The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

ON THE EMBEDDING OF NONLINEAR MULTIPOINT CONSTRAINTS IN THE FINITE ELEMENT METHOD

JENS WACKERFUSS[1] AND JONAS BOUNGARD[1]

[1] Institute of Structural Analysis, Faculty Civil and Environmental Engineering,
University of Kassel,
Mönchebergstraße 7, 34109 Kassel, Germany
emails: wackerfuss@uni-kassel.de and boungard@uni-kassel.de,
https://www.uni-kassel.de/go/baustatik

Key words: Multi-point constraints, nonlinear constraints, master-slave elimination

Summary. *The consideration of constraints is crucial in continuum and structural mechanics, such as in modeling hinges, rigid inclusions, or deformation-dependent Dirichlet boundary conditions. In the finite element method, constraints relate to nodal degrees of freedom and are termed multi-point constraints when involving multiple nodes. A nonlinear formulation of the constraints is required if these degrees of freedom change significantly. Unlike Lagrange multipliers and the penalty method, the master-slave elimination reduces the problem dimension but is limited to linear constraints. We introduce a new master-slave elimination method for arbitrary nonlinear multi-point constraints. The method's derivation is general and can be separated from the specific constraints in algorithmic implementation. Numerical examples demonstrate that the method is as accurate, robust, and flexible as Lagrange multipliers, while being more efficient by reducing the total number of degrees of freedom, particularly beneficial for numerous constraints.*

1 INTRODUCTION

The consideration of constraint conditions plays an important role in many areas of continuum and structural mechanics. For example, in the modeling of shear force hinges in frame structures, in the modeling of rigid inclusions in a body originally modeled as deformable, or in the modeling of deformation-dependent Dirichlet boundary conditions. In the context of the finite element method, the constraints refer to the nodal degrees of freedom. If several nodes are involved, they are referred to as multi-point constraints. These must be formulated non-linearly if the degrees of freedom involved change significantly during the simulation (e.g. as a result of a heavily deformed FE mesh). One method for considering constraints is master-slave elimination, which, in contrast to Lagrange multipliers and the penalty method, offers the advantage of reducing the dimension of the problem. However, the existing master-slave elimination method is limited to linear constraints. Therefore, an extension to nonlinear constraints is of great interest.

The master-slave elimination was originally introduced by Green [10] in finite element analysis for linear constraints. Subsequent research primarily focused on efficient implementation but not on extending the method to nonlinear constraints. Notable studies include Szabo and Tsai [15], Curiskis and Valliapan [8], Abel and Shephard [1], Shephard [14], Chang and Lin [7], and Ainsworth [2]. However, few studies have explored master-slave elimination for nonlinear constraints: Naraswayamy [13] and Dhondt [9] proposed formulations for nonlinear constraints, both with assumptions on the structure of constraints as well as missing consistent linearization. Jelenić and Crisfield [12] presented a method solely capable of handling joints with large rotations, while Areias et al. [3] emphasized efficient sparse system access.

The authors of this contribution introduced an extension of the method to arbitrary nonlinear constraints [6]. In the following, the derivation of this new method is presented and it is compared to existing constraint methods both on the basis of the equations as well as a numerical example involving a large number of constraints. Additionally, both the challenge of selecting the slave dofs as well as the handling of redundant constraints are addressed.

2 GOVERNING EQUATIONS

Starting point is an arbitrary nonlinear boundary value problem (BVP). It is solved by using the finite element method. In the spatial discretization of the weak form of the BVP, the degrees of freedom (dofs) $\mathbf{V} \in \mathbb{R}^{n_{\text{dof}}}$ are introduced. To obtain the solution for the dofs $\mathbf{V}$, the minimum of the discretized functional of the BVP, denoted as $W(\mathbf{V}) \in \mathbb{R}$, has to be found

$$\min_{\mathbf{V}} W(\mathbf{V}) \tag{1}$$

which can be interpreted as an *optimization problem without constraints*. Eq. (1) is solved by finding a set of unknowns $\mathbf{V}$ with a gradient equal to zero, leading to the following nonlinear system of equations with the residual vector $\mathbf{R} \in \mathbb{R}^{n_{\text{dof}}}$:

$$\mathbf{R}(\mathbf{V}) := \left(\frac{\partial W(\mathbf{V})}{\partial \mathbf{V}}\right)^T = \mathbf{0} \tag{2}$$

The nonlinear system Eq. (2) cannot be solved analytically in general. For the numerical solution, the Newton-Raphson method can be used. Then, Eq. (2) has to be linearized, which leads to the following linear system for solving for the unknown increment of the dofs $\Delta\mathbf{V} \in \mathbb{R}^{n_{\text{dof}}}$

$$\mathbf{K}_T \Delta\mathbf{V} = -\mathbf{R} \; ; \qquad \text{with} \quad \mathbf{K}_T(\mathbf{V}) := \frac{\partial \mathbf{R}(\mathbf{V})}{\partial \mathbf{V}} = \frac{\partial^2 W(\mathbf{V})}{\partial \mathbf{V} \partial \mathbf{V}} \tag{3}$$

with the tangential stiffness matrix $\mathbf{K}_T \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$.

