

# Algebraic multigrid methods for constrained linear systems with applications to contact problems in solid mechanics

Mark F. Adams<sup>1</sup>

<sup>1</sup> Sandia National Laboratories, MS 9417, Livermore CA 94551 (mfadams@ca.sandia.gov)

## SUMMARY

This article develops a general framework for applying algebraic multigrid techniques to constrained systems of linear algebraic equations that arise in applications with discretized PDEs. We discuss constraint coarsening strategies for constructing multigrid coarse grid spaces and several classes of multigrid smoothers for these systems. The potential of these techniques is investigated with their application to contact problems in solid mechanics. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: algebraic multigrid, multigrid methods, saddle point problems, parallel multigrid, contact in solid mechanics

## 1. Introduction

This paper investigates the construction of algebraic multigrid (AMG) methods for constrained linear systems—saddle point problems or KKT systems. Saddle point problems, differential equations with equality constraints of the form:

$$Ax \equiv \begin{pmatrix} K & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \equiv b \quad (1)$$

where  $K$  is a large, possibly singular, matrix from a discretized PDE,  $C$  is a matrix of constraint equations,  $u$  are the primal variables and  $\lambda$  are Lagrange multipliers, arise in many applications from contact in solid mechanics to incompressible flow and problems in mathematical optimization with PDE systems. Several approaches have been developed for these problems, given a solver for the primal matrix  $K$ , namely Uzawa's method [4],

---

\*Correspondence to: Sandia National Laboratories, MS 9417, Livermore CA 94551 (mfadams@ca.sandia.gov)

<sup>†</sup>This paper is authored by an employee(s) of the U.S. Government and is in the public domain.

Contract/grant sponsor: Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy; contract/grant number: DE-AC04-94AL85000

projection methods [18] and Schur elimination (or static condensation techniques). These methods *decouple* the primal and dual equations, and though they are useful and have their place in the repertoire of methods for solving these problems, they all suffer some shortcomings (eg, not general, scalable, robust, or costly).

Multigrid methods (eg, [12, 20, 10]) are among the most efficient iterative schemes for solving the linear systems associated with elliptic partial differential equations. Multigrid methods are well known to be theoretically scalable for  $H^1$ -elliptic operators, both scalar problems like Poisson's equation, and systems of PDEs like displacement finite element discretizations of elasticity [7]. Hence, multigrid is a natural choice for solving solid mechanics problems. Multigrid has been applied to structured grid problems for decades [9]. In the past ten years, multigrid methods have been developed for unstructured problems as well. One such method, that has proven to be well suited to elasticity problems, is smoothed aggregation [21, 2, 16]. We use smoothed aggregation as the primal solver, in the Uzawa algorithm, for our contact problems in solid mechanics and as the primal solver in the methods developed in this study.

This paper investigates the development of fully *coupled* algebraic multigrid methods for solving saddle point problems. Multigrid methods that solve the coupled constrained system have the potential to mitigate the problems associated with known approaches (eg, be about as expensive as a solve of a similar non-constrained problem, not require altering the primal system, accommodate any type of constraint efficiently and be robust).

The remainder of this paper introduces a general framework for applying algebraic multigrid techniques to constrained systems of equations. This includes a general framework for designing AMG methods to KKT systems, and coarsening strategies for constraints in §2, and several approaches for constructing smoothers for KKT systems in §3. These techniques are applied to contact problems in solid mechanics, with example problems from an industrial finite element application in §4; and we conclude in §5.

## 2. AMG framework for constrained linear systems

A *multigrid*, or multilevel, method for a particular discretized PDE has two primary components: 1) the coarse grid spaces and 2) the smoother for each level. These coarse grid spaces are used to construct the *restriction* operator  $R$ , which is used to map residuals from the fine grid  $l$  to the coarse grid  $l+1$ , and the *prolongation* operator  $P$  that is used to map corrections from the coarse grid to the fine grid ( $P = R^T$  for all methods considered here and we avoid level superscripts). Note, the columns of  $P$  are a discrete representation of the coarse grid space on the fine grid. The smoother in multigrid is an inexpensive solver (eg, one iteration of Gauss-Seidel) used to reduce the error on each grid that is not effectively reduced by the coarse grid correction; multigrid methods are designed to be applied recursively until the coarsest grid is small enough to be easily solved accurately, usually with an exact solver. See Briggs, and the references therein, for an introduction to multigrid and to methods for common PDEs [10].

