} = c_{ir}^{(r)} / c_{rr}^{(r)}$$

$$c_{ij}^{(r+1)} = \begin{cases} c_{ij}^{(r)} - l_{ir} c_{rj}^{(r)} & (r+1 \leq j \leq N) \wedge [(i,j) \in S_C] \wedge (i \neq j) \\ 0 & (r+1 \leq j \leq N) \wedge [(i,j) \notin S_C] \\ c_{ii}^{(r)} - l_{ir} c_{ri}^{(r)} & i = j \end{cases}$$

In another variant we add removed entries to the diagonal, i.e.

$$c_{ii}^{(r+1)} = c_{ii}^{(r)} - l_{ir} c_{ri}^{(r)} - \sum_{\substack{(i,k) \notin S_C \\ k=r+1}}^{N} l_{ir} c_{rk}^{(r)}$$

Thus, in the matrix form

$$C = EE^T + R = W + R$$

$$R = \sum_{r=1}^{N-1} R^{(r+1)} \quad r^{(r+1)} = \begin{cases} 0 & (i,j) \in S_C,\, i \neq j \\ c_{ij}^{(r)} - l_{ir} c_{rj}^{(r)} & (i,j) \notin S_C \\ \sum_{k=r+1}^{N} l_{ir} c_{rk}^{(r)} & i = j. \end{cases}$$

(The form of R follows from the description of the incomplete Gaussian elimination through lower triangular matrices L_r and from properties of these matrices.)

It is obvious that, in particular, the version with adding to the diagonal in the number of operations does not differ from a complete factorization very much. Its main advantage is in avoiding the fill-in which occurs in the complete factorization. This fact is not important in SKY-LINE format. Therefore, here we also test the complete factorization.

DEFINITION 8.1. C is $\overline{M}$-matrix, if

$$\begin{array}{lll} (1) & c_{ii} > 0 & i = 1, \dots, N-1 \\ (2) & c_{ij} < 0 & i \neq j \\ (3) & \max\{\, j \,|(i \leq j \leq N) \wedge (c_{ij} \neq 0)\} > i & \text{for } 1 \leq i < N \end{array}$$

The following Theorem for this class of matrices and for the second variant of inc. decomposition is proved in [1] .

THEOREM 8.2. The incomplete factorization is a stable process for the diagonal dominant $\overline{M}$-matrix in the following sense:
the number

$$q = \max_{i,j,r} |c_{ij}^{(r)}| / \max_{i,j} |c_{ij}|$$

is bounded from above (even $q = 1$).

In regard to the modes of storage for C, we will proceed in our case from the point of view of the Gaussian elimination and Chol. decomposition described in Sec. 2.4., i.e. we pass through the columns. We carry out the elimination for each entry one at a time. In the variant with adding to the diagonal we add the non-zeros to the diagonal element in the same column. In this case the incomplete factorization also turns out to be more efficient than SSOR decomposition. Generally, it can be said that the number of iterations on particular facets is lower in the preconditioning (on our test example approx. 3 times), however, the calculations of the projection matrix are very expensive.

In the SKY-LINE format it is best to carry out the complete factorization. While in the problems without constraints it would be redundant to perform the iterations after it, for this situation we do not have the solution yet, but we can achieve substantial acceleration of the CGM iterations. Only in this situation is the convergence faster (on test example 2-3 times) than in the case without the preconditioning. Naturally, the disadvantage is the fill-in which arises due to the elimination.

2.9 The Pre-elimination

In previous paragraphs we have shown that in the problem $(\mathcal{P}_d)$ only the contact degrees of freedom, which belong to some contact pair $a_i \in \Gamma_c$ (cf. 2.1.), are constrained in the matrix A. The number of degrees of freedom with this property is often far smaller than the total number of all degrees of freedom. By the elimination of non-constrained degrees of freedom (substructuring, see [9, 8, 22, 19]), we can reduce the number of variables in the minimized functional and therefore carry out the iterations for smaller problem.

We proceed from the problem ($\mathcal{P}_d$), i.e.

$$f(x) = \tfrac{1}{2} x^T C x - x^T d \;\rightarrow\; \min$$
$$Ax \leq 0 \;,$$

$$C = (N \times N),\; A = (M \times N) \;.$$

Suppose that nodes are renumbered so that the constrained components, the number of which is P, $M < P < N$, are placed on the last $N - P$ positions. The minimization problem is equivalent to [5] :

$$\begin{array}{ll} \text{find } x^* \in R^N,\ Ax^* \le 0, & \\ (y - x^*)^T Cx \ge (y - x^*)^T d \qquad \forall y \in R^N, Ay \le 0. & \end{array} \tag{2.9}$$

Write

$$x^* = (x_1^*, x_2^*)^T, \quad x_1^* \in R^L,\ x_2^* \in R^P, \quad L + P = N.$$

Similarly

$$y = (y_1, y_2)^T, \quad d = (d_1, d_2)^T.$$

We divide the matrices A, C into the blocks

$$A = \begin{pmatrix} A_1 & A_2 \end{pmatrix} = \begin{pmatrix} \emptyset & A_2 \end{pmatrix} \quad A_1 = (M \times N), \quad A_2 = (M \times P)$$

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \quad \begin{array}{ll} C_{11} = (L \times L), & C_{12} = (L \times P) \\ C_{22} = (P \times P), & C_{21} = C_{12}^T. \end{array}$$

Choose in (2.9) the vector y as follows

$$y = (x_1^* \pm z_1, x_2^*)^T,\ z_1 \in R^L \text{ arbitrary,}$$

It holds $Ay \le 0$.

Thus

$$z_1^T (C_{11} x_1^* + C_{12} x_2^*) = z_1^T d_1 \quad \forall z \in R^L,$$

i.e.

$$x_1^* = C_{11}^{-1} d_1 - C_{11}^{-1} C_{12} x_2^* = \overline{d}_1 - \overline{C}_{12} x_2^*, \tag{2.10}$$

where

$$\overline{C}_{12} = C_{11}^{-1} C_{12} \quad \text{and} \quad \overline{d}_1 = C_{11}^{-1} d_1.$$

Now choose, in (2.9), the vector y as follows

$$y = (x_1^*, z_2)^T, \quad z_2 \in R^P, \quad A_2 z_2 \le 0.$$

Again, $Ay \le 0$.

We obtain a new inequality

$$(z_2 - x_2^*)^T (C_{12}^T x_1^* + C_{22} x_2^*) \ \ge\ (z_2 - x_2^*)^T d_2, \qquad \forall\, z_2 \in R^P,\ A_2 z_2 \le 0,$$

and after substituting from (2.10)

$$(z_2 - x_2^*)^T (C_{12}^T \overline{d}_1 + (C_{22} - C_{12}^T \overline{C}_{12}) x_2^*) \ \ge\ (z_2 - x_2^*)^T d_2$$

thus,

$$(z_2 - x_2^*)^T \overline{C}_{22} x_2^* \ \ge\ (z_2 - x_2^*)^T \overline{d}_2, \quad \forall\, z_2 \in R^P, A_2 z_2 \le 0, \tag{2.11}$$

where

$$\overline{C}_{22} = C_{22} - C_{12}^T \overline{C}_{12} = C_{22} - C_{12}^T C_{11}^{-1} C_{12}$$

and

$$\overline{d}_2 = d_2 - C_{12}^T \overline{d}_1 = d_2 - C_{12}^T C_{11}^{-1} d_1.$$

The inequality (2.11) is in turn equivalent to the minimization

$$\begin{aligned} \overline{f}(x) = \tfrac{1}{2} x_2^T \overline{C}_{22} x_2 - x_2^T \overline{d}_2 &\rightarrow \min \\ A_2 x_2 \le 0,\ x_2 \in R^P. & \qquad\qquad (\mathcal{P}_{\overline{d}}) \end{aligned}$$

The matrix $\overline{C}_{22}$ and the vector $\overline{d}_2$, and also the matrix $\overline{C}_{12}$ and the vector $\overline{d}_1$, which we use for the calculation of x_1^* according to (2.10) already knowing the minimum x_1^*, can be obtained by the Gaussian elimination to the row L (see Sec. 2.4.) on the system $(C|d)$.

Let the matrix $L_{11}\,(L \times L)$ perform the elimination of the first L unknowns. At first, by forward elimination we obtain

$$\left(\begin{array}{cc|c} R_{11} & R_{12} & d_{1R} \\ \emptyset & \overline{C}_{22} & \overline{d}_2 \end{array}\right) = \begin{pmatrix} L_{11} & \emptyset \\ X_{21} & I \end{pmatrix} \left(\begin{array}{cc|c} C_{11} & C_{12} & d_1 \\ C_{12}^T & C_{22} & d_2 \end{array}\right),$$

where

$$X_{21} = -C_{12}^T C_{11}^{-1},\ R_{11} = L_{11} C_{11},\ R_{12} = L_{11} C_{12},\ d_{1R} = L_{11} d_1.$$

Then, by backward elimination we diagonalize R_{11}, i.e.

$$\left(\begin{array}{cc|c} I & \overline{C}_{12} & \overline{d}_1 \\ \emptyset & \overline{C}_{22} & \overline{d}_2 \end{array}\right) = \begin{pmatrix} U_{11} & \emptyset \\ \emptyset & I \end{pmatrix} \left(\begin{array}{cc|c} R_{11} & R_{12} & d_{1R} \\ \emptyset & \overline{C}_{22} & \overline{d}_2 \end{array}\right)$$

where

$$I = U_{11} R_{11} = U_{11} L_{11} C_{11} \text{ i.e. } U_{11} L_{11} = C_{11}^{-1}.$$

THEOREM 9.1. Let C be symmetric, positive definite matrix with dimension N. Then the matrix $\overline{C}_{22} = C_{22} - C_{12}^T C_{11}^{-1} C_{12}$ is also symmetric and positive definite.

Proof. As C_{11}, C_{22} and also C_{11}^{-1} are symmetric, $\overline{C}_{22}$ is also symmetric. C is positive definite, i.e.

$$0 < x^T C x = x_1^T C_{11} x_1 + x_1^T C_{12} x_2 + x_2^T C_{12}^T x_1 + x_2^T C_{22}^T x_2.$$

Through the choice $x_1 = -C_{11}^{-1} C_{12} x_2$ we obtain

$$0 \le x_2^T (C_{22} - C_{11}^{-1} C_{12}) x_2 = x_2^T \overline{C}_{22} x_2,$$

i.e. $\overline{C}_{22}$ is positive definite. □

It is obvious from this Theorem that the method from Sec.2.6. can be used for the problem $(\mathcal{P}_{\overline{d}})$. Since $\overline{C}_{22}$ and A_2 are stored in the computer memory in the same places as the original (greater) matrices, the relative adresses of entries $\overline{C}_{22}$ and A_2 in CGC differ from absolute ones which are related to the original matrices. Therefore, it is necessary to slightly modify multiplication subroutines.

