13th Copper Mountain Conference on Multigrid Methods

Investigation of Enhancements in Incremental Forming Simulations using Algebraic Multigrid and Schur Complement Methods

F. Schmid and M. Schäfer

Department of Numerical Methods in Mechanical Engineering
Technical University of Darmstadt

Abstract

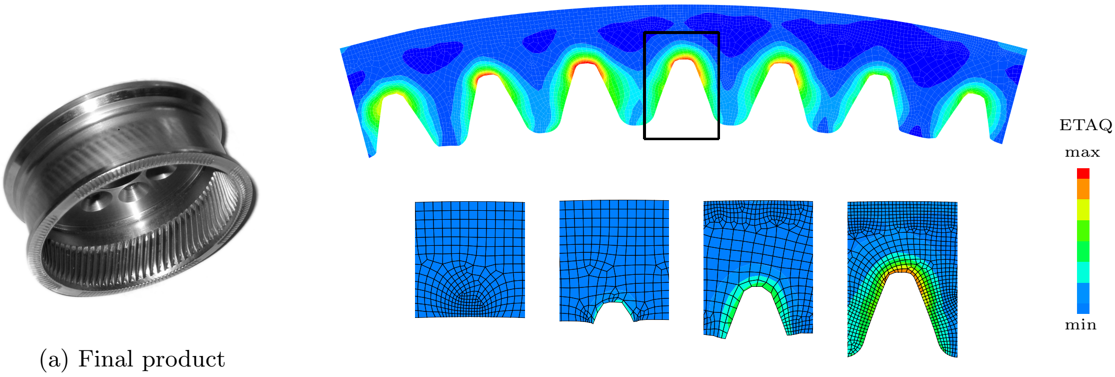
The industrial demand for short production times and a high flexibility of the applied processes lead to the development of incremental forming processes in the recent years. Numerically though, the simulation of those processes encounters difficulties which complicate an efficient simulation with current methods. The requirement of continual mesh adaption, the indispensible presence of contact and non-linear constitutive laws lead to unacceptably high computation times when classical numerical schemes are applied. The main problem encountered, is the solution of the ill-conditioned system matrices upon which iterative solvers struggle with bad convergence rates. This paper investigates the performance of an algebraic multigrid (AMG) solver when different strategies of using certain process characteristics of incremental forming are applied. The main idea of those strategies is to exploit the local restriction of the actual forming on the model, which is the most striking difference to other forming processes. The discussion includes the consideration of those process properties within the AMG setup as well as in an feasible schur complement decomposition.

Keywords: algebraic multigrid, schur complement, incremental forming process, finite elements, structural mechanics

1 Introduction

In the recent years, technology in the field of industrial production had to cope with an increasing need for short production runs, a high product diversification and complexity and thus, continuous developments on metal forming have taken place. Along with aspects from manufacturing the numerical simulation of forming processes has received steady progress and current methods perform well in that field of application [8, 14].

As one of the more recent approaches incremental forming processes try to achieve those industrial demands due to their high flexibility. They are specially designed as a low-cost alternative for difficult product geometries, variable low-series productions or prototyping. In the process, the product is formed in multiple sequentially similar steps of a small and simple die. Plastic forming only occurs in a very small zone of the work piece, while wide regions of it undergo only elastic deformation [5]. Bulk forming as well as sheet metal forming are possible processes for application of the method. In Figure 1 a flow forming simulation of internally geared wheels is depicted as an example of a typical application.



(a) Final product

(b) Numerical simulation at different stages

Figure 1: Typical application *(Cooperation: PtU, Darmstadt)*

The numerical properties of those simulations obstruct an efficient simulation with current methods, though. Beside a high number of unknowns and time steps, large deformations, unsteady boundary conditions and the need for continual mesh adaption are aspects that have to be taken into account. Moreover, nonlinear constitutive laws and the indispensible presence of multibody contact between dies and work pieces complicate the computations and lead to unacceptably high computation times. The main problem encountered is the conditioning of the system matrices upon which iterative solvers struggle with bad convergence. Algebraic multigrid methods have proven to be very efficient on finite element discretizations and to overcome mesh dependency but maintain the good complexity of multigrid methods [13].