The dofs of the initial BVP are now restricted by a set of n_c constraints $\mathbf{c} \in \mathbb{R}^{n_c}$ depending nonlinearly on the dofs:

$$\mathbf{c}(\mathbf{V}) = \mathbf{0} \tag{4}$$

With the nonlinear constraints in Eq. (4), the optimization problem without constraints Eq. (1) is transformed in the following *optimization problem with constraints*:

$$\min_{\mathbf{V}} W(\mathbf{V}) \quad \text{subject to} \quad \mathbf{c}(\mathbf{V}) = \mathbf{0} \tag{5}$$

The numerical solution of Eq. (5) involves an effective nonlinear system which is solved using the Newton-Raphson method. For this purpose, the Jacobian $\mathbf{G} := \frac{\partial \mathbf{c}}{\partial \mathbf{V}} \in \mathbb{R}^{n_c \times n_{\text{dof}}}$ and Hessian $\mathbf{H} = [\mathbf{H}_1, \ldots, \mathbf{H}_i, \ldots, \mathbf{H}_{n_c}]$ with $\mathbf{H}_i := \frac{\partial^2 c_i}{\partial \mathbf{V} \partial \mathbf{V}} \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$ of the constraints are introduced.

3 CLASSIFICATION AND EXAMPLES FOR MULTI-POINT CONSTRAINTS

Before presenting the new constraint method, several constraints are introduced for illustrative purposes. This paper focuses on the so called nonlinear "multi-point constraints". Strictly speaking, the term "multi-point constraints" can be misleading because it could imply that the constraint equations act upon nodes (points), rather than dofs. However, the constraints are formulated with respect to individual dofs. This is investigated in the following by a brief classification of constraints which includes several examples. This illustrates the wide variety of constraints that can be applied in the context of the finite element method.

3.1 Single node constraints

Single node constraints act upon dofs of a single node. They can be handled in the same way as constraints acting upon multiple nodes since the underlying constraint equations are structured the same way. Examples for such constraints are given in Figure 1. They can act upon a single dof or multiple dofs. For the latter, the constraint equation can either depend linearly or non-linearly on the dofs.

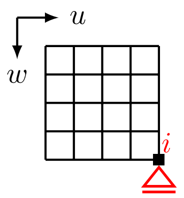
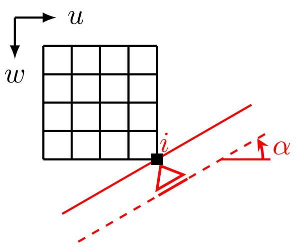
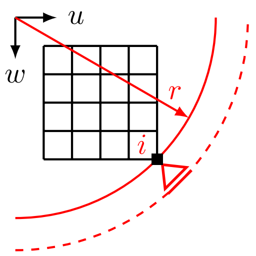
single-dof constraints	**multi**-dof constraints	
linear	linear	non-linear
		
$c(\mathbf{V}) = w_i = 0$	$c(\mathbf{V}) = \cos(\alpha)u_i + \sin(\alpha)w_i = 0$	$c(\mathbf{V}) = \sqrt{u_i^2 + (w_i + r)^2} - r = 0$

Figure 1: Examples for single node constraints

3.2 Multi-node constraints

In most applications, constraints depend on the dofs of multiple nodes. Examples for this are given in Figure 2 and include rigid trusses, rigid beams, special links of dofs like the shear force release or the application of periodic boundary conditions at an representative volume element (RVE) in the FE^2 method. Different types of coupling of multi-node constraints result in different structures of the Jacobian $\mathbf{G}$. Special structures of $\mathbf{G}$ can be used exploited for a more efficient handling of the constraints, which is described in detail in [6].