*Algebraic multigrid* (AMG) can be defined in the broadest terms as a multigrid method that internally constructs the coarse grid operator, usually via a Galerkin process, as opposed to having the coarse grid operators provide by the application. For fine grid  $l$ , given a restriction operator  $R$  and a prolongation operator  $P$ , the Galerkin coarse grid is constructed via the matrix triple product  $A^{l+1} \leftarrow RA^lP$ .

To apply a Galerkin process to the operator in equation 1, care must be taken to insure that the structure of the coarse grid operator is identical to that of the fine grid, to allow the process to be applied recursively. One solution to this problem is to construct the coarse grids as follows

$$\begin{pmatrix} \hat{R}K\hat{P} & \hat{R}C^T\bar{P} \\ \bar{R}C\hat{P} & 0 \end{pmatrix} \Leftarrow \begin{pmatrix} \hat{R} & 0 \\ 0 & \bar{R} \end{pmatrix} \begin{pmatrix} K & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \hat{P} & 0 \\ 0 & \bar{P} \end{pmatrix} \quad (2)$$

An attractive property of this formulation is that the coarse grid spaces are at least formally separated. The operator  $\hat{P}$ , the prolongator from the multigrid method for the discretized PDE in the primal part of the system, is a well studied topic for some classes of PDEs and can be presumed to be available. Only the operator  $\bar{P}$ , the constraint coarse grid spaces, needs to be defined—the remainder of this section discusses the construction of  $\bar{P}$ .

### 2.1. Coarse grid spaces for Lagrange multipliers

The selection of the coarse grid spaces is critical in the construction of any multigrid method; the construction of these spaces for constraint equations has not been address, to our knowledge. First, consider the simplest choice of  $\bar{P}$ : the identity. The identity can be an appropriate choice for some types of constraints, in particular, constraints that have many nonzero values and are few in number (eg, rigid body mode constraints), are probably best served by the identity. Many important classes of constraints, however, require coarsening to avoid complexity problems and to avoid overdetermined or ill-conditioned coarse grid matrices.

### 2.2. Constructing $\bar{P}$ for contact problems in solid mechanics

This section describes our approach for constructing  $\bar{P}$  with particular emphasis on solid mechanics applications with contact. For simplicity we restrict ourselves to piecewise constant constraint coarse grid spaces, hence  $\bar{P}$  is block diagonal with columns that contain zeros and ones (in practice these columns are normalized to have a two norm of 1.0). The attraction of this choice for  $\bar{P}$  is that aggregation techniques can be employed to construct these spaces relatively easily. Aggregation techniques are often used in AMG algorithms, but their application to constraint equations is not obvious because, for one, there is no square matrix, or graph  $G$ , to use for common aggregation strategies. Thus, the first task is to define a suitable graph. The matrix inner product  $G \leftarrow CYC^T$ , where  $Y$  is a matrix to be determined, is one such choice for  $G$ .

The identity is a natural choice for  $Y$ , but in the case of contact constraints this can lead to slow coarsening, via standard “greedy” aggregation strategies, and coarse grid constraint matrices which are nearly rank deficient due to equations being nearly identical under some conditions. Greedy aggregation strategies, based on maximal independent sets [1], are attractive because they are relatively easy to implement in parallel and we have experience in developing them, but partitioning algorithms can also be used for this purpose [2]. We have studied several choices for  $Y$  and have selected  $\hat{P}\hat{P}^T$ . This choice of  $Y$  has the effect of applying the coarsening of the primal equations to the columns of  $C^T$  (ie,  $\hat{P}^T C$ ), and using this matrix inner product to construct the aggregation matrix  $G = C\hat{P}\hat{P}^T C^T$ .

This choice of  $G$  is motivated by the deterioration in convergence that we have observed in coarsening constraints from contact, namely, that the coarse grid constraint equations can become nearly linearly dependent. This choice of  $G$  allows us to directly bound the angle

between coarse grid constraint equations. This is due to the construction of the constraint matrix on grid  $l+1$  from grid  $l$ :  $C^{l+1} \leftarrow \bar{P}^T C^l \hat{P}$  (from equation 2). The dot product, and hence the angle, between two coarse grid constraint equations are closely related to the off diagonal entries (edges) of  $G$ , because  $C^{l+1} (C^{l+1})^T = \bar{P}^T C^l \hat{P} \hat{P}^T (C^l)^T \bar{P} = \bar{P}^T G \bar{P}$ . Because  $\bar{P}$  is a simple aggregation operator, a bound on the minimum angle between constraint equations is closely related to the maximum edge cut in the partitioning of  $G$ . Thus, standard graph partitioning principles, of minimizing the maximum edge cut, can be employed to directly maximize the minimum angle between coarse grid constraint equations and thereby improving the condition of the coarse grid operator vis-a-vis nearly linear dependent constraint equations.