This paper describes the performance of an algebraic multigrid solver upon this class of problems when different strategies are applied to exploit process characteristics which are specific to incremental forming. Those strategies concern the setup of the algebraic multigrid process as well as a feasible schur complement decomposition. The simulations are performed in LARSTRAN/SHAPE, a FE-package for large-strain plasticity in the context of metal forming simulations. The AMG solver at hand is the SAMG package by Fraunhofer Institut Bonn.

2 Governing equations

In this section a brief introduction on the governing equations and the incorporation of the non-linearities will be given. A complete derivation of the equations and the solution schemes can be found in, e.g., [1, 15].

Assume a two-dimensional domain Ω with boundary Γ. In the energy equation heat flux and heat sources are considerd to vanish and only the mechanical process is under consideration. The material is assumed to be isotropic and incompressible. We neglect the acceleration term and the kinematic virtual power respectively due to the fact that the process is assumed to be quasi-static. ρ denotes the density, $\mathbf{v}$ the velocity, $\boldsymbol{\sigma}$ the Cauchy stress tensor and $\mathbf{b}$ the body load. $\mathbf{D}$ is the rate of deformation tensor defined as $\mathbf{D} = \operatorname{sym}(\nabla \mathbf{v})$. ω^{int} is the internal energy per unit volume. A superposed dot denotes the material time derivative. The contact condition is formulated in the unitary contact condition. γ_N denotes the rate of interpenetration of two bodies, t_N describes normal tractions. The die is considered to be rigid and the contact is assumed to be frictionless. The governing equations and their associated boundary conditions then read

$$\begin{array}{rlllll} \nabla\cdot\mathbf{v} = & 0 & \text{in } \Omega, & \mathbf{n}\cdot\boldsymbol{\sigma} = & \bar{\mathbf{t}} & \text{on } \Gamma^t \quad \text{(prescribed tractions)}, \\ \nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{b} = & 0 & \text{in } \Omega, & \mathbf{v} = & \bar{\mathbf{v}} & \text{on } \Gamma^v \quad \text{(prescribed velocities)}, \\ \rho\dot{\omega}^{int} = & \mathbf{D}:\boldsymbol{\sigma} & \text{in } \Omega, & t_N\gamma_N = & 0 & \text{on } \Gamma^c \quad \text{(contact condition)}. \end{array}$$

Forming processes feature a large measure of non-linearity mainly caused by material response and the contact between die and workpiece. Geometric non-linearities are taken into account by using formulations which are invariant to large rotations as suggested in, e.g., [1]. The incorporation of the contact condition into the system equations is performed by a penalty formulation. This is still broadly used throughout engineering practice because of its easy implementation and the way of straightforward programming of the physical means. As a consequence, the contact penalty function imposes a strong non-linearity on the equations to be solved and deteriorates the matrix conditioning.

To provide some insight into the behaviour of the different solution methods under discussion two different constitutive laws are used in this paper which result in a different amount of matrix ill-conditioning: an elasto-plastic material as well as an incompressible rigid-plastic material law.

The first material under consideration is an elasto-plastic material. Its response can be decomposed in an elastic and a plastic deformation. The elastic deformation complies with standard formulations for geometrical non-linearity, whereas the plastic deformation is defined by further properties like an appropriate yield function and a plastic flow rate. Thus, during the computations the material response is iteratively determined by a corresponding stress-strain curve. For further details on the equations and the implementation in LARSTRAN, we refer to [1,4,12].

The second material is mainly used for e.g., hot forging processes. Due to the special material behaviour of those processes, in this rigid-plastic material law another penalty factor is introduced to preserve the incompressibility. Recalling the mass conservation above, for incompressible materials the velocity field is claimed to be divergence-free. Decomposition of $\boldsymbol{\sigma}$ and $\mathbf{D}$ into its deviatoric and volumetric part with the pressure $p = -\frac{1}{3}\sigma_{kk}$ and including Stokes's friction law leads to

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\text{dev}} + \boldsymbol{\sigma}^{\text{vol}} = 2\mu\mathbf{D}^{\text{dev}} - p\mathbf{I}.$$

An according formulation for that matter can be $p = -\kappa$ trace($\mathbf{D}$) with the penalty parameter κ. μ denotes the viscosity. The complete constitutive model can then be stated as

$$\boldsymbol{\sigma}^{\nabla J} = \mathbf{C}_{\mu}^{\sigma J} : \mathbf{D} \quad \text{with} \quad \mathbf{C}_{\mu}^{\sigma J}(\mathbf{D}, \mu, \kappa).$$

A complete derivation of this material model can be found in [16]. Further information on the underlying theory can be found in e.g., [14]. In the case of the rigid-plastic material law, the used penalty formulation deteriorates the matrix conditioning even further and slows down iterative solvers badly. A typical matrix condition of those problems would be between 10^{12} and 10^{13}.