If we omit the elimination part in the process and, for the same reason, the LL^T decomposition in preconditioning by using LL^T (Sec. 2.8.), we get almost equally fast methods. Due to the necessity of renumbering of contact nodes, which has to be performed after contingent renumbering in order to reduce the bandwidth, the Pre-elimination has greater memory requirements than LL^T preconditioning.

There was an attempt to perform this second renumbering implicitly, i.e. instead of Gaussian elimination to the row L, to use a more general version in which "non-contact" degrees of freedom are eliminated in the order that was created directly after the assembling or after the first renumbering (Sec. 2.3-4.). To do this within the SKY-LINE format, it was necessary to store the whole contact columns in the stiffness matrix. After the forward elimination for entries above the diagonal we perform the same process for entries below the diagonal. We adjust the whole contact columns in the backward elimination. However, in our examples the memory requirements were not lower than those for the method with explicit renumbering, not even in the cases with relatively small number of contact pairs. Moreover, the algorithm was slower because of more complicated manipulations during the calculation.

2.10 The Penalization

This method belongs to the ones which transform the problem with constraints to another problem, in which the constraints are no longer present. The principal idea consists of adding the penalization terms to the minimized functional. These terms are zero on the set determined by the constraints and outside they are boundlessly increasing, thus causing the limit solution to be inside the above defined set (The Exterior Method). The Penalization was used several times for solving various other formulations of the Contact Problem. The problem without constraints seems to be simpler, however, it will turn out that too big penalization term prevails numerically over the original functional and therefore, we are not able to obtain the exact solution, even with the use of more strict tolerances.

As the main advantage of the Penalization which, compared with previous methods, should represent the presence of the problem without constraints, we will penalise only the disretized problem i.e. $(\mathcal{P}_d)$.

Define for $\epsilon_p > 0$ the functional

$$g_{\epsilon_p}(x) = f(x) + \frac{1}{2\epsilon_p} \cdot \sum_{j=1}^{M} [(a_j, x)^+]^2 \quad ,$$

where the term $\frac{1}{2\epsilon_p} \cdot \sum_{j=1}^{M} [(a_j, x)^+]^2$ is the penalization functional.

It holds that

$$x \in K_d \;\Leftrightarrow\; (\,(a_j,x) \le 0\,\forall j = 1,\dots,M\,) \;\Leftrightarrow\; \frac{1}{2\epsilon_p}\cdot\sum_{j=1}^{M}[(a_j,x)^+]^2 = 0$$

If f is strictly convex, then $g_{\epsilon_p}(x)$ is strictly convex (since K_d and the penalization functional are convex). Thus, there exists a unique $x^*_{\epsilon_p}$

$$g_{\epsilon_p}(x^*_{\epsilon_p}) \le g_{\epsilon_p}(x) \quad \forall\, x \in R^N\,. \qquad (\mathcal{P}_{\epsilon_{d1}})$$

THEOREM 10.1. Let f be strictly convex. Let x^* be the solution of $(\mathcal{P}_d)$ and $x^*_{\epsilon_p}$ the solution of $(\mathcal{P}_{\epsilon_{d1}})$.
Then $\quad x^*_{\epsilon_p} \to x^*$ in R^N.
The proof is similar to that of Theorem 3.2., [7].

The penalization functional in $g_{\epsilon_p}(x)$ is, however, less suitable for computation, since its derivation of x_i is not in x_i linear. Thus, we introduce M new variables t_j, $t_j \ge 0$ and write the constraints as follows :

$$x \in K_d \quad \Leftrightarrow \quad \sum_{j=1}^{M}((a_j,x)+t_j)^2 = 0 \quad \text{for} \quad t_j \ge 0\,\forall j$$

We create a new functional in the form

$$h_{\epsilon_p}(x,t) = f(x) + \frac{1}{2\epsilon_p}\cdot\sum_{j=1}^{M}((a_j,x)+t_j)^2$$

and consider the following problem

$$\min_{\substack{x\in R^N\\ t_j\ge 0}} h_{\epsilon_p}(x,t) \qquad (\mathcal{P}_{\epsilon_{d2}})$$

The constraints are again in $(\mathcal{P}_{\epsilon_{d2}})$, however, their form allows us to use a very simple method for the minimization, namely the Relaxation method. In addition, the experiments have shown that in this situation this method behaves far better than the conjugate gradient method with constraints (Sec. 6).

THEOREM 10.2. The problem $(\mathcal{P}_{\epsilon_{d2}})$ has a unique solution $(x^*_{\epsilon_p}, t^*_{\epsilon_p})$, where $x^*_{\epsilon_p}$ is the solution of $(\mathcal{P}_{\epsilon_{d1}})$ and $t^*_{\epsilon_p} = (t^*_{\epsilon j})_{j=1}^M$, $\; t^*_{\epsilon j} = (a_j, x^*_{\epsilon_p})^-$.

Proof. For a given x define $t_x = (t_{xj})_{j=1}^M$, $t_{xj} = (a_j,x)^-$.
Using $t_j \ge 0$ and the relations

$$z = z^+ - z^-\,, \quad z^+z^- = 0\,, \quad (z+y)^2 = (z^- - y)^2 + (z^+)^2 + 2z^+y$$

(z^+ and z^- are the positive and negative parts of z, respectively), we get

$$h_{\epsilon_p}(x,t) \geq h_{\epsilon_p} = g_{\epsilon_p}(x) > g_{\epsilon_p}(x^*_{\epsilon_p}) \text{ for } x \neq x^*_{\epsilon_p} \text{ and } t_j \geq 0.$$

At the same time

$$h_{\epsilon_p}(x^*_{\epsilon_p}, t^*_{\epsilon_p}) = g_{\epsilon_p}(x^*_{\epsilon_p}). \qquad \square$$

Denote

$$\begin{aligned} J_i(x) = &\quad h_{\epsilon_p}(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_N, t) \qquad && 1 \leq i \leq N\,, \\ J_j(t) = &\quad h_{\epsilon_p}(x, t_1, \ldots, t_{j-1}, t, t_{j+1}, \ldots, t_N) \qquad && 1 \leq j \leq M\,. \end{aligned}$$

$$\begin{aligned} J'_i(x_i) &= \sum_{\substack{k=1 \\ k \neq i}}^{N} c_{ik} x_k - d_i + \frac{1}{\epsilon_p} \cdot \sum_{j=1}^{M} a_{ji} \left[\sum_{\substack{l=1 \\ l \neq i}}^{N} a_{jl} x_l + t_j \right] + \\ & \left(\frac{1}{\epsilon_p} \cdot \sum_{j=1}^{M} a^2_{ji} + c_{ii} \right) x_i \qquad (2.12) \\ J'_j(t_j) &= \frac{1}{\epsilon_p} (x^t a_j + t_j)\,, \quad t_j \geq 0\,. \qquad (2.13) \end{aligned}$$

The Relaxation method is based on the following iterations [7]

```
$k = 0$
$x_0, t_0$   initial guess (e.g. $x_0 = 0$, $t_0 = 0$)
$DO\ WHILE$ ( $(K < MAXIT).AND.(ERR.GT.\epsilon)$ )

for $i = 1, \ldots, N$

1:
find $x_i^{k+1}$
    $h_{\epsilon_p}(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^{k+1}, x_{i+1}^k, \ldots, x_N^k, t^k) \leq$
            $h_{\epsilon_p}(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x, x_{i+1}^k, \ldots, x_N^k, t^k)$      $\forall\, x \in R$

2:
the calculation of $t^{k+1}$

the calculation of $ERR$ (Sec. 5)
$k = k + 1$
$ENDDO$
```

REMARK 10.1. We perform the step 1 using the equality $J_i'(x_i) = 0$ (see (2.12)). For f being strictly convex the matrix C is positive definite, i.e.

$$\frac{1}{\epsilon_p}\sum_{j=1}^{M} a_{ji}^2 + c_{ii} \geq c_{ii} > 0\,.$$

Step 2: $t_j^{k+1} = [(x^{k+1})^T a_j]^-$ (see (2.13)).

REMARK 10.2. Similarly to SSOR, the relaxation parameter ω may be introduced $(0 < \omega < 2)$.

2.11 The Uzawa saddle point method

Further possibility of transforming the constrained problem to the sequence of unconstrained problems consists in the transformation of the original problem to the saddle point problem. This transformation will be fully employed when considering the friction in the model.

At first, we note the continuous problem, for the Lagrange multipliers which appear here have a concrete meaning. We will clarify it through the additional assumptions on the problem $(\mathcal{P})$: zero Dirichlet boundary condition on Γ_u, the boundary of the region sufficiently smooth. Now $\tau_i \in L^2(\Gamma_c)$ [21]. Let

$$\Lambda = \{\mu \in L^2(\Gamma_c) | \mu \geq 0 \ \text{ a.e. on } \Gamma_c\}$$

Define

$$\Psi(\mathbf{v},\mu) = \int_{\Gamma_c^{kl}} \mu(v_n^k - v_n^l)ds\,, \quad \mathbf{v} \in V,\, \mu \in \Lambda\,.$$

It holds

$$\begin{aligned} \mathbf{v} \in K \quad &\Leftrightarrow \quad \Psi(\mathbf{v},\mu) \leq 0 \quad \forall \mu \in \Lambda\,, \\ \sup_{\Lambda} \Psi(\mathbf{v},\mu) \quad &= \quad \begin{cases} 0 & \mathbf{v} \in K \\ +\infty & \mathbf{v} \notin K \end{cases} \end{aligned}$$

Therefore, we can write the original (primary) problem as

$$\inf_{\mathbf{v}\in V} \mathcal{L}(\mathbf{v}) = \inf_{\mathbf{v}\in V} \sup_{\mu\in\Lambda} (\mathcal{L}(\mathbf{v}) + \Psi(\mathbf{v},\mu)\,) = \inf_{\mathbf{v}\in V} \sup_{\mu\in\Lambda} \mathcal{H}(\mathbf{v},\mu)$$

Through these means, the problem is transformed to seeking the saddle point of the Lagrangian

$$\mathcal{H}(\mathbf{v},\mu) = \mathcal{L}(\mathbf{v}) + \Psi(\mathbf{v},\mu)\,.$$

THEOREM 11.1. ([8]) Let the saddle point of $\mathcal{H}(\mathbf{v},\mu)$ exist. Then its first component solves the problem $(\mathcal{P})$.

The problem $\sup_{\mu\in\Lambda} \inf_{\mathbf{v}\in V} \mathcal{H}(\mathbf{v},\mu)$ is called dual.

THEOREM 11.2. The pair $(\mathbf{u},\lambda)$ is the saddle point iff

$$\sup_{\mu\in\Lambda}\ \inf_{\mathbf{v}\in V}\ \mathcal{H}(\mathbf{v},\mu) = \inf_{\mathbf{v}\in V}\ \sup_{\mu\in\Lambda} \mathcal{H}(\mathbf{v},\mu) = \mathcal{H}(\mathbf{u},\lambda)$$

and the corresponding extremes are attained in $(\mathbf{u},\mu)$.
The proof follows from [21].

