COPPER MOUNTAIN CONFERENCE - STUDENT PAPER COMPETITION
2015; **00**:1–10

Algebraic multigrid for contact problems in saddle point formulation

T. A. Wiesner[1*,†], M. W. Gee[2] and W. A. Wall[1]

[1] *Institute for Computational Mechanics, Technische Universität München, Boltzmannstr. 15, D-85748 Garching, Germany*

[2] *Mechanics and High Performance Computing Group, Technische Universität München, Parkring 35, D-85748 Garching, Germany*

SUMMARY

In this paper, a full aggregation-based algebraic multigrid strategy is developed for structural contact problems in saddle point formulation. The basic idea is to preserve the saddle point structure on all multigrid levels. This concept of extending multigrid methods to saddle point systems is not new and can be found, e.g., in the context of Stokes and Oseen equations in [8, 20]. So far, there is not so much literature on aggregation-based algebraic multigrid solvers for mortar-based contact problems. The main contribution of this work is the development of an interface aggregation strategy for generating the Lagrange multiplier aggregates that are required for coupling the structural equations with the contact constraints. The proposed method is simpler to implement, computationally less expensive than the ideas from [1], and – in the author's opinion – the presented approach is more intuitive for contact problems. Our interface aggregation strategy perfectly fits into the general multigrid framework and can easily be combined with segregated transfer operators which allow to preserve the saddle point structure on the coarse levels. Numerical experiments show the robustness of the new method.

KEY WORDS: Algebraic Multigrid Methods, Structural Contact problems, Mortar methods, Saddle point problems, Coarsening strategy, Lagrange multipliers, Iterative linear solvers

INTRODUCTION

In the meanwhile, mortar finite element methods are well-established as a basis for state-of-the-art contact formulations that are required in many engineering applications. However, the efficient solution of the arising linear systems turns out to be challenging. This paper proposes an aggregation-based algebraic multigrid method with some extensions specifically designed for contact problems in saddle point formulation. The outline of the paper is the following: Section 1 briefly introduces the problem with a special focus on the algebraic saddle point structure of the linear systems. In section 2 the core components of the multigrid method are explained that are necessary to preserve the saddle point structure throughout all multigrid levels. A new interface coarsening strategy is introduced in section 3, which is then shown to be robust for a large numerical example discussed in section 4.

1. ALGEBRAIC STRUCTURE OF CONTACT PROBLEMS

1.1. Strong formulation of contact problem

Since we are only interested in the algebraic block structure of the final linearized discrete systems, it is sufficient to discuss a quasi-static contact problem with only two deformable bodies to derive all mathematical basics concerning the contact kinematics and the contact constraints. That is, the time parameter $t \in [0, T]$ reduces to a load path parameter. Consider two bounded solid bodies, which are represented by $\Omega_0^{(\mathcal{N}_1)}$, $\Omega_0^{(\mathcal{N}_2)} \subset \mathbb{R}^d$ with $d \in \{2, 3\}$ in the reference configuration (cf. Figure 1). This leads

*Correspondence to: T. A. Wiesner, Sandia National Laboratories, PO Box 5800, MS 1320, Albuquerque, NM 87185-1320

†E-mail: tawiesn@sandia.gov

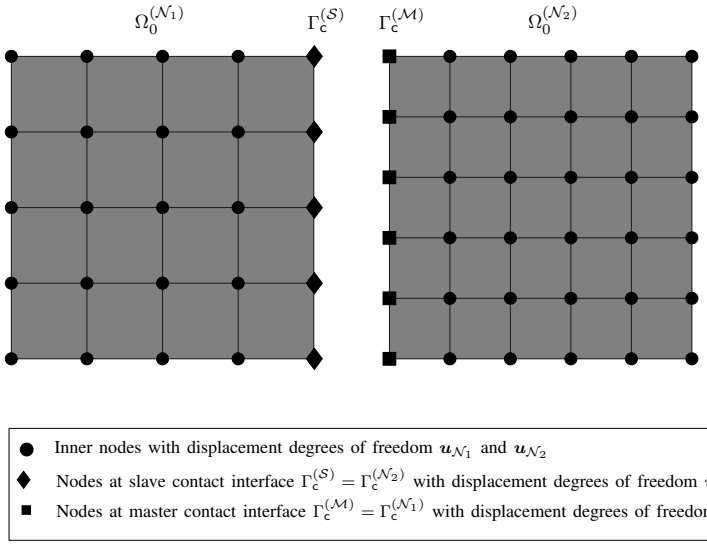


Figure 1. Two solid bodies contact problem in reference configuration.

to the following constrained boundary value problem with $i \in \{1, 2\}$:

$$\operatorname{Div}\boldsymbol{P}^{(\mathcal{N}_i)} + \boldsymbol{f}^{(\mathcal{N}_i)} = \boldsymbol{0} \qquad \text{in } \Omega_0^{(\mathcal{N}_i)} \times [0,T]\,, \tag{1}$$

$$\boldsymbol{u}^{(\mathcal{N}_i)} = \widehat{\boldsymbol{u}}^{(\mathcal{N}_i)} \qquad \text{on } \Gamma_{\mathsf{D}}^{(\mathcal{N}_i)} \times [0,T]\,, \tag{2a}$$

$$\boldsymbol{P}^{(\mathcal{N}_i)}\boldsymbol{N}^{(\mathcal{N}_i)} = \widehat{\boldsymbol{t}}_0^{(\mathcal{N}_i)} \qquad \text{on } \Gamma_{\mathsf{N}}^{(\mathcal{N}_i)} \times [0,T]\,. \tag{2b}$$

$$\text{subject to the constraints} \quad g_{\mathrm{n}}\big(\boldsymbol{x}^{(\mathcal{N}_1)}, t\big) \geq 0 \qquad \text{on } \gamma_{\mathsf{c}}^{(\mathcal{N}_1)} \times [0,T]\,, \tag{3a}$$

$$p_{\mathrm{n}}\big(\boldsymbol{x}^{(\mathcal{N}_1)}, t\big) \leq 0 \qquad \text{on } \gamma_{\mathsf{c}}^{(\mathcal{N}_1)} \times [0,T]\,, \tag{3b}$$

$$p_{\mathrm{n}}\big(\boldsymbol{x}^{(\mathcal{N}_1)}, t\big) g_{\mathrm{n}}\big(\boldsymbol{x}^{(\mathcal{N}_1)}, t\big) = 0 \qquad \text{on } \gamma_{\mathsf{c}}^{(\mathcal{N}_1)} \times [0,T]\,. \tag{3c}$$