3 Solution method and enhanced strategies

Applying a Newton time integration scheme to the discretization of the equations above leads to the solution of ill-conditioned linearized equation systems of form $\mathbf{Ku} = \mathbf{b}$.

Multigrid methods have proven to be very efficient for problems arising from finite element discretizations of partial differential equations, see e.g., [6]. The character of incremental forming processes however requires meshes with continual mesh adaption to a locally refined area of plastic deformation. In turn, a hierarchical grid structure which is required for geometric multigrid might be very difficult to achieve. For that reason we concentrate on algebraic multigrid methods (AMG) which appeared in the 1980's and were first introduced mainly by Brandt, McCormick, Ruge and Stüben. Futhermore, it has been shown that in many cases the method can be successfully applied to problems in which the system matrices appear not to be M-matrices, see e.g. [3, 9, 13]. For acceleration the AMG is mainly used as a preconditioner for a conjugate gradient method as suggested in e.g., [2, 13]. Gauss-Seidel loops are performed for pre- and post-smoothing in a standard V-cycle. Direct sparse Gauss-elimination is used on the coarsest level. It has been shown that those methods perform well on simulations of incremental forming processes compared to other iterative solvers and retain a high robustness and efficiency [11].

To enhance the solution of the systems emerging from incremental forming simulations the exploitation of certain process characteristics is suggested here. The most striking property is that contact as well as plastic forming only occurs in a very small zone of the work piece whereas large regions undergo only elastic forming. The idea is to dispose those critical physical phenomena from the main solution scheme.

The first strategy is to incorporate the process characteristics into the AMG-setup and force certain variables on certain grids in the multigrid process. The second attempt is to apply an appropriate domain decomposition by the schur complement method upon the plastic and elastic zone of deformation respectively. The aim is to achieve better properties for the different domains and thus, enhance the solution process to better convergence rates of the AMG-solver.

To be able to split the problem into subproblems in either way some criteria for an appropriate splitting has to be defined. As depicted in Figure 2 it is assumed that only a small zone of the workpiece is split off the rest of the FE-model which contains all elements with plastic deformation and nodes subject to contact forces from the die.

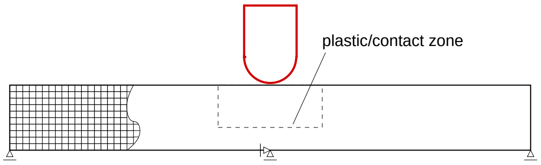
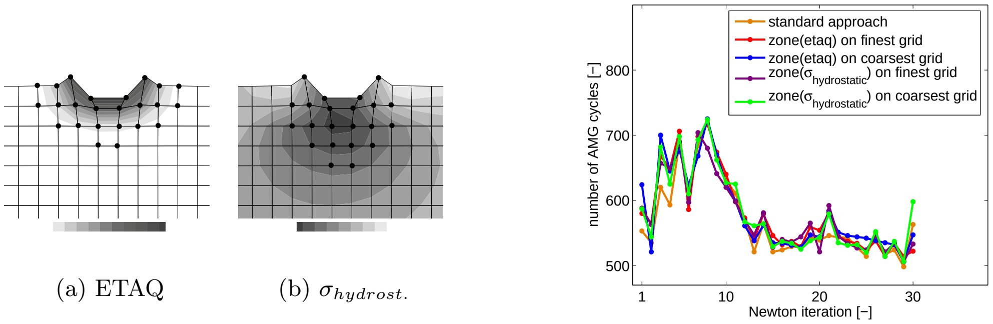


Figure 2: Exemplary splitting; elastic zone, plastic/contact zone resp.