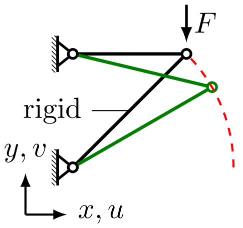
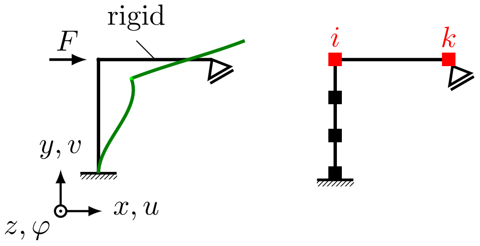
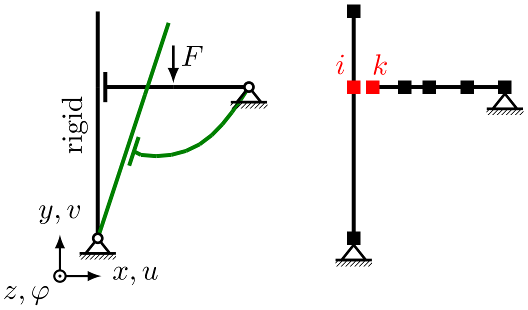
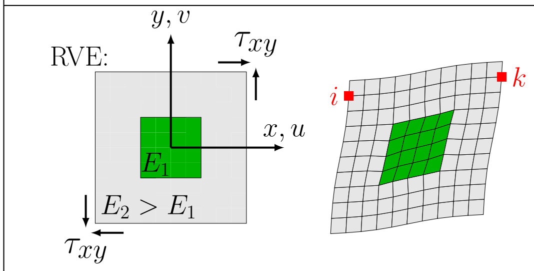
System and deformation	constraints
	$c := d_{ik} - D_{ik} = 0$ with: $d_{ik} = \lVert\mathbf{x}_k - \mathbf{x}_i\rVert = \left\lVert \mathbf{D} + \begin{bmatrix}u_k\\v_k\end{bmatrix} - \begin{bmatrix}u_i\\v_k\end{bmatrix} \right\rVert$ $D_{ik} = \lVert\mathbf{D}\rVert\ ;\quad \mathbf{D} = \mathbf{X}_k - \mathbf{X}_i$
	$c_1 := d_{ik} - D_{ik} = 0$ $c_2 := \varphi_i - \varphi_k = 0$ $c_3 := \varphi_i - \psi_{ik} = 0$ with: $\psi_{ik} = \operatorname{atan2}(a, b)$ $\begin{bmatrix}a\\b\end{bmatrix} = \begin{bmatrix}\bar{N}_x & -\bar{N}_y\\ \bar{N}_y & \bar{N}_x\end{bmatrix} \left(\mathbf{N} + \begin{bmatrix}u_k\\v_k\end{bmatrix} - \begin{bmatrix}u_i\\v_k\end{bmatrix} \right)$ $\mathbf{N} = \mathbf{X}_k - \mathbf{X}_i\ ;\quad \bar{\mathbf{N}} = \mathbf{N}/\lVert\mathbf{N}\rVert$
	$c_1 := \varphi_i - \varphi_k = 0$ $c_2 := \mathbf{n}_i^T \begin{bmatrix}u_i\\v_i\end{bmatrix} - \mathbf{n}_k^T \begin{bmatrix}u_k\\v_k\end{bmatrix} = 0$ with: $\mathbf{n}_i = \mathbf{R}_i\mathbf{N} \quad \mathbf{n}_k = \mathbf{R}_k\mathbf{N}$ $\mathbf{R}_i = \begin{bmatrix}\cos(\varphi_i) & -\sin(\varphi_i)\\ \sin(\varphi_i) & \cos(\varphi_i)\end{bmatrix}$ $\mathbf{R}_k = \ldots;\quad \mathbf{N} = \begin{bmatrix}\cos(\alpha_0)\\ \sin(\alpha_0)\end{bmatrix}$
 RVE: y, v τxy x, u E1 E2 > E1 τxy i k	$\begin{bmatrix}c_1\\c_2\end{bmatrix} := \begin{bmatrix}u_i\\v_i\end{bmatrix} - \begin{bmatrix}u_k\\v_k\end{bmatrix} - \mathbf{A}\,\boldsymbol{\varepsilon} = \begin{bmatrix}0\\0\end{bmatrix}$

Figure 2: Examples for multi-node constraints

4 MASTER-SLAVE ELIMINATION FOR NONLINEAR MULTI-POINT CONSTRAINTS

4.1 Derivation of the master-slave elimination for nonlinear constraints

In this section, the extensive and rigorous derivation of the new method presented in [6] is briefly recalled. The central idea of the new approach is based on the calculation of the constraint forces $\mathbf{C} = [C_1, \ldots, C_i, \ldots C_{n_{\text{dof}}}] \in \mathbb{R}^{n_{\text{dof}}}$ resulting from the constraints $\mathbf{c}$ introduced in Section 2. The constraint forces have to be taken into account in the equilibrium equations Eq. (2) resulting in the modified nonlinear system of equations with the modified residual vector $\mathbf{R}_{\text{mod}}$:

$$\mathbf{R}_{\text{mod}} := \mathbf{R} - \mathbf{C} \tag{6}$$

In Figure 3, this modification of the equilibrium and the residual vector is illustrated. Here, a simple one-dimensional distance constraint between nodes i and k in form of $c(\mathbf{V}_i, \mathbf{V}_k) = u_i - u_k = 0$ is employed.

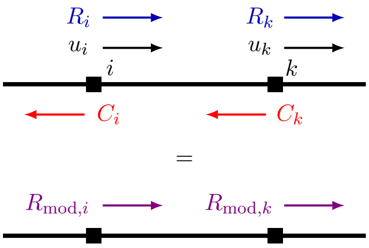


Figure 3: Illustration of the constraint forces

The degrees of freedom are partitioned into $n_{\text{dof}} - n_c$ master dofs, denoted with m, and n_c slave dofs, denoted with s

$$\mathbf{V} \xrightarrow{\text{partition}} \begin{bmatrix} \mathbf{V}_m^T & \mathbf{V}_s^T \end{bmatrix}^T \tag{7}$$

with the subvectors of the master dofs $\mathbf{V}_m \in \mathbb{R}^{n_{\text{dof}} - n_c}$ and the slave dofs $\mathbf{V}_s \in \mathbb{R}^{n_c}$. For each constraint, one slave dof can be identified. For the minimal example in Figure 3, the slave dof could either be the horizontal displacement u_i at node i or the horizontal displacement u_k at node k. The dofs, that are not a slave dof, are denoted as master dofs.