Multigrid theory indicates that aggregates should be composed of strongly connected subdomains and it is common to apply heuristics in the aggregation algorithms to accomplish this [21]. Though, we are not aware of an analysis that proves that this is the case for the aggregation of the constraints, it is intuitive to believe that it is so. For instance, the contact constraints considered in this study are enforced with a simple point-on-surface technique in which a “slave” node on one contact surface is constrained to lie on a “master” quadrilateral face of the other contact surface. Furthermore, our application implements friction by fixing all three degrees of freedom on a slave node to the master surface, this results in three equations that involve the five nodes in each contact constraint. These three equations are however orthogonal, and it is natural to expect that it would be advantageous for the aggregation strategy to interpolate these equations separately (ie, interpolate ‘x’ direction constraints separately from ‘y’ and ‘z’ direction constraints, and so on). This criterion can be achieved by optimizing a standard graph partitioning metric, that of maximizing the edge weights within an aggregate to produce strongly connected subdomains. Note, one complication with our choice of  $Y$  is that the  $\hat{P}$  from some multigrid algorithms will weakly couple these equations, that is, the information of their strict orthogonality is lost.

*2.2.1. Aggregation algorithm for contact in solid mechanics* The choice of piecewise constant interpolation requires a disjoint decomposition of the nodes, or rows, of the provided aggregation graph  $G$ , described in the previous section. First, given tolerance  $\gamma$ , the graph  $G$  is modified by removing all edges  $w_{ij} < \gamma$ . We use a variant of a “greedy” parallel maximal independent set (MIS) algorithm [1]. Define the *edge weight* between two nodes  $i$  and  $j$  as  $w_{ij} = \frac{|G_{ij}|}{\sqrt{G_{ii}G_{jj}}}$ . Define the node weight  $w(i)$  for node  $i$ , by  $w(i) = \sqrt{\sum w_{ij}^2}$ . The node order in the greedy MIS algorithm starts from the highest to the lowest node weight or alternatively these weights are used to assign the “random” numbers used in the parallel randomized MIS algorithm [13].

Thus, given an aggregation graph  $G$ , and a tolerance  $\gamma$ , construct a maximal independent set  $S$ . For convenience of exposition, assume the nodes in  $G$  are ordered with the nodes in  $S$  first and construct initial sets  $C_j$  with each node  $j$  in  $S$  (ie,  $C_j = \{j\}$ ). Next, for all nodes  $i \in G \setminus S$ , add  $i$  to the aggregate  $C_j | \max_{j \in S} w_{ij}$  (the definition of an MIS insures that such a node,  $j \in S$ , exists for all nodes  $i \in G \setminus S$ ). These sets  $C_j$  define the prolongator as follows

$$\bar{P}_{ij} = \begin{cases} |C_j|^{-\frac{1}{2}}, & \text{if } i \in C_j \\ 0, & \text{otherwise} \end{cases}$$

This defines the coarse grid spaces, for the constraint equations, used in this study, with a values  $\gamma = 0.2$  and leaves only the definition of the KKT smoothers to provide a complete

multigrid method.

### 3. Smoothers for constrained linear systems

Smoothers for constrained systems are a simpler matter than the construction of the coarse grid spaces, if for no other reason because previous work on this topic is informative. Schulz used structured geometric multigrid for optimization problems and developed a stable incomplete factorization for M-matrices, by ordering the constraints after all primal equations with which they interact [19]. Note, this technique of ordering the constraints after all of the primal equations with which they interact is a component of most methods. Vanka uses structured geometric multigrid for incompressible flow problems with a multiplicative Schwarz, constraint centric method [22]. Segregated methods, or preconditioned Uzawa, have been investigated by Braess and Sarazin for the Stokes problem [8], and generalized by Zulehner [23]. We investigate variants of these three smoothers: 1) processor block incomplete factorizations (ILU) as preconditioners for a Chebyshev polynomial [3], 2) segregated (Braess) methods and 3) a constraint centric (Vanka) Schwarz methods.

