IT Licentiate theses
2005-006

Robust Preconditioned Iterative Solution Methods for Large-scale Nonsymmetric Problems

ERIK BÄNGTSSON

UPPSALA UNIVERSITY
Department of Information Technology

UPPSALA
UNIVERSITET

Robust Preconditioned Iterative Solution Methods for Large-scale Nonsymmetric Problems

BY
ERIK BÄNGTSSON

Nov 2005

DIVISION OF SCIENTIFIC COMPUTING
DEPARTMENT OF INFORMATION TECHNOLOGY
UPPSALA UNIVERSITY
UPPSALA
SWEDEN

Dissertation for the degree of Licentiate of Technology in Scientific Computing
at Uppsala University 2005

Robust Preconditioned Iterative Solution Methods for Large-scale Nonsymmetric Problems

Erik Bängtsson
Erik.Bangtsson@it.uu.se

Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

ISSN 1404-5117
Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

We study robust, preconditioned, iterative solution methods for large-scale linear systems of equations, arising from different applications in geophysics and geotechnics.

The first type of linear systems studied here, which are dense, arise from a boundary element type of discretization of crack propagation in brittle material. Numerical experiment show that simple algebraic preconditioning strategies results in iterative schemes that are highly competitive with a direct solution method.

The second type of algebraic systems are nonsymmetric and indefinite and arise from finite element discretization of the partial differential equations describing the elastic part of glacial rebound processes. An equal order finite element discretization is analyzed and an optimal stabilization parameter is derived.

The indefinite algebraic systems are of 2-by-2-block form, and therefore block preconditioners of block-factorized or block-triangular form are used when solving the indefinite algebraic system. There, the required Schur complement is approximated in various ways and the quality of these approximations is compared numerically.

When the block preconditioners are constructed from incomplete factorizations of the diagonal blocks, the iterative scheme show a growth in iteration count with increasing problem size. This growth is stabilized by replacing the incomplete factors with an inner iterative scheme with a (nearly) optimal order multilevel preconditioner.

List of Papers

This thesis is a summary of the following papers and report. They will be referred to as Paper A, Paper B, Paper C and Paper D.

A E. Bängtsson and M. Neytcheva. Algebraic preconditioning versus direct solvers for dense linear systems as arising in crack propagation. *Communications in Numerical Methods in Engineering*, 21:73–81, 2005.

B E. Bängtsson and M. Neytcheva. Numerical simulations of glacial rebound using preconditioned iterative solution methods. *Applications of Mathematics*, 50(3):183–201, 2005.

C E. Bängtsson and M. Neytcheva. An agglomerate multilevel preconditioner for linear isostasy saddle point problems. *Accepted for publication in Lecture Notes in Computer Science*, 2005.

D E. Bängtsson. A consistent stabilized formulation for a nonsymmetric saddle-point problem Technical Report 2005-030, Department of Information Technology, Uppsala University, 2005.

Acknowledgments

I would like to thank my supervisor Dr. Maya Neytcheva for her encouragement and the endless hours she has spent on answering my questions, debugging my codes, and correcting my reports and papers. Without her this thesis would never have been written. I would also like to thank my assistant supervisor Prof. Per Lötstedt, for his time and for valuable discussions.

Further I would like to thank Dr. Björn Lund at the Geophysics Department, Uppsala University, for initiating the project on modeling of glacial rebound, and for the time he is spending explaining geophysics and seismology.

All my colleagues at the Department of Scientific Computing, thank you for making it a wonderful workplace, with a spirit of warmth and friendliness that helped me get through the days when nothing seems to go right.

Petra, for love and support.

Contents

1 Introduction

In many fields of science, due to practical, technical, and/or economical obstacles, it is not possible to perform classical experiments to obtain answers to our questions. In geophysics and astrophysics, where the length and time scales are enormous, laboratory or field experiments are impossible to perform due to sheer size. In more earthbound applications, such as manufacturing industry, experiments are avoided because of their cost. It is much less expensive to simulate car crashes than to actually perform them. The feasible alternative that then remains is to model the process you are interested in mathematically and solve the arising partial differential equations (PDE) numerically.

Partial differential equations constitute the foundation of the mathematical physics and they are known not to have analytic solutions, except for a limited number of special cases. This means that the solution to the PDE needs to be approximated, and in order to do this the PDE must be discretized. The field of scientific computing is devoted to this discretization and the efficient solution of the so-arising linear systems of equations,

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is nonsingular, $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{b} \in \mathbb{R}^n$.

The discretization of the PDE is often performed using some well-established technique, such as the finite difference method (FDM), or the finite element method (FEM). Both these methods require a discretization of the entire computational domain $\Omega \subset \mathbb{R}^d$, and they result in an algebraic system of equations with a large and sparse matrix. In some cases the PDE can be reformulated as an integral equation and reduced to the boundary of the computational domain, $\partial\Omega \subset \mathbb{R}^{d-1}$. The arising matrix is of smaller size than in the case of FEM and FDM, but it is on the other hand dense.

The obtained linear system is aimed to be solved with as small computational effort and memory demand as possible. For really large problems ($n > 500000$), the only way to achieve this is to use an optimal, robust, preconditioned, iterative solution method. Below, the particular meaning of the terminology is explicitely stated.

(i) Robustness means that the iterative solver converges independently of the parameters of the underlying problem (such as the Poisson number in elasticity problems and the viscosity in fluid dynamics).

(ii) For the iterative method to be optimal, its rate of convergence, i.e. the number of iterations required for the method to converge, must be independent of the size of A. When this is the case, the overall arithmetic work for the solution method becomes proportional to n if the cost for one iteration is $\mathcal{O}(n)$. The latter holds for sparse A. If the matrix is dense the cost per iteration, and the overall arithmetic work for the iterative solution method, is $\mathcal{O}(n^2)$.

(iii) Furthermore, in order to handle large scale applications, the iterative solution method should work fast in terms of CPU-time. To achieve this, the iterative solution method must be numerically efficient (few arithmetic operations per unknown), and

(iv) consume an amount of memory proportional to n.

(v) The management of data must make beneficial use of the computers memory hierarchy.

(vi) Finally, the iterative solver must be highly parallelizable, i.e. a lot of the computational work of the method can be performed independently of each other.

The target problems considered in this thesis are two different applications from geophysics and geotechnics. The aim of the thesis is to solve the latter using highly efficient preconditioned iterative solution methods which comply to the above-listed goals (i) - (vi). The problem in Paper A originates from a boundary element method (BEM) discretization of a model of crack propagation in brittle material, while the problem in Paper B, C and D originates from finite element (FE) modeling of the lithospheres elastic response to glaciation and deglaciation.

The outline of this summary is as follows. Section 2 contains a short description of the two most used solution techniques for linear systems of equations - direct and iterative methods. Section 3 is a brief introduction to different preconditioning techniques, and in Section 4 preconditioners for the special case when the matrix A admits a 2-by-2-block structure are described. Section 5 is an overview of the four papers constituting this thesis. In Section 6 a viscoelastic extension of the model in Section 5.2 is presented. The summary ends with Section 7, where some conclusions are drawn, and Section 8, which is an outlook into future work.

Some notations. Throughout this thesis, unless stated otherwise, uppercase Roman letters (A, B, C) denote matrices, script uppercase letters ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$) denote block-matrices arising from a discretized system of

PDEs. Lowercase Roman letters (x,y) denote scalars, and bold lowercase Roman letters ($\mathbf{x}$,$\mathbf{y}$) denote vectors.

2 Solution methods for linear systems of equations

When solving Equation (1) the following three objectives need to be met:

S1 The solver must be robust, i.e. $\mathbf{x}$ shall be found regardless of the parameters of the underlying problem, the size of A and the quality of the mesh.

S2 The computational complexity has to be minimized.

S3 The memory requirements of the solver should be small.

The objectives S2 and S3 are especially important when A is large.

Direct methods

One way to solve a linear system is to use a direct method, such as Gaussian Elimination (LU-factorization) for a general matrix, or Cholesky factorization if the matrix is symmetric positive definite. Both are robust and meet S1, but they fail to meet S2 and S3. For dense matrices the computational complexity of these methods is $\mathcal{O}(n^3)$, and the memory demand is $\mathcal{O}(n^2)$. For large n these requirements will make the task to solve Equation (1) impossible even on a large high-performing computer.

When A is sparse, the memory demand to store the matrix itself is $\mathcal{O}(n)$, and the cost to factorize it can be as small as $\mathcal{O}(n \log n)$ if A has a beneficial structure. The memory demand to store the factors L and U is then not larger than the storage cost of the original matrix, $\mathcal{O}(n)$. On the other hand, if A is not well structured, and no special preordering is used, the computational complexity can grow up to $\mathcal{O}(n^3)$ and the memory demands to $\mathcal{O}(n^2)$, due to fill-in elements produced in the factorization.

Iterative methods

The alternative to a direct method is an iterative method. The scheme of a simple iterative solution method can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k(\mathbf{b} - A\mathbf{x}_k) \tag{2}$$

where $\mathbf{x}_{k+1}$ is the current update, $\mathbf{x}_k$ is the previous update, τ_k is a parameter which may or may not be constant, and k is the iteration index. The iterative procedure ends when some termination criterion is fulfilled.

An often used class of iterative solution methods is that of the Krylov subspace methods. The idea is to find an approximate solution $\mathbf{x}_k$ in the Krylov subspace

$$\mathcal{K}^k(A, \mathbf{r}_0) \equiv \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{(k-1)}\mathbf{r}_0\}.$$

where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ is the initial residual.

Among the most used representatives of the Krylov subspace methods are the conjugate gradient method (CG), for symmetric positive definite matrices, and the generalized conjugate gradient (GCG) and generalized minimum residual methods (GMRES) for nonsymmetric matrices. The theory regarding the convergence behavior for these methods is well-established and can for example be found in [1] and [28].

The robustness of an iterative solution method is in general not guaranteed, and S1 is not always met. Even if the iterative solver is determined to converge in exact arithmetics, the finite precision of the floating point representation may influence the convergence. One way to accelerate the convergence, and decrease the number of iterations is to use a proper preconditioner, see Section 3.

The major part of the arithmetic work of a simple iterative method is spent in performing matrix-vector multiplications. This operation has $\mathcal{O}(n)$ complexity for sparse matrices and $\mathcal{O}(n^2)$ for dense matrices. If the method converges rapidly, i.e. the number of iterations required for convergence is much smaller than n, the overall complexity is $\mathcal{O}(n)$, and $\mathcal{O}(n^2)$ respectively, and S2 is met.

Objective S3 is also met by the iterative methods, since, in general only the matrix itself, a few vectors, and the preconditioner, need to be stored. A good preconditioner should by construction have a memory demand of $\mathcal{O}(n)$.

3 Preconditioners

A preconditioner G to A is a matrix or a procedure having the following properties:

P1 The preconditioned version of Equation (1),

$$G^{-1}A\mathbf{x} = G^{-1}\mathbf{b}, \tag{3}$$

is easier to solve than the original problem.

P2 G is constructed at a low cost.

P3 To apply G^{-1} or respectively, to solve a system with G, is inexpensive (typically of the same order as the cost to perform a matrix-vector multiplication).

If not stated otherwise, here G denotes a matrix.

The objective P1 is met if the eigenvalues of $G^{-1}A$ are clustered. In the extreme case $G^{-1} = A^{-1}$, and the iterative method converges in one iteration. This preconditioner, however, does not meet P2 and P3.

The action of the preconditioner transforms the scheme of the iterative method of Equation (2) into

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k G^{-1}(\mathbf{b} - A\mathbf{x}_k), \tag{4}$$

which can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k G^{-1}A(\mathbf{x} - \mathbf{x}_k), \tag{5}$$

where $\mathbf{x}$ is the true solution to Equation (1). If the eigenvalues of $G^{-1}A$ are clustered, $\tau_k G^{-1}A(\mathbf{x} - \mathbf{x}_k)$ resembles the error in the kth iteration.