Since the constraints $\mathbf{c}$ are a function depending on both the master dofs $\mathbf{V}_m$ as well as the slave dofs $\mathbf{V}_s$, the constraint forces $\mathbf{C}$ and, in turn, all quantities in Eq. (6) depend on both master and slave dofs. In order to perform the master-slave elimination and derive a function solely depending on the master dofs, a formula for the slave dofs $\mathbf{V}_s$ depending on the master dofs $\mathbf{V}_m$ is derived based on the *implicit function theorem.*

In order to apply the implicit function theorem, a selection of slave dofs $\mathbf{V}_s$ must exist, that fulfills the regularity condition

$$\det(\mathbf{G}_s) \neq 0 \; ; \qquad \mathbf{G}_s := \frac{\partial \mathbf{c}}{\partial \mathbf{V}_s} \tag{8}$$

Using the *implicit function theorem* leads to the following nonlinear system:

$$\mathbf{R}_{\text{mod}} := \begin{bmatrix} \mathbf{R}_m - \mathbf{G}_m^T \, \mathbf{G}_s^{-T} \, \mathbf{R}_s \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \; ; \quad \mathbf{G}_m := \frac{\partial \mathbf{c}}{\partial \mathbf{V}_m} \; ; \mathbf{R}_m^T := \frac{\partial W}{\partial \mathbf{V}_m} \; ; \mathbf{R}_s^T := \frac{\partial W}{\partial \mathbf{V}_s} \tag{9}$$

The nonlinear system Eq. (9) is solved using the Newton-Raphson method. This results in the following linear system to calculate the increments of the master and slave degrees of freedom $\Delta\mathbf{V}_m$ and $\Delta\mathbf{V}_s$

$$\begin{bmatrix}\mathbf{K}_{\text{mod},mm} & \mathbf{K}_{\text{mod},ms}\\ \mathbf{G}_m & \mathbf{G}_s\end{bmatrix}\begin{bmatrix}\Delta\mathbf{V}_m\\ \Delta\mathbf{V}_s\end{bmatrix} = -\begin{bmatrix}\mathbf{R}_{\text{mod},m}\\ \mathbf{c}\end{bmatrix} \quad ; \quad \mathbf{K}_{\text{mod}}\Delta\mathbf{V} = -\mathbf{R}_{\text{mod}} \ ; \quad \mathbf{K}_{\text{mod}} := \frac{\partial\mathbf{R}_{\text{mod}}}{\partial\mathbf{V}} \tag{10}$$

In a next step, one obtains the linearized, reduced system of equations for the master-slave *elimination* with the reduced dimension $n_{\text{dof}} - n_c$ by static condensation:

$$\underbrace{\left(\mathbf{K}_{\text{mod},mm} - \mathbf{K}_{\text{mod},ms}\mathbf{G}_s^{-1}\mathbf{G}_m\right)}_{=:\mathbf{K}_{\text{red}}}\Delta\mathbf{V}_m = \underbrace{-\mathbf{R}_{\text{mod},m} + \mathbf{K}_{\text{mod},ms}\mathbf{G}_s^{-1}\mathbf{c}}_{=:-\mathbf{R}_{\text{red}}} \tag{11}$$

with the reduced stiffness matrix $\mathbf{K}_{\text{red}} \in \mathbb{R}^{(n_{\text{dof}}-n_c)\times(n_{\text{dof}}-n_c)}$ and the reduced residual vector $\mathbf{R}_{\text{red}} \in \mathbb{R}^{n_{\text{dof}}-n_c}$. If the tangential stiffness matrix of the unconstrained problem $\mathbf{K}_T$ is symmetric, the resulting reduced matrix $\mathbf{K}_{\text{red}}$ is symmetric too. For the special case that all constraints depend linearly on the unknowns $\mathbf{V}$, Eq. (11) coincides with the equations on existing master-slave elimination schemes for linear constraints.

4.2 Automatic selection of slave dofs and redundancy handling

In the master-slave elimination, there are the following challenges with respect to the regularity condition in Eq. (8): a) selection of an admissible set of slave dofs, b) identification and elimination of redundant constraints, c) identification of contradictory constraints and d) selection of slave dofs in the context of coupled Dirichlet boundary conditions and constraints. The consideration of these aspects is particularly relevant in the context of coupled constraints. Due to the nonlinearity of the constraints, a re-evaluation of these aspects over the the course of the simulation is required.

It can be shown that all of these aspects can be considered by the transformation of the augmented system $[\mathbf{G}|\mathbf{c}]$ in its row-echelon form:

$$[\mathbf{G}|\mathbf{c}] \xrightarrow[\text{elimination}]{\text{extended GAUSS-JORDAN-}} \begin{cases}\left[\begin{array}{cc|c}\bar{\mathbf{G}}_s & \bar{\mathbf{G}}_m & \bar{\mathbf{c}}\\ \mathbf{0} & \mathbf{0} & \bar{\mathbf{e}}\end{array}\right]\\ \bar{m},\ \mathbf{P}_{\text{col}},\ \mathbf{Q}\end{cases}$$

with the number of non-redundant constraint $\bar{m}$, the submatrices $\bar{\mathbf{G}}_s \in \mathbb{R}^{\bar{m}\times\bar{m}}$ and $\bar{\mathbf{G}}_m \in \mathbb{R}^{\bar{m}\times(n_{\text{dof}}-\bar{m})}$, the subvectors $\bar{\mathbf{c}} \in \mathbb{R}^{\bar{m}}$ and $\bar{\mathbf{e}} \in \mathbb{R}^{n_c-\bar{m}}$, the column permutation matrix $\mathbf{P}_{\text{col}} \in \mathbb{N}^{n_{\text{dof}}\times n_{\text{dof}}}$ and the transformation matrix $\mathbf{Q} \in \mathbb{R}^{n_c\times n_c}$. The incorporation of this in the new method is presented in detail in [5].