The ILU preconditioner is the processor local level fill ILU method in PETSc with level fill of one; the constraint equations are ordered last on each processor and the processor's primal equations are ordered with nested dissection (note, this is not a provably stable method). The following two sections describe the other two smoothers investigated here: segregated methods in §3.1, and a constraint centric Schwarz method in §3.2.

#### 3.1. Segregated KKT smoothers

Consider a block factorization of the KKT system

$$\begin{pmatrix} K & C^T \\ C & 0 \end{pmatrix} = \begin{pmatrix} K & 0 \\ C & -S \end{pmatrix} \begin{pmatrix} I & K^{-1}C^T \\ 0 & I \end{pmatrix} \quad (3)$$

where  $S = CK^{-1}C^T$  is the Schur complement. This factorization suggests the use of the preconditioning matrix

$$M = \begin{pmatrix} K & 0 \\ C & -S \end{pmatrix}, M^{-1}A = \begin{pmatrix} I & K^{-1}C^T \\ 0 & I \end{pmatrix} \quad (4)$$

and is attractive because the preconditioned system  $M^{-1}A$  is well conditioned in that all of its eigen values are 1.0 and the Jordan blocks of size at most two [15], if an exact solver is used for  $S^{-1}$  and  $K^{-1}$ .

The action of  $K^{-1}$  is approximated with a common smoother for the primal part of the system  $\hat{A}(f)$ . The action of  $(CK^{-1}C^T)^{-1}$  is approximated with an approximate Schur complement  $Q = C^T D^{-1} C^T$ , where  $D$  is an approximation to  $K$  (eg,  $D$  is the diagonal of  $K$ ). The dual smoother,  $\hat{S}(g)$ , used in this study is processor block Jacobi, with exact subdomains solves of  $Q$  on each processor. These choices for the segregated smoother results in the following algorithm, give an initial guess  $\lambda^j$  and  $u^j$ , a primal smoother  $\hat{A}(f)$  and the dual smoother  $\hat{S}(g)$ :

$$\begin{aligned} u^{j+1} &\leftarrow u^j + \hat{A}(f - Ku^j - C^T \lambda^j) \\ \lambda^{j+1} &\leftarrow \lambda^j + \hat{S}(Cu^{j+1} - g) \end{aligned}$$

Note, this is equivalent to one iteration of the preconditioned Uzawa algorithm without regularization [11].

This smoother can be symmetrized with a second application of the primal smoother as discussed by Bank, Welfert and Yserentant [6], as follows,

$$\begin{aligned}\hat{A}(\hat{u}^{j+1} - u^j) &= f - Au^j - C^T \lambda^j \\ \hat{S}(\lambda^{j+1} - \lambda^j) &= C\hat{u}^{j+1} - g \\ \hat{A}(u^{j+1} - \hat{u}^{j+1}) &= -C(\lambda^{j+1} - \lambda^j)\end{aligned}$$

The symmetrized form is used in this

### 3.2. A constraint centric Schwarz KKT smoother

The second smoother that we investigate is a constraint centric overlapping Schwarz method. The approach is similar to that used by Vanka [22], in that overlapping Schwarz subdomains are formed from a non-overlapping decomposition of the constraints. Vanka uses one constraint per subdomain, we use much larger aggregates—all of the constraint equations on each processor (in the current implementation). Each of these sets of equations is augmented with all of the primal equations that interact with any of these constraint equations. Thus, each subdomain is a small KKT system and can be interpreted in physical terms, as the matrix resulting from applying Dirichlet boundary conditions to the nodes in the problem not touched by any of the constraint equations in the aggregate. These subdomain solves are stable because each subdomain is a small, well posed, KKT system and thus a factorization, with the constraints ordered last, is well defined and does not require pivoting.

These aggregates are used to construct an overlapping additive Schwarz component ( $B_d$ ) with exact subdomain solves. An additive formulation is used for ease of parallelization, and the solution for the primal variables is updated only from the local solve on each processor, to negate the effects of “overshooting” the solution in the overlap region. That is, each subdomain solve restricts the residual for the entire overlapping domain (thus requiring inter-processor communication), performs the local subdomains solve (with a factorized matrix that includes rows from other processors), but only updates local nodes and therefore discards solution values that would otherwise be communicated in a full overlapping additive Schwarz approach.