Note that the application of G in Equation (3) is called “left preconditioner”. It is also possible to use a “right preconditioner”, $AG\mathbf{y} = \mathbf{b}$, $\mathbf{x} = G\mathbf{y}$, or to apply a “symmetric preconditioner”, that is, solve $GAG\mathbf{y} = G\mathbf{b}$, $\mathbf{x} = G\mathbf{y}$.

Incomplete Factorization preconditioners

One class of preconditioners that is widely used in commercial codes due to their straight forward implementation are based on pointwise incomplete LU (ILU) of A, or pointwise incomplete Cholesky factorization (IC) when A is symmetric positive definite. The drawbacks of the high arithmetic cost and memory demands of the classical (full) Gaussian Elimination and full Cholesky factorization are avoided by neglecting (some of) the fill-in elements in the factors L and U. When elements in the LU-factors are neglected because they are smaller than a certain threshold, the factorization is called “ILU-by-value”, and when they are omitted because they do not belong to a certain sparsity pattern we have “ILU-by-position”. The choice of the threshold and the sparsity pattern is a balance between the accuracy of the preconditioner and the cost to construct and apply it.

Among the incomplete factorization preconditioners are the numerous ILU-algorithms for nonsymmetric matrices and the IC-methods for symmetric positive definite matrices, see for example [1] and [27].

Sparse Approximate Inverse preconditioners

A preconditioner is said to be multiplicative if it is designed such that $G \approx A^{-1}$, and one class of multiplicative preconditioners is that of the (sparse) approximate inverse (SPAI) preconditioners.

A SPAI preconditioner is constructed as a matrix $G = [g_{ij}]_{i,j=1}^{n}$ with an a priori given sparsity pattern $\mathfrak{S} = \{i, j : g_{ij} \neq 0\}$, e.g. a band matrix. See for example [22] and the references therein.

The Multigrid framework

The multigrid (MG) method was initially introduced as an efficient iterative solution method for algebraic systems arising form the discretization of elliptical PDEs, e.g. the Laplace equation. However MG methods have also shown to be optimal and robust preconditioners for a large class of problems.

The framework of the MG methods is based on a sequence of "grids" $T_{(l)}$, $l = 0, \dots, L$. Let $T_{(l-1)}$ be coarser than $T_{(l)}$. On each level one needs a system matrix $A_{(l)}$, a restriction operator $R_{(l)}^{(l-1)} : T_{(l)} \to T_{(l-1)}$, a prolongation operator $P_{(l)}^{(l+1)} : T_{(l)} \to T_{(l+1)}$, and a pre- and a post-smoother. The smoother is supposed to reduce the high-frequency component of the error. Often used smoothers are simple iterative solution methods, such as the Jacobi method or the Gauss-Seidel method, and usually a few iterations are enough to smooth the error sufficiently.

We demonstrate the MG algorithm on two grids, T_1 and T_0. On the finest grid T_1 a smooth approximation $\mathbf{x}_1$ to the solution is obtained by the pre-smoother. The corresponding residual, or defect, $\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1$ is restricted to the coarser grid T_0 via the the action of the restriction operator, $\mathbf{r}_0 = R_1^0\mathbf{r}_1$. On T_0 an exact solution to the error equation $A_0\mathbf{e}_0 = \mathbf{r}_0$ is computed, and the correction $\mathbf{e}_0$ is prolongated to the fine grid and added to the smooth approximation, $\mathbf{x}_1 = \mathbf{x}_1 + P_0^1\mathbf{e}_0$. The result is post-smoothed to obtain a smooth update of $\mathbf{x}_1$. If the error equation is recursively solved on coarser grids, the V-cycle multigrid algorithm is obtained.

If $T_{(l-1)}$ is a physical grid and T_l is a uniform refinement of it, we are in the framework of the geometric multigrid (GMG). GMG is introduced in [13] as an efficient iterative solution method for elliptic PDEs. On the other hand, if $T_{(l)}$ is taken from the graph of $A_{(l)}$, and $T_{(l-1)}$ from the graph of the weakly coupled elements in $A_{(l)}$, we obtain the framework of the algebraic multigrid (AMG). See for example [32].

In the context of finite element discretization of PDEs, AMG methods based on the agglomeration of element stiffness matrices can be constructed, such as AMGE and AMGe, see [14] and [19].

Preconditioners based on Fast Transforms

Another class of preconditioners with nearly optimal convergence properties is based on Fast Transforms, e.g. Fast and Generalized Fourier Transforms. These methods are applicable if A has a structure such that it is (block)-diagonalized by a Fast Transform, i.e. it is a (block)-circulant or a (block)-Toeplitz matrix, or can be approximated by one. See, for example, [33].

Domain Decomposition preconditioners

The domain decomposition (DD) method, or Schwarz preconditioner, was introduced by Schwarz as a means to show existence of solution to PDEs on complicated domains. In the DD framework the solution is computed independently on different subdomains, and this gives the preconditioner attractive parallelization properties.

Problem based preconditioners

More efficient, but less general, preconditioners can be constructed if one also uses information about the (discretized) underlying problem, such as the PDE, the discretization method and/or the mesh.

The structure of the PDE is reflected in the matrix A. For example, if the matrix arises from the discretization of a system of PDEs (Stokes, Navier-Stokes, and Oseen's), or from a constrained optimization problem, A will exhibit a block structure.

A block structure of A can also be achieved by a permutation or reordering, for example according to a red-black ordering of the unknowns on a regular mesh. Section 4 contains more details on how to construct block preconditioners.

4 Block and block-factorized preconditioners

Block or block-factorized preconditioners are based on some 2×2-block form of A. The exact factorization of A is

$$\begin{aligned} A &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} && (6) \\ &= \begin{bmatrix} S_1 & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ A_{22}^{-1} A_{21} & I \end{bmatrix} && (7) \\ &= \begin{bmatrix} I & 0 \\ A_{21} A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & S_2 \end{bmatrix}, && (8) \end{aligned}$$

where $S_2 = A_{22} - A_{21} A_{11}^{-1} A_{12}$ and $S_1 = A_{11} - A_{12} A_{22}^{-1} A_{21}$ are the Schur complements of A. In the sequel, when it is clear from the context which is the Schur complement that is meant, the subscript is omitted. Utilizing the factorization in Equation (8), a preconditioner to A is then often sought in the form

$$G = \begin{bmatrix} I & 0 \\ A_{21} P^{-1} & I \end{bmatrix} \begin{bmatrix} P & A_{12} \\ 0 & Q \end{bmatrix}, \tag{9}$$

where P is an approximation of A_{11} and Q is an approximation of S.

A block 2×2 structure of A can be obtained in various ways.

(i) It can correspond to a splitting of the unknowns into *fine* and *coarse* due to some mesh hierarchy, some agglomeration technique, or a splitting of the matrix graph into independent sets.

(ii) It can be due to some permutation of the matrix which leads to some desirable properties of the A_{11}- or A_{22}-block. Typically, the goal is that one of the diagonal blocks can be well approximated with a diagonal or narrowbanded matrix.

Multilevel preconditioners

A multilevel (ML) preconditioner is obtained when the system matrix A is recursively split along fine and coarse unknowns according to one of the strategies in (i), e.g., when the fine mesh is a uniform refinement of the coarse mesh, as is depicted in Figure 1 for a quadrilateral and a triangular mesh.

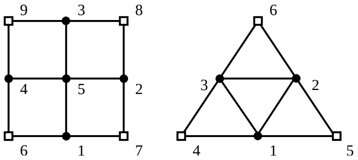


Figure 1: A macroelement on a quadrilateral and a triangular mesh

We demonstrate the idea behind some multiplicative [1] multilevel preconditioning methods on two levels, l and $l-1$, where l denotes the finer level. Representatives of this class are the multiplicative versions of the hierarchical basis (HB) functions preconditioner, the algebraic multilevel iterations (AMLI) method, and the algebraic recursive multilevel solver (ARMS).

The common framework of these block-factorization preconditioners is that the matrix on the fine mesh, $A^{(l)}$, is split along the unknowns corresponding to fine, f, and coarse, c, nodes,

$$A^{(l)} = \begin{bmatrix} A_{ff}^{(l)} & A_{fc}^{(l)} \\ A_{cf}^{(l)} & A_{cc}^{(l)} \end{bmatrix}. \tag{10}$$

From the factorization of $A^{(l)}$, a two-level preconditioner is defined as

$$G^{(l)} = \begin{bmatrix} I & 0 \\ A_{cf}^{(l)} P^{(l)^{-1}} & I \end{bmatrix} \begin{bmatrix} P^{(l)} & A_{fc}^{(l)} \\ 0 & Q^{(l)} \end{bmatrix}, \quad G^{(l-1)} = Q^{(l)}, \tag{11}$$

where $P^{(l)}$ approximates $A_{ff}^{(l)}$, and $Q^{(l)}$ approximates the Schur complement $S^{(l)} = A_{cc}^{(l)} - A_{cf}^{(l)} A_{ff}^{(l)^{-1}} A_{fc}^{(l)}$.

As the name suggests, the HB method uses a hierarchy of basis functions, defined on a sequence of nested meshes. The bases and the nested meshes naturally arise in the context of finite element discretization of PDEs, and the HB method originates in preconditioning of finite element stiffness matrices.

The approximation to the Schur complement on level l, $Q^{(l)}$, is taken as the coarse mesh matrix $A^{(l-1)}$. This matrix is sparse and, in the case of symmetric positive definite matrices, it is a spectrally equivalent approximation to the true Schur complement. The drawback of the HB

[1] Here,"multiplicative" refers to a (block) factorized matrix of the form shown in Equation (7) or (8), and not to a preconditioner G approximating A^{-1} as in Section 3.

method is that the condition number of the preconditioned matrix on the finest level $\kappa({G^{(L)}}^{-1}A^{(L)}$, is growing with the number of levels. One remedy for this growth is to stabilize the method with a matrix polynomial, which leads to the AMLI method. See, for example, [34] for details on the HB and AMLI methods.

In the AMLI method the growth in condition number is stabilized with a properly scaled Chebyshev matrix polynomial of degree ν, $P_\nu(E)$. The stabilization is done by replacing $Q^{(l)}$ in Equation (11) with

$$\widetilde{Q}^{(l)} = A^{(l-1)}[I - P_\nu({G^{(l-1)}}^{-1}A^{(l-1)})],$$

or, if the exact Schur complement can be formed at a low cost, with

$$\widetilde{Q}^{(l)} = S^{(l)}[I - P_\nu({G^{(l-1)}}^{-1}S^{(l)})].$$

The degree of the polynomial P_ν can be chosen to balance the number of levels, and a proper ν leads to an optimal order preconditioning method.

The AMLI method originates in the context of hierarchical basis finite element discretization of PDEs, but in contrast to HB, AMLI can be applied in a purely algebraic fashion. Then the fine-coarse splitting is based on the graph of the matrices $A^{(l)}$, and the Schur complement approximation is formed in some other way than as a coarse mesh matrix, e.g. $P^{(l)}$ is taken as a diagonal or narrowbanded matrix, which can be easily inverted and a sparse $Q^{(l)}$ can be computed at a low cost.

The ARMS method is a purely algebraic method, where the fine-coarse division is based on a splitting of the graph of $A^{(l)}$ into independent sets. The $A^{(l)}_{ff}$-block is approximated by an incomplete factorization, $P^{(l)} = L^{(l)}U^{(l)}$, and $Q^{(l)}$ is taken as $Q^{(l)} = A^{(l)}_{cc} - A^{(l)}_{cf}(L^{(l)}U^{(l)})^{-1}A^{(l)}_{fc}$. See [29] and [11] for details.

