Literature Review on Numerical Methods for Large Thermo-Mechanical Systems

Eva Maquelin
Student number 5144353

MSc Applied Mathematics, Computational Science and Engineering (CSE)
Faculty EEMCS, TU Delft

Supervisor TU Delft: Kees Vuik
Supervisor ASML: Victor Dolk

March 27, 2021

Contents

1 Introduction

This literature review investigates efficient numerical methods for large-scale linear systems of equations, that can be applied in context of model predictive control (MPC) for the thermal and the mechanical part of thermo-mechanical systems. These types of problems are solved at ASML in the next generation Extreme Ultra Violet (EUV) machines. Here, the thermal disturbances induced by the exposure to EUV cause imaging distortions. Such imaging distortions are unwanted, so they have to be actively suppressed. This encounters challenges like dealing with the limited range of a thermal actuator and with the many possibilities of spatially distributed thermal loads. Therefore, high fidelity models are employed. Within ASML, usually the finite element method (FEM) is used to model the thermo-mechanical system, yielding large dynamical systems. For such situations, this review explores efficient numerical methods, in terms of memory, scalability and overall computation time.

The aim is to give an overview of successful algorithms for large-scale linear systems. Recent studies have come up with new ways of combining different solvers, creating even more effective ones. However, as the performance of solvers is generally problem dependent, only an impression of the effectiveness for the specific thermo-mechanical systems can be given.

The report is organized in the following way: the next section presents some well-known basic techniques for solving linear systems. Hereafter, developments are discussed in deflation, algebraic multigrid, model order reduction and domain decomposition methods. The final section gives a conclusion on the found results.

2 Preliminaries

This section introduces methods for solving linear systems of equations, $Ax = b$, that are referred to later in the review. The algorithms itself are not explained – for this, see [1] – but their classification with characteristics and possibly their abbreviation are given.

Methods for solving linear equations can broadly be divided into two categories: direct methods and iterative methods [1]. Direct methods, like LU or Cholesky decomposition, are accurate, reliable and robust. However, they are expensive in terms of memory and computational demands. The accuracy in iterative methods is less than the theoretical machine precision of direct methods for well-conditioned problems. But in many cases, this is justified by the fact that the models themselves also have a certain level of inaccuracy [1].

The two main classes of iterative methods are stationary iterative methods and Krylov subspace methods. Examples of stationary iterative methods are (damped) Jacobi,

Gauss-Seidel and (symmetric) successive over-relaxation ((S)SOR). Examples of Krylov subspace methods are the induced dimension reduction method (IDR(s), see Appendix A), generalized minimum residual method (GMRES), (preconditioned) conjugate gradient ((P)CG), biconjugate gradient (BiCG), conjugate gradient squared (CGS) and BiCG stabilized (BiCGSTAB). Like the direct methods, the stationary iterative methods are not efficient as standalone solvers for large-scale systems, but can be used as preconditioner or smoother. A preconditioner turns the problem into a better conditioned one. A smoother removes fast/(spatially) high frequent error-components [1]. Preconditioning is for instance used in PCG. The convergence of CG depends on the condition number and the amount and distribution of near-zero eigenvalues, since the corresponding eigenvectors do significantly contribute to the solution but may converge slowly. Preconditioning is then used to improve the convergence behavior. The resulting PCG has cheap iterations, is easy to implement and does not require too much memory; though compared to the CG, the preconditioner does increase the work per iteration and memory [2].

Multigrid (MG) methods can also be used as preconditioner to accelerate Krylov subspace methods. They use two complementary processes: smoothing and coarse-grid correction. The smoothing step dampens the high frequent errors components. On the other hand, the coarsening step accelerates the convergence of smooth/low frequent error components. The various MG methods are defined by their choice of smoothing operators, coarsening strategy, interpolation operators and application strategy [3]. The application strategy entails the order in which the coarser grids are visited. A two-grid cycle simply goes from the fine grid to the next coarsest grid and immediately back to the fine grid; but more complex cycles are possible as well. For instance, when at the coarser grid another two-grid cycle is applied before going back to the fine grid, is called the V-cycle; other well known cycles are the W- and F-cycle [4]. In geometric multigrid (GMG) methods, the coarser grids and the transition operators between them are based on the physical properties of the mesh. On the other hand, in algebraic multigrid (AMG) methods, these are based on the matrix-entries, and derived in a purely algebraic sense without explicit knowledge of the geometry [2]. For systems of partial differential equations (PDEs), there are various AMG approaches, like the classical AMG (also known as the Ruge-Stüben AMG) or smoothed aggregation AMG (SA-AMG) [2].

Deflation methods show similarities with the basis of MG methods, but use model order reduction (MOR) techniques instead of coarser grids. MOR techniques can lessen the computational complexity. However, they can destroy the physical interpretation of the states, they are user-intensive and not flexible. Therefore, MOR techniques are not used on their own but considered as a complement, like in deflation methods. To clarify the relation between deflation with multigrid methods, consider a two-grid multigrid cycle solving $Ax = b$. Take as interpolation operator I the deflation-subspace matrix (i.e. the matrix consisting of the deflation vectors), as restriction R its transpose, and as coarse grid operator $A_c := RAI$. If also one preconditioning step is applied with as preconditioner the

deflation matrix[1], then the resulting multigrid is the deflation method. For the deflation method to yield good results, the deflation subspace should contain most of the system's variability; this is usually problem-dependent. Standard choices for deflation vectors are: eigenvectors, recycling solutions (snapshots), subdomain deflation vectors, multigrid and multilevel deflation vectors [5–7].

Proper orthogonal decomposition (POD) is an MOR technique where a basis is constructed via a collection of snapshots. Snapshots are solutions of the system at certain points in time and should capture the dynamics of the system to be solved. They can be obtained via tactics like the recycling deflation approach, moving window approach, and training phase approach [6].

A class of techniques that exploits parallel computing is domain decomposition. These methods rely on the divide-and-conquer strategy, where they aim to solve the problem on the whole domain by dividing it into subdomains and computing solutions on those. This leads to easier to solve subproblems because of e.g. simpler geometry or different modeling equations. The domain decomposition methods are determined by how they deal with the unknown at the interfaces. For small problems, the interface values can be obtained via a block-Gaussian elimination, but this becomes too expensive for larger problems. Then Schwarz Alternating Procedures can used; these alternate between the subdomains, solving the subproblems and updating the boundary conditions. The domain decomposition methods are distinguished by their type of partitioning (along edges, vertices or by elements), the amount of overlap of the subdomains, the processing of interface values and whether the subdomain solution is computed exactly or approximated iteratively [1].

3 Overview of Literature

In this section, literature is discussed on deflation, algebraic multigrid methods, model order reduction techniques and finally domain decomposition.

3.1 Deflation

First, deflation methods are investigated as a way of accelerating iterative methods. Consider solving $Ax = b$ with PCG, where A is symmetric positive semi-definite (SPSD) and has d zero eigenvalues. Then even after preconditioning by M, the spectrum of $M^{-1}A$ can still contain unfavorable eigenvalues that degrade the convergence of PCG and make it less robust. Deflation techniques can be employed to treat these eigenvalues, in order to improve the convergence, robustness and scalability of the method. Deflation works as follows: first solve the deflated system $PA\widehat{x} = Pb$ for $\widehat{x}$, then compute the solution x via

[1]For the definition of the deflation matrix, see Definition 1

$x = Qb + P^T\widehat{x}$. The matrices P and Q used, are defined in the Definition 1, which is given for SPSD coefficient matrices but can be generalized to non-SPSD matrices [7].