As with the segregated smoother, a smoother for the primal part of the system is assumed to be available and is denoted, in matrix form, as  $M^{-1}$  (not to be confused with the  $M$  above). The primal part (domain) of the smoother is defined by  $B_p = R_p^T M^{-1} R_p$  where  $R_p = [I \ 0]$ . The dual part of the smoother is defined by  $B_d = (B_1 + B_2 + \dots + B_k)$ , where  $B_i = \tilde{R}_i^T (R_i K R_i^T)^{-1} R_i$ , for each subdomain  $i$ . The matrices  $R_i$  and  $\tilde{R}_i$  are simple extraction operators (matrices with only ones and zeros), and with the proper node ordering,  $\tilde{R}_i$  is block diagonal, but  $R_i$  is not because of overlap. That is,  $\tilde{R}_i$  is constructed by dropping terms in  $R_i$  that result in communication.

The constraint centric overlapping Schwarz method is constructed with an additive and a multiplicative variant. The multiplicative constraint centric Schwarz smoother is, with a initial guess  $x_0 = [u_0^T \ \lambda_0^T]^T$  and right hand side  $b = [f_0^T \ g_0^T]^T$ , defined as

$$\hat{x} \leftarrow x_0 + (B_p + B_d - B_d A B_p)(b - A x_0) \quad (5)$$

that is, the standard multiplicative Schwarz method with two subdomains. The additive version is defined as

$$\hat{x} \leftarrow x_0 + (B_p + B_d)(b - Ax_0) \quad (6)$$

that is, the standard additive Schwarz method with two subdomains. The additive variant is used as a preconditioner for an appropriately damped iterative scheme—Chebyshev polynomials [3]. The multiplicative variant is used with a multiplicative primal smoother (symmetric Gauss-Seidel).

#### 4. Numerical results with contact problems in solid mechanics

This section investigates the effectiveness of the algebraic multigrid methods for constrained systems (AMG/KKT) developed herein with two problems in solid mechanics. The problems are run with the *Adagio* nonlinear quasi-static solid mechanics application from Sandia National Laboratory [14]. Note, *Adagio*'s implementation of friction results in fixing the slave node on the corresponding master surface, resulting in three Lagrange multipliers per contact interaction.

The solver algorithms are implemented in the linear solver package Prometheus [17], which is built on the parallel numerical package PETSc [5]. Prometheus provides three algebraic multigrid solvers for the primal system—smoothed aggregation is used in this study [21]. Because the smoothers are not, in general, symmetric and the preconditioned systems are not, in general, positive the AMG/KKT methods use GMRES as the solver—preconditioned with one iteration of V-cycle multigrid.

The Uzawa solver uses conjugate gradient (CG), preconditioned with one multigrid V-cycle, as the primal solver. All solves use one pre and post smoothing step, with an exact solver for the coarse grid. The system are solved with a tolerance of  $10^{-12}$ , that is convergence is declared when the solution  $\hat{x}$  satisfies  $|b - A\hat{x}| < 10^{-12}|b|$ .

We investigate our aggregation method for the constraint coarse grid spaces, within our AMG/KKT framework, and four smoothers: 1) ILU preconditioner for a Chebyshev polynomial, 2) the segregated smoother with an additive primal smoother, and both the 3) additive and 4) multiplicative variant of our constraint centric Schwarz method. The ILU preconditioner is the processor local level fill ILU method in PETSc with level fill of one; the constraint equations are ordered last on each processor and the processor's primal equations are ordered with nested dissection (note, this is not a provably stable method).

The performance of our AMG/KKT solvers is compared to an Uzawa implementation which is the standard method used in Prometheus and has been highly refined and thus provides a good standard for comparison. Additionally the augmentation factor for the regularization required with Uzawa's method has been hand optimized to provide the best performance and thus, represents our best effort to solve these systems with Uzawa.