Saddle point preconditioners

Another context where approximations of a Schur complement matrix are required is when we need to precondition saddle point matrices,

$$\mathcal{A} = \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix}, \tag{12}$$

which arise for example when solving Stokes problem, Oseen's problem or constrained optimization problems. For such matrices, one uses a block lower- or upper-triangular,$\mathcal{D}_t$, or a block-factorized preconditioner, $\mathcal{D}_f$, of the form

$$\mathcal{D}_t = \begin{bmatrix} D_1 & 0 \\ B & -D_2 \end{bmatrix}, \qquad \mathcal{D}_f = \begin{bmatrix} I & 0 \\ BD_1^{-1} & I \end{bmatrix} \begin{bmatrix} D_1 & B^T \\ 0 & -D_2 \end{bmatrix}, \tag{13}$$

where D_1 approximates M and D_2 is an approximation of the negative Schur complement $S = C + BD_1^{-1}B^T$. The form of $\mathcal{D}_f$ follows naturally from Equation (9), while the block-triangular preconditioner is motivated by the relation

$$\mathcal{D}_t^{-1}\mathcal{A} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} D_1^{-1}M - I & D_1^{-1}B^T \\ D_2^{-1}B(I - D_1^{-1}M) & D_2^{-1}(C + B^T D_1^{-1}B) - I \end{bmatrix}.$$

The eigenvalues of $\mathcal{D}_t^{-1}\mathcal{A}$ are clustered around unity when D_1 is a good approximation of M, and D_2 is a good approximation of S. For further details on the spectral properties of $\mathcal{D}_t$ and $\mathcal{D}_f$ applied to symmetric matrices, see [4], and for a recent survey on preconditioners for saddle point matrices see [10].

An observation has been made that for symmetric problems [4] the convergence of an iterative method using the block-preconditioners $\mathcal{D}_t$ and $\mathcal{D}_f$ are more sensitive to the quality of D_1 than to the quality of D_2. If D_1 and D_2 are optimal order preconditioners to M and S, then the block preconditioners will also be of optimal order.

To form D_1, as noted in [10], for a general (nonsymmetric) block M, an incomplete factorization is a feasible alternative, possibly combined with a few iterations by an inner iterative solution method. Also multigrid preconditioners for nonsymmetric M are used, e.g. [26].

Schur complement approximations

Unless for some special cases, to explicitely form the true Schur complement is about as expensive as it is to solve the system with $\mathcal{A}$, and the Schur complement is in general a full matrix even for sparse $\mathcal{A}$. In order to fulfill the objectives P2 and P3, D_2 should not only be an accurate approximation to S, but also sparse, and it shall be constructed such that it is easily handled in a parallel environment.

In some cases it is known how to obtain a good quality approximation for the Schur complement. For example, for red-black orderings on regular meshes, A_{11} becomes diagonal and even the exact Schur complement is computed at low cost. In some applications it is enough to approximate A_{11} by its diagonal or by some sparse approximate inverse of A_{11}.

For other problems S can be approximated on a differential operator level, as done for the Oseen's problem in [20]. For the Stokes problem it is known that a good approximation of $BM^{-1}B^T$ is the pressure mass matrix, and for the HB- and AMLI-methods, the usual approximation of S is the coarse mesh stiffness matrix.

A novel approach to construct a Schur complement approximation is proposed in [23] in the context of algebraic multilevel preconditioning of a matrix arising in finite element discretization of (a system of) PDE(s).

The approach arises from the fact that the global stiffness matrix A is assembled from local macroelement matrices A^e. After a splitting of A^e along fine and coarse degrees of freedom, as depicted in Figure 1 for a quadrilateral and a triangular mesh, it takes the following 2×2-block form,

$$A^e = \begin{bmatrix} A^e_{11} & A^e_{12} \\ A^e_{21} & A^e_{22} \end{bmatrix} \begin{array}{l} \}\text{fine} \\ \}\text{coarse.} \end{array}$$

The approximated Schur complement S_a is assembled from exactly computed local Schur complements, $S_a = \sum_e S^e$, $S^e = A^e_{22} - A^e_{21}{A^e_{11}}^{-1}A^e_{12}$, and the so-constructed approximation S_a possesses some attractive properties.

1. S_a inherits the properties of A and automatically generates symmetric or nonsymmetric approximations of S.

2. It is sparse by construction.

3. For symmetric positive definite matrices, it is shown in [23] that S_a is spectrally equivalent to the true Schur complement S.

4. Parallelization techniques applied to handling finite element matrices are automatically applicable for S_a.

We use the approach to assemble a Schur complement approximation two-fold. First, to construct multilevel preconditioners for the diagonal blocks D_1 and D_2. In these cases the local Schur complements are computed exactly on macroelement level after a splitting of the macroelement stiffness matrix along fine and coarse degrees of freedom, as is described in [23].

Second, to assemble D_2. We use that the element stiffness matrices $\mathcal{A}^e$ exhibit the 2×2-block structure of $\mathcal{A}$,

$$\mathcal{A}^e = \begin{bmatrix} M^e & B^{eT} \\ B^e & -C^e \end{bmatrix},$$

and compute local negative Schur complements $S^e = C^e + B^e M^{e-1} B^{eT}$ exactly on each element e. The matrix D_2 is then assembled from the local Schur matrices, $D_2 = \sum_e S^e$.

5 Summary of Papers

5.1 Paper A

Paper A deals with simple algebraic preconditioning for dense linear systems, arising from the discontinuous displacement method (DDM) discretization of crack propagation in brittle material, e.g. rock. Up to the knowledge of the authors, the approach to use an iterative solution method preconditioned by a block-factorized preconditioner to solve a dense matrix arising from a DDM discretization is novel.

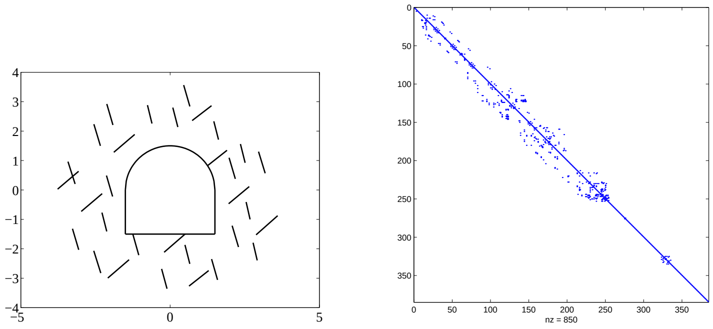
DDM is a BEM-type method, where the crack is expressed in terms of the width of the crack opening instead of in terms of the displacement of the sides of the crack. This decreases the number of unknowns required to describe a crack network by 50 %. See [17] for further details on DDM.

Due to crack singularities, the difference in magnitude between elements of the arising matrix can be enormous, which under a proper ordering of the unknowns leads to a strongly diagonally dominant matrix. However, for fracture networks more complicated than one single crack, it will also contain significant off-diagonal elements.

5.1.1 Preconditioners

Three preconditioners are tested on the arising DDM matrices, a SPAI preconditioner, an ILU-by-value preconditioner, and a full block-factorized preconditioner with approximate blocks (BFP).

Sparse Approximate Inverse preconditioner There exist various methods to compute the entries of the sparse approximate inverse G. One simple idea is to require that for all indices $i, j \in \mathfrak{S}$ there holds $(GA)_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker symbol and $\mathfrak{S}$ is a sparsity pattern. This idea, applied for dense matrices can be found in the literature under different names, one of them being the diagonal block approximate inverse (DBAI) technique , see [16]. DBAI constructs G as a matrix with k diagonals which approximates the inverse of the corresponding band part of A.



(a) A gallery in fractured rock.

(b) $A > 0.01$ after a permutation

Figure 2: The geometry of Problem 5.1.2 and the structure of A.

The efficiency of this type of approximate inverse preconditioner depends on the rate of decay of the off-diagonal elements of A^{-1}, and therefore, on the size of k. If A^{-1} contains significant off-diagonal elements left out of the non-zero structure of G, the preconditioner will not be able to capture those, and will be less efficient. For a theoretical justification of this approach, see [16] and the references therein.

Block-factorized preconditioners (BFP) When A admits a natural 2×2-block form, it can be factorized into the form of Equation (7) or Equation (8). We have utilized such a 2-by-2 block-structure to construct a preconditioner of the form (7), where the block A_{22} is approximated with an incomplete factorization or a diagonal matrix. Using this block we compute an explicit approximation of S_1, which is then solved exactly.

5.1.2 Numerical experiments

In [8], the performance of GMRES, preconditioned with the three preconditioners ILU, SPAI and BFP is illustrated on two problems arising from modeling of stress and fracture propagation around geotechnical constructions.

Problem 5.1.1 (Borehole with four cracks) A circular borehole in homogeneous infinite media is subjected to uniaxial stress and in the wall of the hole four radial cracks are situated.

Problem 5.1.2 (Gallery) A model of a gallery in fractured rock at a depth of 500 m.

The geometry of Problem 5.1.2 is shown in Figure 2, where also the most significant entries of A are shown (A is scaled to unit diagonal). As is depicted in Figure 2, A admits (after permutation) a 2×2-block form with a diagonally dominant A_{22}-block. The test matrices are generated by the DDM method, implemented in a commercial software package [31].

The system is solved with GMRES, preconditioned with either of the three preconditioners and, for comparison, with a direct solver provided in the DDM package. The results in terms of iteration counts and solution time show that, for both Problem 5.1.1 and Problem 5.1.2, the block-factorized preconditioner gives the most robust iterative solver with respect to problem size and preconditioner parameters.

The results also shows that the iterative solution methods are very competitive with the FORTRAN-implemented direct solver, despite the fact that they are implemented in the interpreting language MATLAB. The direct solver is competitive only for the smallest test problems.

5.2 Paper B

This paper deals with numerical simulations of a purely elastic model of the Earths response to glaciation and deglaciation. The lithosphere is modeled as a pre-stressed incompressible solid, and we present an analysis of the variational formulation of the equations of linear elasticity of saddle-point form, including the first order terms arising from the so-called advection of pre-stress.

The arising system of linear equations is of nonsymmetric saddle-point form. The novel idea here is to construct an approximation of the Schur complement of the indefinite matrix by assembling the exact Schur complements of the element matrices, which exhibit the same 2×2 block form as the global matrix itself.

For completeness we include Section 5.2.1 which contains the derivation of the PDE of interest, the moment balance equation for a linearly elastic, isotropic, non-self gravitating pre-stressed solid. This material is not included in [9].

5.2.1 Target problem

This section contains a more detailed description of the modeling of the isostatic, purely elastic response of the Earths lithosphere to glaciation and deglaciation. We derive the governing PDEs from the moment balance equation for an initially pre-stressed solid, and express them in terms of displacements $\mathbf{u}$, and kinematic pressure, p. The discussion follows Sections 2.1 - 2.3 in [18].

Consider an elastic body which occupies a domain $\Omega \in \mathbb{R}^3$ and is pre-stressed under static, in this case gravitational, forces. This stressed configuration is denoted by B, and the moment balance equation in the Eulerian, or spatial frame, reads as

$$\nabla \cdot \sigma_{\mathbf{0}} + \rho_{\mathbf{0}}\mathbf{f} = \mathbf{0} \quad \text{in } \mathbf{\Omega}, \tag{14}$$

where $\sigma_{\mathbf{0}}$ is the initial Cauchy stress tensor, $\mathbf{f}$ is an initial body force and ρ_0 is the initial density.

Under the action of additional, dynamic forces, the body is deformed into the configuration B', and it now occupies a domain Ω'. Under this deformation, a material point initially in position $\mathbf{x}$ will move to a new position $\mathbf{x}'$. Under the assumption that the added dynamical stress is small compared to the initial stress, one can adopt the small strain theory and express the deformations in the coordinate system of configuration B. The description of the deformations in terms of the original, undeformed coordinate system is called Lagrangian, or material, description of the solid.