Consider the inner part of the dual problem only, i.e.

$$\begin{aligned}\inf_{\mathbf{v}\in V} \mathcal{H}(\mathbf{v},\mu) \quad = \quad & \inf_{\mathbf{v}\in V}\{\frac{1}{2}\int_\Omega c_{ijkm}e_{ij}(\mathbf{v})e_{km}(\mathbf{v})d\mathbf{x} - \int_\Omega F_i v_i d\mathbf{x} - \\ & \int_{\Gamma_\tau} T_i v_i ds + \int_{\cup\Gamma_c^{kl}} \mu(v_n^k - v_n^l)ds\}\,, \quad \mu \text{ fixed}\ .\end{aligned}$$

This problem represents the elasticity problem where on the contact boundary the surface tension $\tau = (\tau_n,\tau_t) = (-\mu,0)$ is prescribed.
The saddle point of $\mathcal{H}$ is then $(\mathbf{u},-\tau_n(\mathbf{u}))$ where $\mathbf{u}$ is the solution of $(\mathcal{P})$, $-\tau_n(\mathbf{u})$ describes the corresponding surface loads on Γ_c.

We will desribe the Uzawa algorithm in a more general form.
Let V, L be the Hilbert spaces, $K \subseteq V$, $\Lambda \subseteq L$ non-empty, convex, closed subsets. At the same time we suppose that
either Λ is convex hull with the vertex in $\emptyset_L$ and $K = V$,
or $\quad\Lambda$ is bounded subset of L.
Let $\mathcal{L}$: $V \to R$, Φ: $V \to L$, linear, continuous,
P denote the projection $\quad L \to \Lambda \quad (\|P\mu - \mu\|_L = \min_{\lambda\in\Lambda} \|\lambda - \mu\|_L)$,
and the Lagrangian, whose saddle point $(u,\mu) \in K\times\Lambda$ we seek, have the form

$$\mathcal{H} : V\times L \to R\,, \quad \mathcal{H}(v,\mu) = \mathcal{L}(v) + (\mu,\Phi(v))_L\,.$$

The Uzawa algorithm is given by the following description :

$$\begin{aligned} & \lambda^0 \in \Lambda,\ \text{arbitrary} && (2.14)\\ & \text{Knowing } \lambda^N \in \Lambda, && \\ & \text{we seek } u^N \in K && \\ \mathcal{L}(u^N) + (\lambda^N,\Phi(u^N))_L \quad &= \quad \min_{v\in K}\{\mathcal{L}(v) + (\lambda^N,\Phi(v))_L\} && (2.15)\\ \lambda^{N+1} \quad &= \quad P[\lambda^N + \rho\Phi(u^N)] && (2.16)\end{aligned}$$

The following Theorem holds for the convergence of this algorithm (see e.g. [21],[7],[5]).

THEOREM 11.3. Let $\mathcal{L}(u)$ have the strictly monotonne differential, i.e.

$$D\mathcal{L}(u+h,h) - D\mathcal{L}(u,h) \geq m\|h\|^2 , \quad \forall\, h \in V , \tag{2.17}$$

Let

$$\|\Phi(u) - \Phi(v)\|_L \leq c\|u-v\| \quad \forall\, u,v \in V, \tag{2.18}$$

and let ρ fulfill

$$2m\rho - c^2\rho^2 \geq \beta > 0 \, . \tag{2.19}$$

Assume that the saddle point $(u,\lambda) \in K \times \Lambda$ of the Lagrangian $\mathcal{H}$ exists.
Then the process (2.14)-(2.16) converges in the sense
that $u^N \rightarrow u$ strongly in V.

Moreover, if the saddle point is unique, then $\lambda^N \rightharpoonup \lambda$ weakly in L.

Similarly to the penalization, the problem without constraints should be one of the greatest advantages of the saddle point formulation. Moreover, we require the contact condition to be fulfilled only in the discrete points. Therefore, we introduce the Lagrangian only for the problem $(\mathcal{P}_d)$.

Here,

$$V = K = R^N ,\, u^N \equiv x^N ,\, L = R^M ,\, \Lambda = R^M_+ \equiv \{x \in R^M \,|\, x_i \geq 0 \quad 1 \leq i \leq M\} \, ,$$

the functional $\mathcal{L}(u)$ is represented by $f(x)$, $\Phi(u)$ by the vector Ax
(i.e. $\Psi(u,\lambda) \approx \lambda^T Ax$).
The projection P has the form $P\lambda = \underline{\lambda}$, where $\underline{\lambda} = (\lambda_1^+, \ldots, \lambda_M^+)$.
Thus, we seek the saddle point of $\mathcal{H}(x,\lambda)$,

$$\mathcal{H}(x,\lambda) = f(x) + \lambda^T Ax = \tfrac{1}{2}x^T Cx - x^T d + \lambda^T Ax, \quad x \in R^N ,\, \lambda \in R^M \tag{2.20}$$

If C is positive definite, then all the assumptions of the previous Theorem are fulfilled as we have a finite dimensional problem. The existence and uniqueness of the saddle point is also ensured [21],[5].

The minimization of the functional in (2.15)

$$f_\lambda(x) = \tfrac{1}{2}x^T Cx - x^T(d - A^T\lambda)$$

can be accomplished by a standard conjugate gradient method.

REMARK 11.1. The optimal value of ρ can be theoretically determined, e.g. for the equality problem, we have

$$\rho_{opt} = \frac{2}{(\lambda_{min} + \lambda_{max})} \, ,$$

where $\lambda_{min}, \lambda_{max}$ are the extremal eigenvalues of the matrix $(C^{-1}A_{I^0}^T A_{I^0})$ (see [16] and cf. (2.19); the matrix A_{I^0} is defined in Sec. 6.). However, the calculation of the eigenvalues would be at least as expensive as the whole problem. Therefore, ρ is to be estimated during the computation in a similar way as ϵ_p is in penalization.

REMARK 11.2. We also obtain the values of the multipliers in the CGC algorithm. The criterion for terminating CGC is

$$f'(x^*) + A_J^T\lambda^* = 0 \quad \text{and} \quad \lambda_i^* \geq 0 \quad i \in J \cap I^- \text{ , i.e.}$$

$$Cx^* - d + A^T\underline{\lambda}^* = 0, \quad \underline{\lambda}_i^* = \lambda_i \;\; i \in J, \quad \underline{\lambda}_i^* = 0 \;\; i \in I - J.$$

At the same time $Ax^* \leq 0$. Furthermore,

$$\lambda_i^* \neq 0 \;\Rightarrow\; i \in J \;\Rightarrow\; (Ax^*)_i = 0\,.$$

By virtue of Kuhn-Tucker Theorem (see e.g. [21]) the pair (x^*, λ^*) is the saddle point of the Lagrangian $\mathcal{H}(x, \lambda)$.

2.12 The minimization of the dual functional

The Uzawa method from previous section is relatively slow for greater problems. Therefore, it is reasonable to examine yet another, faster saddle point algorithms.

The conditions for saddle point (x^*, λ^*) of (2.20) are (e.g. [8]):

$$\begin{aligned} Cx^* - d + A^T\lambda^* &= 0 && (2.21)\\ (x^*)^T A^T(\tau - \lambda^*) &\geq 0 \qquad \forall\, \tau \in R_+^M\,. && (2.22) \end{aligned}$$

For models which lead to the positive definite matrix C (c.f. Rem. 2.1.), we may calculate x^* from (2.21) and substitute it into $\mathcal{H}(x, \lambda)$. We get

$$\inf_{x \in R^N} \mathcal{H}(x, \lambda) = \tfrac{1}{2}\lambda^T H\lambda + \lambda^T h + k\,,$$

where

$$H = AC^{-1}A^T, \quad h = AC^{-1}d, \quad \text{and } k = \tfrac{1}{2}d^T C^{-1}d\,.$$

(Up to a constant term, we obtain the same by substituting x^* into (2.22).)

DEFINITION 12.1. By dual functional we call the functional

$$\mathcal{J}'(\lambda) = -\inf_{x \in R^N} \mathcal{H}(x, \lambda)\,.$$

Let diagonal $M \times M$ matrix B, $B = diag(-1, \ldots, -1)$, represent the condition $\lambda \in R_+^M$. Thus, we arrive at problem $(\mathcal{P}_{dd})$:

$$\begin{gathered} \min \mathcal{J}'(\lambda) \\ \text{with constraints} \quad B\lambda \leq 0 \end{gathered}$$

THEOREM 12.1. Let $x^TCx > 0$ for $x \neq 0$ and let the rows A be linearly independent. Then $H = AC^{-1}A^T$ is positive definite.
The proof is obvious, as $A^Ty = 0 \Leftrightarrow y = \emptyset$ and $z^TC^{-1}z > 0$ for $z \neq 0$.

As the matrix B has linearly independent rows, the method of Sec. 6. can be used for solving $(\mathcal{P}_{dd})$. It is obvious that the calculation of the projection can now be simplified. Using the CGM we avoid the fill-in which can arise from the decomposition of C, and this is, in our case, a more essential criterion than a slow down.

The calculation follows the algorithm therein presented. However, we do not store H. Thus, every multiplying of Hz consists of solving the system with the matrix C. We may use the standard conjugate gradient method for the solution of this "inner" problem. Note that during the "outer" iterations (the problem of dimension M) it is necessary to choose more strict tolerance than in Sec. 6. (as much as several orders). The removal of more indices from the active set is also convenient here.

2.13 The Active set method

The idea of this method is similar to the method of Sec. 6. ([6]). We are succesively searching for those saddle points of Lagrangians (2.20) which contain only the equality constraints. At the same time we assume C to be positive definite. Using the notation similar to the one of Sec. 6., we may express the scheme of the method as follows:

```
$k = 0$
$x^0$   init. guess
$J \subset I^0 \cup I^-$   the corresponding set of active constraints

$DO\ WHILE\ (k < MAXIT)$
solve EP (2.26 below)

$IF\ \|\delta\| \approx 0\ THEN$
    $\bar{j} := \min\{\, i \in I^- \cap J |\, \lambda_i = \min_{j \in I^- \cap J} \lambda_j\}$
    $IF\ (\lambda_{\bar{j}} \geq 0)\ THEN$
        $x^* = x^k$        { the solution }
        $GOTO$ 1
    $ELSE$
        $J := J - \{j\}$
    $ENDIF$
$ELSE$
    $\alpha = \min(1, \min_{\substack{i:\ i \notin J \\ a_i^T\delta < 0}} \frac{-a_i^T x^k}{a_i^T \delta})$
    $x^{k+1} = x^k + \alpha\delta$
    correction of $J$
```

```
$ENDIF$
$k = k + 1$
$ENDDO$
{ maximum number of iterations reached }
1 :
$END$
```

Let us now study the equality problem - EP. The Lagrangian has the form

$$\mathcal{H}_J(x, \lambda) = \tfrac{1}{2} x^T C x - x^T d + \lambda^T A_J x \,. \tag{2.23}$$

The conditions for the saddle point are

$$0 = \quad \nabla_x \mathcal{H}(x^*, \lambda^*) = \quad C x^* - d + (\lambda^*)^T A_J \tag{2.24}$$
$$0 = \quad \nabla_\lambda \mathcal{H}(x^*, \lambda^*) = \quad A_J x^* \tag{2.25}$$

Let us introduce in $(k+1)$-th iteration the substitution $\delta = x^* - x^k$. Moreover, let

$$B_J = \begin{pmatrix} C & A_J^T \\ A_J & \emptyset \end{pmatrix} , \qquad y = \begin{pmatrix} \delta \\ \lambda \end{pmatrix} , \qquad f = \begin{pmatrix} d - C x^k \\ \emptyset \end{pmatrix} .$$

Therefore, we can write (2.24,2.25) in matrix form

$$B_J y = f \,, \tag{2.26}$$

where B_J is of type $(L \times L)$, y,f $(L \times 1)$, $L = N + M(J)$, $M(J)$ num. of active constraints. As C is positive definite and the rows of A_J are linearly independent, the matrix B_J is regular [6].