##### 4.1. The “circles” problem

The test first problem, known as the circles problem, is a disk within a ring within a circular cutout and is shown in Figure 1. The middle ring segment has an elastic modulus  $10^2$  times that of the “soft” material of the central disc and outer cutout plate. All materials have a Poisson's ratio of 0.3. This problem has two layers of hexahedral elements through the thickness, 7236

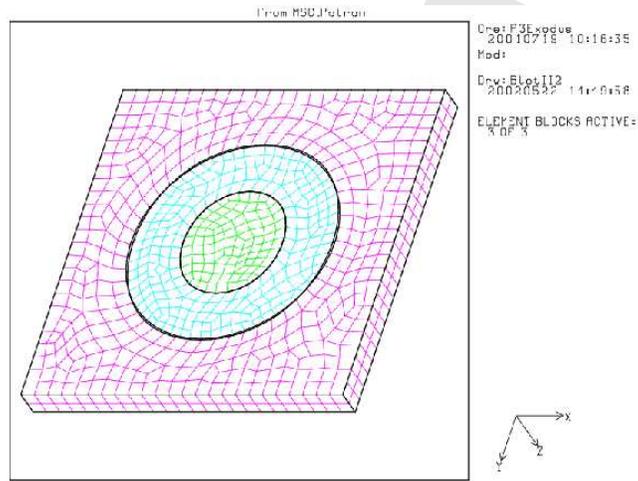


Figure 1. Adagio's "circles" problem

primal equations (minus the Dirichlet boundary conditions) and 180 point on surface contact interactions with a total of 420 Lagrange multipliers. Note, the sliding boundary conditions on the top and bottom surface result in the smoothed aggregation algorithm having the same semantics and the plane aggregation algorithm due the presents of Dirichlet boundary conditions on all elements. Two multigrid levels are used for this problem and only the first linear solve is investigated. The problem is run on eight Sun processor and PETSc achieves about 400 Mflops/second in the matrix vector product on the fine grid matrix.

Figure 2 shows the residual history, as a function of time (measured in units of the time for one matrix vector product on the fine grid), of the four smoothers in the AMG algorithm, the Uzawa solver, and the additive constraint centric smoother used as the solver.

This shows that the smoother alone (as the solver) is stagnating as is expected because it reduces the high frequency error effectively but is slow at reducing the lower frequency content of the error. The AMG/KKT solver with constraint centric smoothers, and the segregated smoother, solve the problem in about two thirds the time of the Uzawa solver; and the ILU smoother is noticeably slower than the other methods.

All four of the AMG/KKT smoothers use the processor subdomains and are hence less powerful as more processors are used for a given problem. This is not necessarily a problem, because smoothers are only responsible for the high frequency content of the error which can be reduced effectively with small Schwarz subdomains. To determine if this is indeed the case, in practice, the iteration counts for this problem on 1,2,4 and 8 processors are tabulated in Table I. Table I shows the iteration counts for constraint centric Schwarz (CCS) method, both the additive and multiplicative variants, the segregated smoother, and the ILU smoother. This data shows that the smoothers are indeed essentially invariant to the number of processors (ie, the size of the subdomains) in this range of processors.

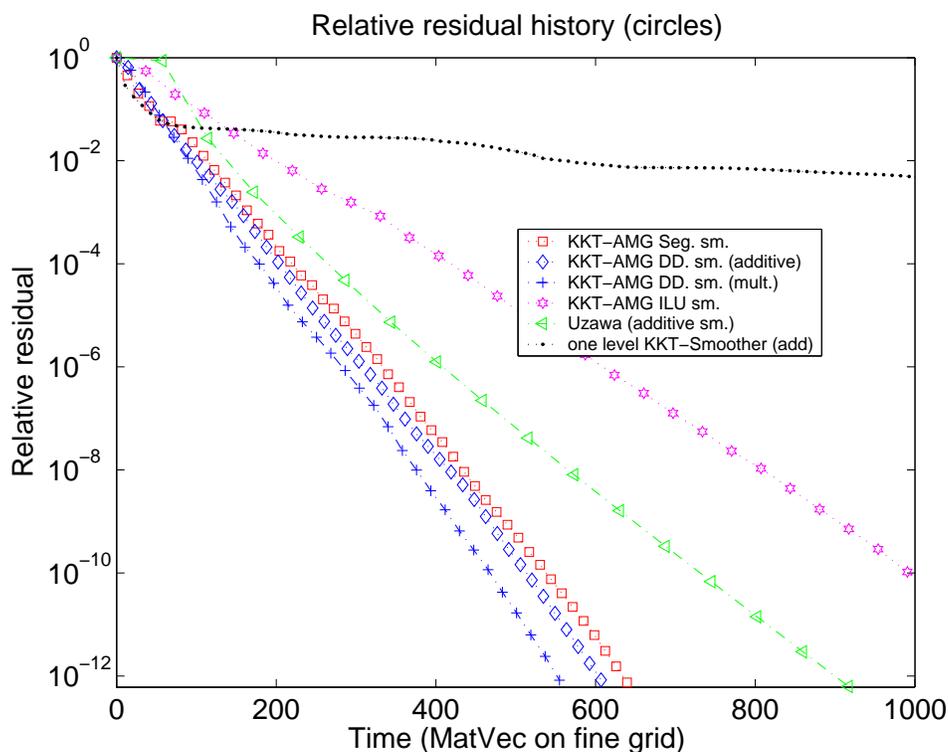


Figure 2. Residual vs. Time (in matrix vector products on the fine grid) for the “circles” problem