Equations (1) to (2b) describe the momentum balance equations with the Dirichlet and Neumann boundary conditions. Herein, $\boldsymbol{P}$ is the first Piola–Kirchhoff stress tensor and $\boldsymbol{f}$, $\widehat{\boldsymbol{t}}_0$ and $\widehat{\boldsymbol{u}}$ represent body forces, known tractions at the Neumann boundary Γ_{N} and prescribed displacements at the Dirichlet boundary Γ_{D} in the reference configuration, respectively. Furthermore, $\boldsymbol{N}^{(\mathcal{N}_i)}$ denotes the outward unit normal vector on $\Gamma_{\mathsf{N}}^{(\mathcal{N}_i)}$. In Figure 1, the Γ_{c} represents the contact boundaries at the contact interface in the reference configuration. In our notation, the slave contact interface $\Gamma_{\mathsf{c}}^{(\mathcal{S})}$ and the master contact interface $\Gamma_{\mathsf{c}}^{(\mathcal{M})}$ are associated with the solid bodies represented by $\Omega_0^{(\mathcal{N}_1)}$ and $\Omega_0^{(\mathcal{N}_2)}$, respectively. The inequalities (3a) to (3c) model the normal contact constraints. Therein, g_{n} defines a so-called gap function which measures the distance of a point $\boldsymbol{x}^{(\mathcal{N}_1)}$ on the slave interface $\gamma_{\mathsf{c}}^{(\mathcal{N}_1)}$ to the projected corresponding point on the master side of the contact interface in the current configuration. In the mathematical formulation one introduces the negative slave side contact traction $\boldsymbol{t}_{\mathsf{c}}^{(\mathcal{N}_1)}$ as Lagrange multiplier, i.e., $\boldsymbol{\lambda} = -\boldsymbol{t}_{\mathsf{c}}^{(\mathcal{N}_1)}$. Therefore, the normal part of the contact stress can be denoted by $\lambda_{\mathrm{n}} := \boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{n}$ and the tangential part by $\boldsymbol{\lambda}_\tau := \boldsymbol{\lambda} - \lambda_{\mathrm{n}}\boldsymbol{n}$ in the following. Here, $\boldsymbol{n}$ denotes the outward unit normal vector in the current configuration. Isoparametric finite elements with first-order and second-order Lagrange interpolation are employed for discretization. The Lagrange multiplier space inherits the $(d-1)$-dimensional mesh emerging from a restriction of the d-dimensional triangulation $\mathcal{T}_{\Omega_0}^{\mathrm{h}}$ of Ω_0 to the slave side of the contact interface. For the sake of brevity the details of the discretization, contact search and nonlinear solution methods are skipped. The interested reader might refer to, e.g., [16, 26, 44].

1.2. Algebraic formulation

The final linear systems arising within a nonlinear iterative solver for the discrete contact problems have the following algebraic block structure:

DOI: unknown/cpm

$$\left(\begin{array}{ccccc:cc} \mathsf{K}_{\mathcal{N}_1\mathcal{N}_1} & \mathsf{K}_{\mathcal{N}_1\mathcal{I}} & \mathsf{K}_{\mathcal{N}_1\mathcal{A}} & 0 & 0 & 0 & 0 \\ \mathsf{K}_{\mathcal{I}\mathcal{N}_1} & \mathsf{K}_{\mathcal{I}\mathcal{I}} & \mathsf{K}_{\mathcal{I}\mathcal{A}} & \mathsf{K}_{\mathcal{I}\mathcal{M}} & 0 & \mathsf{D}_{\mathcal{I}\mathcal{I}}^{\mathsf{T}} & \mathsf{D}_{\mathcal{I}\mathcal{A}}^{\mathsf{T}} \\ \mathsf{K}_{\mathcal{A}\mathcal{N}_1} & \mathsf{K}_{\mathcal{A}\mathcal{I}} & \mathsf{K}_{\mathcal{A}\mathcal{A}} & \mathsf{K}_{\mathcal{A}\mathcal{M}} & 0 & \mathsf{D}_{\mathcal{A}\mathcal{I}}^{\mathsf{T}} & \mathsf{D}_{\mathcal{A}\mathcal{A}}^{\mathsf{T}} \\ 0 & \mathsf{K}_{\mathcal{M}\mathcal{I}} & \mathsf{K}_{\mathcal{M}\mathcal{A}} & \mathsf{K}_{\mathcal{M}\mathcal{M}} & \mathsf{K}_{\mathcal{M}\mathcal{N}_2} & -\mathsf{M}_{\mathcal{I}}^{\mathsf{T}} & -\mathsf{M}_{\mathcal{A}}^{\mathsf{T}} \\ 0 & 0 & 0 & \mathsf{K}_{\mathcal{N}_2\mathcal{M}} & \mathsf{K}_{\mathcal{N}_2\mathcal{N}_2} & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & \mathsf{I} & 0 \\ 0 & \mathsf{N}_{\mathcal{I}} & \mathsf{N}_{\mathcal{A}} & \mathsf{N}_{\mathcal{M}} & 0 & 0 & 0 \\ 0 & \mathsf{F}_{\mathcal{I}} & \mathsf{F}_{\mathcal{A}} & 0 & 0 & 0 & \mathsf{T}_{\mathcal{A}} \end{array}\right) \left(\begin{array}{c} \Delta\boldsymbol{u}_{\mathcal{N}_1} \\ \Delta\boldsymbol{u}_{\mathcal{I}} \\ \Delta\boldsymbol{u}_{\mathcal{A}} \\ \Delta\boldsymbol{u}_{\mathcal{M}} \\ \Delta\boldsymbol{u}_{\mathcal{N}_2} \\ \hdashline \Delta\boldsymbol{\lambda}_{\mathcal{I}} \\ \Delta\boldsymbol{\lambda}_{\mathcal{A}} \end{array}\right) = - \left(\begin{array}{c} \mathbf{r}_{\mathcal{N}_1}^{\boldsymbol{u}} \\ \mathbf{r}_{\mathcal{I}}^{\boldsymbol{u}} \\ \mathbf{r}_{\mathcal{A}}^{\boldsymbol{u}} \\ \mathbf{r}_{\mathcal{M}}^{\boldsymbol{u}} \\ \mathbf{r}_{\mathcal{N}_2}^{\boldsymbol{u}} \\ \hdashline \mathbf{r}_{\mathcal{I}}^{\boldsymbol{\lambda}} \\ \mathbf{r}_{\mathcal{A}}^{\boldsymbol{\lambda},\mathrm{n}} \\ \mathbf{r}_{\mathcal{A}}^{\boldsymbol{\lambda},\tau} \end{array}\right). \tag{4}$$