4 AMG strategy - selective coarsening

This first strategy basically employs a standard AMG solution method on the given problem. The main difference lies in the construction of the coarser grids and the special treatment of the variables carrying the information of the forming zone in the work piece. In the used AMG method which was suggested by e.g., Ruge and Stüben (see, e.g., [3, 9]) the coarsening is accomplished in a pure algebraical manner, i.e. without any information about the underlying geometry. The creation of the coarse grids only depends on the principles of strong influence and dependence (see, e.g., [3, 7]). Dependent on some parameters more or less variables are chosen to be coarse grid variables. For that reason the forming zone might not be well represented in the matrices of the coarser grids. The idea of the AMG-approach therefore is to force those variables containing the numerically critical physical information on certain grid levels and achieve a stronger influence of those variables on the overall solution process. Two computation series were performed to see the effects hoped for. First, those variables were forced to stay on the finest grid, assuming that they could slow down the multigrid solution. In the second approach all those variables were forced to become variables on all levels supposing them to be subject to a better smoothing.

Since the main property to be exploited is the restriction of the plastic zone of deformation, a physical measure describing plasticity is exemplarily chosen for definition of the splitting: The total equivalent plastic strain (ETAQ) and the hydrostatic stress. Other physical variables would be possible. In Figure 3 (a/b) the splitting is illustrated. A black dot denotes a node which is to be forced on a certain grid level whereas all other variables will be subject to the regular coarsening process within the AMG procedure. To evaluate the numerical efficiency of the approach Figure 3(c) depicts the number of AMG V-cycles per Newton iteration in a representative time step of the simulation.



(a) ETAQ

(b) $\sigma_{hydrost.}$

(c) AMG-iterations per Newton step

Figure 3: Setup and definition of regions under consideration

As one can see, the convergence of the algebraic multigrid solver could not be enhanced by the suggested method. Computations in which other physical variables were used to create the splitting show the same results as given here. To apply an own coarsening process for the chosen variables in the splitting did not result in an acceleration of the computations either.

5 Schur complement approach

The second approach deals with an appropriate domain decomposition of the problem. Rather than parallelizing the system, which is the main field of application of the schur complement method, the idea here is to remove the small numerically difficult part of the problem from the whole system. This leaves small systems for the forming zone of the problem and a system almost as large as the original describing the rest of the model. The aim is to improve the conditioning of the matrix for the large domain such that its solution is significantly accelerated.

Theoretical background and Implementation

For the theoretical backround only a brief overview on the methods used is given here. For a more detailed derivation on the mathematical background we refer to e.g., [10].

The problem under consideration is again the model problem depicted in Figure 2. An appropriate splitting into the two subdomains $\mathcal{A}_{\mathcal{N}}$ as the large domain and $\mathcal{A}_{\mathcal{B}}$ as the critical small domain is assumed. A schur complement decomposition of the original system $\mathbf{Ku} = \mathbf{b}$ and its solution assuming $\mathbf{A}$ to be non-singular then reads:

$$\begin{bmatrix} \mathbf{A}_{\mathcal{B}} & \mathbf{0} & \mathbf{E}_{\mathcal{B}} \\ \mathbf{0} & \mathbf{A}_{\mathcal{N}} & \mathbf{E}_{\mathcal{N}} \\ \mathbf{F}_{\mathcal{B}} & \mathbf{F}_{\mathcal{N}} & \mathbf{C}_{\Gamma} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{\mathcal{B}} \\ \mathbf{u}_{\mathcal{N}} \\ \mathbf{u}_{\Gamma} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_{\mathcal{B}} \\ \mathbf{f}_{\mathcal{N}} \\ \mathbf{f}_{\Gamma} \end{Bmatrix} \quad \text{or, for simplicity} \quad \begin{bmatrix} \mathbf{A} & \mathbf{E} \\ \mathbf{F} & \mathbf{C} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_{\mathcal{A}} \\ \mathbf{u}_{\Gamma} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_{\mathcal{A}} \\ \mathbf{f}_{\Gamma} \end{Bmatrix}$$

$$\mathbf{u}_{\mathcal{A}} = \mathbf{A}^{-1}(\mathbf{f}_{\mathcal{A}} - \mathbf{E}\mathbf{u}_{\Gamma})$$

$$\mathbf{S}\mathbf{u}_{\Gamma} = (\mathbf{C} - \mathbf{F}\mathbf{A}^{-1}\mathbf{E})\mathbf{u}_{\Gamma} = \mathbf{f}_{\Gamma} - \mathbf{F}\mathbf{A}^{-1}\mathbf{u}_{\mathcal{A}}$$