The stress and the displacements in the deformed solid can be approximated as

$$\begin{aligned} T(\mathbf{x}, t) &= \sigma_0 + \epsilon\bar{T}(\mathbf{x}, t) \\ \mathbf{x}'(\mathbf{x}, t) &= \mathbf{x} + \epsilon\mathbf{u}(\mathbf{x}, t) \end{aligned} \tag{15}$$

where T is the Piola-Kirchoff stress tensor, $\epsilon\bar{T}(\mathbf{x}, t)$ is its increment, and $\epsilon\mathbf{u}(\mathbf{x}, t)$ is the displacement field. The parameter ϵ is a small real number.

The stress vector t on the surface of a solid element is defined as the force per unit area acting on this surface. A generic stress tensor is a linear transformation T, such that $t = T\mathbf{n}$, where $\mathbf{n}$ is the normal of the surface. When the stress tensor is expressed in the Eulerian frame it is referred to as the Cauchy stress tensor σ.

The force df acting on an infinitesimal solid element is, in the reference frame,

$$df \equiv dA_0 T\mathbf{n}_0,$$

where dA_0 is the undeformed area, and $\mathbf{n}_0$ its normal. In the spatial frame, df is defined as

$$df \equiv dA\sigma t\mathbf{n},$$

where dA is the deformed area and $\mathbf{n}$ its normal.

The force on the reference area df is independent of the coordinate system, and hence, $\sigma\mathbf{n}dA = T\mathbf{n}_0 dA_0$ [24]. After some manipulations we obtain

$$\sigma = j^{-1}\mathbf{F}T, \tag{16}$$

where $\mathbf{F} = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}}$ is the deformation tensor, and $j = det(\mathbf{F})$. Combined with Equation (15) and neglecting higher order terms in ϵ, Equation (16) yields

$$\mathbf{F} = I + \epsilon(\nabla\mathbf{u})^T, \qquad j = 1 + \epsilon\nabla\cdot\mathbf{u}, \qquad j^{-1} = 1 - \epsilon\nabla\cdot\mathbf{u}. \tag{17}$$

From Equation (16) and the first row in Equation (15), the incremental Cauchy stress $\bar{\sigma}$ is given by $\sigma = \sigma_0 + \epsilon\bar{\sigma}$, where

$$\bar{\sigma} = \bar{T} + (\nabla\mathbf{u})^T\sigma_0 - (\nabla\cdot\mathbf{u})\sigma_0. \tag{18}$$

The equations of motion for the body in the deformed state are

$$\nabla\cdot T(\mathbf{x},t) + \rho_0\mathbf{f}(\mathbf{x}',t) = \rho_0\frac{\partial^2\mathbf{x}'(\mathbf{x},t)}{\partial t^2}, \tag{19}$$

and with $T = \sigma_0 + \epsilon\bar{T}$, we obtain

$$\nabla\cdot\sigma_0 + \epsilon\nabla\cdot\bar{T} + \rho_0\mathbf{f}(\mathbf{x}',t) = \epsilon\frac{\partial^2\mathbf{u}(\mathbf{x},t)}{\partial t^2}, \tag{20}$$

since the initial configuration B is at rest.

We assume the Earth to be non-self gravitating, that is, the gravitational potential is not changed with density changes in the lithosphere. This implies that the body force $\mathbf{f}$ does not change with the change of the coordinate system and $\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}')$, see [18] for further details.

Combining Equations (18) and (20), and neglecting body forces (this is balanced by the choice of the boundary conditions), we get

$$\nabla\cdot\bar{\sigma} - \nabla\cdot[(\nabla\mathbf{u})^T\sigma_0] + \nabla\cdot[(\nabla\cdot\mathbf{u})\sigma_0] = \rho_0\frac{\partial^2\mathbf{u}}{\partial t^2} \tag{21}$$

So far, the initial stress σ_0 has been random, but in the sequel, we assume it to be hydrostatic and to only depend on the depth,

$$\sigma_0 = -p_0(\mathbf{x})\mathbf{I}, \tag{22}$$

where $p_0 = \rho \mathbf{g}_0 \cdot \mathbf{x}$ is the hydrostatic pressure, ρ_0 is the material density, and $\mathbf{g}_0$ the gravitational acceleration in equilibrium.

For nearly isostatic equilibrium the acceleration term $\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2}$ is negligible, and if it is neglected, the result reads

$$\nabla \cdot \bar{\sigma} + \nabla(\mathbf{u} \cdot \nabla p_0) - (\nabla \cdot \mathbf{u}) \nabla p_0 = \mathbf{0}. \tag{23}$$

5.2.2 An elastic model

For a linearly elastic and isotropic solid material Hooke's law reads

$$\bar{\sigma} = \mu \varepsilon(\mathbf{u}) + \lambda \nabla \cdot \mathbf{u} \mathbf{I}, \tag{24}$$

where $\varepsilon(\mathbf{u}) = 0.5(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the strain tensor, $\mu = \dfrac{E}{2(1+\nu)}$ and $\lambda = \mu \dfrac{2\nu}{1-2\nu}$ are the Lamé coefficients, and E and ν are Youngs modulus and Poissons ratio, respectively. The parameter λ is well defined for $\nu \in [0, 0.5)$, but as is well known Equation (24) is not well posed in the incompressible limit. Therefore, special care is required when discretizing and solving Equation (23) for $\nu \to 0.5$.

To handle purely incompressible materials, the usual remedy is to introduce the scaled (kinematic) pressure

$$p = \frac{\lambda}{\mu} \nabla \cdot \mathbf{u} = \frac{2\nu}{(1-2\nu)} \nabla \cdot \mathbf{u} \tag{25}$$

as an auxiliary variable, and consider the following coupled differential equation problem

$$\left\{ \begin{array}{rl} -2\nabla \cdot (\mu \nabla \mathbf{u}) - \nabla \times (\mu \nabla \times \mathbf{u}) & \\ -\rho g \left(\nabla \left(\mathbf{u} \cdot \mathbf{e}_d\right) - \mathbf{e}_d \nabla \cdot \mathbf{u}\right) - \mu \underline{\nabla} p & = \mathbf{0} \\ \mu \nabla \cdot \mathbf{u} - \frac{\mu^2}{\lambda} p & = \mathbf{0} \end{array} \right. \tag{26}$$

with boundary conditions

$$\begin{array}{rcll} \mathbf{u}(\mathbf{x}, t) & = & 0 & \mathbf{x} \in \Gamma_D \\ \bar{\sigma} \cdot n & = & \mathbf{l} & \mathbf{x} \in \Gamma_L \\ \bar{\sigma} \cdot n & = & \mathbf{0} & \mathbf{x} \in \Gamma_N. \end{array}$$

On the boundary segment Γ_D (meas$(\Gamma_D) > 0$) homogeneous Dirichlet conditions are imposed, and Γ_L and Γ_N are the parts of the boundary where the load and the homogeneous Neumann conditions are imposed.

For the analysis of the variational form and the finite element approximation of Equation (26), we consider a slightly more general form of the advection term, namely

$$-\nabla(\mathbf{u}\cdot\mathbf{b}) + \mathbf{c}\nabla\cdot\mathbf{u}, \tag{27}$$

where $\mathbf{b}$ and $\mathbf{c}$ are coefficient vectors.

From the properties of the operator ∇ we have that for any two differentiable vector functions $\mathbf{f}$ and $\mathbf{g}$ there holds

$$\nabla(\mathbf{f}\cdot\mathbf{g}) = \underbrace{(\mathbf{f}\cdot\nabla)\mathbf{g}}_{(a)} + \underbrace{(\mathbf{g}\cdot\nabla)\mathbf{f}}_{(b)} + \underbrace{\mathbf{f}\times(\nabla\times\mathbf{g})}_{(c)} + \underbrace{\mathbf{g}\times(\nabla\times\mathbf{f})}_{(d)}, \tag{28}$$

and from Equation (28) we see that the term $\nabla(\mathbf{u}\cdot\mathbf{b})$ is of more general form as compared to, for instance, the first-order term in the linearized Navier-Stokes equations which is of the form (b). In the special case when $\mathbf{b}$ is a constant vector, terms (a) and (c) in (28) vanish.

The target problem now reads

$$\left\{\begin{array}{rrl} -2\nabla\cdot(\mu\nabla\mathbf{u}) - \nabla\times(\mu\nabla\times\mathbf{u}) - \nabla(\mathbf{u}\cdot\mathbf{b}) + \mathbf{c}\nabla\cdot\mathbf{u} & -\mu\underline{\nabla}p & = \mathbf{0} \\ \mu\nabla\cdot\mathbf{u} & -\frac{\mu^2}{\lambda}p & = \mathbf{0}. \end{array}\right. \tag{29}$$

5.3 Variational formulation

The variational formulation corresponding to Equation (29) is defined in terms of the Sobolev spaces $\mathbf{V} = \left(H_0^1(\Omega)\right)^d$, $d = 2,3$, and $P = \{p \in L^2(\Omega);\ \int_\Omega \mu\, p\, d\Omega = 0\}$. It leads to the following mixed variable problem:

$$\begin{array}{l} \text{Find } \mathbf{u}\in\mathbf{V} \text{ and } p\in P \text{ such that} \\ \left\{\begin{array}{rcll} a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) & = & \langle \mathbf{l},\mathbf{v}\rangle, & \forall\mathbf{v}\in\mathbf{V}, \\ b(\mathbf{u},q) - c(p,q) & = & 0, & \forall q\in P, \end{array}\right. \end{array} \tag{30}$$

where

$$
\begin{aligned}
a(\mathbf{u},\mathbf{v}) &= \int\limits_{\Omega} \Big[2\mu \sum_{k=1}^{d} (\nabla u_k)\cdot(\nabla v_k) - \mu(\nabla\times\mathbf{u})\cdot(\nabla\times\mathbf{v}) \\
&\quad - \nabla(\mathbf{u}\cdot\mathbf{b})\cdot\mathbf{v} + (\nabla\cdot\mathbf{u})(\mathbf{c}\cdot\mathbf{v})\Big] d\Omega \\
b(\mathbf{u},p) &= \int\limits_{\Omega} \mu(\nabla\cdot\mathbf{u})p\,d\Omega = -\int\limits_{\Omega} \mu\nabla(p)\cdot\mathbf{u}\,d\Omega \\
c(p,q) &= \int\limits_{\Omega} \frac{\mu^2}{\lambda} p\,q\,d\Omega \\
\langle\mathbf{l},\mathbf{v}\rangle &= \int\limits_{\Gamma_L} \mathbf{v}\cdot\mathbf{l}\,d\Gamma
\end{aligned}
\tag{31}
$$

A solution to the variational problem (30) exists and is unique if $a(\mathbf{u},\mathbf{v})$, $c(p,p)$ and $b(\mathbf{u},p)$ are bounded,

$$
\begin{aligned}
a(\mathbf{u},\mathbf{v}) &\leq \overline{a}\|\mathbf{u}\|_{\mathbf{V}}\|\mathbf{v}\|_{\mathbf{V}} \quad \forall\mathbf{u},\mathbf{v}\in\mathbf{V} && (32)\\
b(\mathbf{v},p) &\leq \overline{b}\|\mathbf{v}\|_{\mathbf{V}}\|p\|_{P} \quad \forall\mathbf{u}\in\mathbf{V},\, p\in P && (33)\\
c(p,q) &\leq \overline{c}\|p\|_{P}\|q\|_{P} \quad \forall p,q\in P, && (34)
\end{aligned}
$$

and if $a(\mathbf{u},\mathbf{u})$ and $c(p,p)$ are coercive,

$$
\begin{aligned}
a(\mathbf{u},\mathbf{u}) &\geq \underline{a}\|\mathbf{u}\|^2_{\mathbf{V}}, \quad \underline{a}>0 \quad \forall\mathbf{u}\in\mathbf{V} && (35)\\
c(p,p) &\geq \underline{c}\|p\|^2_{P}, \quad \underline{c}>0 \quad \forall p\in P. && (36)
\end{aligned}
$$

As is clear from Equation (31) $c(p,q) = 0$, $\forall p,q \in P$ corresponds to $\nu = 0.5$. In this case, , Equation (30) is solvable if

- the conditions in Equation (32) - (34) hold,
- $a(\mathbf{u},\mathbf{u})$ is coercive on the null-space of $b(\mathbf{u},q)$,
- $b(\mathbf{u},q) = 0 \quad \Rightarrow q = 0 \quad \forall\mathbf{u}\in\mathbf{V}$.