The Gaussian elimination algorithm used in Sec. 9. and the possibility of node renumbering after the mesh generation give us one way to solve (2.26). By these means, we obtain a fast method comparable with the complete LL^T decomposition preconditioning (Sec. 9.). However, we have to keep in mind the fill-in in B_J which arises from the elimination as well as the necessity to store the stiffness matrix C. We can reduce the bandwidth of B_J by inserting the component λ_l immediately after x_i, where $i = \max \{ i |\, i = 1, \ldots, N;\ a_{li} \neq 0 \}$.

2.14 The conjugate gradient method with hyperbolic pairs

We describe one iterative method for solving (2.26). As it is well-known, by using the standard conjugate gradient method, we obtain the following algorithm

$$y^0 \ \ldots \ \text{initial guess}$$

$$
\begin{aligned}
p^1 = r^1 &= f - B_J y^0 \\
\text{For } k &= 1, \ldots, N \\
\alpha^k &= (r^k, p^k)/(p^k, B_J p^k) && (2.27) \\
y^{k+1} &= y^k + \alpha^k p^k && (2.28) \\
r^{k+1} &= r^k - \alpha^k B_J p^k && (2.29) \\
\beta^k &= (r^{k+1}, B_J p^k)/(p^k, B_J p^k) && (2.30) \\
p^{k+1} &= r^{k+1} - \beta^k p^k && (2.31)
\end{aligned}
$$

The matrix B_J is regular and symmetric but not positive definite. Thus, it may occur $(p^k, B_J p^k) = 0$ for some $p^k \neq 0$. This difficulty can be rectified by transforming (2.26) to $B_J^2 y = B_J f$ or by using the conjugate gradient method with orthogonalization in $(B_J^2 y, y)$ inner product. We present here a modification of the standard method, suggested in [15], which turned out to be the best.

DEFINITION 14.1. A nonzero vector $y \in R^L$ is said to be singular if $(y, B_J y) = 0$. A pair of vectors $x, y \in R^L$ is said to be a hyperbolic pair if x and y are both singular and $(x, B_J y) \neq 0$.

Then, we may express the algorithm as follows:

Case I : p^k is not singular - use (2.27)-(2.31)

Case II : p^k is singular - use the following:

$$p^{k+1} = B_J p^k - \frac{(B_J p^k, B_J^2 p^k)}{2(B_J p^k, B_J p^k)} p^k \tag{2.32}$$

$$\alpha^k = \frac{(r^k, p^{k+1})}{(p^k, B_J p^{k+1})} \tag{2.33}$$

$$x^{k+1} = x^k + \alpha^k p^k \tag{2.34}$$

$$\alpha^{k+1} = \frac{(r^k, p^k)}{(p^k, B_J p^{k+1})} \tag{2.35}$$

$$x^{k+2} = x^{k+1} + \alpha^{k+1} p^{k+1} \tag{2.36}$$

$$r^{k+2} = r^k - \alpha^k B_J p^k - \alpha^{k+1} B_J p^{k+1} \tag{2.37}$$

$$p^{k+2} = r^{k+2} - \frac{(r^{k+2}, B_J p^{k+1})}{(p^k, B_J p^{k+1})} p^k \tag{2.38}$$

REMARK 14.1. In the Case II, p^k, p^{k+1} is a hyperbolic pair.

THEOREM 14.1. The algorithm defined above converges to the solution of (2.26) in L steps or less.
Proof. See [15].

REMARK 14.2. A direction vector p^k is treated as singular if

$$\left| \frac{(p^k, B_J p^k)}{(p^k, p^k)} \right| \leq \epsilon; \qquad \text{we take } \epsilon = 10^{-4}\,.$$

REMARK 14.3. An obvious advantage of the iterative method is again the possibility of using the SPARSE format for the storage of B_J.

Chapter 3

Numerical tests

3.1 First test example

The comparison test of all above methods was carried out on a personal computer with MS-FORTRAN 5.0 compiler, for the model, which simulates a contact between three bodies (Figs. 3.1-3.4).

These bodies together occupy the rectangle region $1000 \times 800[m]$. The distribution of surface tension $\mathbf{P}$ is prescribed on the top and bottom side. The displacements $\mathbf{u}_{0L}$ and $\mathbf{u}_{0R}$ are prescribed on the left and right side. The gravity g and density ρ form the body forces $F_2 = -\rho g$. The first body is enclosed by lines $1 - \ldots - 14 - 1$, the second by $15 - \ldots - 25 - 15$ and the third by $26 - \ldots - 39 - 26$. The values for boundary conditions were taken as follows:

$$u_{0L1} = 0.2,\ u_{0L2} = 0.0,\ u_{0R1} = -0.2,\ u_{0R2} = 0.0\,[m],$$
$$P_1 = 0.0,\ P_2 = -0.8d{+}08[Nm^{-2}].$$

Furthermore, $g = 0.1d{+}02[ms^{-2}]$, $\rho = 0.7d{+}04[kgm^{-3}]$. The elastic parameters were: $E = 0.1d{+}12[Nm^{-2}]$, $\mu = 0.3$. We assume the linear Hooke's law to be valid.

After the triangulation there are 128 nodes, 182 elements, 218 degrees of freedom and 18 constraints. The number of stored entries of the stiffness matrix in our model is: 1347 for SPARSE format, 3003 for SKY-LINE (without the renumbering ≈ 10000), and ≈ 14000 for SKY-LINE with the Pre-elimination (already after the first renumbering).

The speed of the projection gradient methods (Sec.6-9.,12.) as well as the Active set method (Sec.13-14.) can be affected by suitable initial guess and particularly also by the number of removed indices (as much as several times). Some loss of accuracy can be expected [13] in non-convex corners. We listed the displacements (in $[m]$) in nodes 34 and 77 (cf. Fig.3.1). The TIME is in seconds.

The first table compares the elementary CG Method in both of the formats and the Pre-elimination. The Reverse Cuthill-McKee renumbering which also increased the speed of the method is done for SKY-LINE. For the Pre-elimination, second value is the computational time excluding the elimination part.

The tables (3.2),(3.3) show the influence of preconditioning on the CGC method in both of the formats. In SKY-LINE format, we have got the fastest tested method by "preconditioning" using the complete decomposition (LL^T). It is necessary to store this

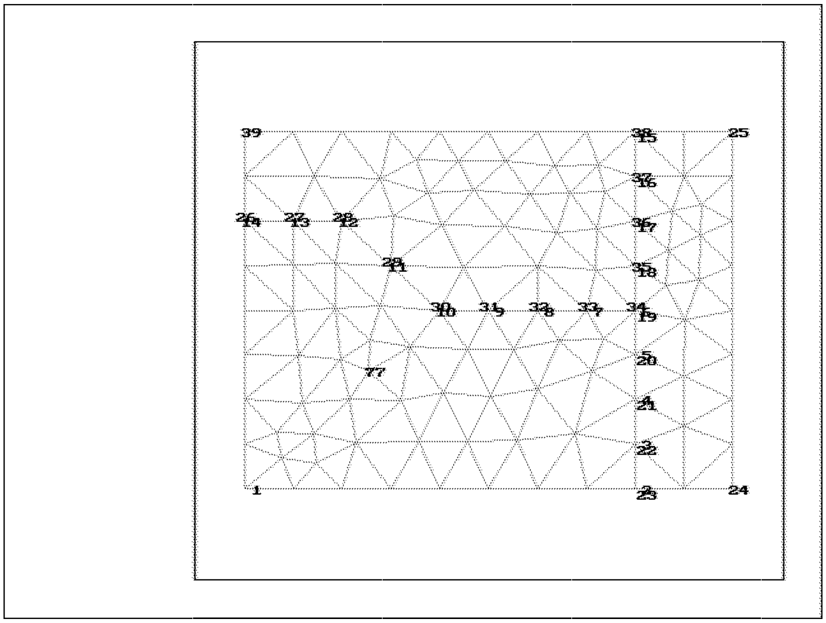

Figure 3.1:

Table 3.1: CGC in both formats

	$u_1(34)$	$u_2(34)$	$u_1(77)$	$u_2(77)$	TIME
SKY-LINE	-1.65	-5.56	-1.73D-2	-1.91	12
SPARSE	-1.65	-5.56	-1.73D-2	-1.91	15
PRE-ELIM	-1.65	-5.56	-1.73D-2	-1.91	11/3

Table 3.2: The preconditioning - SKY-LINE

SKY-LINE	$DIAG$	SOR	ILL^T	ILL^TD	LL^T
TIME	20	36	23	22	5/3

Table 3.3: The preconditioning - SPARSE

SPARSE	SOR	ILL^T	ILL^TD
TIME	35	29	29

factor. However, in most cases the bandwidth of the stiffness matrix is proportional to the number of contact degrees of freedom (Rem. 2.2.3). Therefore, considering the possibility of renumbering as well, there are smaller memory requirements than for the Pre-elimination. In the case of this preconditioning, we actually calculate with the projection of unity matrix. Often only one iteration is performed on each facet. Similarly to the Pre-elimination, second value is the computational time which excludes the LL^T decomposition.

Least efficient turned out to be the SOR preconditioning which is almost independent on ω (we take $\omega = 1$). The incomplete factorization (ILL^T) and i.f. with adding to the diagonal (ILL^TD) were faster but still did not reach the speed of the elementary method (without precond.).

The preconditioning for SPARSE format had similar behaviour. In this case, it was not convenient to create a complete factorization.

Greater efficiency of classical preconditioners may be supposed when there is a greater number of elements in the model, due to the increase of the condition number of stiffness matrix [1]. Numerical values are the same as in Tab.3.1 and are not listed for the sake of greater amount of variants.