The subscripts $\mathcal{N}_1$ and $\mathcal{N}_2$ denote the inner nodes of the two solid bodies away from the contact interface. The subscript $\mathcal{M}$ represents the nodes at the master interface. The nodes at the slave interface are split into the set of inactive nodes $\mathcal{I}$ and active nodes $\mathcal{A}$. Here, *inactive* means, that the nodes are currently not in contact. The set of inactive and active nodes is dynamically determined by a contact search algorithm (cf. [26]). The solution vector contains displacement increments $\Delta\boldsymbol{u}$ and Lagrange multiplier increments $\Delta\boldsymbol{\lambda}$ associated with all nodes at the slave interface. In rows 7 and 8, the matrix in (4) contains the (linearized) contact constraints in normal and tangential direction relative to the current contact interface. Row 6 represents the inactive nodes. The upper right block in the matrix couples the structural equations and the contact constraints using the mortar matrices D and M. For a detailed description and derivation the reader is referred to [26–29, 44]. Equation (4) perfectly fits into the definition of a generalized saddle point problem with the typical block structure

$$\begin{pmatrix} \mathsf{K} & \mathsf{C}_1^{\mathsf{T}} \\ \mathsf{C}_2 & -\mathsf{Z} \end{pmatrix} \begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}^{\boldsymbol{u}} \\ \mathbf{r}^{\boldsymbol{\lambda}} \end{bmatrix}. \tag{5}$$

Linear systems with saddle point structure make special preconditioning and solving strategies necessary, which are aware of the special block structure of the underlying system matrix. The interested reader might refer to [5] as a starting point to dive into the theory of saddle point problems with an elaborate classification and a general overview of solving strategies for saddle point problems. There is a wide variety of classes of (block) preconditioning techniques for saddle point problems (see, e.g., [2, 5, 6, 11, 13–15, 23–25, 31, 35, 40, 45, 49]). This list is by far not complete.

1.3. Multigrid for saddle point problems

Multigrid methods are known to be among the most efficient preconditioning or solution methods for certain classes of linear systems. This paper presents a multigrid strategy for the class of saddle point matrices arising from mortar contact problems as introduced in the previous sections. The proposed multigrid method is based on the (smoothed) aggregation algebraic multigrid algorithms (cf. [30, 36–39, 44]) with special extensions for block matrices and some minor contact-specific adaptions. In contrary to geometric multigrid methods, algebraic multigrid methods (e.g., [32]) have the advantage that they do not rely on geometric user-provided mesh information, but use only algebraic information from the provided matrix in (5) as input. The class of aggregation-based algebraic multigrid methods uses so-called aggregates to coarsen the fine level problem. Each aggregate is an agglomeration of neighboring fine-level nodes which represents one virtual node on the next coarser level. The aggregation information is then used to build the transfer operators (prolongator P and restrictor R) which are used to recursively coarsen the fine level problem. A detailed introduction to multigrid methods is given in different textbooks (e.g., [12, 19, 33]).

In general, there are two distinct ways to apply multigrid ideas in context of saddle point problems:

Nested multigrid approach: Multigrid methods can serve as local single field smoothers or solvers within well-known Schur complement based block preconditioners such as the SIMPLE method (cf. [25]) and variants. The coupling of the primary variables and the constraint variables is only considered on the finest level in the outer (SIMPLE) iteration. This approach is already known in the literature and, e.g., described by [18] and [32] for the Navier–Stokes equations. The implementation is very easy and allows to use the multigrid components in a standalone fashion within the Schur complement solver.

Full multigrid approach: A truly monolithic algebraic multigrid method for saddle point problems is presented in the following. It is specifically designed for problems arising from structural contact problems but the general concept applies to other saddle point problems, too. The idea is to keep the saddle point structure on all multigrid levels such that the contact constraints are considered on all levels. We apply Schur complement based level smoothers which have the advantage that the contact constraints are sufficiently considered on all levels.

For the nested approach the existing standard theory for saddle point problems as well as multigrid methods applies in a straightforward way. In contrast, for the full multigrid approach the multigrid hierarchy may suffer from stability issues. Thus, the stability of the discretization scheme plays an important role for building coarse representations of the fine level problem in context of multigrid methods. In the work

by Wabro [41] a coupled AMG method is developed and analyzed for a stabilized mixed finite element discretization of the Oseen equations, which utilizes special techniques to preserve stability on the coarser levels by scaling the standard Galerkin product. However, at least for the Stokes problem, extensive numerical studies in [20] reveal that uniform inf–sup stability (cf. [3, 22]) of the coarse level operators is not necessary for obtaining a successful multigrid preconditioner (see also, e.g., statements on successful preconditioners in [32]).

Special multigrid methods have been developed for mortar finite element problems in saddle point formulation. The multigrid theory for this particular class of saddle point problems has evolved starting from special multigrid methods for mortar finite element methods (e.g., [7, 17]) to mortar finite element methods in saddle point formulation (e.g., [9, 10]). Recent works on specific multigrid methods for structural contact problems in saddle point formulation are based on the ideas from the mortar finite element method and introduce a very flexible multigrid framework that can be applied to a wide variety of different mortar situations including structural contact problems (e.g., [42, 43]).

Most of the literature available on multigrid for contact problems is primarily on geometric multigrid methods with abundant work on saddle point smoothers (cf. [46, 48]). However, in this paper, the usage of a smoothed aggregation algebraic multigrid method is proposed for the contact problems based on the mortar discretization.

2. FULL MULTIGRID APPROACH FOR CONTACT PROBLEMS

The basic idea of the full multigrid approach is to preserve the saddle point structure on all multigrid levels by using appropriate transfer operators. This means, that one has to deal with coarse saddle point problems on each level which makes appropriate level smoothers necessary.