$\mathbf{S}$ in that context is the Schur complement matrix, $\mathbf{C}_{\Gamma}$ describes the variables on the interface. Due to the size of extracted zone containing the plastic deformation and the contact, the matrices $\mathbf{A}_{\mathcal{B}}$ and $\mathbf{S}$ are assumed to be small and solved directly. Thus, for large models, any operation on $\mathbf{A}$ can be interpreted as an operation on $\mathbf{A}_{\mathcal{N}}$. Since the size of $\mathbf{A}_{\mathcal{N}}$ is almost the size of the original problem, the main issue for an efficient algorithm is to circumvent the inversion of $\mathbf{A}$. To do so, two methods are under consideration: a Block-Gaussian-Elimination solving substitute systems (Algorithm 1) and a preconditionend conjugate gradient method for which $\mathbf{S}$ has not to be formed explicitly (Algorithm 2) as suggested in [10].

Algorithm 1 Block-Gaussian-Elimination

```
solve substitute systems: $\mathbf{A}\tilde{\mathbf{E}} = \mathbf{E}$, $\mathbf{A}\tilde{\mathbf{f}}_{\mathcal{A}} = \mathbf{f}_{\mathcal{A}}$ for $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{f}}_{\mathcal{A}}$ resp.
compute $\tilde{\mathbf{f}}_{\Gamma} = \mathbf{f}_{\Gamma} - \mathbf{F}\tilde{\mathbf{f}}_{\mathcal{A}}$ and $\mathbf{S} = \mathbf{C} - \mathbf{F}\tilde{\mathbf{E}}$
solve $\mathbf{S}\mathbf{u}_{\Gamma} = \tilde{\mathbf{f}}_{\Gamma}$
compute $\mathbf{u}_{\mathcal{A}} = \tilde{\mathbf{f}}_{\mathcal{A}} - \tilde{\mathbf{E}}\mathbf{u}_{\Gamma}$
```

Considering again only the large system, the solution of the above system leads to the solution of $(k + 1)$ linear systems of size n for Algorithm 1. k denotes the number of degrees of freedom on the coupling boundary and n is the dimension of $\mathcal{A}_{\mathcal{N}}$.

Instead of Algorithm 1 the inversion of $\mathbf{A}$ can be avoided by usage of a solution scheme in which only matrix-vector products of the schur-matrix are needed. One of those methods is the preconditioned conjugate gradient method with a preconditioner induced by the original matrix (cp. e.g., [10]). The solution process then leads to Algorithm 2.

$\mathbf{M}_K$ in the algorithm denotes an appropriate preconditioner to the original system matrix $\mathbf{K}$. For the computations shown a Symmetric Gauss-Seidel preconditioning was applied. The matrix $\mathbf{R}_{\mathcal{A}}$ in that context describes the restriction operator of all variables onto the variables on the coupling interface. In Algorithm 2 the number of linear systems to solve is dependent on the convergence of the inner CG-method (steps 4 to 11 in Algorithm 2). For the overall solution of the problem $(m + 2)$ linear systems of dimension n have to be solved. m here denotes the number of inner CG-iterations and n the dimension of $\mathcal{A}_{\mathcal{N}}$.

Algorithm 2 CG-method with induced preconditioning and schur-complement

```
solve substitute system: $\mathbf{A}\tilde{\mathbf{f}}_\mathcal{A} = \mathbf{f}_\mathcal{A}$ for $\tilde{\mathbf{f}}_\mathcal{A}$
compute $\tilde{\mathbf{f}}_\Gamma = \mathbf{f}_\Gamma - \mathbf{F}\tilde{\mathbf{f}}_\mathcal{A}$
compute $\mathbf{r}_0 := \tilde{\mathbf{f}}_\Gamma$, $\mathbf{z}_0 = \mathbf{M}_S^{-1}\mathbf{r}_0 = \mathbf{R}_\mathcal{A}\mathbf{M}_K^{-1}\mathbf{R}_\mathcal{A}^T\mathbf{r}_0$ and $\mathbf{p}_0 := \mathbf{z}_0$
for $j = 0, 1, \ldots$ until convergence do
    solve $\mathbf{A}\tilde{\mathbf{w}}_j = \mathbf{E}\mathbf{p}_j$
    compute $\mathbf{w}_j = \mathbf{C} - \mathbf{F}\tilde{\mathbf{w}}_j$
    compute $\alpha_j := (\mathbf{r}_j, \mathbf{z}_j)/(\mathbf{w}_j, \mathbf{p}_j)$, $\mathbf{u}_{\Gamma,j+1} := \mathbf{u}_{\Gamma,j} + \alpha_j\mathbf{p}_j$
    compute $\mathbf{r}_{j+1} := \mathbf{r}_j - \alpha_j\mathbf{w}_j$, $\mathbf{z}_{j+1} := \mathbf{R}_\mathcal{A}\mathbf{M}_K^{-1}\mathbf{R}_\mathcal{A}^T\mathbf{r}_{j+1}$
    compute $\beta_j := (\mathbf{r}_{j+1}, \mathbf{z}_{j+1})/(\mathbf{r}_j, \mathbf{z}_j)$
    compute $\mathbf{p}_{j+1} := \mathbf{z}_{j+1} + \beta_j\mathbf{p}_j$
end for
solve $\mathbf{A}\mathbf{u}_\mathcal{A} = \mathbf{f}_\mathcal{A} - \mathbf{E}\mathbf{u}_\Gamma$
```