4.3 Properties and comparison of methods

In the following, the new master-slave elimination scheme is compared to penalty and multiplier methods. For this, their advantages and disadvantages are briefly recalled. An excellent overview for such methods can be found in the textbook by Belytschko et al. [4].

The penalty method (PM) is easy to implement because the structure of its effective linear system does not differ from an unconstrained problem Eq. (3). In addition, the method does not need any modifications for redundant constraints but can handle them inherently. However, it is well known that the method suffers from two major drawbacks related to the choice of the additional penalty factor: The constraints are never satisfied exactly and small penalty factors violate them heavily, while large penalty factors approximately satisfy the constraints. However, large penalty factors worsen the condition number of the resulting matrix.

It is well known that Lagrange multipliers (LM) satisfy the constraints exactly but the introduction of additional dofs enlarges the dimension of the problem. Moreover, the saddle point structure of the resulting system poses an additional challenge to the equation solver, leading to numerical problems in some relevant cases.

Like the existing master-slave elimination schemes for linear constraints, the new master-slave elimination scheme for nonlinear constraint satisfies the constraints exactly and reduces the dimension of the resulting linear system by the number of constraints, see Eq. (11). As mentioned above, the master-elimination requires the selection of an appropriate set of slave dofs. This poses a challenge but can be done with minor additional cost which is illustrated in [5]. The master-slave elimination changes the structure of the resulting linear system fundamentally. For the implementation in a finite element code, this requires resizing and reallocation of the arrays. Since all matrices involved in the method are sparse, they can be stored efficiently in compressed sparse row format (CSR). In turn, calculations as well as reallocation of these matrices can be carried out very efficiently with external libraries such as MKL (Intel) [11]. Numerical examples show that the change of size and the need for reallocation only have minor impact on the overall computation time. The method presented in this paper operates at the global level of a FE-code, not on local level. This has the huge advantage of being independent to the underlying finite element formulation. This allows for an easy implementation of new types of constraints because no modification of existing element types has to be performed, in contrast to the local formulation. The comparison shows that the different methods have different dimensions of the resulting linear system. This, combined with additional computational operations, results in different computational complexity of the methods, see Section 4.4.

4.4 Estimation of computational costs

Compared to other constraint methods, the master-slave elimination relies on heavy manipulation of the residual vector and the tangential stiffness matrix. In order to assess the overall computational costs, the reduction of the dimension also has to be taken into account. In the following, the overall computational cost is estimated. This is done by a computation of the computational complexity (denoted with the Landau symbol $\mathcal{O}$). Here, three computational procedures are considered: First, the calculation of the inverse of the Jacobian related to the slave dofs ${\mathbf{G}_s}^{-1}$ which is used many times in the master-slave elimination scheme, see Eq. (11). Second, the calculation of modified and reduced quantities. Third, the solution of the reduced linear system Eq. (11). For the exact computation of the computational complexity, the use of direct solvers is assumed. The three computational procedures are as follows:

1. Considering the worst case scenario, the inversion of $\mathbf{G}_s$ is of order $\mathcal{O}\left((n_c)^3\right)$, i.e. computationally expensive. However, in many practical cases, the special structure of constraints leads to a reduction of the actual computational costs. Additionally, for systems with very few constraints, the calculation of the inverse ${\mathbf{G}_s}^{-1}$ is computationally inexpensive. If the automatic detection of slave dofs and redundancy analysis discussed in Section 4.2 is used,

the inversion is replaced by Gauß-Jordan elimination. For $n_{\text{dof}} > n_c$, this algorithm is of order $\mathcal{O}\left((n_c)^2(n_{\text{dof}})\right)$. However, in many practical cases, the Jacobian has a large number of zero column associated to dofs that are not involved in any constraints. Hence, the actual computational complexity is smaller.

2. Only the matrix-matrix multiplications are taken into account because other operations are of neglectable costs. Here, the following products are of interest, see Eq. (11):

Product	Computational complexity $\mathcal{O}$
${\mathbf{G}_s}^{-1}\,\mathbf{G}_m$	$(n_c)^2(n_{\text{dof}} - n_c)$
$\mathbf{G}_m^T\mathbf{G}_s^{-T}\mathbf{K}_{ss}$	$(n_c)^3$
$\mathbf{G}_m^T\mathbf{G}_s^{-T}\mathbf{K}_{sm}$	$(n_c)^2(n_{\text{dof}} - n_c)$
$\mathbf{K}_{\text{mod},ms}\,{\mathbf{G}_s}^{-1}\,\mathbf{G}_m$	$(n_c)^2(n_{\text{dof}} - n_c)$

Thus, the total complexity of all matrix-matrix multiplications in the master-slave elimination is $\mathcal{O}\left(\max((n_c)^2(n_{\text{dof}} - n_c), (n_c)^3)\right)$. In the penalty method and for augmented Lagrange multipliers, the only product to consider is $\mathbf{G}^T\,\mathbf{G}$ which is of complexity $\mathcal{O}\left((n_{\text{dof}})^2 n_c\right)$. The actual computational costs are lower in practice due to sparsity of the matrices involved. In addition, there exist efficient, parallelized implementations in libraries like the Math Kernel Library (MKL) by Intel [11].

