Designing compliant mechanisms with stress constraints using sequential linear integer programming

Mats Werme

Optimization and Systems Theory, Department of Mathematics, KTH, SE-100 44, Stockholm, Sweden
e-mail: werme@math.kth.se

1. Abstract
In this paper the problem of designing compliant mechanisms with stress constraints is addressed. The well-known difficulty of high stresses in mechanism designs obtained by topology optimization methods is handled by the introduction of stress constraints in the problem formulation. By the use of a sequential linear integer programming method the stress constraints can be handled explicitly. The obtained numerical results indicate that mechanism designs with relatively low stresses can be found with the method.

2. Keywords: Stress constraints, Topology optimization, Compliant mechanisms.

3. Introduction
This paper deals with compliant mechanism design by using a topology optimization technique that can handle stress constraints. A mechanism is a mechanical device that has the purpose of transferring motion and/or force from a source to an output, see [1]. When the mechanism gains its mobility from the flexibility of its components the mechanism is called a compliant mechanism. The original problem setting for compliant mechanisms is usually, even if truss designs occur, see e.g. [2], on a continuum design domain. In order to handle the problem numerically the common approach is to discretize the design domain into a ground structure with \( n \) finite elements. With this discretization, the finite element method, see e.g. [3], can be used to compute the displacements in the nodes and the stresses in the elements. The problem is then to decide which finite elements that should be filled with material and which should be kept as void. A natural choice is to define the design variable vector \( \mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \{0,1\}^n \) as \( x_j = 1 \) when the \( j \)th element is filled with material and \( x_j = 0 \) when the \( j \)th element is void.

In most cases the resulting topology optimization problem will be a non-linear integer program having a non-convex continuous relaxation. This means that finding the global optimum will be extremely hard in practice. A common heuristic approach is to first impose a lower bound \( \varepsilon > 0 \) on the design variables to ensure that the stiffness matrix never becomes singular. Next, the constraint \( \mathbf{x} \in \{\varepsilon, 1\}^n \) is relaxed so that the design variables become continuous and thus standard techniques from non-linear continuous optimization can be applied. In order to obtain solutions that are close to \( \varepsilon^{-1} \) designs the element stiffness matrices corresponding to design variables that have intermediate values between \( \varepsilon \) and 1 are modeled so that they are given low stiffness in comparison to the weight of the finite element. This approach is seen in interpolation models such as SIMP, see e.g. [4], [5] and [6]. For more information on topology optimization see e.g. [7] and the references therein.

As stated above this paper deals with compliant mechanism design. The main advantages of compliant mechanisms, as opposed to rigid body mechanisms that attain their mobility from e.g. hinges, bearings and sliders, is that they can be built using fewer parts, require fewer assembly processes and need no lubrication. These properties make compliant mechanisms well-suited for MicroElectricalMechanical Systems (MEMS), see e.g. [8], [9], [10] and [11]. Since it is desirable to have large displacements in a mechanism, non-linear modelling, see e.g. [2], [12], [13] and [14], will result in more accurate displacements. In this paper however, the focus is on the adding of stress constraints to the problem formulation and therefore the easier, linear, modelling will be used in the computation of the displacements. In principle, the used sequential linear integer programming (SLIP) method, see [15], can be applied if non-linear modelling is used but it is an open issue how to perform the required sensitivity calculations efficiently in this case.