2.1. Block smoothers

General block smoothers can be expressed by

$$\begin{bmatrix}\Delta \boldsymbol{u}^{k+1}\\ \Delta \boldsymbol{\lambda}^{k+1}\end{bmatrix} = \begin{bmatrix}\Delta \boldsymbol{u}^{k}\\ \Delta \boldsymbol{\lambda}^{k}\end{bmatrix} + Q^{-1}\left(\begin{bmatrix}\mathbf{r}_{\boldsymbol{u}}^{k}\\ \mathbf{r}_{\boldsymbol{\lambda}}^{k}\end{bmatrix} - \begin{pmatrix}\mathsf{K} & \mathsf{C}_1^{\mathsf{T}}\\ \mathsf{C}_2 & -\mathsf{Z}\end{pmatrix}\begin{bmatrix}\Delta \boldsymbol{u}^{k}\\ \Delta \boldsymbol{\lambda}^{k}\end{bmatrix}\right), \tag{6}$$

where Q describes the 2×2 block preconditioning matrix approximating the 2×2 block operator in (5). In structural contact simulations the interesting but challenging part is the coupling of the different solid blocks at the contact interface. Mathematically, the contact problem is governed by the contact constraint equations. Since the coupling of the structural blocks takes only place in the level smoother, constraint smoothers (cf. [21]) are preferred, which put some special focus on the consideration of the constraint equations. Guided by ideas of the SIMPLE preconditioner (e.g., [25]) the Q is chosen as

$$Q := \begin{pmatrix}\mathsf{K} & 0\\ \mathsf{C}_2 & -\widetilde{S}\end{pmatrix}\begin{pmatrix}I & \widetilde{\mathsf{K}}^{-1}\mathsf{C}_1^{\mathsf{T}}\\ 0 & \frac{1}{\alpha}I\end{pmatrix} = \begin{pmatrix}\mathsf{K} & \mathsf{K}\widetilde{\mathsf{K}}^{-1}\mathsf{C}_1^{\mathsf{T}}\\ \mathsf{C}_2 & \left(1-\frac{1}{\alpha}\right)\mathsf{C}_2\widetilde{\mathsf{K}}^{-1}\mathsf{C}_1^{\mathsf{T}} - \frac{1}{\alpha}\mathsf{Z}\end{pmatrix} \approx \begin{pmatrix}\mathsf{K} & \mathsf{C}_1^{\mathsf{T}}\\ \mathsf{C}_2 & -\mathsf{Z}\end{pmatrix} \tag{7}$$

with $\alpha > 0$. In (7) the $\widetilde{S}$ denotes an approximation of the Schur complement $S := \mathsf{Z} + \mathsf{C}_2\mathsf{K}^{-1}\mathsf{C}_1^{\mathsf{T}}$ with a cheap and easy-to-invert approximation $\widetilde{\mathsf{K}}$ of the block K.

With $\widetilde{S} = \alpha\mathsf{Z} + \alpha\mathsf{C}_2\widetilde{\mathsf{K}}^{-1}\mathsf{C}_1^{\mathsf{T}}$ properly scaled, the error matrix of the preconditioner reduces to

$$\mathsf{E}_{\text{SIMPLE}} = A - Q = \begin{pmatrix}0 & \mathsf{C}_1^{\mathsf{T}} - \mathsf{K}\widetilde{\mathsf{K}}^{-1}\mathsf{C}_1^{\mathsf{T}}\\ 0 & 0\end{pmatrix}. \tag{8}$$

The zero row in (8) shows that an appropriate approximation of the Schur complement S allows to exactly satisfy the contact constraints within one smoothing sweep. Therefore, in our case, the preferred choice are the block approximate smoothers such as the SIMPLE-based methods.

Variants of the SIMPLE method like SIMPLEC as introduced by [34] can be understood as an enhancement of the classical SIMPLE method. The general idea is to provide better approximations for the inverse of the block K. Instead of just using the diagonal of K for calculating the approximate inverse of K, the diagonal matrix containing the row sums of $\left|\mathsf{K}\right| = \left(|a_{ij}|\right)_{i,j=1,\dots,n_{\mathsf{K}}}$ is used. That is, $\widetilde{\mathsf{K}}$ is defined as

$$\widetilde{\mathsf{K}} = \operatorname{diag}\left(\sum_{j=1}^{n_{\mathsf{K}}} |a_{ij}|\right), \quad i = 1, \dots, n_{\mathsf{K}}. \tag{9}$$

The default choice for $\widetilde{S}$ is consequently $\widetilde{S} = \alpha\mathsf{Z} + \alpha\mathsf{C}_2\widetilde{\mathsf{K}}^{-1}\mathsf{C}_1^{\mathsf{T}}$ with $\widetilde{\mathsf{K}}$ as defined in (9). So, the method is very similar to the classical SIMPLE method (cf. [14]). To keep the computational costs low, one does not solve for the internal block inverses exactly, but only applies a fixed number of smoothing sweeps with

a relaxation-based method. As a naming convention, the prefix "Cheap" is added to the name of the block smoothing method to indicate the usage of a cheap approximation for finding the inverse of the diagonal blocks. A more theoretical discussion on the mathematical consequences of approximations for the Schur complement S can be found in [48].

2.2. Segregated transfer operators

To keep the characteristic saddle point block structure (5) on all multigrid levels, the common approach is to use *segregated* transfer operators

$$P_{\ell+1} = \begin{pmatrix} P^{\boldsymbol{u}} & 0 \\ 0 & \widehat{P}^{\boldsymbol{\lambda}} \end{pmatrix}_{\ell+1} \quad \text{and} \quad R_{\ell+1} = \begin{pmatrix} R^{\boldsymbol{u}} & 0 \\ 0 & \widehat{R}^{\boldsymbol{\lambda}} \end{pmatrix}_{\ell+1}, \tag{10}$$

as, e.g., introduced in [8] or [1]. The segregated block transfer operators (10) are put together from the transfer operator blocks for the different physical and mathematical fields (e.g., displacement field and Lagrange multipliers). Here, $P^{\boldsymbol{u}}$ and $R^{\boldsymbol{u}}$ describe the transfer operator blocks corresponding to the stiffness matrix block K in (5). The transfer operators $\widehat{P}^{\boldsymbol{\lambda}}$ and $\widehat{R}^{\boldsymbol{\lambda}}$ define the level transfer for the Lagrange multipliers.

The block diagonal structure in (10) guarantees that the primary displacement variables and the secondary Lagrange multipliers are not "mixed up" on the coarser levels. That is, the coarse level matrix still has the same block structure with a clear distinction of momentum and constraint equations as for the fine level problem.