Table I. AMG/KKT iteration counts vs. number of processors

Smoothers	1	2	4	8
Segregated	51	57	52	47
CCS (additive)	47	43	47	42
CCS (multiplicative)	35	34	37	31
ILU	30	33	35	32
Uzawa	100	122	127	106

#### 4.2. The forging problem

The second test problem is shown in Figure 3. This is from a simulation of a forging process and has about 127K degrees of freedom and 64 constraint equations after the first five for time steps, which are measured for this study. This is a nonlinear pseudo-static analysis with five “time” steps, with the first contact occurring in the third step.

The forge problem is run on eight processors of an SGI Origin 2000 and the matrix vector

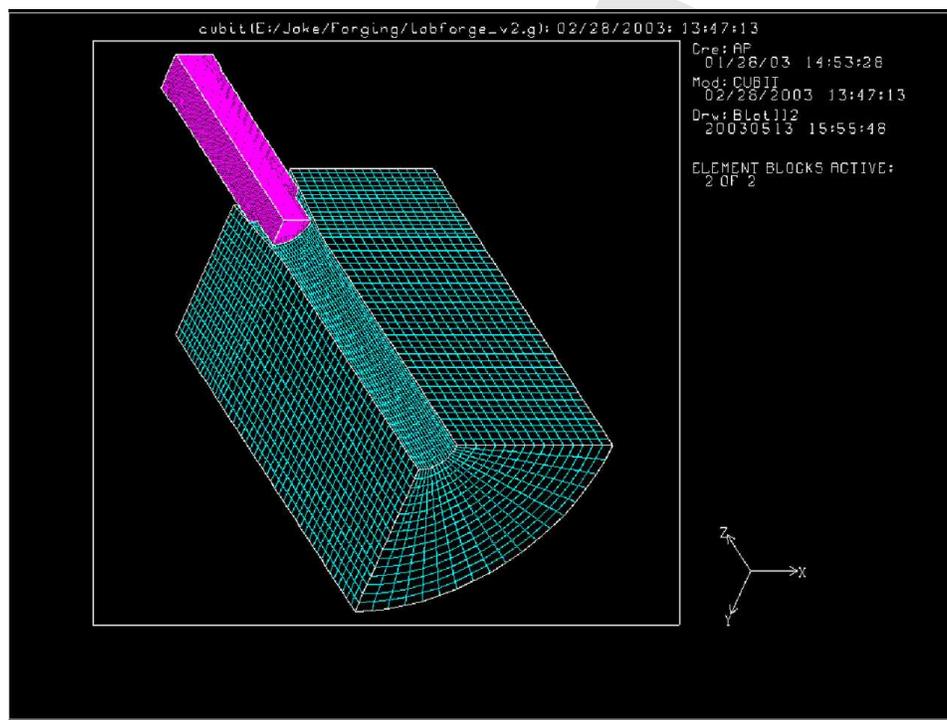


Figure 3. Adagio “forge” problem

product on the fine grid runs at about 500 Mflops/sec. Adagio uses a non-linear CG method that can be “preconditioned” with a linear solver, Prometheus is used in this case with a residual tolerance of  $10^{-4}$ . Three levels of smoothed aggregation multigrid are used. Table II shows the total (“end-to-end”) time for the simulation along with the time for the solution phase (with the number of linear solves) and the solver setup phase.

Table II. Solve and run times (sec) for the forge problem

Smoothers	end-to-end	setup	solve (# of solves)
Segregated	1722	307	839 (235)
CCS (additive)	1877	327	951 (240)
CCS (multiplicative)	2302	308	1424 (239)
ILU	4746	841	3270 (242)
Uzawa	1758	297	893 (238)

This data shows that the ILU smoother is not competitive and that the other two smoothers are competitive with a well optimized Uzawa solver.