Definition 1. Let the coefficient matrix A be as above, and the deflation-subspace matrix $Z \in \mathbb{R}^{n\times k}$ with full rank and $k < n - d$ be given. Then the Galerkin matrix is defined as $E := Z^T AZ \in \mathbb{R}^{k\times k}$ where Z must be chosen such that E is invertible, the correction matrix is $Q := ZE^{-1}Z^T \in \mathbb{R}^{n\times n}$ and the deflation matrix is $P := I - AQ \in \mathbb{R}^{n\times n}$.

The columns of the deflation-subspace matrix Z are called the deflation/projection vectors and are such that E is nonsingular (which is the case if $\mathcal{N}(A) \not\subset \mathcal{R}(Z)$). Eigenvectors can be used as deflation vectors and can be effective in reducing the effective condition number and is expected to accelerate the convergence. However, eigenvectors are often expensive to compute and dense. Meanwhile, the deflation-subspace matrix is wanted to be sparse yet give good approximations of the eigenvectors [7].

Solving large, ill-conditioned systems of equations is investigated in the paper *POD-Based Deflation Method For Reservoir Simulation [6].* It is written by Gabriela Diaz Cortes in 2019 and concerns the simulation of single- and two-phase flow through highly heterogeneous porous media. The paper notes that, usually, preconditioned iterative methods are employed to solve these problems, but that there are other options like reduced order models (ROM) or deflation methods that can be combined with iterative methods. Common options for deflation vectors are expensive to compute (like eigenvectors of the system matrix) or problem dependent (like subdomain vectors). Therefore, the paper introduces a deflation method based on POD. This method requires collecting a set of snapshots from which a POD basis is constructed. Then, the POD basis vectors are used as deflation vectors. The article employs the POD-based deflation method in order to accelerate the Krylov subspace iterative method PCG preconditioned with incomplete Cholesky factorization (DICCG), for the porous media flow problem. Moreover, the article considers multiple options for obtaining the snapshots, compares different deflation methods and compares various two-level preconditioned conjugate gradient (2L-PCG) methods using POD-based deflation.

The numerical analysis shows that for the incompressible single-phase problem, the POD-based deflation subspace works better than eigenvectors of the system matrix and preconditioned matrix, and better than subdomain deflation vectors. For both the compressible single-phase problem and the incompressible two-phase problem, the POD-based deflation method used in DICCG reduces the number of iterations compared to ICCG. Finally, the deflated PCG can be seen as a 2L-PCG method and is compared to other such methods (PCG, DEF1, DEF2, A-DEF1, A-DEF2, BNN, R-BNN1, R-BNN2, ROM, SROM). 2L-PCG methods consist of a traditional single-level preconditioner (IC in this case) and a second-level preconditioner (e.g. the deflation matrix). All methods show a

similar performance in terms of work and number of iterations when using POD-based deflation vectors, reducing the number of ICCG iterations. With 70-72% of the ICCG work, Balancing-Neumann-Neumann (BNN) is the most expensive. The methods that performed best are the deflated PCG (implemented as DEF1, DEF2) and R-BBN2 (obtained by removing the matrices from BNN), which require only 37-40% of the ICCG work.

The advantages of the POD-based deflation method are that it is problem independent: while it is introduced for reservoir simulations, it can be adapted to any time-varying problem. Moreover, it is linear solver independent: while it is applied with PCG, it can be implemented for various preconditioners and linear solvers. The article shows that among the various 2L-PCG methods considered, deflated PCG worked best, when all used POD-based deflation. For further research, G. Diaz Cortes [6] suggests among others a theoretical study of using POD-bases as deflation vectors, investigating the influence of time-stepping, using a solution as initial guess and further development of the methodology (combining parallelization, machine learning, ROM and preconditioners like MG methods together with deflation methods) [6].

Another paper that investigates 2L-PCG methods is the thesis *Two-Level Preconditioner Conjugate Gradient Methods with Applications to Bubbly Flow Problems [7]* from 2008, focussing on deflation methods (DPCG or DEF). As in the previous article, incomplete Cholesky is used as the traditional preconditioner. Deflation can be used as the second preconditioner, but also multigrid or domain decomposition methods can be employed. From an algebraic point of view, these three variants of 2L-PCG methods are rather close or even equivalent.

For the choice of deflation vectors, first approximated eigenvector deflation, recycling deflation and subdomain deflation are compared. The vectors should give good approximations of the eigenvectors corresponding to unfavorable eigenvalues of the coefficient matrix. The preferred deflation vectors depend on the specific problem, but in general, subdomain deflation works best and is used throughout the paper. In this deflation variant, the deflation vectors are determined algebraically: the computational domain is split into multiple subdomains, each corresponding to one or more deflation vectors. Moreover, subdomain deflation does not require any prior knowledge of the density field and can be implemented and parallelized in a straightforward manner. Other deflation approaches are level-set deflation and level-set-subdomain deflation, and could also be attractive. The numerical analysis of the methods is based on bubbly flow problems. Compared to PCG, DEF decreases the computational cost for most test cases and is scalable in terms of iterations and CPU time.

Besides deflation, also some other 2L-PCG methods are investigated. Namely, the additive coarse-grid correction (AD), balancing Neumann-Neumann (BNN) and its reduced variants (R-BNN), and multigrid V(1,0), V(0,1) and V(1,1)-cycles. The difference between the methods is small, so similar convergence behavior is predicted. However, the paper

notes that it has been proven that the deflation variant is theoretically expected to converge faster than AD and BNN. The BNN and multigrid cycles are more robust than DEF and R-BNN. Multigrid V(1,0)-cycle preconditioner is the best method based on effectiveness, efficiency and robustness for a class of problems. Usually, the V(1,1)-cycle is faster than the 2L-PCG methods but requires more work per iteration. When comparing DEF and multigrid V(1,1)-cycle, considering various approaches for the multigrid algorithms, it turns out that DEF and V(1,1)-cycle based on Dendy's blackbox multigrid preconditioner are the most robust and efficient 2L-PCG methods [7].

A different option for deflation vectors is using multigrid vectors, which are the columns of the matrix representing the interpolation operator. The paper *Accelerating the shifted Laplace preconditioner for the Helmholtz equation by multilevel deflation [8]* (2016) uses deflation with multigrid vectors. Two-level deflation is a deflation method where the deflation vectors (i.e. the columns of the deflation-subspace matrix) are multigrid vectors. The deflation-subspace matrix then represents the interpolation operator from the coarse to the fine grid. Multilevel deflation is used if in the two-level method, the coarse grid problem remains too large to solve exactly via matrix inversion. The multilevel extension is achieved by introducing a shift and solving the coarse grid problem by going to even coarser grids, until the problem is small enough.