3. The computational cost of the solution of the effective linear system, see e.g. Eq. (11), depends primarily on its dimension. In the master-slave elimination, the dimension is $(n_{\text{dof}} - n_c)$. Therefore, the computational complexity is $\mathcal{O}\left((n_{\text{dof}} - n_c)^3\right)$.

Two variants of the method have to be distinguished: (a) manual selection of slave dofs, redundancy analysis; (b) automatic selection of slave dofs and redundancy analysis. The computational complexity of the three aspects as well as the overall computational complexity are summarized in Table 1. For primarily mathematical reasons for the exact computation of the total computational complexity, two cases for the number of constraints n_c have to be distinguished for (a) respectively for (b): For (a), case I, $n_c < n_{\text{dof}}/2$, and case II, $n_c \leq n_{\text{dof}}/2$ have to be distinguished. For (b), case III, $n_c < 0.43 \cdot n_{\text{dof}}$, and case IV, $n_c \leq 0.43 \cdot n_{\text{dof}}$ have to be distinguished. For all cases, the overall computational complexity is determined.

Table 1: Overall computational complexity $\mathcal{O}$ of constraint methods for direct solvers

Method	matrix-matrix mult.	${\mathbf{G}_s}^{-1}$	$\mathbf{K}_{\text{eff}}\Delta\mathbf{V} = -\mathbf{R}_{\text{eff}}$	overall
Penalty method	$(n_{\text{dof}})^2 n_c$	–	$(n_{\text{dof}})^3$	$(n_{\text{dof}})^3$
Lagrange mult.	–	–	$(n_{\text{dof}} + n_c)^3$	$(n_{\text{dof}} + n_c)^3$
MS elim. (a), case I, I*	$(n_c)^2(n_{\text{dof}} - n_c)$	$(n_c)^3$	$(n_{\text{dof}} - n_c)^3$	$(n_{\text{dof}} - n_c)^3$
MS elim. (a), case II, II*	$(n_c)^3$	$(n_c)^3$	$(n_{\text{dof}} - n_c)^3$	$(n_c)^3$
MS elim. (b), case III, I*	$(n_c)^2(n_{\text{dof}} - n_c)$	$(n_c)^2(n_{\text{dof}})$	$(n_{\text{dof}} - n_c)^3$	$(n_{\text{dof}} - n_c)^3$
MS elim. (b), case IV, II*	$(n_c)^3$	$(n_c)^2(n_{\text{dof}})$	$(n_{\text{dof}} - n_c)^3$	$(n_c)^2(n_{\text{dof}})$

The master-slave elimination exhibits the lowest computational complexity. This is true for an arbitrary number of constraints n_c and both variants of the method. Both the penalty method as well as the multiplier methods are computationally more complex. For case II and IV, the reduction of the computational complexity in the master-slave elimination compared to other methods is even larger compared to case I and III. The computational complexity gives a good estimate of the actual computational cost. It has to be mentioned that the effective computational cost depends heavily on the specific implementation. For a more practical estimation of the computational costs, two modifications of case I and case II are introduced: Case I*, $n_c \ll n_{\text{dof}}$, and case II*, $n_c = \mathcal{O}(n_{\text{dof}})$. For case I*, as mentioned above, the calculation of the inverse $\mathbf{G}_s{}^{-1}$ is computationally inexpensive which renders the master-slave elimination to be very efficient. For case II*, the master-slave elimination is also more efficient than existing methods since the same arguments hold as for case II. In summary, the master-slave elimination can be estimated to be more efficient than the other methods.

5 EXAMPLE

The aim of the numerical example is to compare the results of the proposed master-slave elimination scheme with existing constraint methods in the regime of a large number of constraints. For this purpose, we analyze a slender plane elastic structure under shear condition (Young's modulus E, Poisson's ratio ν and thickness t), see Fig. 4. It is clamped at the bottom and supported by a sliding clamping at the top. The system is loaded at the top with load F. All geometry and stiffness data are stated in Fig. 4. The middle part of the structure, marked with dark gray in Fig. 4, is subsequently modified to be rigid by introducing constraints. The ratio of the rigid part of the domain is $\alpha = 80\,\%$, see Fig. 4. For the numerical simulation, the system is discretized with 4×40 geometrically nonlinear quadrilateral EAS elements with 4 element nodes and 7 internal variables, see left side of Fig. 4, resulting in a total of $n_{\text{dof}} = 410$ dofs. The load F is applied as a uniform distributed load at the top edge and is increased incrementally in 10 steps.