The mechanism design problem is in a sense the opposite of the maximum stiffness (minimum compliance) problem. In the latter one seeks a design that has as small displacements as possible and in the former one seeks as large displacements as possible in certain directions. The compliance problem has
been well studied in the literature, see e.g. [7] and the references therein, and because of the similarity in problem structure, if small displacements are assumed, the techniques used for the minimum compliance problem can often be applied also for the mechanism design problem. The handling of stress constraints, however, is not as well studied as the “standard” compliance problem but different techniques have been described in the literature. In [16] the minimum weight problem with stress constraints is considered. Here, the variables are kept continuous but the stress criterion is adapted in order to obtain relevant stress limits in terms of models that allow for a density description of the structure. The use of a certain failure criterion is seen in [17], where it is used together with an ε-relaxation, see [18]. Here, the sensitivity analysis is performed on the continuum problem and the discretization is done afterwards. In [19] a minimum stress problem is treated by the use of the homogenization method, see e.g. [20]. Another approach is seen in [21] where the 0-1 requirement is relaxed and a concave quadratic function is added to the objective function in order to penalize intermediate values on the design variables. The resulting stress constrained minimum weight problem is then solved with an interior point method, see e.g. [22]. In [23] the minimum weight problem with stress and displacement constraints is treated. Here the design variables are kept binary and the displacement vector is used as a variable in the optimization problem in a SAND (Simultaneous ANalysis and Design) approach. With the use of additional force-like variables the non-linear integer program can be recast as a linear mixed integer program and then solved to global optimality. Of course, in practice, only problems with a relatively small number of design variables can be solved to global optimum. The same problem, with an additional constraint on the compliance, is studied and solved to local optimality in a certain sense in the hierarchical approach in [24]. In [15] some different stress constrained problems were solved to local optimum by the use of a sequential quadratic integer programming method as well as the above mentioned SLIP method. However, none of these techniques have, to the knowledge of the author, been applied to the mechanism design problem.

A common result when using topology optimization to design mechanisms is that the resulting mechanism is a so-called lumped compliant mechanism meaning that almost all deformation is taking place in small hinge-like regions, see e.g. [14]. These hinges are often made up of two solid finite elements connected at only one node. If a pure 0-1 formulation is used such designs can be avoided by the introduction of linear constraints, but still weak parts can appear in the structure since this will increase the movement of the mechanism. These hinge-like regions give rise to high stresses and may break the mechanism and are therefore undesirable. One possible way to deal with this is to introduce stress constraints in the problem formulation. The main goal of this paper is to formulate the mechanism design problem with stress constraints and then use the above mentioned SLIP method to solve the problem. The reason for choosing this method is twofold. First, the SLIP method keeps the design variables binary, thus eliminating the so called singularity problem, see [25], caused by the discontinuity of the stress constraint at zero density, see [26]. Second, the SLIP method handles the stress constraints explicitly. Further, the stress constraints need only be considered in those elements that have \( x_j = 1 \), i.e. those elements that are filled with material.

The paper is organized as follows. In Section 4 the notation and basic relations used in the paper are described. In Section 5 the considered mechanism problem is formulated mathematically. In Section 6 the subproblems needed in the SLIP method are derived and in Section 7 the obtained numerical results are presented. Finally, in Section 8 some conclusions are drawn.

4. Notation and basic relations

It is assumed that the continuum structure has been divided into \( n \) finite elements giving it a total of \( d \) degrees of freedom, dofs. The element stiffness matrix, in global coordinates, corresponding to element \( j \) is given by \( K_j \in \mathbb{R}^{d \times d} \). The design variable vector \( \mathbf{x} \in \{0, 1\}^n \) is defined as \( x_j = 1 \) if the \( j \)th element is filled with material while \( x_j = 0 \) if the \( j \)th element is void. This means that the global stiffness matrix, \( \mathbf{K}(\mathbf{x}) \in \mathbb{R}^{d \times d} \), can be written

\[
\mathbf{K}(\mathbf{x}) = \sum_{i=1}^{n} x_i \mathbf{K}_i.
\]

Even though this paper deals with compliant mechanism design the focus is mainly on the incorporation of stress constraints in the problem formulation. Therefore linear modelling will be used instead of the more accurate non-linear modelling in the computation of the displacements. When linear modelling is
used the displacement vector \( \mathbf{u} \in \mathbb{R}^d \) corresponding to a given external load vector \( \mathbf{p} \in \mathbb{R}^d \) is given as the solution to the equilibrium equations

\[
\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{p},
\]

provided that these equations have a solution.

If the design has white elements, i.e. some variables have \( x_j = 0 \), the global stiffness matrix may contain some rows where all matrix elements are zero. Since the stiffness matrix is symmetric the corresponding columns will then also be zero. These zero rows and columns are a consequence of nodes hanging in “mid-air” in the finite element discretization, see Figure 1. This means that it is possible to partition the global stiffness matrix as

\[
\mathbf{K}(\mathbf{x}) = \begin{bmatrix}
\bar{\mathbf{K}} & \mathbf{O} \\
\mathbf{O} & \mathbf{O}
\end{bmatrix},
\]

where \( \bar{\mathbf{K}} \in \mathbb{R}^{d \times d} \) is the matrix that corresponds to the “active dofs”. Assuming that the external load vector \( \mathbf{p} \) does not have a non-zero component among the “passive dofs” means that (2) can be written as

\[
\begin{bmatrix}
\bar{\mathbf{K}} & \mathbf{O} \\
\mathbf{O} & \mathbf{O}
\end{bmatrix}
\begin{bmatrix}
\mathbf{\hat{u}} \\
\mathbf{u}_0
\end{bmatrix} =
\begin{bmatrix}
\mathbf{\hat{p}} \\
\mathbf{0}
\end{bmatrix}.
\]