Whereas for many coupled problems (e.g., Fluid-Structure-Interaction problems, Thermo-Structure-Interaction problems) it is straightforward to generate $\widehat{P}^{\boldsymbol{\lambda}}$ and $\widehat{R}^{\boldsymbol{\lambda}}$, in case of structural contact problems this is a non-trivial task, as one cannot use the Z block to generate valid aggregation information for the Lagrange multipliers due to an insufficient pattern of the Z block. Consequently, one needs a special routine for finding aggregates for the Lagrange multipliers $\boldsymbol{\lambda}$ to be able to build the (non-smoothed) transfer operators $\widehat{P}^{\boldsymbol{\lambda}}$ and $\widehat{R}^{\boldsymbol{\lambda}}$. For an interface coupled problem it seems natural to apply an interface aggregation method which extends the aggregation to the contact interface for coarsening the Lagrange multipliers.

3. COARSENING STRATEGY

3.1. Aggregation strategy for the displacement variables

To preserve a meaningful saddle point structure on the coarse levels it is important to keep the two solid blocks separated in the matrix representation on the coarse levels. One can use the standard aggregation strategy with a modified K block from (5) as input to build the displacement aggregates $\mathscr{A}_\ell{}^{\boldsymbol{u}}$ which do not cross the contact interface (see Figure 2). This way the two solid blocks are not "melted" together in the coarse level matrices. In an efficient implementation the modified matrix is not built explicitly, but the off-diagonal entries are dropped on the fly.

3.2. Aggregation strategy for the Lagrange multipliers

In contrast to geometric multigrid methods, there is not so much literature on aggregation-based AMG methods for contact problems in saddle point formulation. The only publication, the author is aware of covering all aspects of smoothed aggregation methods for structural contact problems in saddle point formulation, is Adams in [1], which also discusses a special aggregation strategy for the Lagrange multipliers. To find aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$ for the Lagrange multipliers, Adams proposes to apply the standard aggregation algorithm to the graph of a suitable matrix representing the Lagrange multipliers. However, this approach has some drawbacks: First, the graph used for the aggregation of the Lagrange multipliers $\boldsymbol{\lambda}$ has to be built explicitly to serve as input for the standard aggregation algorithm. Secondly, one has to run the aggregation algorithm sequentially both for the displacement degrees of freedom and for the Lagrange multipliers. For the second run of the aggregation method one might have to use a different set of aggregation parameters to obtain optimal results, which further increases the complexity for the user. Algorithmically, the resulting aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$ for the Lagrange multipliers are built independently from the displacement aggregates $\mathscr{A}_\ell{}^{\boldsymbol{u}}$.

In this work, a different approach is proposed to build aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$ for the Lagrange multipliers. Instead of explicitly building some helper matrix for the aggregation routine, interface aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$ for the Lagrange multipliers are directly generated using the aggregation information for the displacement variables (see Figure 2). The resulting interface aggregates for the Lagrange multipliers are by construction consistent with the corresponding displacement aggregates.

The exact aggregation procedure is described in Algorithm 1. Assuming that the standard aggregates $\mathscr{A}_\ell{}^{\boldsymbol{u}}$ for the displacement degrees of freedom are available, new aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$ are built by collecting the corresponding Lagrange multiplier degrees of freedom relative to the displacement aggregates at the slave contact interface. Beside the displacement aggregates $\mathscr{A}_\ell{}^{\boldsymbol{u}}$ only the mortar matrix D is needed

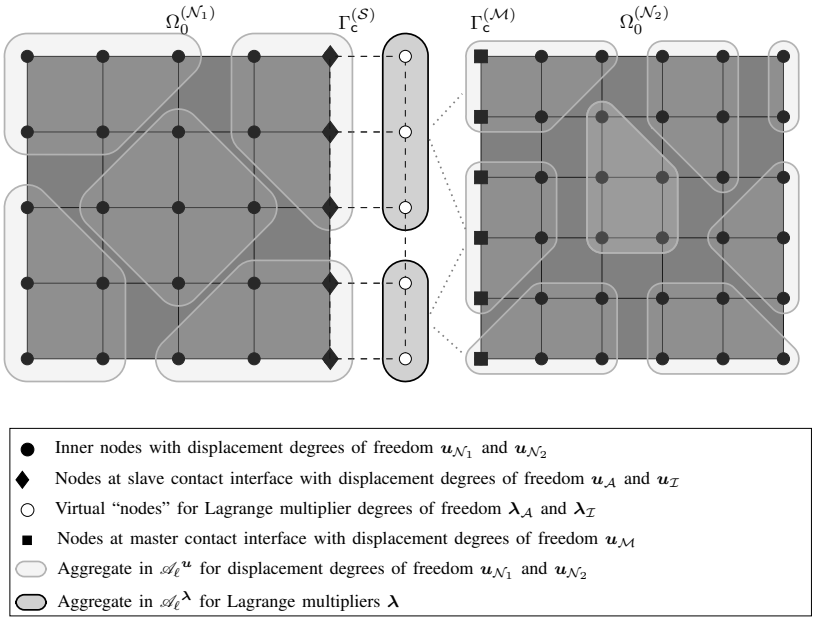


Figure 2. Aggregation for contact problems in saddle point formulation

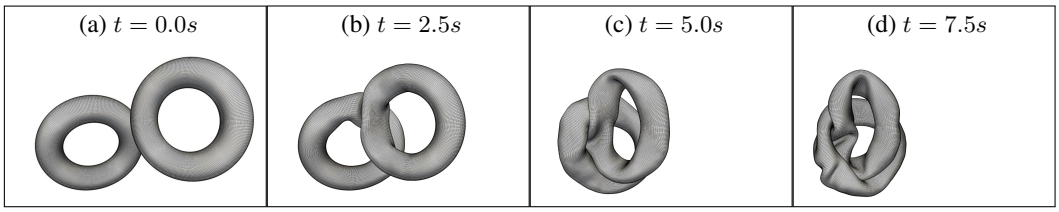


Figure 3. Two tori impact example – Characteristic stages of deformation.

to algebraically reconstruct the contact interface and find the associated Lagrange multipliers. The new aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$ for the Lagrange multipliers can be interpreted as the natural extension of the displacement aggregates $\mathscr{A}_\ell{}^{\boldsymbol{u}}$ at the interface. This way, one can keep the ratio of coarse level nodes at the slave contact interface and the coarse Lagrange multipliers constant, which also balances the ratio of contact constraints and inner structural displacement degrees of freedoms over all multigrid levels.