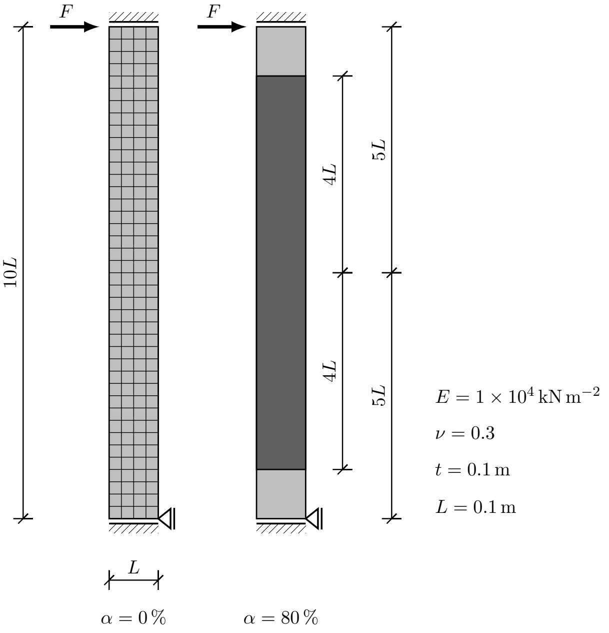


Figure 4: Shear test: Geometry, parameters and mesh

Two variants to model the rigid part of the structured mesh are analyzed: In model A, the rigidity is modeled purely by the nonlinear distance constraints

$$c = \Delta\ell_{ik} = L_{ik} - \ell_{ik} = \|\mathbf{X}_i - \mathbf{X}_k\| - \|\mathbf{x}_i - \mathbf{x}_k\| = 0 \tag{12}$$

preventing a change in length between two nodes i and k, with the initial length L_{ik} and the position vector $\mathbf{X}$ in reference configuration and length ℓ_{ik} and position vector $\mathbf{x}$ in current configuration. Here, the nodes of each edge in the finite element mesh are coupled with distance constraints and, additionally, diagonal distance constraints are considered for the bottom and left element rows of the FE-mesh, which act analogously to a bracing (visualized by red lines at the left side of Fig. 5). The connectivity of the constraints is modeled deliberately in such a way that the constraints are not redundant.In contrast, model B exhibits redundant constraints. Similarly to model A, only nonlinear distance constraints are used. In this model, the constraints are chosen for each element separately. The constraints related to a single element ensure that such an element is rigid. In the total domain, this leads to redundancy, see Fig. 5.

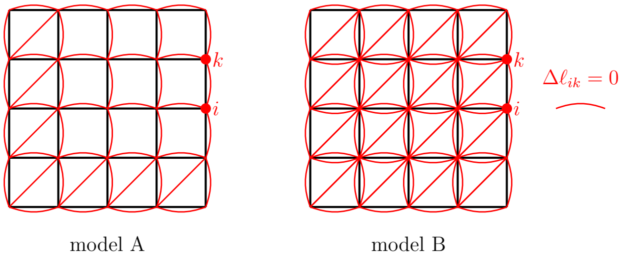


Figure 5: Shear test: Rigid domain modeled solely by non-redundant nonlinear distance constraints (model A) and by redundant nonlinear distance constraints (model B)

It has to be mentioned that the rigid part can be modeled simply by constraints that directly couple the lower edge of the upper elastic part with the upper edge of the lower elastic part. Such a model exhibit fewer number of finite elements and constraints. However, this model is not used here for two reasons: First, even in practice they are reasons to not remove the rigid part, e.g. to enable certain areas of the FE-mesh to be modeled once as rigid and once as elastic during the simulation without the need of re-meshing. Secondly, in this example, the worst case for the master-slave elimination of many fully coupled constraints is examined deliberately.

To verify the implementation of the proposed method as well as the slave dofs identification and redundancy treatment and to examine its accuracy, the load-displacement curve is computed for both models using the master-slave elimination and other constraint methods for model A. For model A, no automatic slave dof identification and redundancy treatment was performed. For model B, the automatic selection of slave dofs as well as an automatic identification and elimination of redundancy is employed. The resulting load-displacement diagram is depicted on the top left of Figure 6. All methods lead to the same results.

To examine the influence of the proposed slave dofs identification and redundancy treatment on the convergence behavior, the Euclidean norm of the residual vector $\|\mathbf{R}\|$ within the iterative process is computed for all models using the master-slave elimination. It is depicted on top right of Figure 6 (representative for 2 of the 10 load steps). For a comparison, a curve representing the optimal convergence rate of $p = 2$ is also shown. In the numerical simulation, the Newton-Raphson scheme is aborted as soon as the relative criterion $\|\mathbf{R}\|^{(k)}/\|\mathbf{R}\|^{(1)} \leq 10^{-12}$ is satisfied (the superscripts indicating the iteration step). The diagram shows that the master-slave elimination exhibits optimal, quadratic convergence.

To study the influence of the proposed slave dof identification and redundancy treatment on the condition number κ of the resulting matrix, the condition number was computed. The results are depicted at the bottom of Figure 6. Here, the strong influence of the slave dof identification and redundancy treatment on the condition number κ can be obtained. For model B, the condition number κ is identical and of the same order than the unconstrained system. The condition number is 2 orders of magnitude smaller in comparison to model A without slave dof identification and redundancy treatment.