From (4) it is clear that \( \mathbf{u}_0 \) can be chosen arbitrarily and that the displacements in the active dofs, \( \mathbf{\hat{u}} \), can be computed as the solution to

\[
\bar{\mathbf{K}} \mathbf{\hat{u}} = \mathbf{\hat{p}},
\]

provided that these equations have a solution. In fact, the solution to this system of equations is unique if and only if \( \mathbf{x} \in \mathcal{S} \), where \( \mathcal{S} \) is the set of stable designs defined as

\[
\mathcal{S} = \{ \mathbf{x} \in \{0,1\}^n \mid \text{rank}(\mathbf{K}(\mathbf{x})) = \text{nzc}(\mathbf{K}(\mathbf{x})) \text{ and } \mathbf{p} \in \mathcal{R}(\mathbf{K}(\mathbf{x})) \},
\]

where \( \mathcal{R}(\mathbf{K}(\mathbf{x})) \) denotes the column space of \( \mathbf{K}(\mathbf{x}) \) and \( \text{nzc}(\mathbf{K}(\mathbf{x})) \) is the number of non-zero columns in \( \mathbf{K}(\mathbf{x}) \).

Defining \( \mathbf{u}_0 = \mathbf{0} \) means that the displacement vector \( \mathbf{u}(\mathbf{x}) \in \mathbb{R}^d \) given by

\[
\mathbf{u}(\mathbf{x}) = \begin{bmatrix}
\mathbf{\hat{u}} \\
\mathbf{0}
\end{bmatrix}
\]

is unique if \( \mathbf{x} \in \mathcal{S} \). Moreover, it is a solution to (2). With this displacement vector, the average squared von Mises stress in element \( j \) is given by

\[
\sigma_j^2(\mathbf{x}) =
\begin{cases}
\mathbf{u}(\mathbf{x})^T \mathbf{S}_j \mathbf{u}(\mathbf{x}), & \text{if } x_j = 1 \\
0, & \text{if } x_j = 0,
\end{cases}
\]

Figure 1: Nodes hanging in “mid-air”.
where the matrix $S_j$ is defined in [28]. This means that the average von Mises stress in element $j$ can be defined as

$$\sigma_j(x) = x_j \sqrt{u(x)^T S_j u(x)}.$$  \hspace{1cm} (9)

5. The considered problem

One example of a mechanism (half of a symmetric gripper) is given in Figure 2. In this case the idea is to apply force, given by $p$, at some part of the structure in order to make another part move in the direction defined by $r \in \mathbb{R}^d$, as much as possible. The “standard” mechanism design problem with a volume constraint, given by the upper bound $V$ on the amount of material that can be used, can be written

$$\max_x r^T u(x) \quad \text{s.t.} \quad e^T x \leq V, \quad x \in S,$$  \hspace{1cm} (10)

where $u(x)$ is given by (7) and (5) and $e = (1, \ldots, 1)^T \in \mathbb{R}^n$. The vector $r$ indicates in which directions the maximum displacements are desired, see Figure 2.

Usually the topology optimization approach to mechanism design introduces joints in the structure since a joint is very weak and thus allows large displacements. However, a joint will usually give rise to high stresses that may break the mechanism and is therefore undesirable. Since high stresses are unwanted stress constraints can be added to the above problem formulation resulting in the following problem

$$\max_x r^T u(x) \quad \text{s.t.} \quad e^T x \leq V, \quad \sigma_i(x) \leq \sigma_{\text{max}}, \quad i = 1, \ldots, n \quad x \in S,$$  \hspace{1cm} (11)

where $\sigma_{\text{max}}$ is a predefined stress limit. This formulation, however, makes it crucial to have a feasible starting point with respect to the stress constraints in the iterative method used in this paper (actually it is sufficient that there exists a feasible solution in the 1-neighbourhood of the starting point, but this is usually harder to know in advance). A way of dealing with infeasibility with respect to the stress constraints is to introduce an artificial variable $s \geq 0$ and then formulate the stress constraints as

$$\sigma_i(x) - \sigma_{\text{max}} s \leq \sigma_{\text{max}}, \quad i = 1, \ldots, n.$$  \hspace{1cm} (12)