4. TWO TORI IMPACT EXAMPLE

Inspired by some similar analysis in [47] the problem setup of the two tori impact example with geometry and load conditions from [26] is used. There are two thin-walled tori with a Neo-Hookean material model ($E = 2250$, $\nu = 0.3$, $\rho_0 = 0.1$) with a major and minor radius of 76 and 24 units and a wall thickness of 4.5 units. The right torus in Figure 3 lies in the xy-plane and is accelerated by a body force towards the left torus, which is rotated around the y-axis by 45 degrees. The simulation runs 200 time steps with a time step size of $0.05s$ using a generalized-α time integration scheme. The chosen finite element mesh consists of 284672 first-order hexahedral elements with 350208 nodes. All the simulations have been run on 16 cores (spread over 2 Intel Xeon E5-2670 Octocore CPUs). With the rather complex geometry and contact configuration, that heavily changes over time, this example can be considered as a representative test for the robustness and efficiency of the tested numerical methods.

The nonlinear stopping criteria for the internal (non-smooth) Newton method (see,e.g., citepopp2012diss) are chosen as

$$\|\Delta \boldsymbol{u}\|_e < 10^{-7} \ \wedge \ \left(\left\| \frac{\mathbf{r}_i^{\boldsymbol{u}}}{\mathbf{r}_0^{\boldsymbol{u}}} \right\|_e < 10^{-8} \ \wedge \ \left\| \mathbf{r}_i^{\boldsymbol{\lambda}} \right\|_e < 10^{-4} \right). \tag{11}$$

Algorithm 1: Aggregation algorithm for Lagrange multipliers.

Procedure `LagMultAggregation(`$\mathscr{A}_\ell{}^{\boldsymbol{u}}$,$\mathsf{D}$`)`

Initialize empty set and counter for aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$
$\mathscr{A}_\ell{}^{\boldsymbol{\lambda}} \leftarrow \emptyset,\ l \leftarrow 0$

Initialize empty mapping of displacement aggregates to Lagrange multiplier aggregates
$\mathsf{d}(k) \leftarrow \emptyset \quad \forall k = 1, \ldots, m_{\mathscr{A}_\ell{}^{\boldsymbol{u}}}$

Loop over slave DOFs (rows of D*)*
for $i \in \mathcal{D}_{\mathcal{S}}$ **do**

Find displacement node $n^{\boldsymbol{u}}$ *id corresponding to displacement DOF* i
$n^{\boldsymbol{u}} \leftarrow \mathrm{n}(i)$

Find aggregate index k *that contains displacement node* $n^{\boldsymbol{u}}$
Find k with $\mathscr{A}_\ell{}^{(k)} \in \mathscr{A}_\ell{}^{\boldsymbol{u}}$ where $n^{\boldsymbol{u}} \in \mathscr{A}_\ell{}^{(k)}$

Loop over all Lagrange multipliers j
for $j \in \mathcal{D}_{\boldsymbol{\lambda}}$ **do**

Check whether Lagrange multiplier j *is coupled with row* i
if $\mathsf{D}_{i,j} \neq 0$ **then**

Find pseudo node n^λ *for Lagrange multiplier* j
$n^\lambda \leftarrow \mathrm{n}(j)$

Check whether to build a new Lagrange multiplier aggregate
if $\mathsf{d}(k) = \emptyset$ **then**

Increment internal aggregation counter
$l \leftarrow l + 1$

Build a new aggregate and add Lagrange multiplier node n^λ
$\mathscr{A}_\ell{}^{(l)} \leftarrow \{n^\lambda\}$

Associate displacement aggregate k *with Lagrange multiplier aggregate* l
$\mathsf{d}(k) \leftarrow \{l\}$

Add new aggregate to set of Lagrange multiplier aggregates $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$
$\mathscr{A}_\ell{}^{\boldsymbol{\lambda}} \leftarrow \mathscr{A}_\ell{}^{\boldsymbol{\lambda}} \cup \mathscr{A}_\ell{}^{(l)}$

else

Extend aggregate $0 \leq \mathsf{d}(k) \leq l$ *with pseudo node*
$\mathscr{A}_\ell{}^{(\mathsf{d}(k))} \leftarrow \mathscr{A}_\ell{}^{(\mathsf{d}(k))} \cup \{n^\lambda\}$

end

end

end

end

Return aggregates for Lagrange multipliers
return $\mathscr{A}_\ell{}^{\boldsymbol{\lambda}}$

Here, $\mathbf{r}_i^{\boldsymbol{u}}$ and $\mathbf{r}_i^{\boldsymbol{\lambda}}$ denotes the (nonlinear) residual for the displacement variables and Lagrange multipliers in the i-th Newton iteration. The quantity $\Delta\boldsymbol{u}$ describes the solution increment for the displacement variables in the current Newton iteration.

Within each Newton step, a preconditioned GMRES solver is used with the convergence criterion

$$\left\| \frac{\mathbf{r}^k}{\mathbf{r}^0} \right\|_e < 10^{-8} \tag{12}$$

for the full residual vector $\mathbf{r}^k = \begin{bmatrix} \mathbf{r}^{\boldsymbol{u}} \\ \mathbf{r}^{\boldsymbol{\lambda}} \end{bmatrix}$ in the linear iteration step k. Table I gives an overview of the different tested preconditioner variants. It includes variants with the full multigrid approach, the nested multigrid approach and a SIMPLE based variant without multigrid at all. For the full AMG variants the transfer operators for the displacement blocks are varied. Particularly, non-smoothed transfer operators (PA-AMG) are compared with smoothed aggregation transfer operators (SA-AMG).

Figure 4a shows the accumulated number of iterations for solving all linear systems within the Newton method in one time step. Obviously, the SIMPLE based methods need more linear iterations than the AMG

Full multigrid based methods	SIMPLE based methods
PA-AMG (CheapSIMPLE) Transfer operators: PA-AMG Level smoother: 1 CheapSIMPLEC(0.8) – Pred. smoother: 1 SGS (0.8) – Corr. smoother: ILU (0)	**CheapSIMPLE (SGS)** Transfer operators: – Block prec.: 2 CheapSIMPLEC(0.8) – Pred. smoother: 3 SGS (0.8) – Corr. smoother: ILU (0)
SA-AMG (CheapSIMPLE) Transfer operators: SA-AMG (0.4) Level smoother: 1 CheapSIMPLEC(0.8) – Pred. smoother: 1 SGS (0.8) – Corr. smoother: ILU (0)	**CheapSIMPLE (SA-AMG)** Block prec.: 2 CheapSIMPLEC(0.8) – Pred. smoother: AMG – Transfer op.: SA-AMG (0.4) – Level sm.: 2 SGS (0.8) – Corr. smoother: ILU (0)

Table I. Two tori impact example – Different AMG variants.

based methods. In this example there is nearly no difference between the non-smoothed transfer operator variant PA-AMG (CheapSIMPLE) and the smoothed transfer operator variant SA-AMG (CheapSIMPLE). Furthermore, there is no clear and obvious correlation between the number of linear iterations and the number of active nodes. This shows, that the full multigrid method is robust and efficient with regard to the increasing complexity of the contact configuration over time. This is in contrast to cheaper mehtods such as the SIMPLE based methods where one can see a significant drop in the linear iterations for the last time steps in the simulation, which may correspond to the small number of nodes in contact.