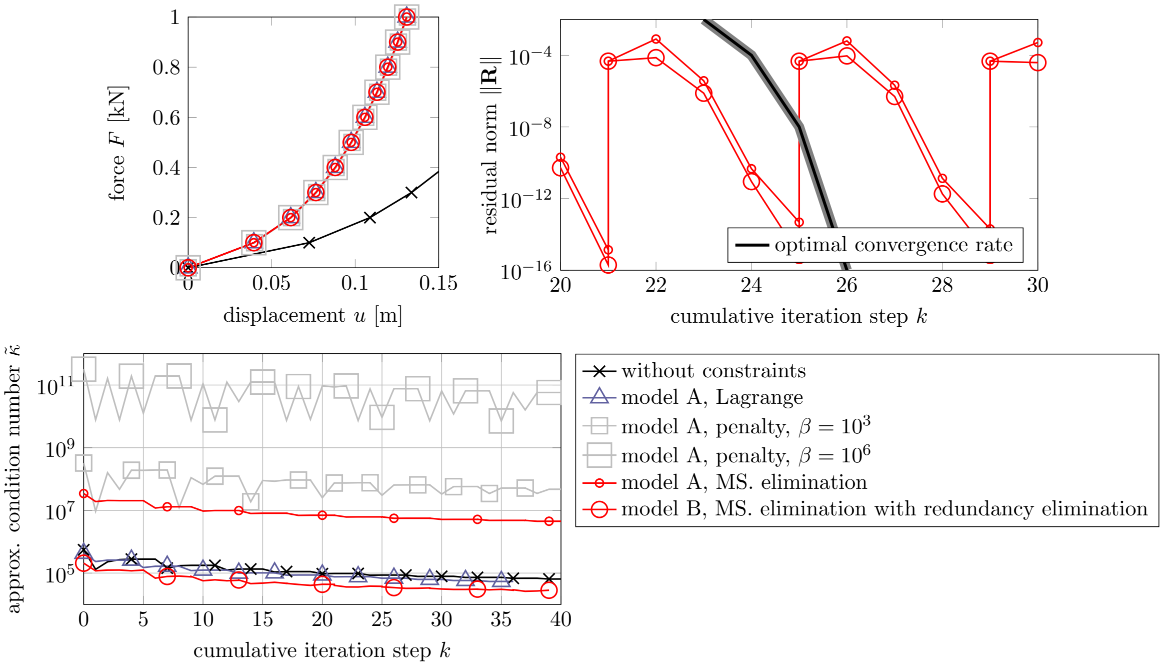


Figure 6: Load-displacement diagrams (top left), residual norm (top right) and condition number of resulting matrix (bottom) (model A and B)

6 CONCLUSIONS

We presented a new master-slave elimination scheme for arbitrary nonlinear multi-point constraints. It satisfies the constraints exactly and leads to a beneficial reduction of the problem dimension. This results in a reduced computational complexity compared to existing constraint methods which makes the method more efficient, especially for a large number of constraints. Additionally, numerical results indicate that the method is as accurate, robust, and flexible as Lagrange multipliers.

REFERENCES

[1] JF Abel and MS Shephard. An algorithm for multipoint constraints in finite element analysis. *International Journal for Numerical Methods in Engineering*, 14(3):464–467, 1979.

[2] M Ainsworth. Essential boundary conditions and multi-point constraints in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6323–6339, 2001.

[3] P Areias, T Rabczuk, D Dias-da Costa, and EB Pires. Implicit solutions with consistent additive and multiplicative components. *Finite Elements in Analysis and Design*, 57:15–31, 2012.

[4] T Belytschko, WK Liu, B Moran, and KI Elkhodary. *Nonlinear finite elements for continua and structures*. Wiley, 2. ed. edition, 2014.

[5] J Boungard and J Wackerfuß. Identification, elimination and handling of redundant non-linear multi-point constraints, in preparation.

[6] J Boungard and J Wackerfuß. Master-slave elimination scheme for arbitrary nonlinear multi-point constraints (published online). *Computational Mechanics*, 2024.

[7] SC Chang and TW Lin. Constraint relation implementation for finite element analysis from an element basis. *Advances in Engineering Software (1978)*, 10(4):191–194, 1988.

[8] JI Curiskis and S Valliappan. A solution algorithm for linear constraint equations in finite element analysis. *Computers & Structures*, 8(1):117–124, 1978.

[9] G Dhondt. *The Finite Element Method for Three–Dimensional Thermomechanical Applications*. Wiley, 2004.

[10] BE Greene. Application of generalized constraints in the stiffness method of structural analysis. *AIAA Journal*, 4(9):1531–1537, 1966.

[11] Intel. *Intel oneAPI Math Kernel Library. Developer Reference*, 2022.

[12] G Jelenić and MA Crisfield. Non-linear 'master-slave' relationships for joints in 3-d beams with large rotations. *Computer Methods in Applied Mechanics and Engineering*, 135(3-4):211–228, 1996.

[13] OS Narayanaswamy. Processing nonlinear multipoint constraints in the finite element method. *International Journal for Numerical Methods in Engineering*, 21(7):1283–1288, 1985.

[14] MS Shephard. Linear multipoint constraints applied via transformation as part of a direct stiffness assembly process. *International Journal for Numerical Methods in Engineering*, 20(11):2107–2112, 1984.

[15] BA Szabo and CT Tsai. The quadratic programming approach to the finite element method. *International Journal for Numerical Methods in Engineering*, 5(3):375–381, 1973.