The Helmholtz equation is a physical model for waves. After discretization, the resulting coefficient matrix is complex, symmetric, non-Hermitian and indefinite. Most iterative methods do not give good results for this equation. The paper considers complex shifted Laplacian preconditioners (CSLP) which via damping make the system easily solvable by for example multigrid methods. However, the number of the outer Krylov subspace iterations increases with the wave number. As a solution, (multilevel) deflation techniques are introduced, in two possible ways. The first option is deflate the CSLP preconditioned system (first precondition, then deflate). The second option is to deflate the original Helmholtz operator and combine it multiplicatively with the CSLP preconditioner (first deflate, then precondition). Both options are preconditioned multilevel Krylov methods (which resemble MG methods, but the coarse-grid problem is solved by a Krylov method), using a flexible Krylov method for every level. The first option yields better clustering, but also has a higher computational cost. Numerical results confirm that deflation reduces the amount of iterations needed [8].

Also concerning the Helmholtz equation is the very recent (2020) article *Scalable Convergence using Two-Level Deflation Preconditioning for the Helmholtz Equation [9].* It concerns solving the Helmholtz equation via deflation. Deflation techniques for accelerating the convergence of Krylov subspace methods have been investigated in recent research like the previous article. For larger wave numbers, the CSLP shifts eigenvalues of the

preconditioned system towards zero. The combination of CSLP with a two-level-deflation preconditioner seemed promising in terms of reducing this phenomenon. However, for large wave numbers, small eigenvalues reappear. This is due to a misalignment of the near-singular eigenmodes of the fine- and coarse-grid operators. Therefore, the article suggests using higher-order approximation schemes to construct the deflation vectors. The resulting method yields better results than the other deflation-based preconditioners for the Helmholtz equation, and the convergence is almost independent of the wave number [9].

The paper *Comparison of the deflated preconditioned conjugate gradient method and algebraic multigrid for composite materials [2]* from 2011 focuses on parallel preconditioners for finite element problems in structural mechanics. In structural mechanics, many applications involve composite materials, where large discontinuities in material properties cause many small eigenvalues. The problems are translated via FEM to large linear systems. These are difficult to solve since the small eigenvalues slow down the convergence of iterative methods like PCG using a simple preconditioner. There exists a correlation between the number of rigid body modes (RBM) and the number of small eigenvalues of the stiffness matrix. Therefore, the RBM of elements with homogeneous material properties are used as deflation vectors, in order to remove those small eigenvalues, resulting in a more stable and robust method. Nowadays, parallel computing is the standard in finite element software packages, so only parallel algorithms are discussed in this paper. As the stiffness matrix is SPD and PCG is well parallelizable, the PCG method is employed. The paper compares deflated PCG (DPCG) and PCG preconditioned with an AMG method (more information on AMG is given in the next section). As preconditioner, two variants of AMG smoothed aggregation (SA-AMG) methods are employed, and compared to diagonal scaling (where one choses diagonal matrices D_1, D_2 such that the condition number of D_1AD_2 is minimal) in numerical simulations on asphalt concrete. For these problems, the DPCG method proves to be efficient, robust and parallelizable, as it decouples regions with homogeneous material properties via a subdomain deflation technique.

The DPCG method is compared to using SA-AMG – the best AMG adaptation – as preconditioner, which is known to be a successful parallel preconditioner for various structural mechanics applications. The article compares PCG with as preconditioners diagonal scaling, default and optimal SA-AMG, against DPCG with as preconditioners diagonal scaling and default SA-AMG; all implemented in a parallel setting. These methods are applied to meshes derived from real-life samples of asphaltic material and the following is observed.

- DPCG preconditioned with diagonal scaling has a low cost per iteration and reaches the solution much faster than PCG with diagonal scaling or even with default SA-AMG. It is expected that this is due to the fact that deflation and scaling are complementary operations, operating on respectively the lower and upper part of the spectrum.

- DPCG preconditioned with default SA-AMG reduces the number of iterations compared to diagonal scaling, but does not reach the solution faster.
- For the larger test problems considered (roughly 3 million degrees of freedom), PCG preconditioned with optimized SA-AMG does outperform the DPCG preconditioned with diagonal scaling. However, PCG with optimized SA-AMG needs significantly more software development effort than the much simpler DPCG with diagonal scaling.

While the DPCG has higher memory demands and computation costs, these are worth it since the deflation ensures convergence even for highly ill-conditioned problems, giving more accurate results than PCG. Thus, DPCG is efficient, scalable, robust; it improves the convergence and computation time, and is easily implemented and parallelized [2].

RBM are also used in the article *On the use of rigid body modes in the deflated preconditioned conjugate gradient method [10]* (2013), which considers mechanical problems with materials that have strongly varying stiffness. The linear equations arising from problems with large discontinuities in material properties, are ill-conditioned systems. Namely, the discontinuities cause small eigenvalues that can deteriorate the convergence of iterative methods; like noted in the discussion of the previous article considered. This paper considers as iterative method DPCG. The RBM of the sets of elements with homogeneous material properties are used to construct the deflation space, to remove those small eigenvalues corresponding to the slowly converging solution-components. However, a different preconditioner is employed than in the previous article. Namely, in the previous article diagonal scaling and smoothed aggregation were used as preconditioners, while now diagonal scaling and incomplete Cholesky are considered. The convergence of the resulting DPCG is independent of the discontinuities in the material properties.

RBM deflation broadens the scope of applicability of deflation techniques, as it can be used for discretized coupled PDEs. Using RBM to accelerate computations has also been employed in AMG methods, among others. The parallel implementation of DPCG was compared with SA-AMG, which is a state-of-the-art solver for these mechanical problems. Numerical experiments showed that DPCG is quite competitive with SA-AMG. For most applications, when using sparse deflation vectors, the cost of speeding up via deflation is that DPCG is takes roughly 30% more time per iteration than PCG [10].

3.2 Algebraic Multigrid

In the 1980s, the classical AMG methods were introduced, which are efficient for M-matrices and Poisson models. AMG determines the transition operators between grids (restriction and interpolation/prolongation) automatically. The methods are based on the observation that the near-kernel of the operator A can be approximated by a constant vector. But this

limits their use in for example elasticity problems [3,11]. Smoothed aggregation AMG (SA-AMG) is an AMG method modified to solve elasticity equations [2]. Also, methods like the adaptive AMG (αAMG), Bootstrap AMG (BAMG) and AMG based on element interpolation (AMGe) have been developed to solve problems where the classical and smoothed AMG do not work well. Contrary to classical AMG, for these methods there are no restrictions on the near-kernel of A. Instead, they approximate the near-kernel adaptively. The main idea of adaptive AMG is finding algebraically smooth modes (vectors that are not damped by relaxation) of A. In elasticity problems, RBM can be used to define an approximation of (part of) the near-kernel beforehand. This can be exploited to create effective methods in e.g. aggregation-based AMG, domain decomposition, and deflation methods as was shown in the previous section [3].

Focussing on parallel implementation, there are two general-purpose parallel AMG codes, namely BoomerAMG[2] for classical AMG and ML[3] for SA-AMG. It is shown that SA-AMG generally performs better than classical AMG. A generalized version of SA-AMG is even more robust, but also more expensive. The parallel smoother employed in SA-AMG has a big impact on the performance of the methods. It is shown that when comparing parallel hybrid Gauss-Seidel orderings with polynomial (Chebyshev) smoothers, the polynomial smoothers are preferred. Therefore, the article [2] discussed in the previous section used Chebyshev smoothers for both the default and optimal version of SA-AMG [2].