Furthermore, Equation (30) is stable if the following inf-sup (or Ladyzhenskaya-Babuška-Brezzi or LBB) conditions are fulfilled,

$$
\inf_{\mathbf{u}\in\mathbf{V}} \sup_{\mathbf{v}\in\mathbf{V}} \frac{a(\mathbf{u},\mathbf{v})}{\|\mathbf{u}\|_{\mathbf{V}}\|\mathbf{v}\|_{\mathbf{V}}} \geq \underline{a}' > 0, \tag{37}
$$

and

$$\inf_{q \in P} \sup_{\mathbf{v} \in \mathbf{V}} \frac{b(\mathbf{u}, q)}{\|\mathbf{v}\|_{\mathbf{V}} \|q\|_P} \geq \underline{b} > 0. \tag{38}$$

Note that when $a(\mathbf{u}, \mathbf{v})$ is coercive, Equation (37) is automatically satisfied. See, for example, [15] for details.

In [9], we show that the bilinear forms in Equation (30) are bounded, but that $a(\mathbf{u}, \mathbf{v})$, in general is not coercive due to the first order terms. For the special case when $\mathbf{b} = \mathbf{e}_d$ and $\nabla \cdot \mathbf{u} = 0$, $a(\mathbf{u}, \mathbf{v})$ is coercive. The ellipticity of $c(p, p)$ is straightforwardly seen.

5.3.1 Finite element discretization

To discretize Equation (30), let $\mathbf{V}^h$ and P^h be finite element subspaces of $\mathbf{V}$ and P correspondingly, and $\mathbf{u}_h$, $\mathbf{v}_h$, p_h and q_h be the discrete counterparts to $\mathbf{u}$, $\mathbf{v}$, p and q. The discrete formulation of (30) then reads:

$$\begin{array}{l} \text{Find } \mathbf{u}_h \in \mathbf{V}_h \text{ and } p_h \in P_h \text{ such that} \\ \begin{array}{rcll} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) & = & \langle \mathbf{l}, \mathbf{v}_h \rangle & \forall \mathbf{v}_h \in \mathbf{V}^h, \\ b(\mathbf{u}_h, q_h) - c(p_h, q_h) & = & 0, & \forall q_h \in P^h. \end{array} \end{array} \tag{39}$$

As is well known, in order to obtain a stable discrete formulation, the finite element spaces $\mathbf{V}^h$ and P^h cannot be chosen arbitrarily. They either have to form a stable pair, or Equation (39) needs to be stabilized.

A stable pair of finite element spaces for Equation (39) is a tuple $\mathbf{V}^h \times P^h$, having the properties that

1. $a(\mathbf{u}_h, \mathbf{u}_h) > \alpha \|\mathbf{u}_h\|_{\mathbf{V}^h}, \forall \mathbf{u}_h \in \mathbf{V}^h$,
2. $c(p_h, p_h) > \beta \|p_h\|_{P^h}, \forall p_h \in P^h$, and
3. the discrete counterpart to the LBB-condition (38),

$$\sup_{\mathbf{u}_h \in \mathbf{V}^h} \frac{b(\mathbf{u}_h, p_h)}{\|\mathbf{u}_h\|_{\mathbf{V}^h}} \geq \gamma_h \|p_h\|_{P^h}, \geq \gamma_0 \|p_h\|_{P^h}, \forall p_h \in P^h, \tag{40}$$

   is satisfied.

See, for example, [12] for details.

One way to circumvent the discrete LBB-condition on the finite element spaces is to stabilize Equation (39), and use an unstable pair of elements. This gives us the freedom to choose the finite element spaces

$\mathbf{V}_h$ and P^h in a way that is preferable from a computational complexity point of view, e.g. such that the problem size is reduced, compared to when satisfying the inf-sup condition in Equation (38),

A stabilized and consistent equal order FE discretization of Equation (39) can be achieved by adding the equation

$$-\sigma_h \int_\Omega \mu \nabla q \cdot \nabla p = \sigma_h \int_\Omega \mathbf{f} \cdot \nabla q + \sigma_h \sum_{\tau_k} \int_{\tau_k} 2\mu \Delta \mathbf{u} \cdot \nabla q \\ + \sigma_h \int_\Omega \nabla(\mathbf{b} \cdot \mathbf{u}) \cdot \nabla q - \sigma_h \int_\Omega (\nabla \cdot \mathbf{u})(\mathbf{c} \cdot \nabla q),$$

to the second equation of (30), where σ_h is some suitably determined stabilization parameter. A derivation of the optimal choice of σ_h is found in [6], and it is shown that $\sigma_h = \mathcal{O}(h^2)$ gives an optimal stability estimate of the form

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + h\|p - p_h\|_0 \leq h^2 C \|\mathbf{l}\|_0. \tag{41}$$

The finite element discretization of the stabilized version of Equation (39) leads to a linear algebraic system

$$\mathcal{A} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} \equiv \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} = \begin{bmatrix} \mathbf{r}_h \\ \mathbf{s}_h \end{bmatrix}. \tag{42}$$

The system matrix $\mathcal{A}$ admits a saddle point form and is unsymmetric indefinite. The nonsymmetricity is due to the discretized first order (advection) terms in the block M. The system in Equation (42) is further solved by preconditioned iterative solution methods.

5.3.2 Numerical experiments

We apply the preconditioners $\mathcal{D}_t$ and $\mathcal{D}_f$ from Equation (13), page 13, on the following realistic benchmark problem.

Problem 5.3.1 A 2D flat Earth model, which is symmetric with respect to $x = 0$, is subjected to a Heaviside load of a 1000 km wide and 2 km thick ice sheet. The size of the domain is 10 000 km width and 4 000 km depth and the boundary conditions are homogenous Dirichlet conditions on the boundary $y = -4000$ km and symmetry conditions on the boundary $x = 0$. Homogenous Neumann conditions are imposed on the boundary $x = 10000$ km and on the boundary segment $y = 0, x > 1000$ km. The Young modulus of the solid is 400 GPa, the Poisson ratio is 0.5 (the material is incompressible), and its density is 3000 $kg\ m^{-3}$. The density of the ice is 981 $kg\ m^{-3}$.

In the experiments we use GMRES as an iterative scheme, preconditioned by either $\mathcal{D}_t$ or $\mathcal{D}_f$. The iterations are terminated when the residual norm is decreased by six orders of magnitude compared to the initial residual.

Two approximations for the (negative) Schur complement matrix S are tested, one symmetric and one nonsymmetric. The symmetric approximation of the Schur complement S, S_m, is chosen as $S_m = C + M_p$, where M_p is the pressure mass matrix. To form a nonsymmetric approximation for S we assemble a matrix S_a from exact Schur complements of the local element stiffness matrices, as described in Section 4. The construction is computationally cheap and numerical tests show that S_a is as good approximation to S as S_m.

The blocks D_1 and D_2 are formed as incomplete LU factorizations of M and S_i, $i = m, a$, employing `ILUT` [27].

Iteration counts The numerical results in [9] reveal that the performance of $\mathcal{D}_f$ and $\mathcal{D}_t$ is more sensitive to the quality of the approximation of M. The observed growth in iteration counts with the problem size is due to the choice of D_1 as an incomplete factorization of M. The increase in the number of iterations can be stabilized with a better choice of preconditioner for M and S of multilevel or multigrid type. This can be seen from the comparisons with $D_1 = M$, when the diagonal block is solved with exactly.

Further, the results show that $\mathcal{D}_f$ is a more robust preconditioner than $\mathcal{D}_t$, and that both are relatively insensitive to the quality of the factorization of, and the choice of approximation to, the Schur complement. Finally, the results show some increase in iteration count with increasing ν. The growth is, however, acceptable.

CPU time comparisons In [9] we also perform CPU-time comparisons using our code and a commercial FEM package for Problem 5.3.1 with identical geometry, mesh and physical parameters. The only slight difference between the runs is in the boundary conditions. On the far boundaries ($x = 10000$ and $y = -4000$) the package imposes bilinear, infinite elements instead of standard homogeneous Neumann and Dirichlet conditions. The package is run on two different systems, an AMD Athlon 2.5 GHz processor, and a dual Itanium 1.5 GHz processor. The benefit from using an appropriately preconditioned iterative method instead of a direct solver is clearly seen from the timing results.

5.4 Paper C

Paper C is a continuation and extension of Paper B and targets the same nonsymmetric saddle point problem. The arising algebraic system is solved using a generalized conjugate gradient-minimized residual (GCG-MR) method, preconditioned with a block-triangular preconditioner of the form $\mathcal{D}_t$ in Equation (13), page 13. The novelty of this paper is that the blocks D_1 and D_2 are solved by a nearly optimal inner solution method, namely, an AMLI-preconditioned GCG-MR method.

The AMLI preconditioner is recursively defined and is of the form

$$G^{(l)} = \begin{bmatrix} I & 0 \\ G_{21}^{(l)} {P^{(l)}}^{-1} & I \end{bmatrix} \begin{bmatrix} P^{(l)} & G_{12}^{(l)} \\ 0 & Q^{(l)} \end{bmatrix}$$
$$G^{(l-1)} = Q^{(l)}, \quad l = L, L-1, \dots, l_0. \tag{43}$$

On each level l the matrices $Q^{(l)}$ are obtained from assembly of local, exactly computed, macroelement Schur complement matrices.

As already mentioned, the latter framework is originally proposed in [23] and theory is derived in the case of symmetric positive definite matrices. Up to the knowledge of the authors, this is the first time it is applied in the context of preconditioning for nonsymmetric saddle point problems.

A rigorous theory for the AMLI methods for nonsymmetric matrices is not yet derived. One reason for that is that in the nonsymmetric case there is no straightforward way to define an analogous parameter to the constant γ in the strengthened Cauchy-Bunyakowsky-Schwarz inequality, which is the main tool for proving optimal convergence for the classical AMLI methods. See for example [5]. However, from the numerical experiments in [7] it is seen that the method works well for the considered nonsymmetric problems.

Symmetric preconditioners for M In order to define a preconditioner D_1 for M in Equation (42), let us order the displacements $\mathbf{u}$ using the so-called *separate displacement ordering* (sdo), i.e., let all displacements in the x-direction be ordered first. This introduces a 2×2 block structure in M,

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

Recall that M is a non-symmetric matrix which entries are given by $m_{ij} = a(\mathbf{v}_i, \mathbf{v}_j)$. The bilinear form $a(\mathbf{u}, \mathbf{v})$ is a sum of two terms,

$$a(\mathbf{u}, \mathbf{v}) = \widehat{a}(\mathbf{u}, \mathbf{v}) + \bar{a}(\mathbf{u}, \mathbf{v}),$$

where

$$\widehat{a}(\mathbf{u}, \mathbf{v}) = \int_\Omega [2\mu \sum_{k=1}^{2} (\nabla u_k) \cdot (\nabla v_k) - \mu(\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v})]$$

and

$$\bar{a}(\mathbf{u}, \mathbf{v}) = \int_\Omega \left[-\nabla(\mathbf{u} \cdot \mathbf{b})\mathbf{v} + (\nabla \cdot \mathbf{u})(\mathbf{c} \cdot \mathbf{v}) \right].$$

In Problem 5.3.1 $\mathbf{b} = \mathbf{c} = \rho g \mathbf{e}_d$, where $\mathbf{e}_d$ is the unit vector directed downwards. As both the density ρ and the acceleration of gravity g are orders of magnitude smaller than the Lamé coefficient μ, the bilinear form $a(\mathbf{u}, \mathbf{v})$ is dominated by the elastic part $\widehat{a}(\mathbf{u}, \mathbf{v})$. This motivates the choice of the preconditioner $\widehat{\mathcal{D}}$, where the AMLI preconditioner in Equation (43) is generated by $\widehat{a}(\mathbf{u}, \mathbf{v})$. That is, the entries of the macroelement stiffness matrix $\widehat{D}^{(e)}$ are given by $\widehat{D}^{(e)}_{ij} = \widehat{a}(\mathbf{v}_i, \mathbf{v}_j)$. After a fine-coarse splitting, $\widehat{D}^{(e)}$ is of the form

$$\widehat{D}^{(e)} = \begin{bmatrix} \widehat{D}^{(e)}_{ff} & \widehat{D}^{(e)}_{fc} \\ \widehat{D}^{(e)}_{cf} & \widehat{D}^{(e)}_{ff} \end{bmatrix}, \tag{44}$$

and the four blocks in Equation (44) are used to assemble the matrices $P^{(l)}$, $G^{(l)}_{12}$, $G^{(l)}_{21}$, and $Q^{(l)}$.