Every design, $x$, is feasible with respect to (12). The reason for this is that the artificial variable $s$ makes sure that no matter how large the (finite) stresses are, one can always choose an $s$ such that the design becomes feasible. Replacing the stress constraints in (11) with the new constraints (12) will
then be equivalent to removing the stress constraints from (11) completely since every design is feasible with respect to (12). By adding a penalty term \(-C s\) to the objective a penalization is done on the “infeasibility” with respect to the stress constraints. This gives the following problem

\[
\begin{align*}
\max_{x,s} & \quad r^T u(x) - C s \\
\text{s.t.} & \quad e^T x \leq V, \\
& \quad \sigma_i(x) - \sigma_{\text{max}} s \leq \sigma_{\text{max}}, \quad i = 1, \ldots, n \\
& \quad x \in S, \ s \geq 0.
\end{align*}
\] (13)

The penalty parameter \(C > 0\) is then chosen to be so large that \(s\) will typically become zero unless there is no feasible solution with \(s = 0\).

In order to avoid certain designs, such as checkerboards, see e.g. [7], or too thin structures, a set of linear inequalities defined by \(Ax \leq b\) is added to the problem formulation. For more details regarding the unwanted designs and the linear constraints used to avoid them see [15]. With these linear constraints the problem becomes

\[
\begin{align*}
\max_{x,s} & \quad r^T u(x) - C s \\
\text{s.t.} & \quad e^T x \leq V, \\
& \quad \sigma_i(x) - \sigma_{\text{max}} s \leq \sigma_{\text{max}}, \quad i = 1, \ldots, n \\
& \quad Ax \leq b, \\
& \quad x \in S, \ s \geq 0.
\end{align*}
\] (14)

If a Lagrange multiplier \(\lambda > 0\) is introduced for the volume constraint, the following Lagrangian relaxation of (13) is obtained:

\[
\begin{align*}
\max_{x,s} & \quad L(x, \lambda) - C s \\
\text{s.t.} & \quad \sigma_i(x) - \sigma_{\text{max}} s \leq \sigma_{\text{max}}, \quad i = 1, \ldots, n \\
& \quad Ax \leq b, \\
& \quad x \in S, \ s \geq 0,
\end{align*}
\] (15)

where \(L(x, \lambda) = r^T u(x) - \lambda (e^T x - V)\) is the Lagrange function.

If \((\hat{x}, \hat{s})\) is an optimal solution to (15) for some fixed \(\lambda > 0\), then \((\hat{x}, \hat{s})\) is an optimal solution also to (13) for the particular right hand side \(V = e^T \hat{x}\). This implies that \(\lambda\) can be used as a tuning parameter to obtain optimal solutions to (13) for various right hand sides \(V\). In the following \(\lambda\) will be considered to be a given constant.

6. Formulating the subproblems

The sequential linear integer programming (SLIP) method described in [15] will be used to solve (15). The SLIP method solves a problem on the form

\[
\begin{align*}
\text{maximize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq f_{i}^\text{max}, \quad i = 1, 2, \ldots, m \\
& \quad x \in X,
\end{align*}
\] (16)

where \(X\) is a given subset of \(\{0, 1\}^n\), to a local optimum in a 1-neighbourhood. I.e. it finds a point \(x^*\) such that \(f_0(x^*) \leq f_0(x)\), \(\forall x \in F\) such that \(\|x - x^*\|_1 \leq 1\), where \(F\) denotes the feasible set of (16).

Define

\[
\begin{align*}
 f_0(x,s) &= r^T u(x) - \lambda e^T x - C s = h(x) - C s \quad \text{and} \\
 f_i(x,s) &= \sigma_i(x) - \sigma_{\text{max}} s, \quad i = 1, \ldots, n.
\end{align*}
\] (17)

Note that the term \(\lambda V\) in (15) is a constant and can be omitted from the objective function.