The article *Algebraic Multigrid Based On Element Interpolation (AMGe) [11]* (2002) introduces AMGe. In MG methods, the interpolation operator must be constructed such that its range can well-approximate the error components remaining after relaxation, called the smooth error components. This brings forward the problem of determining smooth error components. Here, two local measures are suggested to find local representations of such error components.

As mentioned, classical AMG was designed for M-matrices, where the strength of connections (SoC) is easily measured. The SoC is used in the coarsening process as it can be used to determine which variables represent smooth error components. However, for many problems (when the matrix is not an M-matrix) the SoC is not as easily measured, and AMG has to be modified. There are many different AMG versions, constructed for specific problems. Generally, these methods work well on the problems for which they are designed, but break down in different situations or it cannot be determined beforehand whether or not they will be effective. Therefore, the AMGe method – for solving PDEs discretized by Ritz-type FEMs – is introduced, with the aim of having a method that is robust: "it is expected to perform well in more general problems involving high aspect ratios, so they should be widely applicable for problems based on unstructured grids having

[2]Contained in the Hypre package (https://hypre.readthedocs.io/en/latest/solvers-boomeramg.html)
[3]Algebraic multi-level preconditioner package, contained in Trilinos project (https://trilinos.github.io/ml.html)

thin domains or regions". But this wider applicability comes at the cost of needing more information than in classical AMG, namely access to the element stiffness matrices. These matrices are used to localize two existing global measures M_1 and M_2, discussed in more detail later. The resulting local measures are in turn used to find smooth error components and to determine the interpolation operator. The restriction matrix is then chosen to be the transpose of interpolation matrix, and the Galerkin coarse-grid operator is used. The method derived from M_1 is denoted AMGe1, and from M_2 is AMGe2.

Consider the linear system $Au = f$ arising from finite element discretizations, where A is SPD and can be expressed as the sum of known finite element stiffness matrices. In MG methods, the relaxation and coarse-grid correction must be complement to each other, i.e. errors not reduced by the relaxation must be reduced by the coarse-grid correction and vice versa. This article uses a simple point-wise method as relaxation – like Richardson, damped Jacobi or Gauss-Seidel – which reduces error components in the direction of eigenvectors corresponding to large eigenvalues quickly, while those corresponding to small eigenvalues are reduced slowly. In classical AMG, multigrid components are constructed based on the properties of M-matrices causing smooth error components to vary slowest in the direction of strong dependence. Classical AMG is not necessarily restricted to M-matrices, but does require this feature associated with M-matrices regarding smooth error components. To get a more robust method, this feature is here replaced by the requirement of the interpolation operator to approximate eigenvectors of A with an error bound proportional to the size of its corresponding eigenvalue; to ensure it is complement to the relaxation. Note that in the next article [3] discussed, a method is investigated that instead uses a more generally applicable SoC definition. The M_1 and M_2 measure how well this new requirement on the interpolation operator is satisfied, but do so globally. In practice, we want to measure the quality of the interpolation locally, hence the measures are localized. The requirement then relates the interpolation quality to local eigenvectors. To localize the measures, access is needed to the finite element stiffness matrices. The interpolation matrix is then constructed by solving a constrained min-max problem, that aims to minimize the bound on the localized measure while maximizing sparsity of the coarse-grid operator.

In the numerical analysis, two test cases are considered: a Poisson equation discretized on stretched quadrilaterals, and a plane-stress cantilever beam with various thicknesses of the beam. Three versions of standard AMG, AMGe1, AMGe2, and the CWF method (presented by Chang et al. [12]) are applied. The coarse grids are constructed the same way for AMG and AMGe, but in AMGe the interpolation matrix is based on elements and hence is different than for AMG. In the stretched quadrilaterals problem, the AMGe methods improve upon one of the three AMG variants. While the other two AMG variants give results similar to AMGe, it could not be determined beforehand how they would perform. On the other hand, from AMGe it was expected that they would yield good results. For the plane-stress cantilever beam problems, the AMGe methods are independent of the beam size or improve for thinner beams, while the AMG methods and even GMG become less stable for thinner beams.

For both test cases, AMGe methods outperformed the AMG methods, yielding better convergence rates. As further research the article suggests that the construction of the coarse grid could be improved by using the information in the local measures [11].

In *Solving Tensor Structured Problems with Computational Tensor Algebra [13]* (2010), a tensor multigrid (TMG) algorithm is presented. Tensors are here considered as objects with more than two dimensions, like higher dimensional matrices (a more detailed description of tensors is given in MOR section). Multidimensional problems are viewed as tensor structured problems and a framework is introduced to solve them via tensor algebra. Usually, multidimensional problems are reshaped into matrices and vectors, which can be solved with the common methods of matrix algebra, and then the solution is changed back into its original form. However, this way the multidimensional structure of the problem is lost, possibly deteriorating the convergence of the solvers. On the other hand, the new framework preserves the structure and data coherence of the problems. A tensor space is an outer product of multiple vector spaces. In this paper, the vector spaces have a unique predefined order. This makes the tensor multiplication commutative. Also, note that all the properties of a tensor are gathered in its component values and indices. This framework enables automated optimization of solving algorithms. Moreover, the decomposition of the tensor multiplication allows parallelization of the computations.

To solve the tensor equations, tensor modifications of well-known solvers are considered. Direct tensor methods, like tensor extensions of Gauss elimination or LU decomposition, are not effective for large systems of tensor equations, similar to the non-tensor case. Then, use iterative tensor solvers. Stationary iterative tensor solvers, like the tensor Jacobi method, reduce the amount of memory needed, but do not converge for all problems. Therefore, Krylov subspace tensor solvers, like tensor CG, are developed, which perform better for most problems. As final class, the tensor multigrid (TMG) methods are introduced. Recall that AMG derives meshes and operators based on the matrix entries only, but risks the loss of data coherence which could deteriorate the convergence. On the other hand, GMG does preserve spatial coherence in multidimensions, but is problem dependent. Via tensors, the two methods are combined in the TMG algorithm. TMG is similar to AMG, but due the the use of tensors, it can preserve the multidimensional structure and spatial data coherence like in GMG. TMG and AMG are compared numerically for a test-problem of image reconstruction, where TMG converges faster and gives a better solution.

The paper introduced a tensor framework in which to solve multidimensional problems. This, contrary to the matrix-approach, maintained the structure of the problems and ensured commutativity of tensor multiplications. This resulted in computationally efficient solvers, benefitting from automatic expression analysis, and separability properties [13].

The paper *A robust adaptive algebraic multigrid linear solver for structural mechanics [3]* (2019) considers numerical simulations of structural mechanics problems via finite elements. This requires solving large, ill-conditioned linear systems of equations, hence a robust and efficient linear solver is sought. Direct solvers are only able to reach superlinear complexity at most. However, iterative solvers combined with AMG preconditioners can reach up to linear complexity if the problem is sufficiently regular. However, these methods do not guarantee convergence and need more user-knowledge for an efficient setup. Therefore, the paper presents an adaptive AMG method which is designed to improve its usability and efficiency for structural problems.

The article continues on the the adaptive Smoothing and Prolongation based Algebraic Multigrid (aSP-AMG) method proposed in [14]. The modifications made improve its applicability for large-scale and challenging SPD linear systems resulting from linear elasticity PDEs. The aSP-AMG method automatically constructs an approximation of the near-kernel of the system matrix. The paper proposes a new way of determining the interpolation operators in a least-squares sense. Moreover, it automatically tunes the accuracy of the Adaptive Factorized Sparse Approximate Inverse (aFSAI), which is used as a smoother and has a very high degree of parallelism. Coarsening algorithms rely on the SoC, which measures the connection between nodes. aSP-AMG uses an affinity-based SoC, which is wider applicable than the commonly used definition of SoC. It needs a test space, which is a matrix with smooth modes as its columns. The Simultaneous Rayleigh Quotient Minimization by Conjugate Gradients (SRQCG) is used to compute the test space, as its initial convergence is often faster than that of other eigensolvers like the Lanczos algorithm. For the adaptive prolongation, an interpolation operator is constructed – using the dynamic pattern least squares (DPLS) algorithm – that is close to the set of test vectors.

In the sensitivity analysis the most important configuration parameters for aSP-AMG, their useful range and prospective default value are determined. From this, it is shown that the majority can be set to a default value without compromising their performance. Then, the aSP-AMG (as preconditioner for PCG) is compared to state-of-the-art MG methods like GAMG (geometric agglomerated algebraic multigrid) and BoomerAMG. They are applied to real-world structural problems, chosen for their large size and sources of ill-conditioness. In most cases, the aSP-AMG method has smallest computation time. Moreover, it still yields good results when using the default parameter values, hence decreasing the need for fine tuning.

As next steps, the paper suggests among others extending to a block version, where x, y and z unknowns of each physical node are grouped together [3].

3.3 Model Order Reduction

The goal of MOR techniques is to reduce the complexity of large systems of equations, resulting in a computationally efficient model yet still giving an accurate representation of the original system [15].

A tensor decomposition approach to data compression and approximation of ND systems [15] (2010) introduces a new MOR technique for systems with multiple independent variables by combining the techniques of tensor decomposition and POD. The resulting method has as benefit that the multidimensional structure of the original model is maintained. Earlier work mostly focussed on the evolution of systems with as only independent variable the time. This paper aims to obtain reduced order models for multidimensional systems, by explicitly taking the structure of the various independent variables into account. While POD can already be applied to such systems, it only separates space and time and no further structure is assumed regarding the space-domain. So basically, POD is a method considering two variables. POD is generalized by assuming a more general Cartesian structure for the independent (spatial) variables and employing tensors, so that the multidimensionality of these variables is taken into account. Also, prior knowledge of the structure of the problem can be used to improve the approximations.

The paper first introduces the original POD method. Consider the PDE describing a linear distributed system in a signal evolving over multiple independent variables. The domain is assumed to have a Cartesian structure, which is in most cases the product of a spatial and temporal domain. A Hilbert space of functions on the spatial domain is considered, and (given some assumptions) every solution allows a spectral expansion in the orthonormal basis functions of this Hilbert space. Given some $r > 0$, the lower rank r approximation of the solution is defined by a truncation of this expansion. The reduced order model is defined by the set of lower rank approximations satisfying the Galerkin projection for all functions in the finite dimensional projection space being the span of the first r basis functions of the Hilbert space. This way, the original PDE is reduced to an approximate model consisting of r ODEs. The quality of this reduced model is determined by its choice of basis functions for the Hilbert space. For POD, these are determined empirically, from a set of measured/simulated data representative for the model. The basis functions are the solution to a constrained optimization problem, and they minimize the integrated error between the solution and all its truncated rank r approximations. Moreover, the POD basis is determined by the eigenfunctions of the data correlation operator. In many applications, FEM can be used to discretize the PDE, yielding a difference equation. The finite element solutions are defined on a domain of finite cardinality and also the (solution) Hilbert space becomes finite dimensional. Finding the POD basis functions then is simply an algebraic eigenvalue/singular value decomposition problem.

As mentioned, the original POD puts all spatial variables together. Therefore, modifications based on tensors are introduced to preserve the multidimensional nature of the sys-

tem. The data in multiple independent variables can be associated with a tensor. Namely, assume that the spatial domain itself also has a Cartesian structure. Then the data can be stored in a multidimensional array, which in turn defines a tensor. The projection spaces are constructed via orthonormal decompositions of the tensor, and are then used to construct reduced order models via Galerkin projections on equation residuals. Here, tensors are defined as multilinear functionals. A tensor decomposition is an expansion of the tensor in more elementary building blocks, like low-rank tensors. For matrices, decomposition is done via SVD, but for higher order tensors this is not as straightforward. The paper considers the Tucker decomposition of tensors and proposes an algorithm to compute the the singular values, vectors and singular value core tensor. The tensor SVD (TSVD) is then used to determine a projection basis to construct reduced order models.

While the TSVD decomposes the entire tensor, one might only be interested in the orthonormal bases for the vector space of functions on the spatial domain, not the temporal domain. Therefore, the TSVD can be modified such that only these vector spaces are orthonormalized, removing unnecessary constraints from the optimization problem (namely the constraints orthonormalizing vector spaces of functions defined on temporal domain). This results in more accurate models since there is more freedom to capture information of the original data in the POD basis. The new construction for decomposing a tensor is by dropping these constraints and then computing the dedicated[4] singular values and vectors of the tensor. The dedicated representation is used to define the dedicated modal truncation.

To conclude, the paper adapted the POD method such that the multidimensional nature is explicitly considered. First, it was shown how the new POD basis functions are computed via a lower rank decomposition of a tensor. Second, it was shown how prior knowledge can be used to yield better approximations, illustrated by examples [15].

The article *Tensor-based reduced order modeling in reservoir engineering: An application to production optimization [16]* from 2015 introduces tensor-based MOR for reservoir engineering. In reservoir simulations, the evolution of saturation and pressure over time and space are simulated, generally represented with a three-dimensional Cartesian plane. Empirical subspaces are created via tensor representations of flow profiles. The paper considers the Single Directional Modal-rank decomposition (SDM) for the decomposition of tensors. The flow equations are then projected independently in every physical dimension onto these empirical tensor subspaces via Galerkin projections. This results in reduced order approximations of the original mass and momentum conservation equations. This tensor MOR technique is employed in the setting of water flooding to compute gradient-based optimal production strategies via tensor-based reduced order adjoints. In such optimization problems to maximize the financial output of a reservoir, large

[4] *dedicated* stands for dropping certain orthonormalization constraints in the computation of the singular values and vectors

numerical reservoir models are used as equality constraints. These constraints can be solved using MOR. The original POD method has limitations in reservoir engineering. Namely, the obtained reduced reservoir models are often unstable or inaccurate, due to the highly nonlinear systems with gravity terms in the flow equations. To overcome these problems, tensors are used to maintain the spatial correlation that get lost in the classical POD projection spaces; like noted in the previous article.

In the numerical analysis, both POD and tensor MOR techniques are applied to a reservoir model. Projection onto the POD subspaces means the loss of the diffusive-convective nature of the model. On the other hand, the tensor MOR technique does represent most of the dynamical characteristics of the model; but the accuracy of the tensor models depends on method for generating the empirical projection spaces. The computational gains were only 5% to 10% for POD and tensor MOR, compared to the full model. This is rather low and is a known limitation of projection-based methods, and methods are known to improve on this like trajectory piecewise linearization (TPWL) [17]. Also, the MOR techniques are applied to optimization problems related to water flooding. There, POD and tensor-based MOR are compared for the financial performance. POD encounters limitations in terms of accuracy, while the tensor strategy yields better results and is close to the optimal strategy for the full model.

The advantages of using tensor representations over POD-based MOR techniques are the higher approximation accuracy and their application in optimization problems [16].

3.4 Domain Decomposition

Domain decomposition (DD) methods are generally inefficient as stand-alone solvers, but can yield efficient (parallel) preconditioners. DD divides the domain into multiple subdomains, consequently splitting the problem into multiple subproblems which are coupled through their interface-values. The coupling is then replaced by an iterative process. Efficient DD preconditioners consist of both local and global components. The strong connections between neighboring subdomains are captured in the local part. The global component (a.k.a. the "coarse space correction") allows cheap communications among all subdomains. In homogeneous DD, every subdomain has the same model and discretization, while in heterogeneous DD these can vary per subdomain [18].

The thesis *Domain decomposition preconditioners: theoretical properties, applications to the compressible Euler equations, parallel aspects [18]* (2003) suggests efficient parallel preconditioners based on DD techniques, for problems with unstructured grids. Domain decomposition preconditioners based on both overlapping and non-overlapping decompositions are proposed and applied to the compressible Euler equations (CEE, a classic aerodynamic problem) in a parallel setting. Moreover, grid adaptation is used to improve the results and optimize the use of computational resources. The final framework is called

$\alpha\Psi NKS$, where the α stands for grid adaptation, ΨN for the pseudo-transient continuation Newton method, K for Krylov subspace method and S for the DD preconditioner (either Schwarz or Schur complement based).

The large class of homogeneous DD methods is split into two groups: overlapping and non-overlapping subdomains. The overlapping a.k.a. Schwarz group ensures communication among the subdomains through the overlapping areas. However, these overlapping areas are not enough to get scalability. Considering the simplest one-level Schwarz preconditioners, they are not scalable – i.e. the performance degrades as the number of processors increases – since information is exchanged only in this local manner. The Schwarz methods are made scalable by introducing a coarser grid, here constructed via (smoothed) aggregation/agglomeration procedures. This is an algebraic procedure which does not require a geometric definition of the grid, but operates on the matrix entries. The advantage is that it can be used on unstructured grids and does not require any user-input besides the matrix and the dimension of the coarse space. The resulting two-level Schwarz preconditioners have an additional coarse level correction term on the subdomains. Typically, the group of nodes in a subdomain form an aggregate. Among the two-level Schwarz methods the additive and multiplicative Schwarz methods are considered. In the context of parallel computing, the additive Schwarz method is preferred over the multiplicative one.

In the non-overlapping group, the unknowns are divided into two sets: those on the interface between or internal nodes within subdomains. By "condensing" the internal unknowns, the Schur complement (SC) matrix is formed. The solution is determined by computing the interface unknowns and then computing the internal unknowns for each subdomain by solving the independent subproblems. The SC method is derived from element- (EO) or vertex-oriented (VO) non-overlapping decompositions. In EO respectively VO decompositions, the domain is decomposed such that each element respectively vertex of the grid belongs to a different subdomain. The advantage of the VO decompositions is that the local operators are derived directly from the assembled global matrix, while for EO decompositions this involves the problem dependent assembly process. The VO decomposition suggested to use is such that one subdomain is connected to all the other disconnected subdomains; allowing fast information-exchange and the definition of global preconditioners. For the resulting SC matrix, various preconditioners are investigated, like the Swiss carpet preconditioner which is a Dirichlet-Neumann domain decomposition method[5]. The Swiss carpet preconditioner for VO is compared to the state-of-the-art BNN preconditioner for EO. While BNN requires fewer iterations to converge, its preconditioning phase is much more expensive in both CPU-time and memory. Also, a class of approximate Schur complement (ASC) preconditioners is suggested since the exact SC matrix is dense and expensive to compute.

The CEE is discretized via a multidimensional upwind residual distribution (MURD) scheme, which is applicable to unstructured grids. The resulting large system of nonlinear

[5]For more information on the Swiss carpet preconditioner, see [19]

equations which is solved via the Newton method. At each Newton iteration, a linear system with the Jacobian matrix is solved via a Krylov subspace method with a preconditioner based on DD techniques. To get even better results and improve the use of computational resources, a-posteriori adaption cycles are employed. These mesh adaptation procedures decrease the mesh size when the solution is not yet accurate enough, and decrease the number of elements where the solution is accurate. The resulting parallel algorithm $\alpha\Psi NKS$ combines a non-linear solver, Krylov accelerator, DD preconditioners (of both the Schwarz and SC-based group), and mesh adaptation procedures. The numerical results for aeronautical test problems show that this algorithm improves on the results from MURD schemes.

Finally, the numerical simulations are performed, on distributed memory parallel computers. First, the Krylov accelerators Bi-CGSTAB, CGS, TFQMR, GMRES(25) and GMRES(60) are applied to a test problem with as preconditioner the one-level Schwarz preconditioner with ILU(0) and one-element of overlap among the subdomains. The size of the overlap is a compromise between the number of iterations required for convergence and the CPU-time, and this minimal overlap used is optimal. Regarding the Krylov accelerators, it is found that GMRES is preferred if there are no memory restrictions, otherwise TFQMR should be used. Next, within the two-level Schwarz methods, the additive and hybrid version are compared, also with the one-level Schwarz method, based on CPU-time. The numerical results show that for low CFL numbers, the one-level Schwarz preconditioner is preferred. Otherwise, the hybrid two-level Schwarz preconditioner is preferred. Finally, the SC-based methods are analyzed. For the test cases considered, the SC solver requires more CPU-time than methods operating on the unreduced matrix. But note that the performance depends on the computer architecture and processors used in the simulations. It is suspected that – due to the cost of solving the internal problems – the SC solver only performs well when the there are few unknowns compared to the number of processors, or when the the CPU speed is high compared to the communication speed. When investigating the ASC preconditioner, its scalability turns out to be rather good, but it requires more CPU-time than the two-level hybrid Schwarz preconditioner. Though, both methods have superlinear speedups, and the ASC is an effective method for the test-problem. Overall, the numerical analysis showed that combining Krylov accelerators with Schwarz preconditioners and a mesh adaptation procedure yields effective and scalable methods for solving CEE on unstructured grids [18].

The paper *Comparison of domain decomposition methods for elliptical partial differential problems with unstructured meshes [20]* (2012) parallelizes finite element simulations via the non-overlapping DD methods: Schur Complement (SC) method and the Finite Element Tearing and Interconnecting (FETI) method. The paper considers finite element problems concerning two-dimensional linear elliptic PDEs and introduces a parallel solution method.

The FEM requires solving large systems of equations, where time can be saved by

distributing computations over multiple parallel processors. First, the finite element domain is partitioned into subdomains such that each contains approximately the same number of nodes. Usually, the number of subdomains equals the number of processors. To ensure the computational load is divided equally over the processors and to minimize the communication required amongst them, minimize the number of subdomains per processor and the number of common elements between them. The paper uses a combination of Gmsh[6] and METIS[7] algorithm for the domain discretization and partitioning. Then, a DD method splits the large-scale problem into several smaller interconnecting subproblems. The independent subproblems can be solved in parallel.

Consider the linear problem $Kx = b$ where K is SPD, arising from discretizing a static field on the domain. Two DD methods are investigated to solve this problem. First, the SC method is used. After dividing the domain into subdomains, each subdomain is assigned to an independent processor and the equation is split into blocks. The unknowns on the interface boundaries of the subdomains are computed via the Schur complement matrix; this is called the coarse grid problem, as it only uses unknowns of the interior boundaries. While these cannot be computed in parallel, the system is much smaller than the original problem. Then, the unknowns in each subdomain can be computed in parallel. Note that after computing the unknowns at the subdomains interface boundaries, their values must be communicated to other processors in order to compute the unknowns in the subdomains. The cost of this communication is worth it if the problem is large enough. Next, the FETI method is investigated, which is a powerful and popular solver. It replaces $Kx = b$ by a system of equations incorporating a vector of Lagrange multipliers introduced for enforcing a constraint on the subdomain interface boundaries. In the resulting equations floating subdomains can be encountered, which are subdomains where there are not enough Dirichlet boundary conditions. To ensure solvability also in these cases a condition is added, and some algebraic manipulations then yield the FETI interface problem, which is to be solved for the Lagrange multiplier vector. This can be done via an iterative algorithm, or as is done here via a direct solver based on splitting the vector.

In the numerical analysis, two problems are considered: single-phase transformer and parallel-plate capacitor, which are both static problems resulting in elliptic PDEs. The former problem is considered for two sizes of degrees of freedom (DOFs). The problems are solved by both DD methods on a massively parallel computer, for various amounts of processors and unknowns per subdomain. Due to the symmetry of the test problems, they only need to be solved on a quarter of their domain. However, when many processors are used, FETI benefits from solving the full problem as this does not contain floating subdomains, and is done with a mesh size as large as the quarter problem. The FETI method is faster than Schur, except when many processors (around 6 or higher) are used, then Schur is faster on the quarter problem. However, when FETI solves the full problem

[6]Gmsh is an open source three-dimensional finite element mesh generator (https://gmsh.info)

[7]METIS is a set of serial programs for partitioning graphs and finite element meshes, and for producing fill reducing orderings for sparse matrices (http://glaros.dtc.umn.edu/gkhome/views/metis)

it is faster again. Note, both methods are for all the number of processors considered (2 up to 8) faster than the sequential computation. The speedup by adding processors compared to two processors is also investigated. For the single-phase transformer, a larger number of DOFs gives a greater speed-up for the full FETI method. For the Schur method, its speedup increases when there are more processors, for all test cases.

This paper compared execution time and speedup for the FETI and SC method for two problems. The parallel FEM works properly with both FETI and SC, namely, for both the execution time decreases when more processors are employed. The speedup for eight processors (compared to two processors) was seven times and five times for SC and FETI, respectively. For further research, the paper suggests investigating more complex, large two- and three-dimensional problems, and constructing preconditioned iterative solvers for the FETI method, which can handle subdomains with Neumann boundary conditions [20].

4 Conclusion & Discussion

This literature review has given an overview of various methods used to solve large systems of equations. Deflation, algebraic multigrid (AMG), model order reduction (MOR) and domain decomposition (DD) have been reviewed to determine what might be good methods for solving large thermo-mechanical models.

First, papers on deflation methods as a way of accelerating iterative methods have been investigated. The deflation techniques treat the unfavorable eigenvalues that degrade the convergence of the iterative method. [7] shows that subdomain deflation works best in general, compared to eigenvector deflation and recycling deflation. [6] introduces POD-based deflation methods for reservoir simulation, which improve upon deflation subspaces based on eigenvectors of the system matrix and preconditioned matrix, and subdomain deflation vectors. The deflated PCG preconditioned with incomplete Cholesky (DICCG) reduces the number of iterations compared to ICCG, and works well compared to various other 2L-PCG methods as well. [8, 9] both use multigrid vectors (i.e. the columns of the interpolation matrix) as deflation vectors. Combining this deflation with CSLP for the Helmholtz equation reduces the amount of iterations needed. Using higher-order approximation schemes to construct the deflation vectors yields convergence almost independent of wave number. [2, 10] use RBM of elements with homogeneous material properties as deflation vectors, for composite material problems in structural mechanics. DPCG preconditioned with diagonal scaling is faster than PCG with diagonal scaling or even default SA-AMG. While DPCG has higher memory demands and computational costs, these are worth it since the deflation ensures convergence even for highly ill-conditioned problems.

Second, some papers on AMG methods have been discussed. AMG is a class of multigrid methods where the hierarchy of meshes and the transition operators between them are derived in a purely algebraic sense using only the matrix-entries, not requiring explicit knowledge of the geometry. [11] introduces AMG based on element interpolation (AMGe) for solving PDEs discretized by Ritz-type FEMs. Both of the two versions of AMGe outperform the AMG methods considered. [13] presents tensor multigrid (TMG), in which AMG and GMG are combined. TMG converges faster and gives better solutions than AMG in the test-problem of image reconstruction. Finally, in [3], the adaptive smoothing and prolongation based AMG (aSP-AMG) method is considered for structural mechanics problems. When used as preconditioner to PCG, aSP-AMG has smaller computation time compared to the state-of-the-art MG methods like GAMG and BoomerAMG.

Third, papers investigating MOR have been considered. MOR techniques aim to reduce the computational complexity while still giving an accurate representation of the original system. [15,16] both use a tensor-based MOR technique for systems with multiple independent variables, by combining techniques of tensor decomposition and POD. The resulting method preserves the multidimensional structure of the original model. In reservoir engineering, the tensor MOR outperforms POD for an example problem, however both gains are not very large. In optimization problems, the tensor strategy is close to the optimal strategy for the full model.

Finally, papers focussing on DD have been treated. DD methods exploit parallel computing by a divide-and-conquer strategy. They solve the problem by splitting the domain into subdomains and computing solutions on those. [18] investigates efficient parallel preconditioners based on DD techniques for problems on unstructured grids. One- and two-level Schwarz preconditioners and (approximate) Schur complement based methods are used to accelerate a Krylov subspace method. This, combined with a mesh adaptation procedure yields effective and scalable methods for solving compressible Euler equations on unstructured grids. [20] investigates DD methods for elliptical PDEs with unstructured grids. The parallel FEM works properly with both SC method and FETI method, with a greater speedup (compared to two processors) for SC than for FETI.

With respect to the objective of investigating efficient numerical methods (in terms of memory, scalability, and overall computation time) that can be exploited in the context of model predictive control for both the thermal as the mechanical part of the system, the following techniques discussed in this review paper could be useful. POD-based deflation, and also the tensor-based MOR technique combining tensors and POD can be investigated as it maintains the multidimensional structure of a system. For the mechanical part of the system, deflation based on RBM can be considered. Within the AMG methods, SA-AMG, AMGe, TMG or aSP-AMG could be considered. DD techniques could be used as well, like the two-level Schwarz and (approximate) Schur complement methods.

References

[1] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2 edition, 2003. D000789161/00.

[2] T. B. Jönsthövel, M. B. Van Gijzen, S. MacLachlan, C. Vuik, and A. Scarpas. Comparison of the deflated preconditioned conjugate gradient method and algebraic multigrid for composite materials. *Computational Mechanics*, 50:321–333, 2011.

[3] A. Franceschini, V. A. Paludetto Magri, G. Mazzucco, N. Spiezia, and C. Janna. A robust adaptive algebraic multigrid linear solver for structural mechanics. *Comp. Methods in Appl. Mech. and Eng.*, 2 2019.

[4] C. Vuik and D. J. P. Lahaye. Scientific computing (wi4201), 2019. Delft University of Technology Faculty of Electrical Engineering, Mathematics and Computer Science.

[5] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of Two-Level Preconditioners Derived from Deflation, Domain Decomposition and Multigrid Methods. *Journal of Scientific Computing*, 39:340–370, 6 2009.

[6] G. Diaz Cortes. POD-Based Deflation Method For Reservoir Simulation. 2019. Dissertation at Delft University of Technology.

[7] J. M. Tang. Two-Level Preconditioned Conjugate Gradient Methods with Applications to Bubbly Flow Problems. 2008. Dissertation at Delft University of Technology.

[8] A. H. Sheikh, D. Lahaye, L. Garcia Ramos, R. Nabben, and C. Vuik. Accelerating the shifted Laplace preconditioner for the Helmholtz equation by multilevel deflation. *Journal of Computational Physics*, 322:473–490, 10 2016.

[9] V. Dwarka and C. Vuik. Scalable convergence using two-level deflation preconditioning for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 42:A901–A928, 2020.

[10] T. B. Jönsthövel, M. B. Van Gijzen, C. Vuik, and A. Scarpas. On the use of rigid body modes in the deflated preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 35, 2013.

[11] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge. Algebraic Multigrid Based On Element Interpolation (AMGe). *SIAM Journal on Scientific Computing*, 22, 01 2002.

[12] Q. Chang, Y. S. Wong, and H. Fu. On the Algebraic Multigrid Method. *Journal of Computational Physics*, 125(2):279 – 292, 1996.

[13] O. Morozov and P. R. Hunziker. Solving Tensor Structured Problems with Computational Tensor Algebra. *arXiv*, 01 2010.

[14] V. A. Paludetto Magri, M. Ferronato, A. Franceschini, and C. Janna. A novel AMG approach based on adaptive smoothing and prolongation for reservoir simulations. European Association of Geoscientists and Engineers, EAGE, 2018.

[15] F. Van Belzen and S. Weiland. A tensor decomposition approach to data compression and approximation of ND systems. *Multidimensional Systems and Signal Processing*, 23:209–236, 2010.

[16] E. Insuasty, P. M. J. Van Den Hof, S. Weiland, and J. D. Jansen. Tensor-based reduced order modeling in reservoir engineering: An application to production optimization. *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, 2015.

[17] M. A. Cardoso and L. J. Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229:681–700, 2 2009.

[18] M. Sala. Domain decomposition preconditioners: theoretical properties, application to the compressible Euler equations, parallel aspects. 2003. Dissertation at École Polytechnique Fédérale de Lausanne.

[19] A. Quarteroni, M. Sala, and A. Valli. The Swiss-Carpet Domain Decomposition Preconditioner, 2004.

[20] D. Marcsa and M. Kuczmann. Comparison of domain decomposition methods for elliptic partial differential problems with unstructured meshes. *Przeglad Elektrotechniczny*, 88:1–4, 01 2012.

[21] P. Sonneveld and M. B. Van Gijzen. IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations. *SIAM Journal on Scientific Computing*, 31:1035–1062, 2008.

A IDR(s)

The induced dimension reduction method (IDR(s)) is a Krylov subspace method for large non-symmetric linear systems of equations, introduced in the paper *IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations [21]* (2008). IDR(s) is based on the induced dimension reduction (IDR) theorem. It is a robust and efficient short-recurrence method, hence no excessive computational power and memory are needed. Moreover, in exact arithmetic the true solution is reached in at most $N + N/s$ matrix-vector products, where N is the problem size and s the codimension of a fixed subspace.

Consider a linear system of equations $Ax = b$, where A is a general matrix and r_0 the initial residual. Then it is impossible for a method to have both an optimal minimization of some error norm over the Krylov subspace $\mathcal{K}^n(A, r_0)$ as well as short recurrences; like CG has for Hermitian and positive definite A. The development of Krylov methods for such general problems has focussed on two different ways of generalizing CG. In the first class, short recurrences are not mandatory. A well-known method of this class is GMRES, which has as drawback that if the number of required iterations is large, the memory and computations costs become too high. The second class does enforce short recurrences but not the optimality property. A typical example is the Bi-CG method, which is equivalent to CG in the symmetric case but at about twice the cost. Other examples in this class are CGS, Bi-CGSTAB and TFQMR. The latter two are ways to stabilize CGS since while CGS is faster than Bi-CG, it exhibits erratic convergence behavior. Developments in fast Krylov methods with short recurrences were centered around Bi-CG-type methods. However, different to both these classes, new methods can be developed based on the IDR theorem.

IDR(s) generates residuals located in a sequence of nested subspaces $\mathcal{G}_j = (I - \omega_j A)(S \cap \mathcal{G}_{j-1})$ of decreasing dimensions. Here ω_j are nonzero scalars and S denotes a certain proper subspace of $\mathbb{C}^N$ which may be assumed to be left nullspace of some $N \times s$ matrix P. By the IDR theorem, such subspaces exist and under mild conditions $\mathcal{G}_j = \{0\}$ for some $j \leq N$. By the extended IDR theorem, the decrease in dimension per step is between zero and s, where zero being very unlikely and s being the case in practice. There is quite some freedom in the translation of the IDR theorem into an algorithm; namely, the choice of P, ω_j, and the computation of the intermediate residuals.

Numerical experiments are performed in the article, where IDR(s) is compared with the best known Bi-CG-type methods: Bi-CG, CGS, Bi-CGSTAB, BiCGstab(l), and QMR. While it is not a limited memory method, GMRES is also included as it is optimal with respect to the amount of matrix-vector multiplications (so no method can improve on this). Since IDR(1) is mathematically equivalent to Bi-CGSTAB – in the sense that at even steps they yield the same residuals – they are similar in stability and the amount of computations and memory needed. If $s > 1$, then IDR(s) performed better than Bi-CGSTAB. While increasing s slightly increased the cost per iteration, it significantly decreased in the number of iterations required in all of the experiments and for most came close to the optimal performance of GMRES. Even for problems with a highly nonsymmetric or indefinite matrix, IDR(s) was efficient. Overall, the numerical experiments showed that IDR(s) performs similarly or better than most Bi-CG-type methods [21].