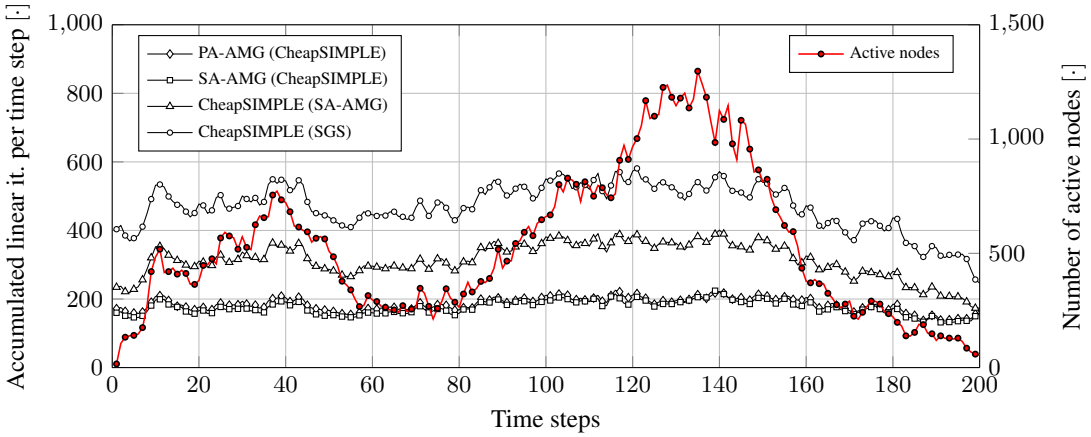
When looking at the corresponding solver timings over the time steps in Figure 4b, one finds the CheapSIMPLE (SA-AMG) method to be very close to the AMG based methods PA-AMG (CheapSIMPLE) and SA-AMG (CheapSIMPLE). For the AMG based methods one sweep with a CheapSIMPLEC method is applied on each level, which internally uses 1 sweep with a symmetric Gauss–Seidel iteration for the primary variable and 1 ILU sweep for the constraint equation. This way, quite a lot of time is invested in the coupling on all levels with the comparably expensive ILU method. In contrary to the AMG based method, the CheapSIMPLE (SA-AMG) method uses 2 sweeps with a CheapSIMPLE preconditioner for the coupling (on the finest level only). Internally, a 3 level AMG multigrid is used with 2 symmetric Gauss–Seidel sweeps for the level smoother and an ILU sweep for the constraint correction equation. These parameters have been found to result in a reasonably low number of linear iterations. For this example the experiment shows that the CheapSIMPLE (SA-AMG) method needs twice as many iterations as the SA-AMG (CheapSIMPLE) method, but the costs per iteration are only half of the costs of the SA-AMG (CheapSIMPLE). Nevertheless, the AMG based methods seem to have a small advantage, when the number of nodes in contact increases.

5. CONCLUSION

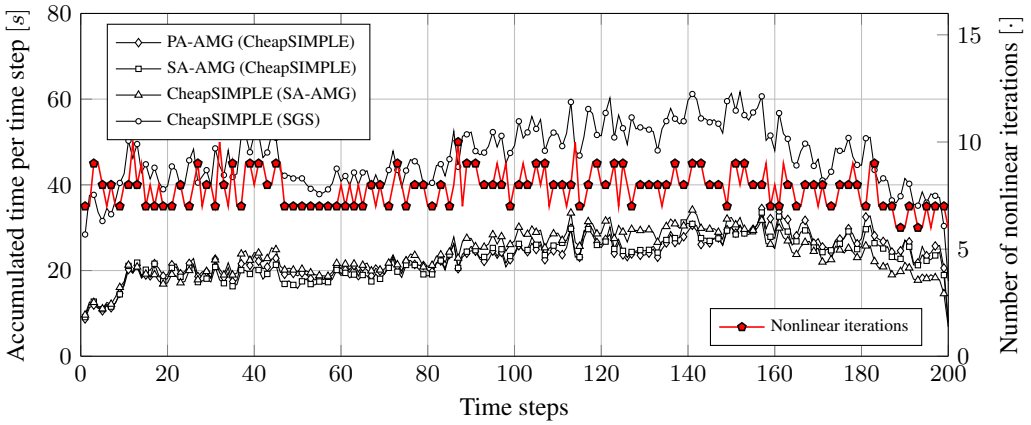
In this paper, we describe a full multigrid scheme designed for saddle point problems arising from contact problems using mortar coupling blocks. The full multigrid scheme has the advantage that the contact constraints are considered on all multigrid levels which significantly reduces the number of iterations. It gives the user full control over the coupling process by appropriately choosing the smoother parameters which might be useful for complex large contact problems in industrial applications. Additionally, we propose a novel aggregation method for the Lagrange multipliers which reuses existing aggregation information at the contact interface. In our opinion this is a more intuitive approach for contact interface problems than the method presented in [1]. We demonstrate the robustness and efficiency of the overall multigrid method for a large example with increasingly complex contact configurations over time.

REFERENCES

[1] M. F. Adams. Algebraic multrigrid methods for constrained linear systems with applications to contact problems in solid mechanics. *Numerical Linear Algebra with Applications*, 11(2-3):141–153, 2004.
[2] Owe Axelsson and Maya Neytcheva. A general approach to analyse preconditioners for two-by-two block matrices. *Numerical Linear Algebra with Applications*, 20(5):723–742, 2013.
[3] Ivo Babuška. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.
[4] Faker Ben Belgacem. The mortar finite element method with Lagrange multipliers. *Numerische Mathematik*, 84(2):173–197, 1999.
[5] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
[6] Michele Benzi and Valeria Simoncini. On the eigenvalues of a class of saddle point matrices. *Numerische Mathematik*, 103(2):173–196, 2006.



(a) Accumulated number of linear GMRES iterations for all nonlinear iterations per timestep.



(b) Accumulated timings for the solution phase for all nonlinear iterations per timestep.

Figure 4. Two tori impact example – Results for different saddle point preconditioner variants.

[7] D. Braess, M. Dryja, and W. Hackbush. A multigrid method for nonconforming FE-discretisations with application to non-matching grids. *Computing*, 63(1):1–25, 1999.
[8] D. Braess and R. Sarazin. An efficient smoother for the stokes problem. *Applied Numerical Mathematics*, 23:3–19, 1997.
[9] Dietrich Braess and Wolfgang Dahmen. Stability estimates of the mortar finite element method for 3-dimensional problems. *East-West Journal of Numerical Mathematics*, 6:249–264, 1998.
[10] Dietrich Braess, Wolfgang Dahmen, and Christian Wieners. A multigrid algorithm for the mortar finite element method. *SIAM Journal on Numerical Analysis*, 37(1):pp. 48–69, 2000.
[11] James H. Bramble, Joseph E. Pasciak, and Apostol T. Vassilev. Analysis of the inexact uzawa algorithm for saddle point problems. *SIAM Journal on Numerical Analysis*, 34(3):pp. 1072–1092, 1997.
[12] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, second edition edition, 2000.
[13] E. de Sturler and J. Liesen. Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. part i: Theory. *SIAM Journal on Scientific Computing*, 26(5):1598–1619, 2005.
[14] Howard Elman, V. E. Howle, John Shadid, Robert Shuttleworth, and Ray Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 227(3):1790–1808, January 2008.
[15] Howard C. Elman and Gene H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM Journal on Numerical Analysis*, 31(6):pp. 1645–1661, 1994.
[16] Markus Gitterle, Alexander Popp, Michael W. Gee, and Wolfgang A. Wall. Finite deformation frictional mortar contact using a semi-smooth Newton method with consistent linearization. *International Journal for Numerical Methods in Engineering*, 84(5):543–571, 2010.
[17] Jayadeep Gopalakrishnan and Joseph E. Pasciak. Multigrid for the mortar finite element method. *SIAM Journal on Numerical Analysis*, 37(3):1029–1052, 2000.

[18] Michael Griebel, Tilman Neunhoeffer, and Hans Regler. Algebraic multigrid methods for the solution of the NavierStokes equations in complicated geometries. *International Journal for Numerical Methods in Fluids*, 26(3):281–301, 1998.
[19] Wolfgang Hackbusch. *Multi-Grid Methods and Applications*, volume 4 of *Series in Computational Mathematics*. Springer, 1st edition edition, 1985.
[20] Ales Janka. Smoothed aggregation multigrid for a Stokes problem. Institute of Analysis and Scientific Comuting, Ecole Polytechnique Federale de Lausanne, 8 2006.
[21] Carsten Keller, Nicholas I. M. Gould, and Andrew J. Wathen. Constrained preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis & Applications*, 21(4):1300 – 1317, 2000.
[22] Olga A Ladyzhenskaya and Richard A Silverman. *The mathematical theory of viscous incompressible flow*, volume 76. Gordon and Breach New York, 1969.
[23] Jan Mandel. On block diagonal and Schur complement preconditioning. *Numerische Mathematik*, 58(1):79–93, 1990.
[24] Y. Notay. A new analysis of block preconditioners for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 35(1):143–173, 2014.
[25] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15:1787–1972, 1972.
[26] A. Popp. *Mortar Methods for Computational Contact Mechanics and General interface Problems*. PhD thesis, Technische Universität München, 2012.
[27] Alexander Popp, Markus Gitterle, Michael W. Gee, and Wolfgang A. Wall. A dual mortar approach for 3D finite deformation contact with consistent linearization. *International Journal for Numerical Methods in Engineering*, 83(11):1428–1465, 2010.
[28] Michael A. Puso. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 59(3):315–336, 2004.
[29] Michael A. Puso and Tod A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(68):601 – 629, 2004.
[30] Marzio Sala and Raymond S. Tuminaro. A new Petrov-Galerkin smoothed aggregation preconditioner for nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 31(1):143–166, 2008.
[31] V. Simoncini. Block triangular preconditioners for symmetric saddle-point problems. *Applied Numerical Mathematics*, 49(1):63 – 80, 2004. Numerical Algorithms, Parallelism and Applications.
[32] K. Stüben. *Multigrid*, Appendix A: An introduction to algebraic multigrid, pages 413–533. U. Trottenberg and C.W. Oosterlee and A. Schüller, 2001.
[33] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.
[34] JP Van Doormaal and GD Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical heat transfer*, 7(2):147–163, 1984.
[35] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65(1):138–158, 1986.
[36] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.
[37] Petr Vaněk. Acceleration of convergence of a two-level algorithm by smoothing transfer operators. *Applications of Mathematics*, 37:265–274, 1992.
[38] Petr Vaněk, Marian Brezina, and Jan Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik*, 88:559–579, 2001.
[39] Petr Vaněk, Ales Janka, and Herve Guillard. Convergence of algebraic multigrid based on smoothed aggregation ii: Extension to a Petrov-Galerkin method. *INRIA Technical Report 3683*, 1999.
[40] C. Vuik, A. Saghir, and G. P. Boerstoel. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. *International Journal for Numerical Methods in Fluids*, 33(7):1027–1040, 2000.
[41] Markus Wabro. Coupled algebraic multigrid methods for the Oseen problem. *Computing and Visualization in Science*, 7:141–151, 2004.
[42] C. Wieners and B. Wohlmuth. Duality estimates and multigrid analysis for saddle point problems arising from mortar discretizations. *SIAM Journal on Scientific Computing*, 24:2163–2184, 2003.
[43] C. Wieners and B.I. Wohlmuth. A general framework for multigrid methods for mortar finite elements. Technical Report 415, Universität Augsburg, Institut für Mathematik, 1999.
[44] T.A. Wiesner. *Flexible Aggregation-based Algebraic Multigrid Methods for Contact and Flow Problems*. PhD thesis, Technische Universität München, 2015.
[45] H. Wobker and S. Turek. Numerical studies of Vanka–type smoothers in computational solid mechanics. *Advances in Applied Mathematics and Mechanics*, 1(1):29–55, 2009.
[46] B. Wohlmuth. Multigrid methods for saddlepoint problems arising from mortar finite element discretizations. *ETNA. Electronic Transactions on Numerical Analysis*, 11:43–54, 2000.
[47] Bin Yang and TodA. Laursen. A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations. *Computational Mechanics*, 41(2):189–205, 2008.
[48] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65:227–246, 2000.
[49] Walter Zulehner. Analysis of iterative methods for saddle point problems: A unified approach. *Mathematics of Computation*, 71:479–505, 2002.