## 5. Conclusions

We have presented a general framework for applying algebraic multigrid techniques to constrained systems of linear algebraic equations that shows promise as an effective approach to construct solvers for saddle point problems. We have developed one AMG method within this framework, along with three classes of smoothers, and applied them to contact problems in solid mechanics. Two of the smoothers, the segregated and constraint centric Schwarz methods, are at least competitive to a well optimized Uzawa solver on two test problems. The third smoother, ILU, is not competitive—but we have not explored the design space of ILU methods, to a significant degree, and hence ILU methods should not be discarded from consideration in our opinion. We have demonstrated that our framework for applying algebraic multigrid techniques to constrained linear systems has the potential of supporting robust, scalable and fast solvers for this class of constrained linear systems.

## ACKNOWLEDGEMENTS

The authors would like to thank the Adagio development team at Sandia National Laboratories for providing the test problems in this study.

## REFERENCES

1. M. F. ADAMS, *A parallel maximal independent set algorithm*, in Proceedings 5th Copper Countain Conference on Iterative Methods, 1998.
2. ———, *Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics*, International Journal for Numerical Methods in Engineering, 55 (2002), pp. 519–534.
3. M. F. ADAMS, M. BREZINA, J. J. HU, AND R. S. TUMINARO, *Parallel multigrid smoothing: polynomial versus Gauss–Seidel*, J. Comp. Phys., (2003).
4. K. ARROW, H. L., AND U. H., *Studies in nonlinear programming*, Stanford University Press, 1958.
5. S. BALAY, W. D. GROPP, L. C. MCINNES, AND B. F. SMITH, *PETSc 2.0 users manual*, tech. rep., Argonne National Laboratory, 1996.
6. R. E. BANK, B. D. WELFERT, AND H. YSERENTANT, *A class of iterative methods for solving mixed finite element equations*, Numer. Math., 56 (1990), pp. 645–666.
7. D. BRAESS, *On the combination of the multigrid method and conjugate gradients*, in Multigrid Methods II, W. Hackbusch and U. Trottenberg, eds., Berlin, 1986, Springer-Verlag, pp. 52–64.
8. D. BRAESS AND R. SARAZIN, *An efficient smoother for the Stokes problem.*, Appl. Numer. Math., 23 (1997), pp. 3–20.
9. A. BRANDT, *Multi-level adaptive solutions to boundary value problems*, Math. Comput., 31 (1977), pp. 333–390.
10. W. L. BRIGGS, V. E. HENSON, AND S. MCCORMICK, *A multigrid tutorial, Second Edition*, SIAM, Philadelphia, 2000.
11. H. C. ELMAN AND G. G. H., *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SAIM J. Numer. Anal., 31 (1994), pp. 1645–1661.
12. W. HACKBUSCH, *Multigrid methods and applications*, vol. 4 of Computational Mathematics, Springer-Verlag, Berlin, 1985.
13. M. LUBY, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 4 (1986), pp. 1036–1053.
14. J. A. MITCHELL, A. S. GULLERUD, W. M. SCHERZINGER, R. KOTERAS, AND V. L. PORTER, *Adagio: non-linear quasi-static structural response using the SIERRA framework*, in First MIT Conference on Computational Fluid and Solid Mechanics, K. Bathe, ed., Elsevier Science, 2001.
15. M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.
16. G. POOLE, L. Y.C., AND M. J., *Advancing analysis capabilities in ansys through solver technology*, Electronic Transactions in Numerical Analysis, 15 (2003), pp. 106–121.

17. *Prometheus*. [www.cs.berkeley.edu/~madams](http://www.cs.berkeley.edu/~madams).
18. P. SAINT-GEORGES, Y. NOTAY, AND G. WARZEE, *Efficient iterative solution of constrained finite element analysis*, *Comput. Methods Appl. Mech. Engrg.*, 160 (1998), pp. 101–114.
19. V. SCHULZ, *Incomplete indefinite decompositions as multigrid smoothers for KKT systems*, *ISNM*, 133 (1999), pp. 257–266.
20. U. TROTTEBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, London, 2001.
21. P. VANĚK, J. MANDEL, AND M. BREZINA, *Convergence of algebraic multigrid based on smoothed aggregation*, *Numerische Mathematik*, 88 (2001), pp. 559–579.
22. S. VANKA, *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*, *J. Comp. Phys.*, (1986), pp. 138–158.
23. W. ZULEHNER, *A class of smoothers for saddle point problems*, *Computing*, 65 (2000), pp. 227–246.