The idea in the iterative SLIP method is to use first order approximations, \(f_i\), of the functions \(f_i, \ i = 0, \ldots, n\) to generate a sequence of linear subproblems. Given an iteration point, the next
iteration point in the process is chosen to be the solution to the corresponding subproblem. Since the functions \( f_i, \ i = 0, \ldots, n \) are linear in \( s \) only the non-linear part which is dependent only on \( x \) needs to be approximated. Given an iteration point \( (x^{(k)}, s^{(k)}) \), where \( k \) is the iteration number, the first order approximations are defined as

\[
\begin{align*}
\tilde{f}_0^{(k)}(x, s) &= h_0(x^{(k)}) + g_0(x^{(k)})^T(x - x^{(k)}) - Cs, \\
\tilde{f}_i^{(k)}(x, s) &= \sigma_i(x^{(k)}) + g_i(x^{(k)})^T(x - x^{(k)}) - \sigma_{\max} s, \quad \text{for } i = 1, \ldots, n
\end{align*}
\]

with

\[
\begin{align*}
g_0(x^{(k)}) &= \left( \frac{\delta h_0(x^{(k)})}{\delta x_1} \ldots \frac{\delta h_0(x^{(k)})}{\delta x_n} \right)^T, \\
g_i(x^{(k)}) &= \left( \frac{\delta \sigma_i(x^{(k)})}{\delta x_1} \ldots \frac{\delta \sigma_i(x^{(k)})}{\delta x_n} \right)^T, \quad \text{for } i = 1, \ldots, n
\end{align*}
\]

and where

\[
\begin{align*}
\frac{\delta h_0(x^{(k)})}{\delta x_j} &= \frac{h_0(x^{(k)} + \xi_j e_j) - h_0(x^{(k)})}{\xi_j}, \\
\frac{\delta \sigma_i(x^{(k)})}{\delta x_j} &= \frac{\sigma_i(x^{(k)} + \xi_j e_j) - \sigma_i(x^{(k)})}{\xi_j}, \quad \text{for } i = 1, \ldots, n
\end{align*}
\]

where \( \xi_j = 1 - 2x_j^{(k)} \in \{-1, 1\} \) and \( e_j = (0, \ldots, 1, \ldots, 0)^T \in \mathbb{R}^n \). The linear integer subproblem in a given point \( (x^{(k)}, s^{(k)}) \) can then be written

\[
\max_{x,s} h_0(x^{(k)}) + g_0(x^{(k)})^T x - g_0(x^{(k)})^T x^{(k)} - Cs
\]

\[
\text{s.t. } \sigma_i(x^{(k)}) + g_i(x^{(k)})^T x - g_i(x^{(k)})^T x^{(k)} - \sigma_{\max} s \leq \sigma_{\max}, \quad i = 1, \ldots, n
\]

\[
\|x - x^{(k)}\|_1 \leq M, \\
Ax \leq b,
\]

\[
x \in \{0,1\}^n, \quad s \geq 0,
\]

for some \( 0 \leq M \leq n \) which defines a neighbourhood of the current iteration point \( x^{(k)} \). Define

\[
J_0 = \{ j | x_j^{(k)} = 0 \} \quad \text{and } \quad J_1 = \{ j | x_j^{(k)} = 1 \}
\]

and note that \( \|x - x^{(k)}\|_1 = \sum_{i=1}^n |x_i - x_i^{(k)}| = \sum_{i \in J_0} x_i + \sum_{i \in J_1} (1 - x_i) \). This together with the possibility to check whether a neighbouring design is in \( S \) in the sensitivity calculations, this is discussed in [15], makes (21) a linear integer program. Since \( h_0(x^{(k)}) - g_0(x^{(k)})^T x^{(k)} \) is a constant it does not influence the optimal solution to (21) only the optimal value and therefore the following problem can be solved instead

\[
\max_{x,s} \quad g_0(x^{(k)})^T x - Cs
\]

\[
\text{s.t. } \quad g_i(x^{(k)})^T x - \sigma_{\max} s \leq \sigma_{\max} - \sigma_i(x^{(k)}) + g_i(x^{(k)})^T x^{(k)}, \quad i = 1, \ldots, n
\]

\[
\|x - x^{(k)}\|_1 \leq M, \\
Ax \leq b,
\]

\[
x \in \{0,1\}^n, \quad s \geq 0.
\]

The next iteration point \( (x^{(k+1)}, s^{(k+1)}) \) is then chosen as the optimal solution to (23).