Improvement of the matrix conditioning

As mentioned, the main idea of the presented solution scheme is to improve the matrix condition of the system $A_\mathcal{N}$ such that its solution by an iterative solver is accelerated so much, that solving a couple of those systems is still more efficient than solving the original problem. In Table 1 the condition numbers of the different systems are given for the two constitutive laws discussed here.

	$\mathbf{K}$	$\mathbf{A}_\mathcal{N}$	$\mathbf{A}_\mathcal{B}$	$\mathbf{C}_\Gamma$
elasto-plastic material law	$5.08 \cdot 10^8$	$3.85 \cdot 10^3$	$1.07 \cdot 10^7$	$2.21 \cdot 10^1$
rigid-plastic material law	$8.29 \cdot 10^{12}$	$5.34 \cdot 10^9$	$2.24 \cdot 10^{10}$	$1.86 \cdot 10^7$

Table 1: Matrix condition numbers of the different systems

First of all, it becomes obvious that the matrix conditioning of the large system $\mathbf{A}_\mathcal{N}$ is improved in both cases. Considering the rigid-plastic material law, the improvement is much less, though. Obviously, in that case the influence of the penalty formulation enforcing the incompressibility is of significant size in the whole model rather than in the forming zone only. Despite this result, the improvement of the matrix conditioning is a lot larger, when a elasto-plastic material law is considered. According to the mathematical formulation, only the contact penalty is deteriorating the matrix conditioning and thus, only those variables subject to contact can become critical variables. Due to the chosen domain decomposition all those variables reside in the domain $\mathcal{A}_\mathcal{B}$. The result obtained here confirms the potential of the discussed solution scheme. In the next section the solver behaviour on those systems will be discussed.

Performance results and required accuracy

In general, the matrix condition number gives quite a clear picture on the convergence speed of most iterative solvers. Given the results from the section above one would expect a speed up of the used AMG-solver as well. In both algorithms discussed above the decoupled systems have to be solved. For the discussion of the solver performance this will be compared to the solution of the original system. The size of both systems is very similar. The actual residual reduction of the AMG-CG solution is depicted in Figure 4 for both materials under consideration. The described effect of a faster convergence becomes obvious. Other solvers tested (Jacobi/CG, SSOR/CG) showed a similar behaviour.

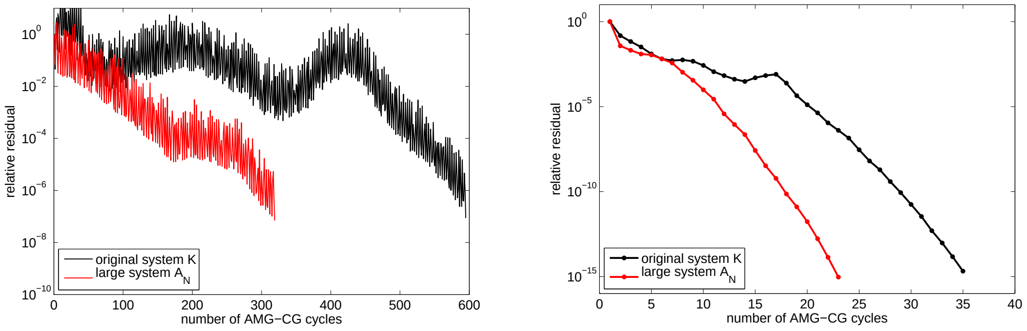


Figure 4: AMG solver convergence upon linear systems $\mathbf{Ku} = \mathbf{b}$ and $\mathbf{A}_{\mathcal{N}}\tilde{\mathbf{f}}_{\mathcal{A}} = \mathbf{f}_{\mathcal{A}}$ for rigid-plastic (left) and elasto-plastic (right) material

Taking the acceleration achieved by itself the domain decomposition under consideration achieves the desired effects. Recalling the described algorithms above, though, the main problem becomes obvious: to circumvent the matrix inversion of the decoupled systems, a couple substitute systems have to be solved.