One of the Korn's inequalities assert that, for some positive number $K = K(\Omega)$, depending only on the domain and not the Lamé coefficients, the inequality

$$K(\Omega)\widetilde{a}(\mathbf{u}, \mathbf{u}) \leq \widehat{a}(\mathbf{u}, \mathbf{u}) \leq 2\widetilde{a}(\mathbf{u}, \mathbf{u}) \tag{45}$$

holds, where $\widetilde{a}(\mathbf{u}, \mathbf{v})$ is the scaled vector Laplacian

$$\widetilde{a}(\mathbf{u}, \mathbf{v}) = \int_\Omega 2\mu \sum_{k=1}^{2} (\nabla u_k) \cdot (\nabla v_k).$$

See, for example, [2].

Equation (45) motivates the choice of the preconditioner $\widetilde{\mathcal{D}}$, in which the inner AMLI preconditioner is generated by the bilinear form $\widetilde{a}(\mathbf{u}, \mathbf{v})$.

This is equivalent to precondition the inner solution method by a block-diagonal matrix

$$\widetilde{D}_1 = \begin{bmatrix} \widetilde{D}_1^{(1)} & \\ & \widetilde{D}_1^{(2)} \end{bmatrix},$$

where $\widetilde{D}_1^{(i)} = \int_\Omega 2\mu(\nabla u_i) \cdot (\nabla v_i)$. Note that the blocks $\widetilde{D}_1^{(i)}$ are different due to the boundary conditions on $\mathbf{u}$.

Numerical results The matrix $\mathcal{A}$ is solved using the generalized conjugate gradient-minimized residual (GCG-MR) method, preconditioned with $\mathcal{D}_t$ in Equation (13), and solved until a relative stopping criterion 10^{-6} is achieved.

The block D_2 is obtained from assembly of local exact Schur complement matrices on the elements, and it is a nonsymmetric approximation of the true, also nonsymmetric, Schur complement.

The diagonal blocks of $\mathcal{D}$, D_1 and D_2 are solved with GCG-MR, preconditioned with the block-factorized multilevel preconditioner in Equation 11, to some relative stopping criteria τ and 10^{-6}, correspondingly. The block $P^{(l)}$ in Equation (43) is approximated by an incomplete LU-factorization (ILUT), see [27].

The numerical tests in [7] illustrate the performance of the proposed preconditioner $\mathcal{D}_t$, depending on the accuracy of the inner solver for D_1, the Poisson number, the problem size and the number of levels in the inner multilevel preconditioners.

The proposed preconditioner $\mathcal{D}_t$ is of optimal order when the inner iterative solution method for M is preconditioned by $\widetilde{D}_1$. The number of inner and outer iterations are constant and the overall computation time nearly scales with the size of the problem.

However, $\mathcal{D}_t$ is not entirely robust in the incompressible limit, since some growth in iteration count can be observed as $\nu \to 0.5$. This result is similar to what is observed in [9].

The results also show that there is a trade-off between the overall computation time to solve with $\mathcal{A}$, and the accuracy of the inner solvers and the number of levels in the inner multilevel preconditioners.

On each level in the multilevel preconditioner there is an overhead in solution time from the two solves with $P^{(l)}$, and the matrix-vector multiplications with A_{12} and A_{21}. This overhead is reduced by a short recursion, and the number of levels in the short recursion is balanced by the cost to solve a larger algebraic system on the coarsest level l_0.

The optimal number of levels depends on the size of the matrix on the finest level n_L. The experiments show that for Problem 5.3.1 with $n_L \leq 500000$ the optimal number of levels is three to four, depending on the preconditioner in the inner iterative solution method for D_1.

The accuracy in the inner iterative solution method for M also affects the overall solution time. A less accurate solution is obtained in a few inner iterations, but leads to a larger iteration count for the outer iterative solution method. The results from the numerical experiments show that the shortest overall solution time for A is given by a termination criterion $\tau = 0.5$ for the inner solver for M.

The inner iterative solution method for S meets the termination criterion 10^{-6} in one or two iterations, regardless of the problem size, the Poisson number, or the number of levels.

5.5 Paper D

As discussed in Paper B, in order to circumvent the LBB-condition on the spaces $\mathbf{V}$ and P, we use a stabilized equal order finite element discretization of Equation (30). This paper contains a derivation of estimates of the approximation error for $\mathbf{u}$ and p, and related to those, an optimal stabilization parameter σ_h. The derivation follows the technique in [3] for the stationary Stokes problem. Up to the knowledge of the author, this is a novel result for the finite element discretization of the equations of linear isostasy.

The derivation is done in two steps. First, we bound the error in the approximation of the displacement and pressure field in H^1-norm. Thereafter, via the Aubin-Nietsche trick [12], we use the dual problem to Equation (30) to find an error estimate in L^2-norm.

5.5.1 Error estimates in H^1-norm

Consider the following variational problem

$$\text{Seek } \mathbf{u} \in \mathbf{V} \text{ and } p \in P \text{ such that}$$
$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \mathbf{f}(\mathbf{v}), & \forall \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{u}, q) - c(p, q) = 0, & \forall q \in P, \end{cases} \tag{46}$$

where the bilinear forms $a(\cdot,\cdot)$, $b(\cdot,\cdot)$ and $m(\cdot,\cdot)$ are as follows

$$\begin{aligned}
a(\mathbf{u},\mathbf{v}) &= \textstyle\int_\Omega \Big[2\mu \sum_{k=1}^{d} \nabla u_k \cdot \nabla v_k - \mu(\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) \\
&\quad -\nabla(\mathbf{u}\cdot\mathbf{b})\cdot\mathbf{v} + (\nabla\cdot\mathbf{u})(\mathbf{c}\cdot\mathbf{v})\Big] \\
b(\mathbf{v},p) &= \textstyle\int_\Omega \mu(\nabla\cdot\mathbf{v})p = -\int_\Omega \mu\mathbf{v}\cdot\nabla p = -b(p,\mathbf{v}) \\
c(p,q) &= \textstyle\int_\Omega \frac{\mu^2}{\lambda} pq
\end{aligned}$$

Above, μ and λ are scalars (problem dependent parameters) which are assumed to be piecewise constant on Ω.

To perform the stability analysis, we assume the solution $[\mathbf{u},p]$ to Equation (46) to be bounded by the given data $\mathbf{f}$, i.e.

$$|\mathbf{u}|_2 + |p|_1 \leq \|\mathbf{f}\|_0. \tag{47}$$

We further assume that the bilinear forms in Equation (46) are bounded, and that $a(\mathbf{u},\mathbf{v})$ and $c(p,q)$ are coercive.

Equation (46) is stabilized in the following way. First we scalar multiply the first row of Equation (26) with ∇q, $q \in H_0^1(\Omega)$, and obtain

$$\begin{aligned}
-2\mu\Delta\mathbf{u}\cdot\nabla q - \mu\nabla\times(\nabla\times\mathbf{u})\cdot\nabla q - \nabla(\mathbf{b}\cdot\mathbf{u})\cdot\nabla q \\
+ (\nabla\cdot\mathbf{u})(\mathbf{c}\cdot\nabla q) - \mu\nabla q\cdot\nabla p = \mathbf{0}.
\end{aligned} \tag{48}$$

Integration of Equation (48) over Ω yields

$$-d(p,q) = (\mathbf{f},\nabla q) + e(q,\mathbf{u};\mathbf{b},\mathbf{c}), \tag{49}$$

where

$$\begin{aligned}
d(p,q) &= \int_\Omega \mu\nabla p\cdot\nabla q \\
e(q,\mathbf{u};\mathbf{b},\mathbf{c}) &= \sum_{\tau_k}\int_{\tau_k} 2\mu\Delta\mathbf{u}\cdot\nabla q + \int_\Omega \mu\nabla\times(\nabla\times\mathbf{u})\cdot\nabla q \\
&\quad + \int_\Omega \nabla(\mathbf{b}\cdot\mathbf{u})\cdot\nabla q - \int_\Omega (\nabla\cdot\mathbf{u})(\mathbf{c}\cdot\nabla q),
\end{aligned} \tag{50}$$

Above,τ_k denotes the kth finite element in the discretization Ω_h of Ω, i.e. $\Omega_h = \bigcup_k \tau_k$.

Finally, we multiply Equation (49) with a stabilization parameter σ, and add the result to the second row of Equation (46), that is

Find $\mathbf{u} \in \mathbf{V}$ and $p \in P$ such that

$$\left\{ \begin{array}{rcll} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) & = & \mathbf{f}(\mathbf{v}) & \forall \mathbf{v} \in \mathbf{V} \\ b(\mathbf{u}, q) - c(p, q) - \sigma d(p, q) & = & \sigma(\mathbf{f}, \nabla q) + e(q, \mathbf{u}; \mathbf{b}, \mathbf{c}) & \forall q \in P. \end{array} \right. \tag{51}$$

Equation (51) is consistent with Equation (46) for any value of σ. The discrete formulation of Equation (51) is

Find $\mathbf{u}_h \in \mathbf{V}_h \subset \mathbf{V}$ and $p_h \in P_h \subset P$ such that

$$\left\{ \begin{array}{rcll} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) & = & \mathbf{f}(\mathbf{v}_h) & \forall \mathbf{v}_h \in \mathbf{V}_h \\ b(\mathbf{u}_h, q_h) - c(p_h, q_h) - \sigma d(p_h, q_h) & = & (\mathbf{f}, \nabla q_h) & \\ & + & e(q_h, \mathbf{u}^*; \mathbf{b}, \mathbf{c}) & \forall q_h \in P. \end{array} \right. \tag{52}$$

where $\mathbf{u}^*$ is an approximation of $\mathbf{u}$.