Note that in order to make the SLIP method efficient the computation of the approximating linear functions, and thus the difference quotients in (20), has to be efficient. This can be achieved by using the sensitivity calculations described in [27] since linear modelling of the displacements is done. In principle though, the SLIP method can be employed for non-linear modelling, but then the efficient computation of the quotients in (20) is an open issue.
7. Numerical results

The considered ground structure and load case is given in Figure 2. Symmetry is forced meaning that only half of the mechanism needs to be modeled and used in the computations. In Figure 3-Figure 5 the whole obtained mechanisms are presented.

The load was a distributed unit load, Young’s modulus was set to 1, the Poisson’s ratio was 0.3 and the spring coefficients were set to 0.1. The method was implemented in a hierarchical way where one starts with a coarse mesh, solves the problem on this mesh and then translates the obtained solution to a starting point in a new finer mesh and so on. In the numerical examples square nine node (Q9) finite elements have been used, see e.g. [3]. A refinement consists of dividing each finite element into four new square finite elements. As in [15] a mesh is considered relatively fine from the second refinement and onwards, meaning that basically shape optimization is performed from the second refinement. The number of finite elements in each mesh were \( n = (18, 72, 288, 1152) \) and the neighbourhood size was set to \( M = (1, 2, 4, 4) \) for the four different meshes respectively.

First, the problem was solved with the penalty parameter set to \( C = 0 \) and the Lagrange multipliers were chosen to be

\[
\lambda = \frac{1}{n_l} \cdot |r^T u(x_l^{(0)})|,
\]

where \( x_l^{(0)} \) denotes the initial design in the \( l \)th mesh refinement and \( n_l \) is the number of finite elements in the \( l \)th mesh refinement. The obtained solutions in the hierarchical implementation are presented in Figure 3. Let the reference solution for mesh refinement \( l \) be given by \( x_l^{ref} \) and let \( \sigma_l^{ref} \) be the highest stress in this obtained solution. Now, (15) is solved again with

\[
\sigma_l^{\text{max}} = \begin{cases} 
\sigma_l^{\text{ref}}, & \text{if } l = 0, \\
0.75 \cdot \sigma_l^{\text{ref}}, & \text{if } l > 0,
\end{cases}
\]

where \( l \), again, indicates the level of mesh refinement. The penalty parameter \( C_l \) was chosen to be

\[
C_l = n_l \cdot |r^T u(x_l^{\text{ref}})|.
\]

![Figure 3: The obtained results for the gripper in Figure 2 when \( C = 0 \).](image)
The obtained solutions in the hierarchical implementation are presented in Figure 4. Let the obtained solution to (15) with stress constraints in the $l$th mesh refinement be denoted $x_l^\sigma$. A comparison of the reference solution $x_l^{ref}$ and the stress constrained solution $x_l^\sigma$ is given in Table 1 and Table 2. It can be seen in Table 1 and Table 2 that the use of stress constraints can lower the maximum stress to about 75\% of the maximum stress in the solution obtained without stress constraints (recall that $\sigma_l^{\text{max}}$ is set to $0.75 \cdot \sigma_l^{\text{ref}}$). As can be expected the mechanism becomes less effective, which is the price to pay for the lowering of the stresses.

The deformation of the obtained designs in the finest mesh are presented in Figure 5. In fact, all obtained designs were grippers in their respective mesh.

8. Conclusions
In this paper the problem of designing compliant mechanisms with stress constraints is addressed. The well-known difficulty of high stresses in mechanism designs obtained by topology optimization methods
is handled by the introduction of stress constraints in the problem formulation.

Since a pure 0-1 formulation is used linear inequalities can be used to remove checkerboards and thus also hinges connected in only one node. Other regions with high stresses can be avoided by the adding and explicit handling of stress constraints in the problem. This is possible by the use of the sequential linear integer programming method. The numerical results indicate that the approach can lower the stresses and still obtain a functional mechanism.

An open question for future research is the development of efficient computational methods for the sensitivities of displacements if non-linear modelling is used.

Acknowledgements
The author would like to thank Krister Svanberg for valuable discussions and comments improving the quality of the paper.

9. References