Converging those systems down to machine precision requires about the same amount of solver cycles for each those systems as shown in Figure 4. On the other hand, solving the linear systems at any stage of the algorithms to a larger relative residual or setting an upper limit for the maximum number of solver cycles for the substitute systems, the overall residual at the end of the computations becomes larger as well. A relative residual of less than 10^{-8} leads in general to a wrong overall solution (see also e.g., [10]). Moreover, in case of Algorithm 2 the PCG-method to solve the Schur system does not converge at all below this criteria. In Figure 5 the described results are depicted.

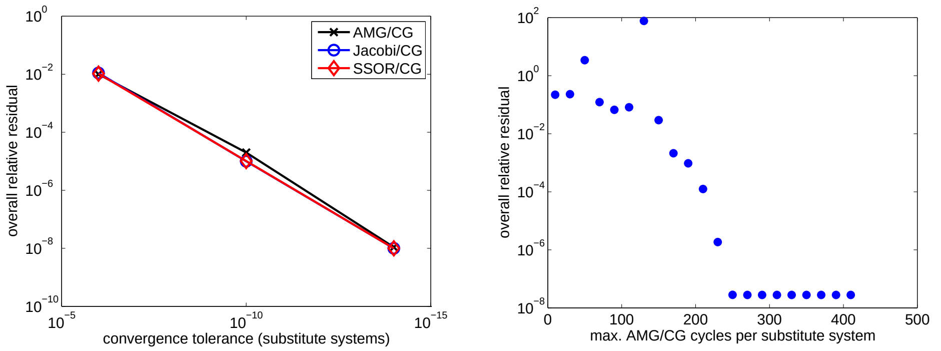


Figure 5: Variation of accuracy (Algorithm 2, left) and maximum number of cycles (Algorithm 1, right) in substitute system solution and overall residual

The described behaviour is in principle independent of the applied solution method to solve the substitute systems. In case of algorithm 2 it is as well independent of the chosen preconditioner of the CG-method to solve the schur system. The AMG/CG solver however shows the best overall complexity compared to the other iterative solvers tested (SSOR/CG, Jacobi/CG, Gauss-Seidel).

Overall efficiency

Taking the results from above into account it becomes obvious that even though the convergence speed of the algebraic multigrid solver could be enhanced, the overall efficiency of the method has not improved compared to the solution of the original system. Table 2 summarizes the number of cycles for some representative computations of the rigid-plastic case. In the same way as it is shown for the rigid-plastic case it applies to the elastic-plastic computations.

For the given problem it becomes obvious that the suggested solution scheme did not succeed to a higher efficiency compared to the original system. Even though the conditioning of the matrices to solve could be improved, the bypass of the matrix inversion by the algorithms presented is too expensive to pay in matters of iteration cycles and therefore computation time. Comparing the two algorithms and assuming an appropriate convergence tolerance for the substitute systems provided, the Block-Gaussian-Elimination (Algorithm 1) seems more robust because of two reasons. First, the number of variables on the coupling interface can always be limited to a small amount due to the process character of incremental forming and the small zone of plasticity and contact. Second, the computations reveal that the convergence of the PCG-method for solution of the schur system in Algorithm 2 is not always guaranteed or sufficiently fast. On the other hand, provided that there is a stronger preconditioner for the Schur-CG-solution, Algorithm 2 might outperform the other approach.

Algorithm 1: $k+1$ systems to solve ($k :=$ number of variables on the interface)			
prescribed max. number of cycles per subst. system	70	180	250
total number of cycles for solution	4816	10348	14276
average relative residual (subst. systems)	$\sim 10^{-9}$	$\sim 10^{-12}$	$\sim 10^{-14}$
overall relative residual	$\sim 10^{-1}$	$\sim 10^{-3}$	$\sim 10^{-8}$

Algorithm 2: $m+2$ systems to solve ($m :=$ number of PCG-cycles for Schur system)			
prescribed convergence tolerance in subst. system	10^{-6}	10^{-10}	10^{-14}
total number of cycles for solution (AMG/CG)	224	18970	27494
total number of cycles for solution (SSOR/CG)	3884	110634	163743
total number of cycles for solution (Jacobi/CG)	6374	334097	440324
overall relative residual	$\sim 10^{-2}$	$\sim 10^{-5}$	$\sim 10^{-8}$

Original system: 1 system to solve	AMG/CG	SSOR/CG	Jacobi/CG
Total number of cycles for solution	586	3843	9374
overall relative residual	$\sim 10^{-8}$	$\sim 10^{-8}$	$\sim 10^{-8}$

Table 2: Iteration statistics to the desired overall residual reduction

6 Conclusions and future work

The present paper considers different solution strategies to enhance the numerical simulation of incremental forming processes. The basic idea is to exploit one of the main characteristics of those processes: There is a very small zone of numerically difficult plastic deformation and contact compared to the total size of the model. To accelerate the solution of the emerging linear systems this property induces a seperate treatment of both regions by extracting the few troublesome terms from the rest of the system. Two approaches were presented: A selective coarsening strategy within the setup phase of an algebraic multigrid solution and an appropriate domain decomposition using the schur complement method. While the selective coarsening approach did not enhance the efficiency of the solution, the suggested domain decompositions improved the properties of the system matrices to solve significantly. Thus, the convergence of several iterative solvers was accelerated. The whole solution process of the algorithms suggested could not beat the solution of the original system in efficiency, though. The reason for that lies in the algorithmic bypass of any matrix inversion during the whole solution process.

Future research will be done to examine the impact of stronger preconditioners to the CG-method, which was used to solve the schur system and its influence on the overall convergence. Moreover, more sophisticated criteria will be investigated for the splitting of the given domain to accelerate the computations further.

Acknowledgements

The work presented in this paper was funded by the German Research Foundation (DFG) in the framework of SPP 1146: Modelling of incremental forming processes.

References

[1] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures.* John Wiley & Sons Ltd, Chichester, 2000.

[2] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182:418–477, 2002.

[3] W.L. Briggs, V.E. Henson, and S. McCormick. *Multigrid Tutorial.* SIAM, Philadephia, 2nd edition edition, 2000.

[4] R. Diez, U. Hindenlang, and A. Kurz. *LARSTRAN User's manual – LARSTRAN documentation.* Stuttgart, 1996.

[5] P. Groche, F. Heislitz, M. Jöckel, S. Jung, C. Rachor, and T. Rathmann. Modelling of incremental forming processes. In *Proceedings of NAFEMS World Congress*, 2001.

[6] W. Hackbusch. *Multigrid Methods and Applications.* Springer, 1985.

[7] V.E. Henson. An algebraic multigrid tutorial. In *Conference Proceedings.* Copper Mountain Conference, 2003.

[8] S. Kobayashi, S. Oh, and T. Altan. *Metal Forming and the Finite Element Method.* Oxford University Press, 1989.

[9] J.W. Ruge and K. Stüben. Algebraic multigrid. In S. McCormick, editor, *Multigrid Methods*, Frontiers in Applied Mathematics. SIAM, 1987.

[10] Y. Saad. *Iterative Methods for Sparse Linear Systems.* 2000.

[11] F. Schmid and M. Schäfer. Performance of Algebraic Multigrid Methods for Simulation of Incremental Forming Processes. In *Proceedings of European Multigrid Conference*, pages 12 pages, CD–ROM, Delft, 2005.

[12] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity.* Springer, 1998.

[13] K. Stüben. Algebraic multigrid (amg), an introduction with applications. Technical Report 70, GMD Report, St. Augustin, 1999.

[14] R.H. Wagoner and J.L. Chenot. *Metal Forming Analysis.* Cambridge University Press, 2001.

[15] P. Wriggers. *Nichtlineare Finite-Elemente-Methoden.* Springer, 2001.

[16] H. Wüstenberg. *Effektive numerische Verfahren zur Behandlung inkompressibler viskoser Strömungsvorgänge.* PhD thesis, Fakultät für Luft- und Raumfahrttechnik der Universität Stuttgart, 1986.