Under the assumption that $\sum_k \tau_k \|\Delta(\mathbf{u} - \mathbf{u}_h^*)\|_{L^2(\tau_k)} = \mathcal{O}(1)$, $\|\mathbf{u} - \mathbf{u}_h^*\|_1 = \mathcal{O}(1)$, and $\sigma = \mathcal{O}(h^2)$, one finds, after some manipulations, that,

$$\begin{array}{rcl} \|\mathbf{u}_I - \mathbf{u}_h\|_1 & \leq & h\|\mathbf{f}\|_0 = \mathcal{O}(h) \\ \|p_I - p_h\|_1 & \leq & \|\mathbf{f}\|_0 = \mathcal{O}(h^0) = \mathcal{O}(1) \end{array} \tag{53}$$

where $\mathbf{u}_I = \Pi_{\mathbf{V}_h}^{\mathbf{V}} \mathbf{u}$ and $p_I = \Pi_{P_h}^{P} p$ are the interpolants defined by the interpolation operators

$$\Pi_{\mathbf{V}_h}^{\mathbf{V}} : \mathbf{V} \rightarrow \mathbf{V}_h \qquad \text{and} \qquad \Pi_{P_h}^{P} : P \rightarrow P_h.$$

Now, we can invoke the triangle inequality to bound the discretization errors

$$\begin{array}{rcl} \|\mathbf{u} - \mathbf{u}_h\|_1 & \leq & \|\mathbf{u} - \mathbf{u}_I\|_1 + \|\mathbf{u}_I - \mathbf{u}_h\|_1 = \mathcal{O}(h) + \mathcal{O}(h) = \mathcal{O}(h) \\ \|p - p_h\|_1 & \leq & \|p - p_I\|_1 + \|p_I - p_h\|_1 = \mathcal{O}(h) + \mathcal{O}(1) = \mathcal{O}(1). \end{array} \tag{54}$$

5.5.2 Error estimates in L^2-norm

The tool to find an L^2-bound of the error for the displacement field is the dual problem:

Find $\eta \in \mathbf{V}$ and $\xi \in P$ such that

$$\left\{ \begin{array}{rcll} a(\alpha, \eta) + b(\alpha, \xi) & = & (\mathbf{u} - \mathbf{u}_h, \alpha) & \forall \alpha \in \mathbf{V} \\ b(\eta, \zeta) - c(\xi, \zeta) & = & 0 & \forall \zeta \in P \end{array} \right. \tag{55}$$

If we chose $\alpha = \mathbf{u} - \mathbf{u}_h$, we find that

$$\begin{aligned}\|\mathbf{u} - \mathbf{u}_h\|_0 \leq\ & hC'\|\mathbf{u} - \mathbf{u}_h\|_1 \\ & + hC''\|p - p_h\|_0 + \sigma C'''\mu\|p - p_h\|_1 \\ & + \sigma C''''\|\mathbf{f}\|_0\end{aligned} \tag{56}$$

for some constants C' - C'''' independent of h. The calculations are tedious but straightforward.

Similarly, to find a bound on $\|p - p_h\|_0$, one can consider the dual problem for the pressure field:

$$\begin{array}{l}\text{Find } \theta \in \mathbf{V} \text{ and } \chi \in P \text{ such that} \\ \left\{\begin{array}{rcll} a(\alpha,\theta) + b(\alpha,\chi) & = & \mathbf{0} & \forall\alpha \in \mathbf{V} \\ b(\theta,\zeta) - c(\chi,\zeta) & = & (p - p_h, \zeta) & \forall\zeta \in P \end{array}\right.\end{array} \tag{57}$$

If $\zeta = p - p_h$, we find that

$$\|p - p_h\|_0 \leq h\|p - p_h\|_1. \tag{58}$$

After combining Equations (56) and (58), the required L^2-estimate follows,

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + h\|p - p_h\|_0 \quad \leq h^2\|\mathbf{f}\|_0 = \mathcal{O}(h^2). \tag{59}$$

6 A viscoelastic model

A viscoelastic material is characterized by the ability to absorb an applied load instantly (an elastic response), and on a long enough time scale, relax the stress in the body by a viscous flow.

In what follows, we use the constitutive relations for a Maxwellian viscoelastic solid, as presented in [30]. The instantaneous response of the material is elastic, and the strain history is approximated as a superposition of Hookean responses for all previous times τ. This yields the following expression for the incremental Cauchy stress tensor

$$\begin{aligned}\sigma(\mathbf{x},t) =& \lambda(\mathbf{x},t,t)\nabla\cdot\mathbf{u}(\mathbf{x},t,t)I + \mu(\mathbf{x},t,t)\varepsilon(\mathbf{u}(\mathbf{x},t)) \\ & - \int_0^t \lambda_\tau(\mathbf{x},t,\tau)\nabla\cdot\mathbf{u}(\mathbf{x},\tau)I + \mu_\tau(\mathbf{x},t,\tau)\varepsilon(\mathbf{u}(\mathbf{x},\tau))d\tau,\end{aligned} \tag{60}$$

where the subscript indicates differentiation with respect to τ. The parameters $\lambda(\mathbf{x}, t, \tau)$ and $\mu(\mathbf{x}, t, \tau)$ are the viscoelastic counterparts to the elastic Lamé coefficients,

$$\lambda(\mathbf{x}, t, \tau) = \lambda_E(\mathbf{x}) e^{-\frac{(t-\tau)}{\tau_0}} \quad \mu(\mathbf{x}, t, \tau) = \mu_E(\mathbf{x}) e^{-\frac{(t-\tau)}{\tau_0}}, \tag{61}$$

where $\tau_0(\mathbf{x}) = \frac{\mu_E(\mathbf{x})}{\eta(\mathbf{x})}$ is the Maxwell time, and η is the dynamic viscosity, see [21]. In the sequel the space dependence of the parameters is omitted for notational simplicity.

Equation (60) is substituted into Equation (23), the moment balance equation for a pre-stressed solid earth in Section 5.2.1, and the corresponding variational problem reads:

$$\begin{aligned}
&\text{Find } \mathbf{u}(t) \in \mathbf{V} \quad \forall t \in \mathcal{I} \text{ such that} \\
&a(\lambda(t,t), \mu(t,t), \mathbf{u}(t), \mathbf{v}) \\
&\quad - \int_0^t \widehat{a}(\lambda_\tau(t,\tau), \mu_\tau(t,\tau), \mathbf{u}(\tau), \mathbf{v}) = l(\mathbf{v}, t) \quad \forall \mathbf{v} \in \mathbf{V} \text{ and } \forall t \in \mathcal{I},
\end{aligned} \tag{62}$$

where $\mathbf{V} \subset (H^1(\Omega))^d$, $d = 2, 3$ and $\mathcal{I} = [0, T]$. The bilinear forms in Equation (62) are

$$\begin{aligned}
&a(\lambda(t,\tau), \mu(t,\tau), \mathbf{u}(t), \mathbf{v}) \\
&\ = \widehat{a}(\lambda(t,\tau), \mu(t,\tau), \mathbf{u}(t), \mathbf{v}) + \bar{a}(\mathbf{b}, \mathbf{c}, \mathbf{u}(t), \mathbf{v}),
\end{aligned}$$

where

$$\begin{aligned}
&\widehat{a}(\lambda(t,\tau), \mu(t,\tau), \mathbf{u}(t), \mathbf{v}) = \\
&\int_\Omega \lambda(t,\tau) \nabla \cdot \mathbf{u}(t) \nabla \cdot \mathbf{v} + \mu(t,\tau) \varepsilon(\mathbf{u}(t)) : \varepsilon(\mathbf{v})
\end{aligned} \tag{63}$$

and

$$\begin{aligned}
&\bar{a}(\mathbf{b}, \mathbf{c}, \mathbf{u}(t), \mathbf{v}) \\
&= \int_\Omega -\nabla(\mathbf{u}(t) \cdot \mathbf{b}) \cdot \mathbf{v} + (\nabla \cdot \mathbf{u}(t))(\mathbf{c} \cdot \mathbf{v}))
\end{aligned} \tag{64}$$

The colon operation (:) is the tensor scalar product, $a : b = \sum_i \sum_j a_{ij} b_{ij}$.

Discretization in time The interval $\mathcal{I}$ is discretized in the instances t_j, $j = 0, 1, \ldots, M$. In the j:th time step, Equation (62), can be rewritten as:

$$
\begin{aligned}
&\text{Find } \mathbf{u}(t_j) \in \mathbf{V} \quad \forall t_j \in \mathcal{I} \text{ such that}\\
&a(\lambda(t_j,t_j),\mu(t_j,t_j),\mathbf{u}(t_j),\mathbf{v})\\
&\quad - \int_{t_{j-1}}^{t_j} \widehat{a}(\lambda_\tau(t_j,\tau),\mu_\tau(t_j,\tau),\mathbf{u}(\tau),\mathbf{v}) = l(\mathbf{v},t_j)\\
&+ \quad \int_0^{t_{j-1}} \widehat{a}(\lambda_\tau(t_{j-1},\tau),\mu_\tau(t_{j-1},\tau),\mathbf{u}(\tau),\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V} \text{ and } \forall t \in \mathcal{I}
\end{aligned}
\tag{65}
$$

by splitting the time integral at $t = t_{j-1}$, and move the part with known quantities to the right-hand-side. Next, the time integral on the left-hand side of Equation (65), is numerically integrated with, e.g., the trapezoidal rule, and we get

$$
\begin{aligned}
&\text{Find } \mathbf{u}(t_j) \in \mathbf{V} \quad \forall t_j \in \mathcal{I} \text{ such that}\\
&a(\lambda(t_j,t_j),\mu(t_j,t_j),\mathbf{u}(t_j),\mathbf{v})\\
&\quad - \frac{t_j - t_{j-1}}{2}\widehat{a}(\lambda_\tau(t_j,t_j),\mu_\tau(t_j,t_j),\mathbf{u}(t_j),\mathbf{v}) = l(\mathbf{v},t_j)\\
&\quad + \frac{t_j - t_{j-1}}{2}\widehat{a}(\lambda_\tau(t_j,t_{j-1}),\mu_\tau(t_j,t_{j-1}),\mathbf{u}(t_{j-1}),\mathbf{v})\\
&\quad + \int_0^{t_{j-1}} \widehat{a}(\lambda_\tau(t_{j-1},\tau),\mu_\tau(t_{j-1},\tau),\mathbf{u}(\tau),\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V} \text{ and } \forall t \in \mathcal{I}
\end{aligned}
\tag{66}
$$

Due to the special form of the viscoelastic Lamé coefficients, Equation (61), Equation (66) simplifies to

$$
\begin{aligned}
&\text{Find } \mathbf{u}(t_j) \in \mathbf{V} \quad \forall t_j \in \mathcal{I} \text{ such that}\\
&(1 - \frac{t_j - t_{j-1}}{2\tau_0})\widehat{a}(\lambda(t_j,t_j),\mu(t_j,t_j),\mathbf{u}(t_j),\mathbf{v})\\
&\quad + \bar{a}(\mathbf{b},\mathbf{c}),\mathbf{u}(t_j),\mathbf{v}) = l(\mathbf{v},t_j)\\
&\quad + \frac{t_j - t_{j-1}}{2\tau_0}\widehat{a}(\lambda(t_j,t_{j-1}),\mu(t_j,t_{j-1}),\mathbf{u}(t_{j-1}),\mathbf{v})\\
&\quad + \int_0^{t_{j-1}} \widehat{a}(\lambda_\tau(t_{j-1},\tau),\mu_\tau(t_{j-1},\tau),\mathbf{u}(\tau),\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V} \text{ and } \forall t \in \mathcal{I}.
\end{aligned}
\tag{67}
$$

For notational simplicity, in Equation (67) and Equation (68), we assume that the Maxwell time τ_0 is constant on the whole domain Ω.