In the penalization (Tab.3.4), it is necessary to choose the parameter ϵ_p correctly. The values $\epsilon_p = 1.d - 11$ and $1d. - 12$ when the penalization term was 1-2 orders greater than the entries in the stiffness matrix turned out to be the most convenient. The correct estimate of ϵ_p is probably the greatest drawback of this method. We have chosen the relaxation parameter $\omega = 1.5$.

Almost the same holds for the Uzawa method. A parameter ρ was succesively increased by order till the value when the oscillations occured. The most optimal values are approximately one order under the oscillations. The properties of this method did not improve the introduction of the penalization term into the inner iterations (Augmented Lagrangian, see [16] for equality problem). In our case we have taken the penalization term from Sec. 2.10. Regarding the speed of the Uzawa algorithm, we have also tried to test the method for dual functional, which gives more acceptable results.

By these means, we have placed the information about the behaviour of the methods onto a relatively simple example. It can be used when considering the friction in a model or for contingent solving of more complex physical problems.

Table 3.4:

	$u_1(34)$	$u_2(34)$	$u_1(77)$	$u_2(77)$	TIME
PEN 1.D-11	-1.79	-5.84	-1.48D-2	-1.92	174
PEN 1.D-12	-1.67	-5.57	-1.80D-2	-1.91	712
UZAWA	-1.68	-5.57	-2.06D-2	-1.92	395
DUAL	-1.65	-5.56	-1.73D-2	-1.91	115

Table 3.5:

	$u_1(34)$	$u_2(34)$	$u_1(77)$	$u_2(77)$	TIME
ASM-E	-1.65	-5.56	-1.73D-2	-1.91	8
CGH	-1.65	-5.53	-1.82D-2	-1.91	25

The last table compares the variants to the Active set method. The elimination version (ASM-E) is almost as fast as the LL^T preconditioning. We have to store the matrix B_J (2.26) which is to be eliminated (in this model up to 4151 entries). In our case the elimination represents only $O(L^2)$ operations. Even more optimal should be the creation of corresponding factors [6],[11]. However, we still do not avoid a fill-in for the constraint matrix A_J. We can use the SPARSE format in the iterative Conjugate gradient method with hyperbolic pairs (CGH), .

3.2 Three cantilever bodies

In the case of our computational possibilities (personal computer, MS FORTRAN 5.0 compiler), the iterative conjugate gradient method with constraints is more optimal even though it is slower than the elimination method. Moreover, we may also consider the semicoercive case (Sec. 3., in what follows). The SPARSE format allows us to solve problems with more than 4000 degrees of freedom (we suppose, that number of constraints is far lower). There is a possibility to increase the memory capabilities on the PC by using a different compiler which also uses a memory above 640K (e.g. SF FORTRAN).

To ilustrate good behaviour of a mathematical formulation of the problem, we have created several other models. For each model, we display these values: NV-number of nodes, NEL-n. of elements, NEQ-n. of degrees of freedom, NCP-n. of constraints, LIC-n. of stored entries in the stiffness matrix, LJA-n. of stored entries in the constraint matrix, TIME- solution time for the CGC method in seconds. This time depends not only on the size of the problem but also on the geometry and boundary conditions.

The first model containing 3 cantilevers is depicted in Fig. 3.5 [12]. A surface

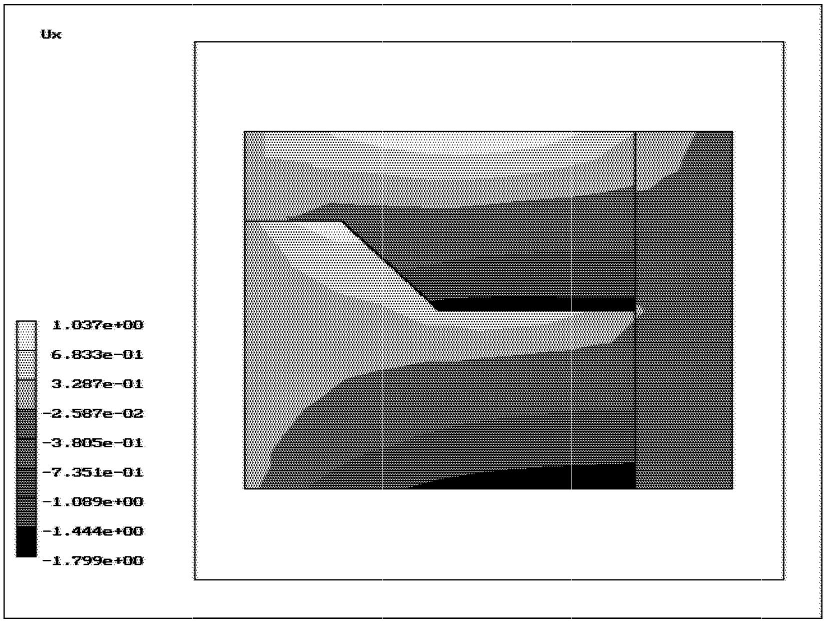

Figure 3.2:

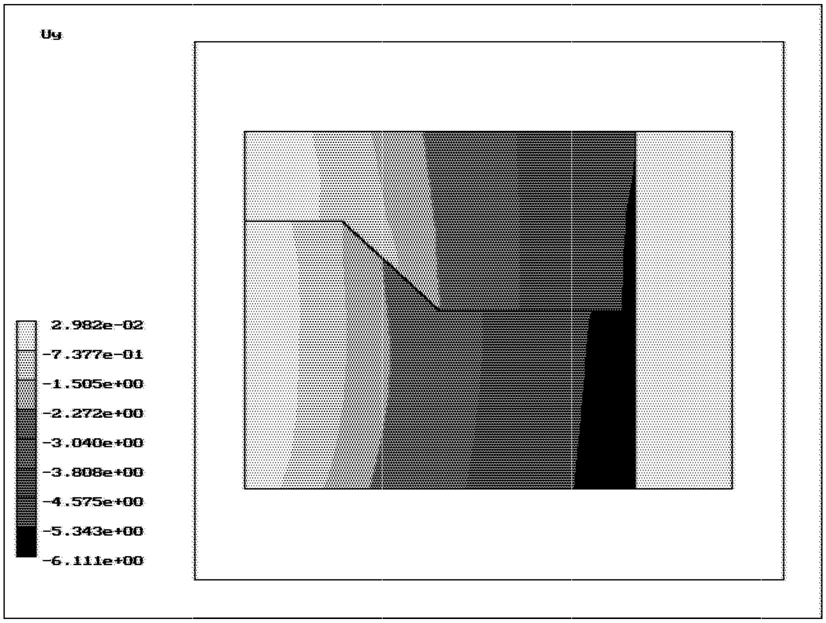

Figure 3.3:

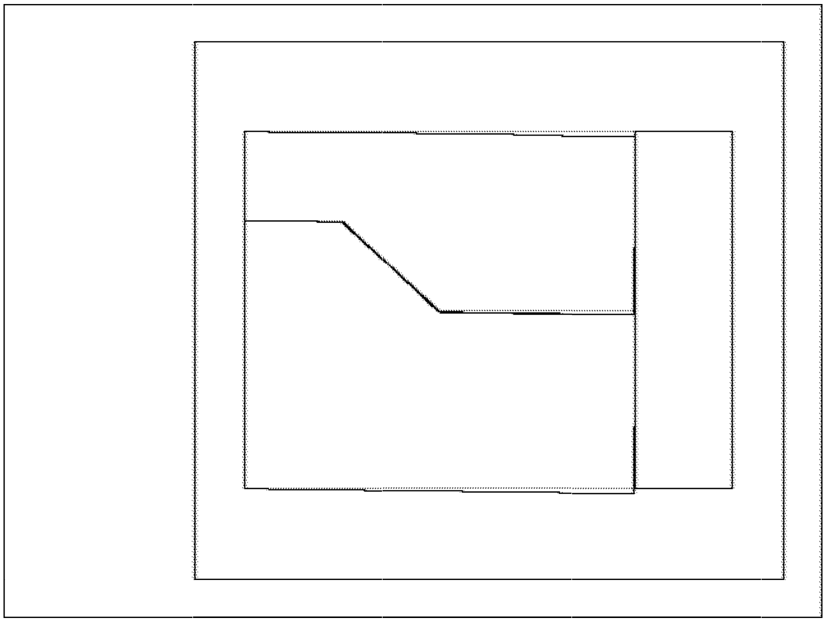

Figure 3.4:

Table 3.6:

NV	NEL	NEQ	NCP	LIC	LJA	$TIME$
424	702	812	17	5514	68	130

pressure $P_2 = -0.9d{+}09[Nm^{-2}]$ is prescribed on top of the highest body. This body and the lowest body are fixed on the left while the middle body is fixed on the right. The material properties are $E = 0.1d{+}12[Nm^{-2}]$, $\mu = 0.3$.

This is the example containing more than two bodies where at most two bodies stick in one point.

Here we have also tried to test the SF FORTRAN compiler with the solution time 26 seconds. However, the assembling of the stiffness matrix was slower in comparison to the MS FORTRAN.

3.3 A simple model of the human hip joint

In this section, a model of the human hip joint is analysed (Fig. 3.6). This analysis was done in co-operation with the Orthopaedic Clinic of the 3rd Faculty of Medicine [3] and may be useful for modelling a human hip joint replacement after surgical reconstruction of a dysplastic acetabulum.

The geometry was taken from an X-ray photograph. The weight of the human body is distributed along the boundary lines $91 - \ldots - 106 - \ldots - 109$ with the value

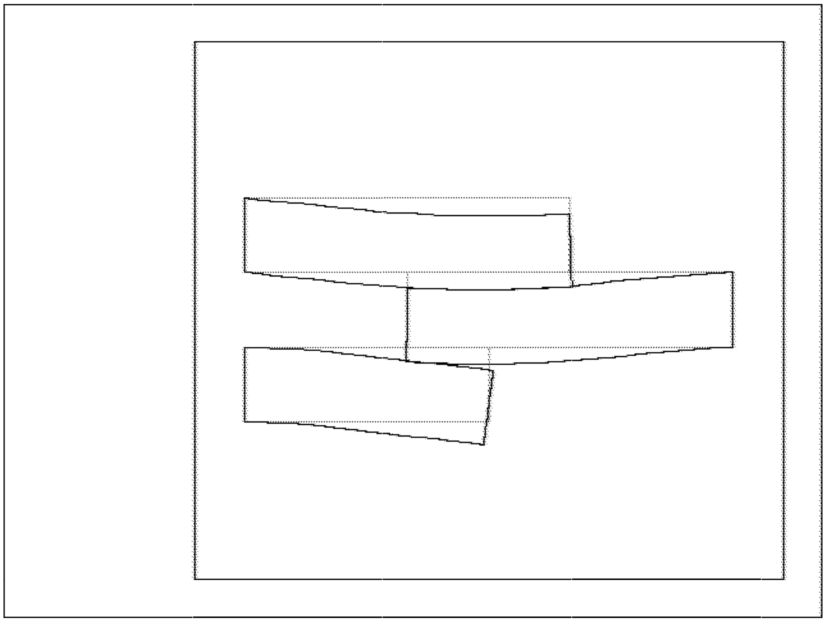

Figure 3.5:

Table 3.7:

$MODEL$	NV	NEL	NEQ	NCP	LIC	LJA	$TIME$
1	180	234	350	8	2131	32	41
2	204	268	398	15	2435	60	43
3	690	1094	1359	15	9064	60	182

$P_1 = 0.0$, $P_2 = -0.5d{+}05[Nm^{-2}]$. Point force $F_1 = -0.607d{+}04$, $F_2 = -0.345d{+}04[N]$, caused by the abductors acts at vertex 116. The oposite force acts at vertex 2.

The bottom of the structure is fixed, i.e. $\mathbf{u} \equiv 0$ along boundary lines $28 - 29$ and $44 - 45$. We prescribed the condition $u_n = 0$ along line $90 - 91$. This means that we have a semicoercive case now. The contact boundary is located between pairs $61, 76$ and $68, 69$.

The elastic parameters were taken as $E = 0.1d{+}11[Nm^{-2}]$, $\mu = 0.295$ [2]. We assume the linear Hooke's law to be valid and that the type of deformation is a plane stress.

We created three triangulations: (1) coarse (Fig. 3.6), (2) finer only on the contact boundary (Fig. 3.7) and (3) finer in the whole structure (Figs 3.8 - 3.11). Our computations are summarized in Tab.3.7.

In Fig. 3.7 we demonstrate the resultant displacements with the scale factor 30. The distributions of stresses for the finest triangulation are depicted in Figs. 3.8 - 3.11.

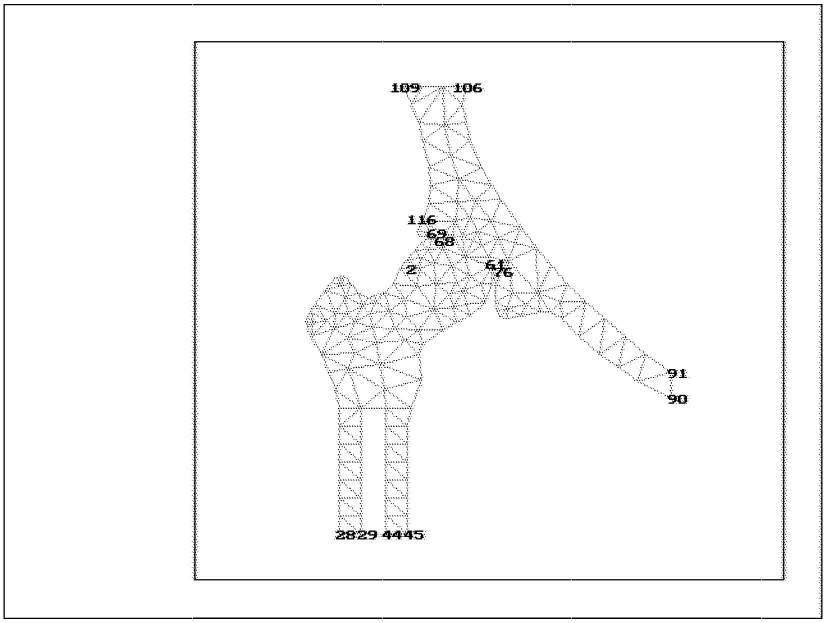

Figure 3.6:

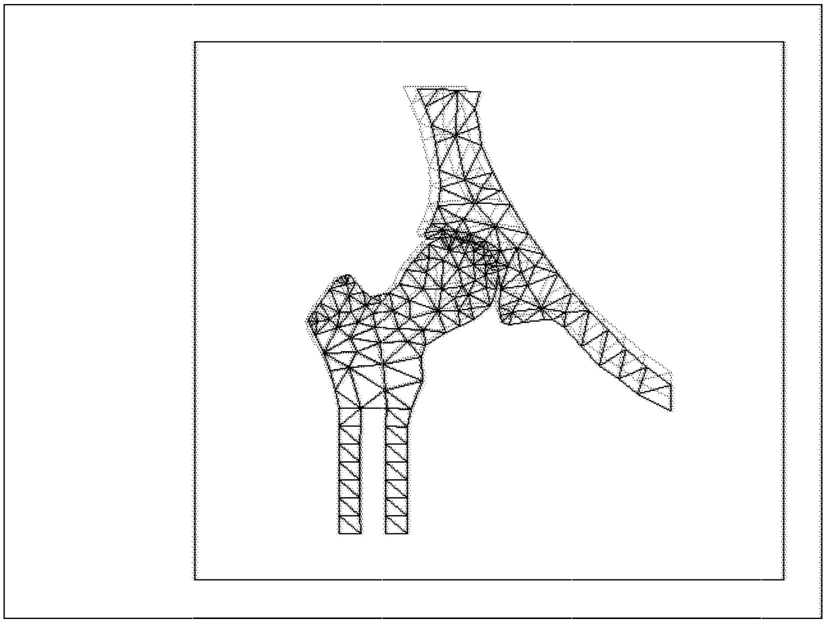

Figure 3.7:

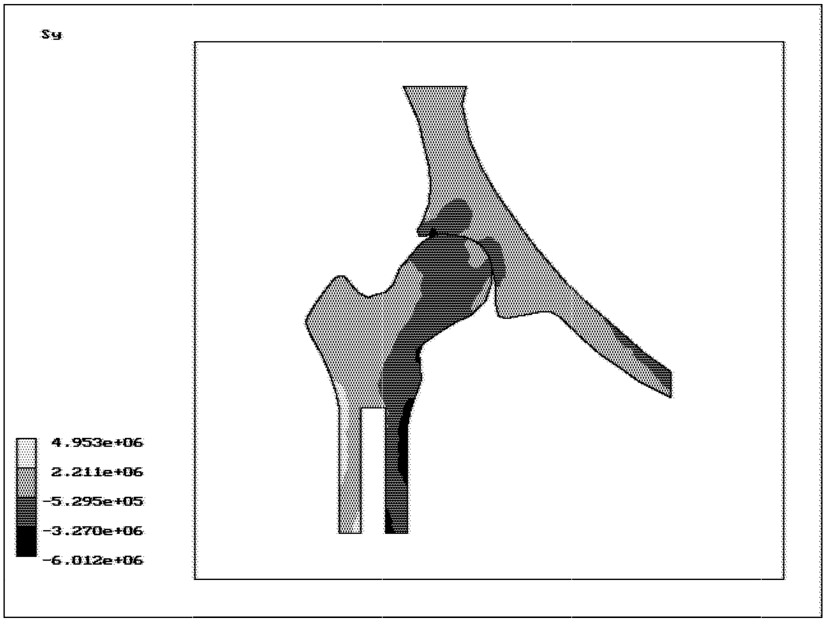

Figure 3.8:

For the stress equivalent we have used

$$\tau_e = \sqrt{\tau_{11}^2 + \tau_{22}^2 - \tau_{11}\tau_{22} + 3\tau_{12}^2}.$$

We have compared our results with [2]. Naturally, small differences exist. They can be caused by different input data. Only the upper part of the structure is considered in [2]. There are no contact conditions, only the linear elastic model is calculated. The top line is fixed and the weight of the human body is transformed into the reaction forces, acting in the joint.

3.4 A more complicated geodynamical model

This model relates to the one in Sec.1. It simulates the motion of litospheric plates in the Earth and can be regarded as a quasistatic study of a dynamic tectonic plate model which mathematically describes the collision zones in the sense of new global tectonics [18]-[19].

The whole structure occupies approximately the region $0.7d + 05 \times 0.6d + 05$ $[m^2]$ (Fig. 3.12), and again contains three bodies in contact $(1-2-23-21-18-24-1)$, $(24-18-21-39-40-24)$ and $(39-21-23-68-39)$. There are 19 subregions in these 3 bodies. Each of them has different values of $E[Nm^{-2}]$, μ, $\rho[kgm^{-3}]$, varying from $E = 0.27d{+}11$, $\mu = 0.23$ and $\rho = 0.24d{+}04$, to $E = 0.18d{+}11$, $\mu = 0.33$ and $\rho = 0.34d{+}04$.

The part $24-1-2-23$ of the boundary is fixed. Along the lines $24-40$ and $23-68$ we have prescribed the Dirichlet boundary condition $\mathbf{u}_{0L}$, $\mathbf{u}_{0R}$ which express

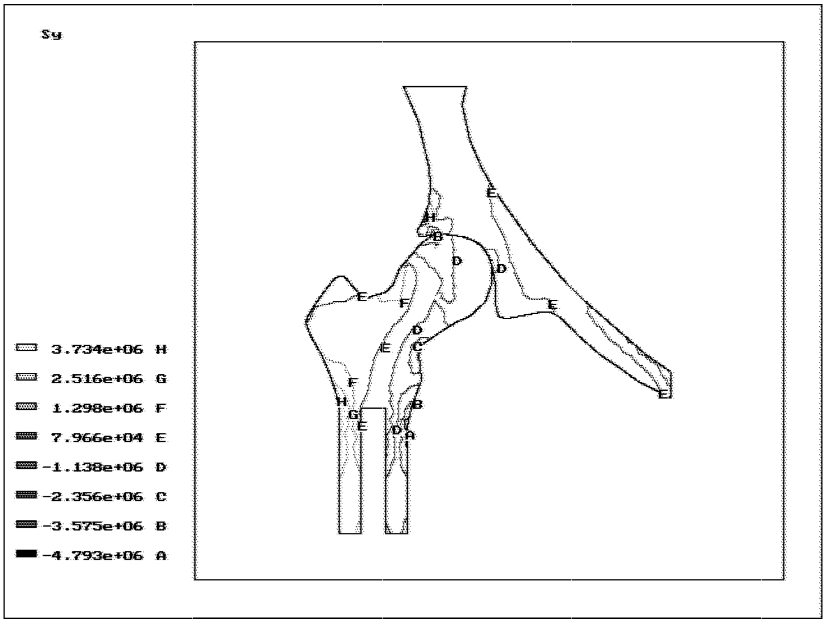

Figure 3.9:

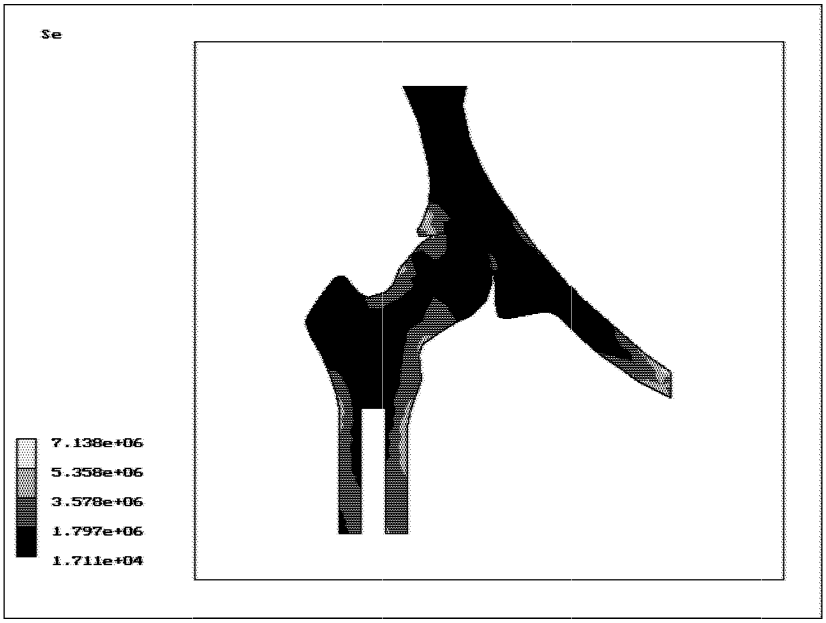

Figure 3.10:

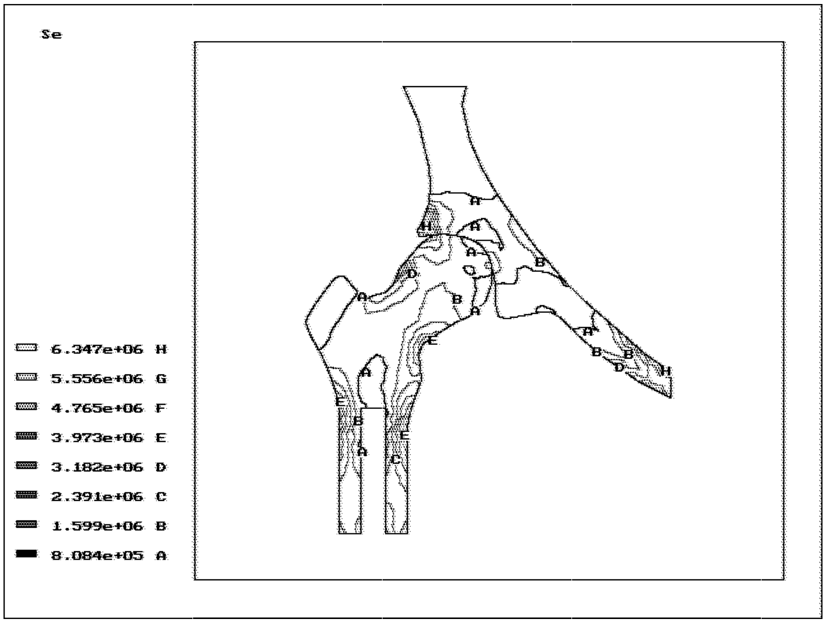

Figure 3.11:

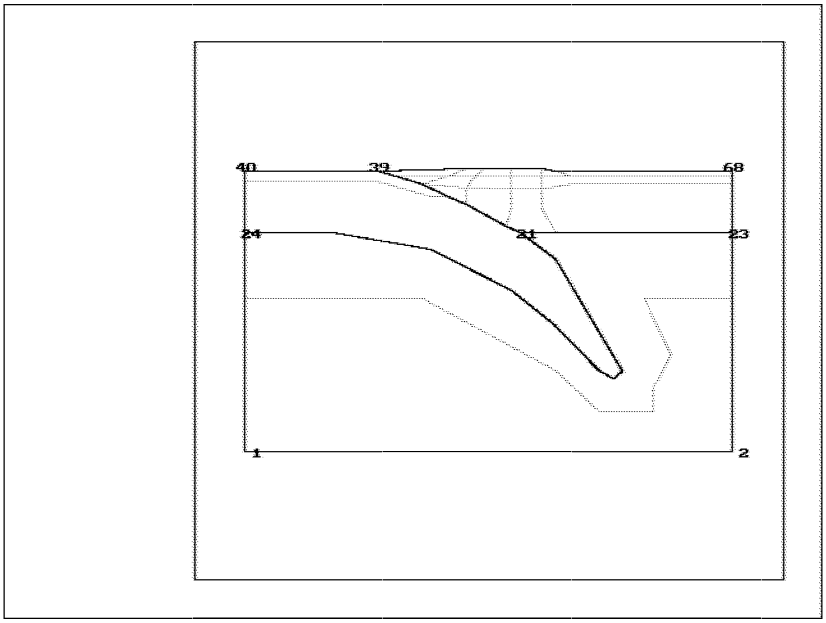

Figure 3.12:

Table 3.8:

$MODEL$	NV	NEL	NEQ	NCP	LIC	LJA	$TIME$
1	307	461	566	34	3637	136	37
2	307	461	566	34	3637	136	38
3	307	461	566	34	3637	136	39
4	309	463	570	33	3659	132	54

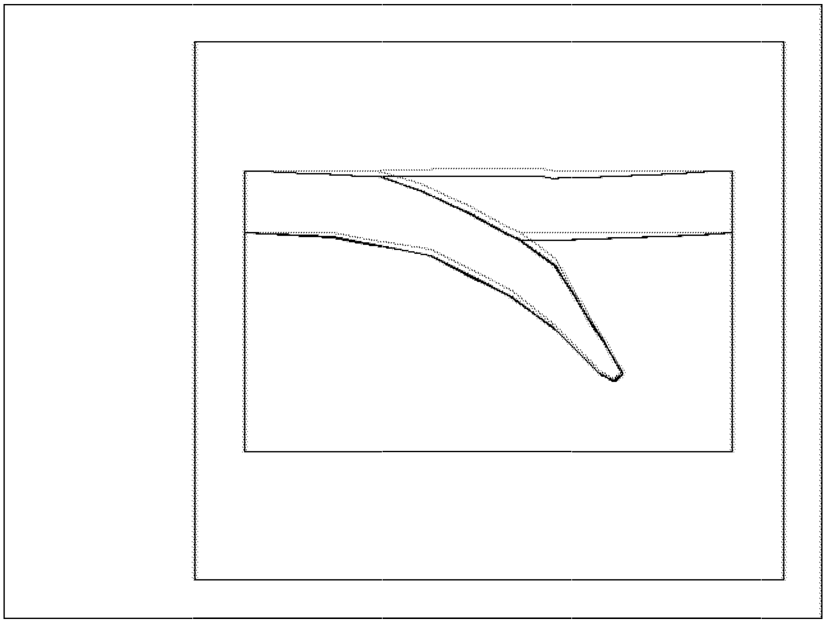

Figure 3.13:

the state of litospheric plates in various time steps. We have prescribed these values for $\mathbf{u}_{0L}$, $\mathbf{u}_{0R}$:

MODEL 1:	$\mathbf{u}_{0L1} = 0.5d{+}03$,	$\mathbf{u}_{0R1} = -0.5d{+}02\,[m]$	Fig. 3.13
MODEL 2:	$\mathbf{u}_{0L1} = 2.5d{+}03$,	$\mathbf{u}_{0R1} = -2.5d{+}02\,[m]$	Fig. 3.14
MODEL 3:	$\mathbf{u}_{0L1} = 0.5d{+}04$,	$\mathbf{u}_{0R1} = -0.5d{+}03\,[m]$	Fig. 3.15
MODEL 4:	$\mathbf{u}_{0L1} = 2.5d{+}04$,	$\mathbf{u}_{0R1} = -0.0d{+}00\,[m]$	Fig. 3.16

The statistics for this example is in Tab. 3.8.

For the last model, we have slightly modified the contact boundary which resulted different values of the parameters. Surface and contour plots for τ_e (Sec.3.) are depicted in Figs. 3.17-3.18 and principal stresses (e.g. [21]) in Fig.3.19.

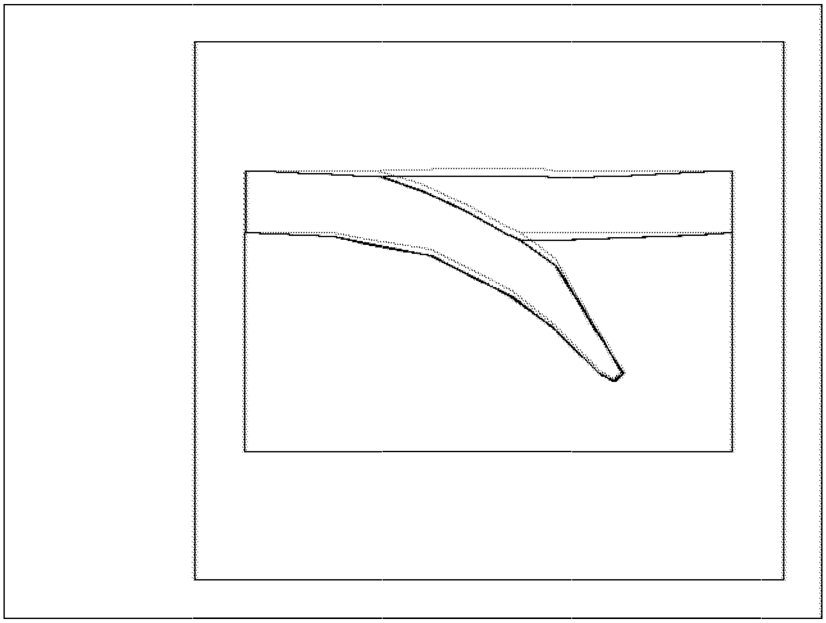

Figure 3.14:

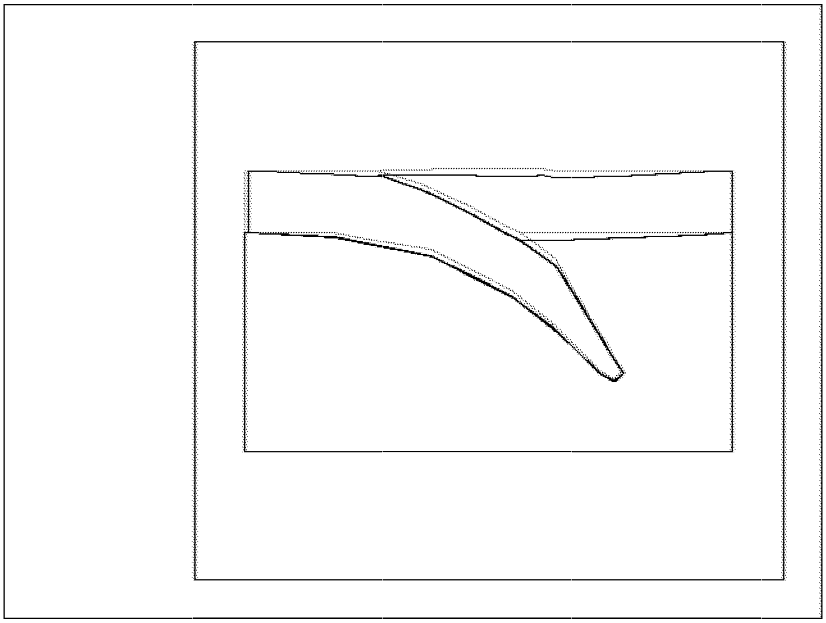

Figure 3.15:

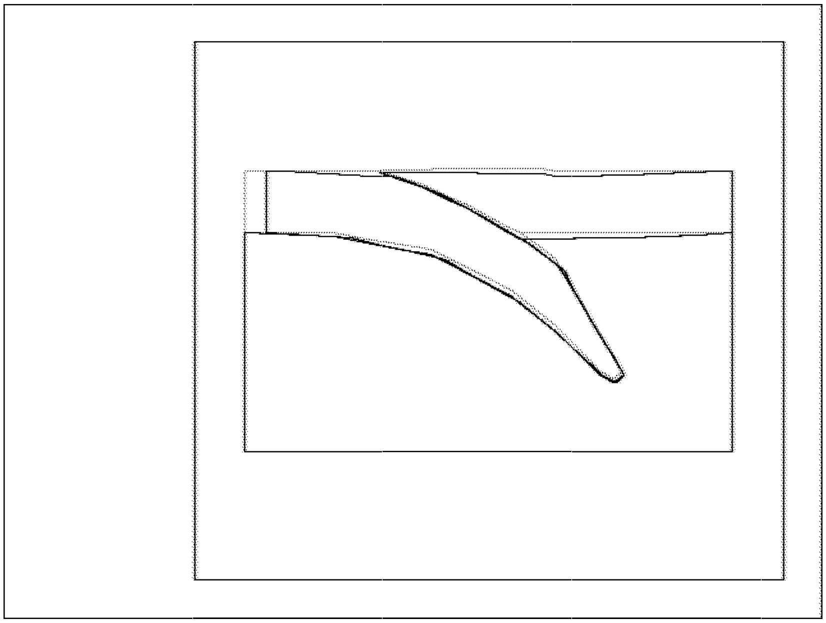

Figure 3.16:

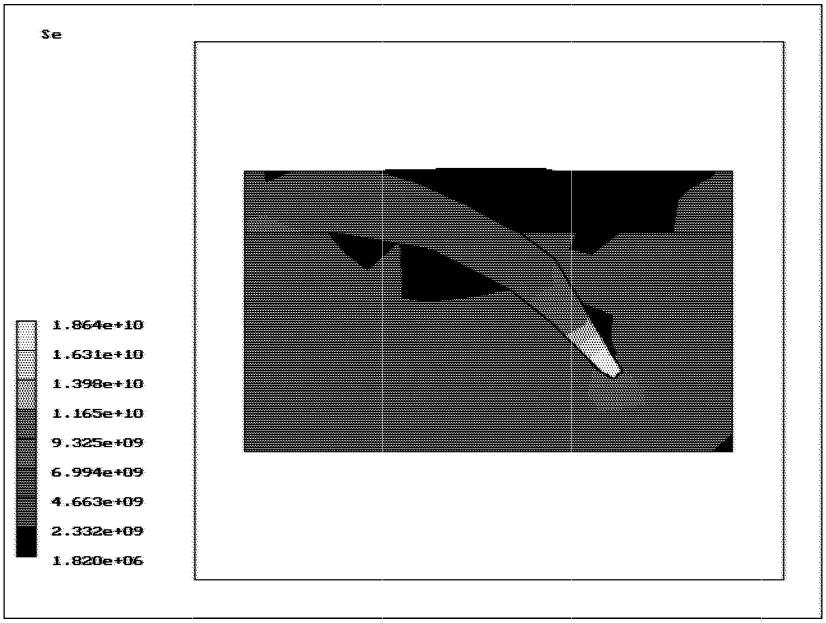

Figure 3.17:

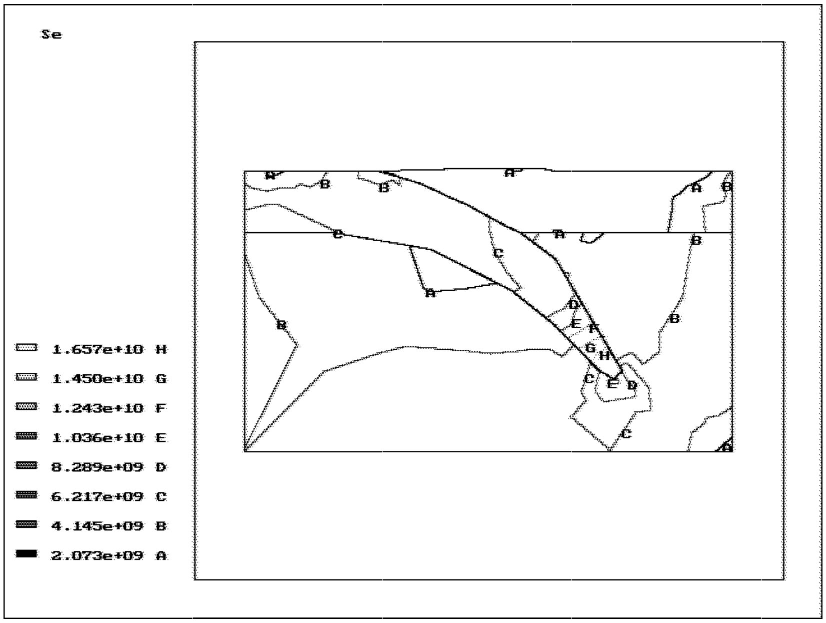

Figure 3.18:

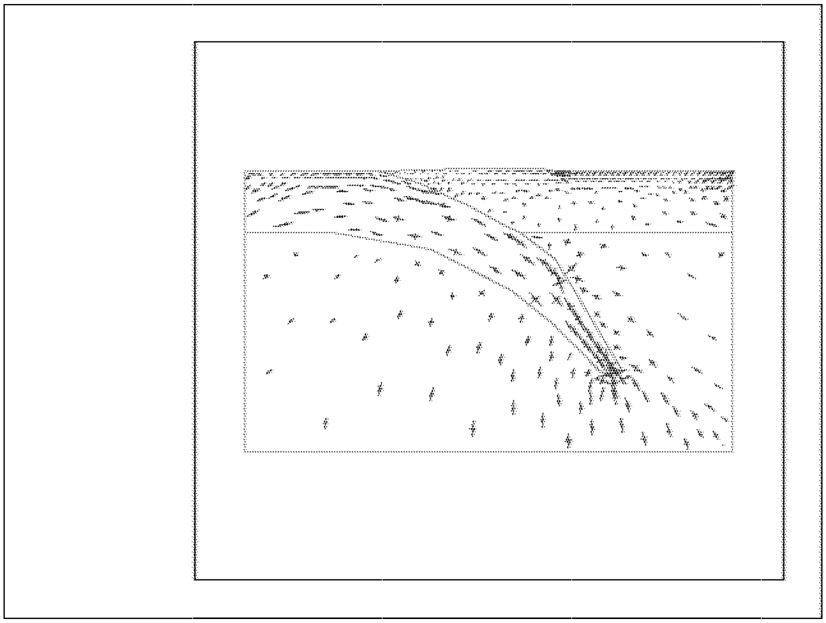

Figure 3.19:

Bibliography

[1] Axelsson O., Barker V.A. : *Finite Element Solution of Boundary Value Problems. Theory and Computation.* Academic Press, New York, London 1984

[2] Bartoš M. : *The Biomechanical Study of Acetabular Component Fixation in TEP Implantation. (In Czech)* Dissertation, LFUK Prague 1988

[3] Bartoš M., Kestřánek Z. : *Numerical Solution of the Contact Problem. Application to a Simple Model of the Human Hip Joint.* Submitted to J.Comput.Appl.Math. 1994

[4] Bittnar Z., Řeřicha P. : *Finite Element Method in the Dynamics of Structures. (In Czech)* SNTL, Prague 1981

[5] Cea J. : *Optimization - Theory and Algorithms.* Springer Verlag, Berlin 1978

[6] Fletcher R. : *Practical Methods of Optimization, Vol.2: Constrained Optimization.* J. Wiley and Sons, New York 1981

[7] Glowinski R., Lions J.L., Tremolièrs R. : *Numerical Analysis of Variational Inequalities.* North-Holland, Amsterdam 1981

[8] Haslinger J., Hlaváček I., Lovíšek J., Nečas J. : *The Solution of Variational Inequalities in Mechanics.* Alfa, Bratislava 1983

[9] Haslinger J., Tvrdý M. : *Approximation and Numerical Solution of Contact Problems with Friction.* Aplikace Matematiky 28, 1983, pp 55-74

[10] Horák J. : *Mathematical Modelling of the System "Moving Cutting Tool-Steady Rock". (In Czech)* TR HoÚ ČSAV, Ostrava 1990

[11] Frank P.D., Healy M.J., Mastro R.A. : *Implementation for Large Quadratic Programs with Small Active Sets.* Journal of Optimization Theory and Applications, Vol.69, No.1, 1991, pp 109-127

[12] Kestřánek Z. : *Comparison of Methods for Solving Contact Problem in Thermoelasticity.* In: Numerical Methods in Continuum Mechanics, Proc. of the Internat. Scient. Conf., Editional Centre VSDS Žilina, Stará Lesná-Slovakia 1994, pp 128-135

[13] Kočvara M. : *The Solution of Elasticity Problem on Polyedra. (In Czech)* In: Programs and Algorithms of Numerical Mathematics 4. Proc. Math. Inst. of Czech. Acad. Sci., 1988, pp 27-34

[14] Kolář V. et al. : *The Computation of Two- and Three Dimensional Structures by The Finite Element Method. (In Czech)* SNTL, Prague 1978

[15] Luenberger D.G. : *Hyperbolic Pairs in the Method of Conjugate Directions.* SIAM J.Appl.Math. Vol.17, No.6, 1969, pp 1263-1267

[16] Míka S., Šulcová I. : *The Saddle Point Algorithms. (In Czech)* In: Programs and Algorithms of Numerical Mathematics 5. Proc. Math. Inst. of Czech. Acad. Sci., 1990, pp 114-144

[17] Nedoma J. : *On One Type of Signorini Problem without Friction in Linear Thermoelasticity.* Aplikace Matematiky 28, 1983, pp 393-407

[18] Nedoma J. : *On the Signorini Problem with Friction in Linear Thermoelasticity. Quasicoupled 2D Case.* Aplikace Matematiky 32, 1987, pp 186-199

[19] Nedoma J. : *Finite Element Analysis in Nuclear Safety.* TR-ICS Prague 1993

[20] Nedoma J. : *Finite Element Analysis of Contact Problems in Thermoelasticity. The Semi-Coercive Case.* J.Comp.Appl.Mat. 50, 1994, pp 411-423

[21] Nečas J., Hlaváček I. : *Mathematical Theory of Elastic and Elasto-plastic Bodies. An Introduction.* North Holland, Amsterdam 1981

[22] Oden J.T., Kikuchi N. : *Contact Problems in Elasticity.* TICOM Report 79-8 July 1979. The University of Texas at Austin.

[23] Pshenichnyj B.N., Danilin J.M. : *Numerical Methods in Extremal Problems.* Mir Publishers, Moscow 1978