The integral memory term in Equation 67 is also easily computed thanks to the exponential form of the viscoelastic Lamé coefficients. Below we write the exponential part of λ and μ outside the bilinear form $\widehat{a}$ for clarity,

$$
\begin{aligned}
&\int_0^{t_{j-1}} \widehat{a}(\lambda_\tau(t_{j-1},\tau), \mu_\tau(t_{j-1},\tau), \mathbf{u}(\tau), \mathbf{v}) = \\
&\qquad \int_0^{t_{j-2}} \frac{1}{\tau_0} e^{\frac{t_{j-1}-\tau}{\tau_0}} \widehat{a}(\lambda_E, \mu_E, \mathbf{u}(\tau), \mathbf{v}) \\
&\qquad + \int_{t_{j-2}}^{t_{j-1}} \frac{1}{\tau_0} e^{\frac{t_{j-1}-\tau}{\tau_0}} \widehat{a}(\lambda_E, \mu_E, \mathbf{u}(\tau), \mathbf{v}) = \\
&\qquad e^{\frac{t_{i-1}-t_{i-2}}{\tau_0}} \int_0^{t_{j-2}} \frac{1}{\tau_0} e^{\frac{t_{j-2}-\tau}{\tau_0}} \widehat{a}(\lambda_E, \mu_E, \mathbf{u}(\tau), \mathbf{v}) \\
&\qquad + \int_{t_{j-2}}^{t_{j-1}} \frac{1}{\tau_0} e^{\frac{t_{j-1}-\tau}{\tau_0}} \widehat{a}(\lambda_E, \mu_E, \mathbf{u}(\tau), \mathbf{v}).
\end{aligned}
\tag{68}
$$

Discretization in space The discrete formulation of Equation (62), reads

$$
\begin{aligned}
&\text{Find } \mathbf{u}_h(t) \in \mathbf{V}^h \quad \forall t_j \in \mathcal{I} \text{ such that} \\
&a(\lambda(t,t), \mu(t,t), \mathbf{u}_h(t), \mathbf{v}_h) \\
&\quad - \int_0^t \widehat{a}(\lambda_\tau(t,\tau), \mu_\tau(t,\tau), \mathbf{u}_h(\tau), \mathbf{v}_h) = l(\mathbf{v}_h, t) \quad \forall \mathbf{v}_h \in \mathbf{V}^h \text{ and } \forall t \in \mathcal{I},
\end{aligned}
\tag{69}
$$

and the discrete version of Equation (67), with constant τ_0, is

$$
\begin{aligned}
\left[(1 - \frac{\Delta t_j}{2\tau_0})\widehat{A}(t_j, t_j) + \bar{A}(t_j, t_j)\right] \mathbf{u}_j^h = \mathbf{l} + \frac{\Delta t_j}{2\tau_0} \widehat{A}(t_j, t_{j-1}) \mathbf{u}_{j-1}^h \\
+ \int_0^{t_{j-1}} \widehat{A}(t_{j-1}, \tau) \mathbf{u}_\tau^h,
\end{aligned}
\tag{70}
$$

where $\widehat{a}_{ik}(t_j, t_l) = a(\lambda(t_j, t_l), \mu(t_j, t_l), \mathbf{v}_i^h, \mathbf{v}_k^h)$, $\bar{a}_{ik} = \bar{a}(\mathbf{b}, \mathbf{c}, \mathbf{v}_i^h, \mathbf{v}_k^h)$, and $l_i = l(\mathbf{v}_i^h)$.

The assumption for a constant Maxwell time τ_0 in Equation (67), (68), and (70) means no loss of generality, since, we need to perform only a matrix-vector multiplication. This is easily implemented even if the matrices $\widehat{A}$ and $\bar{A}$ are assembled only on subdomains with constant coefficients (where the subdomains can be as small as a single finite element).

7 Conclusions

In Papers A, B, C, D and in the extensions, included in this summary, we have shown how efficient block-factorized preconditioners can be constructed and applied within the context of two problems with a different origin and different properties.

In the first problem, the matrix arises from a BEM-type discretization of crack propagation problems. Numerical examples reveal that iterative solution methods are efficient and competitive alternatives to direct solvers, especially when the system matrix admits (or can be permuted into) a proper 2-by-2-block form and a full-block factorized preconditioner with approximate blocks can be employed.

In the second problem, the matrix, which is nonsymmetric and indefinite, arises from a saddle-point formulation of the purely elastic isostatic model of glacial rebound. Analysis reveals that the bilinear form $a(\cdot,\cdot)$ associated with the displacements is in general not coercive. For the special case when the pre-stress is hydrostatic and the solid is incompressible, which is the case for the isostatic model, $a(\cdot,\cdot)$ is elliptic.

Stability analysis of a simple equal order finite element approximation of this nonsymmetric saddle-point problem shows that the choice of a stabilization parameter $\sigma_h = \mathcal{O}(h^2)$, leads to a consistent and stable discretization.

The numerical experiments illustrate that the known block-triangular and indefinite preconditioners exhibit a robust behavior, provided that good approximations D_1 of the pivot block M, and good approximations D_2 to the (negative) Schur complement matrix S, can be found. A cheap nonsymmetric approximation of S is assembled from element Schur matrices, and numerically this approximation shows to work well.

The blocks D_1 and D_2 are solved with an inner iterative solver, preconditioned with an algebraic multilevel preconditioner. For both blocks, approximated (coarse mesh) Schur complements are assembled from element Schur complements.

The numerical experiments show also that for not strongly nonsymmetric blocks M, the choice of a block-diagonal, spectrally equivalent and symmetric preconditioner for M gives an outer preconditioner that is robust and scalable.

8 Future Work

The numerical results for the saddle point matrices have been given as is in this thesis, without thorough theoretical justifications. The first part of the future plans for this research project is therefore to estimate the eigenvalue distribution of $\mathcal{D}^{-1}\mathcal{A}$, and estimate the convergence rate of the preconditioned iterative solver, following the lines of [4].

The second step is to study and simulate the viscoelastic response of the lithosphere to glaciation and deglaciation. For this, we need to analyze and implement the time-stepping scheme in Equation (70), and how the performance of the iterative methods, used for the elastic case, have to be amended in the viscoelastic case.

The third step is to change the model of the lithosphere as an isostatic viscoelastic solid, and take into account effects such as stress induced by temperature differences in the Earth, and plate tectonic processes. A model for this can be found in [25] and related work.

References

[1] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1996.

[2] O. Axelsson. On iterative solvers in structural mechanics; separate displacement orderings and mixed variable methods. *Mathematics and Computers in Simulation*, 50:11–30, 1999.

[3] O. Axelsson, V. Barker, M. Neytcheva, and B. Polman. Solving the stokes problem on a massively parallel computer. *Mathematical Modelling and Analysis*, 4:1–22, 2000.

[4] O. Axelsson and M. Neytcheva. Preconditioning methods for constrained optimization problems. *Numerical Linear Algebra Applications*, 10:3–31, 2003.

[5] O. Axelsson and P. Vassilevski. Algebraic multilevel preconditioning methods. I. *Numerische Mathematik*, 56(2-3):157–177, 1989.

[6] E. Bängtsson. A consistent stabilized formulation for a nonsymmetric saddle-point problem. Technical Report 2005-030, Department of Information Technology, Uppsala University, 2005.

[7] E. Bängtsson and M. Neytcheva. An agglomerate multilevel preconditioner for linear isostasy saddle point problems. *Accepted for publication in Lecture Notes in Computer Science*, 2005.

[8] E. Bängtsson and M. Neytcheva. Algebraic preconditioning versus direct solvers for dense linear systems as arising in crack propagation. *Communications in Numerical Methods in Engineering*, 21:73–81, 2005.

[9] E. Bängtsson and M. Neytcheva. Numerical simulations of glacial rebound using preconditioned iterative solution methods. *Applications of Mathematics*, 50(3), 2005.

[10] M. Benzi, G. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Mathematica*, pages 1–137, 2005.

[11] E. Botta and F. Wubs. Matrix renumbering ILU: an effective algebraic multilevel ILU preconditioner for sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(4):1007–1026, 1999.

[12] D. Braess. *Finite elements. Theory, fast solvers, and applications in solid mechanics.* Cambridge University Press, second edition, 2001.

[13] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[14] M. Brezina, A. Cleary, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, and J. Ruge. Algebraic multigrid based on element interpolation (AMGe). *SIAM Journal on Scientific Computing*, 22(5):1570–1592, 2000.

[15] F. Brezzi and K.-J. Bathe. A discourse on the stability conditions for mixed finite element formulations. *Computer Methods in Applied Mechanics and Engineering*, 82:27–57, 1990.

[16] K. Chen. An analysis of sparse approximate inverse preconditioners for boundary integral equations. (english). *SIAM Journal on Matrix Analysis and Applications*, 22(4):1058–1078, 2001.

[17] S. L. Crouch. Solution of plane elasticity problems by the displacement discontinuity method. *International Journal for Numerical Methods in Engineering*, 10:301–343, 1976.

[18] L. Inoveckỳ. Postglacial relaxation of the earth's models in cylindrically symmetric geometry. Master's thesis, Charles University, Prague, 2003.

[19] J. Jones and P. Vassilevski. AMGE based on element agglomeration. *SIAM Journal on Scientific Computing*, 23(1):100–133, 2001.

[20] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 24(1):237–256, 2002.

[21] V. Klemann, P. Wu, and D. Wolf. Compressible viscoelasticity: stability of solutions for homogeneous plane-Earth models. *Geophysical Journal International*, 153:569–585, 2003.

[22] L. Kolotilina and A. Yeremin. Factorized sparse approximate inverse preconditionings. *SIAM Journal on Matrix Analysis and Applications*, 14:45–58, 1993.

[23] J. Kraus. Algebraic multilevel preconditioning of finite element matrices using local schur complements. *Numerical Linear Algebra with Applications*, 12:1–19, 2005.

[24] W. Lai, D. Rubin, and E. Krempl. *Introduction to Continuum Mechanics*. Butterworth Heinemann, third edition, 1993.

[25] J. Nedoma. *Numerical Modelling in Applied Geodynamics*. John Wiley & Sons, 2000.

[26] A. Ramage. A multigrid preconditioner for stabilised discretisations of advection-diffusion problems. *Journal of Computational and Applied Mathematics*, 110:187–203, 1999.

[27] Y. Saad. ILUT: A dual threshold incomplete lu factorization. *Numerical Linear Algebra with Applications*, 1(4):387–402, 1994.

[28] Y. Saad. Iterative methods for sparse linear systems. http://www-users.cs.umn.edu/ saad/books.html, January 2000. Second Edition with corrections.

[29] Y. Saad and B. Suchomel. ARMS: an algebraic recursive multilevel solver for general sparse linear systems. *Numerical Linear Algebra with Applications*, 9:359–378, 2002.

[30] S. Shaw, M. Warby, C. Dawson, and M. Wheeler. Numerical techniques for the treatment of quasistatic viscoelastic stress problems in linear isotropic solids. *Computer Methods in Applied Mechanics and Engineering*, 118:211–237, 1994.

[31] B. Shen. *FRACOD^{2D} Two Dimensional Fracture Propagation Code, version 1.1, User's manual.* `http://www.fracom.fi`.

[32] K. Stüben. *Multigrid*, chapter An Introduction to Algebraic Multigrid, pages 413–479. Academic Press, 2001.

[33] P. Sundqvist. *Numerical Computations with Fundamental Solutions.* PhD thesis, Department of Information Technology, Uppsala University, May 2005.

[34] P. Vassilevski. On two ways of stabilizing the hierarchical basis multilevel methods. *SIAM Review*, 39(1):18–53, March 1997.

Recent licentiate theses from the Department of Information Technology

2003-015	Erik Berg: *Methods for Run Time Analysis of Data Locality*
2004-001	Niclas Sandgren: *Parametric Methods for Frequency-Selective MR Spectroscopy*
2004-002	Markus Nordén: *Parallel PDE Solvers on cc-NUMA Systems*
2004-003	Yngve Selén: *Model Selection*
2004-004	Mohammed El Shobaki: *On-Chip Monitoring for Non-Intrusive Hardware/Software Observability*
2004-005	Henrik Löf: *Parallelizing the Method of Conjugate Gradients for Shared Memory Architectures*
2004-006	Stefan Johansson: *High Order Difference Approximations for the Linearized Euler Equations*
2005-001	Jesper Wilhelmsson: *Efficient Memory Management for Message-Passing Concurrency — part I: Single-threaded execution*
2005-002	Håkan Zeffer: *Hardware-Software Tradeoffs in Shared-Memory Implementations*
2005-003	Magnus Ågren: *High-Level Modelling and Local Search*
2005-004	Oskar Wibling: *Ad Hoc Routing Protocol Validation*
2005-005	Peter Nauclér: *Modeling and Control of Vibration in Mechanical Structures*
2005-006	Erik Bängtsson: *Robust Preconditioned Iterative Solution Methods for Large-Scale Nonsymmetric Problems*

UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden