UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 296

Robust Preconditioners Based on the Finite Element Framework

ERIK BÄNGTSSON

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2007

ISSN 1651-6214
ISBN 978-91-554-6870-5
urn:nbn:se:uu:diva-7828

Dissertation presented at Uppsala University to be publicly examined in room 2247, building 2, Polacksbacken, Lägerhyddsvägen 2, Uppsala, Friday, May 11, 2007 at 10:15 for the degree of Doctor of Philosophy. The examination will be conducted in English.

Abstract
Bängtsson, E. 2007. Robust Preconditioners Based on the Finite Element Framework. Acta Universitatis Upsaliensis. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 296. 84 pp. Uppsala. ISBN 978-91-554-6870-5.

Robust preconditioners on block-triangular and block-factorized form for three types of linear systems of two-by-two block form are studied in this thesis.

The first type of linear systems, which are dense, arise from a boundary element type of discretization of crack propagation problems. 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, which are sparse, indefinite and nonsymmetric, arise from a finite element (FE) discretization of the partial differential equations (PDE) that describe (visco)elastic glacial isostatic adjustment (GIA). The Schur complement approximation in the block preconditioners is constructed by assembly of local, exactly computed Schur matrices. The quality of the approximation is verified in numerical experiments.

When the block preconditioners for the indefinite problem are combined with an inner iterative scheme preconditioned by a (nearly) optimal multilevel preconditioner, the resulting preconditioner is (nearly) optimal and robust with respect to problem size, material parameters, number of space dimensions, and coefficient jumps.

Two approaches to mathematically formulate the PDEs for GIA are compared. In the first approach the equations are formulated in their full complexity, whereas in the second their formulation is confined to the features and restrictions of the employed FE package. Different solution methods for the algebraic problem are used in the two approaches. Analysis and numerical experiments reveal that the first strategy is more accurate and efficient than the latter.

The block structure in the third type of algebraic systems is due to a fine-coarse splitting of the unknowns. The inverse of the pivot block is approximated by a sparse matrix which is assembled from local, exactly inverted matrices. Numerical experiments and analysis of the approximation show that it is robust with respect to problem size and coefficient jumps.

Keywords: FEM, iterative solution method, algebraic multilevel preconditioner, sparse approximate inverse, block preconditioner, Schur complement approximation, nonsymmetric saddle point matrix, isostatic glacial adjustment, pre-stress advection, elasticity, viscoelasticity, (in)compressible solid, ABAQUS, BEM/DDM

Erik Bängtsson, Department of Information Technology, Box 337, Uppsala University, SE-75105 Uppsala, Sweden

© Erik Bängtsson 2007

ISSN 1651-6214
ISBN 978-91-554-6870-5
urn:nbn:se:uu:diva-7828 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-7828)

Till Petra

List of Papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

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

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

III Bängtsson, E., Neytcheva, M. (2006) An agglomerate multilevel preconditioner for linear isostasy saddle point problems. *Lecture Notes in Computer Science*, 3743:113–120, *Proceedings of the* 5*th International Conference on Large-scale Scientific Computations 2005*.

IV Bängtsson, E., Neytcheva, M. Preconditioning of nonsymmetric saddle point systems as arising in modelling of visco-elastic problems, Submitted to *Electronic Transactions on Numerical Analysis*, Nov 2006.

V Bängtsson, E., Lund, B. A comparison between two solution techniques to solve the equations of linear isostasy, Submitted for publication to *International Journal for Numerical Methods in Engineering*, Jan 2007. Also available as *Technical Report 2006-051*, Department of Information Technology, Uppsala University, 2006.

VI Bängtsson, E., Neytcheva, M. (2007) Finite element block-factorized preconditioners, *Technical Report 2007-008*, Department of Information Technology, Uppsala University.

Reprints were made with permission from the publishers.

Contents

1. Introduction

The human being is a curious creature and despite, or maybe due to, a long history of scientific research, we continue to ask ourselves questions about various phenomena we observe in the world around us. The classical way to obtain answers to those questions is to perform experiments, observe the outcome, and draw conclusions. But, in many fields of science, due to practical, technical, or economical obstacles, this approach is not possible. For example, in geophysics and astrophysics, the length and time scales are enormous, and 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 high cost. Clearly, 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 of interest mathematically, which usually involves partial differential equations (PDE) as model tools, and solve the so-arising equations numerically.

Partial differential equations constitute the foundation of Mathematical Physics, and in general, except for a limited number of special cases, their analytical solutions are not known. This means that the solution to the PDE must be approximated, and in order to do this the PDE is to be discretized. The field of Scientific Computing is devoted to this discretization and the efficient solution of the so-arising linear or nonlinear systems of equations. In both cases, the need arises to solve a linear equation of the type

$$A\mathbf{x} = \mathbf{b}, \tag{1.1}$$

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

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 computational effort and memory demand as small as possible. For 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 latter terminology is explicitely stated.

(i) Robustness means that the iterative solver converges independently of the values 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, that is, the number of iterations required for the method to converge up to a given tolerance, must be independent of the number of degrees of freedom (n). When this is the case, and provided that the cost for one iteration is $\mathcal{O}(n)$, the overall arithmetic work for the solution method becomes proportional to n. The latter holds for sparse matrices. 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 be fast in terms of CPU-time. To achieve this, the iterative solution method must be computationally efficient (requiring few arithmetic operations per unknown), and

(iv) it should require an amount of memory proportional to n.

(v) The management of data should make beneficial use of the memory hierarchy of the computer.

(vi) Finally, the iterative solution method should be highly parallelizable. That is, portions of the computational work of the method can be easily performed independently of each other, or with a minimum of interprocessor communication.

In the papers comprising this thesis, we have focused on the development of robust preconditioners for linear systems of a block-factorized form. This form may be available due to the structure of the underlying PDE, due to a reordering of the degrees of freedom, or a combination of the two. The target problem addressed in Paper I originates from a boundary element method (BEM) discretization of a model of crack propagation in brittle material, while the problem in Papers II – V originates from finite element (FE) modeling of the lithospheres elastic and viscoelastic response to glaciation and deglaciation. Paper VI is devoted to various means to reduce the computational complexity of block-factorized two- or multilevel preconditioners.

The outline of this thesis is as follows. Chapter 2 contains a brief introduction to the finite element method (FEM), to direct and iterative solution methods for linear systems of equations, and to the concept of preconditioning. Chapter 3 is devoted to some different classes of preconditioners

for the case when A admits a two-by-two block form. In Chapter 4 the six papers that constitute this thesis are described, and this summary is concluded in Chapter 5 where an outlook into future work is presented.

Some notations. Throughout this thesis, unless stated otherwise, uppercase Roman letters (A, B, C) denote matrices, script uppercase letters ($\mathscr{A}$, $\mathscr{B}$, $\mathscr{D}$) 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. Linear Systems of Equations

In this chapter solution methods for linear systems of equations are discussed. Also included is a brief description of the finite element method because a FE discretization of a PDE is one possible origin of a linear system of equations. Furthermore, some properties for the so-arising FE matrices are utilized in the solution of the corresponding linear systems.

The outline of this chapter is as follows. Section 2.1 contains a brief introduction of the finite element method, and in Section 2.2, direct and iterative solution methods for linear systems of equations are presented and discussed. Further, in Section 2.3, the concept of preconditioning is presented together with an overview of often used preconditioning techniques.

2.1 The finite element method

The purpose of the presentation in this section is not to give a thorough description of the finite element method, but rather to introduce concepts and notations related to the FE framework that are used throughout the thesis. For a comprehensive and detailed account for the finite element method, the reader is referred to a textbook on the subject, such as [29].

To introduce the FE method, let us consider the variational problem:

$$\begin{aligned} &\text{Find } u \in V \text{ such that} \\ &a(u,v) = f(v) \quad \forall\, v \in V, \end{aligned} \tag{2.1}$$

where V is an appropriate Sobolev space. The bilinear form $a(u,v)$ depends on the discretized PDE, and for example, for the Laplace equation it is of the form

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega.$$

The functional $f(v)$ on the right-hand side of Equation (2.1) typically looks like

$$f(v) = \int_\Omega g \cdot v \, d\Omega.$$

In the finite element (FE) method, an approximation u_h to u is sought in the finite dimensional subspace $V_h \subset V$, and v is replaced by a discrete test function $v_h \in V_h$. The FE problem now reads:

$$\begin{aligned} &\text{Find } u_h \in V_h \subset V \text{ such that} \\ &a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h. \end{aligned} \tag{2.2}$$

Typically, V_h is spanned by nodal basis functions $\phi_i(x)$ with local support, and u_h is given as linear combination of these functions, that is

$$u_h(x) = \sum_{i=1}^{n} u_i \phi_i(x).$$

The basis functions have the property that $\phi_i(x_j) = \delta_{ij}$, where x_j is the coordinate of the j:th node in the discretization, and δ_{ij} is the Kronecker symbol. The linear system of equations that arises after the FE discretization is of the form

$$A\mathbf{u} = \mathbf{f},$$

where $a_{ij} = a(\phi_i, \phi_j)$ $f_i = f(\phi_i)$, and $\mathbf{u} = [u_1, u_2, \ldots, u_n]$. The local support of the basis functions ensures that the system matrix A is sparse.

In the finite element method, the discretized domain Ω_h is a union of N disjoint elements Ω_e, or for short, e. The system matrix A is constructed by assembly of element matrices A^e,

$$A = \sum_{e=1}^{N} P_e A^e P_e^T. \tag{2.3}$$

In Equation (2.3), P_e and P_e^T denote prolongation and restriction operators from the elementwise numbering to the global numbering of the degrees of freedom. In the sequel, when there is no risk for confusion, the sum in Equation (2.3) will be denoted by $\sum_e$, and the prolongation and restriction matrices will be omitted.

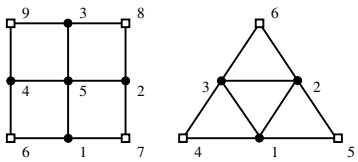


Figure 2.1: A macroelement on a quadrilateral and a triangular mesh

In Chapter 3 and on, the uppercase letter E will, unless stated otherwise, denote a macroelement on the finite element mesh. See Figure 2.1 where typical macroelements on a quadrilateral and a triangular mesh are shown.

2.2 Solution methods

When solving $A\mathbf{x} = \mathbf{b}$ the following three objectives need to be met:

S1 The solution method 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 kept low.

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

Furthermore, the solution methods should be easy to use on parallel computer platforms. This aspect is not focused on in this presentations, but the considered methods are parallelizable using well-known techniques.

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 when the matrix is symmetric positive definite (spd). The system matrix is factorized as $A = LU$ for a general matrix, or as $A = C^T C$ for an spd matrix. Both these methods are robust and meet **S1**, but in many practical and important cases they fail to meet **S2** and **S3**. When A is dense the computational complexity of these methods is $\mathcal{O}(n^3)$ and the memory demand is $\mathcal{O}(n^2)$ (cf. for example [39]). For large n these requirements will make the task to solve Equation (1.1) 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 computational complexity for the factorization of A depends on the so-called bandwidth β of the matrix as $\mathcal{O}(\beta^2 n)$. In many cases, β can be reduced by a proper reordering of A. The computational labour of this reordering depends on the origin and the structure of the matrix. For example, when A is spd the reordering can be computed in $\mathcal{O}(m\,nnz(A))$ operations using the reverse Cuthill–McKee algorithm, cf. [38]. The number m is the so-called maximum degree of the vertices in the graph of A, and $nnz(A)$ is the number of nonzero elements in A. The memory demand to store the factors L and U are of the same magnitude as the storage cost of the original matrix A, $\mathcal{O}(n)$.

On the other hand, if the bandwidth cannot be reduced substantially by some reordering, 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

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k(\mathbf{b} - A\mathbf{x}_k) \tag{2.4}$$

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.

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

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

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 [2] and [56].

An iterative solution method is in general not guaranteed to converge, and **S1** is not always met. Even if the method is theoretically determined to reach the solution in exact arithmetics, the finite precision of the computer representation of floating point numbers may destroy the convergence. One way to accelerate the convergence rate, decrease the number of iterations, and avoid the deterioration of the convergence is to use a proper preconditioner, see Section 2.3, and Chapter 3.

The major part of the arithmetic work of a simple iterative method is spent in performing matrix-vector multiplications, and to solve systems with the preconditioner. The former operation has $\mathscr{O}(n)$ complexity for sparse matrices and $\mathscr{O}(n^2)$ for dense matrices. In order to keep the computational complexity of the method low, the cost for the preconditioning step should not exceed $\mathscr{O}(n)$, or $\mathscr{O}(n^2)$, respectively. When the method converges rapidly, that is, the number of iterations required for convergence is independent of n, the overall complexity is $\mathscr{O}(n)$, and $\mathscr{O}(n^2)$ 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 $\mathscr{O}(n)$.

2.3 Preconditioners

This section contains a brief introduction to the concept of preconditioning, together with short descriptions of some well known preconditioning techniques.

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

P1 The preconditioned version of the linear system $A\mathbf{x} = \mathbf{b}$,

$$G^{-1}A\mathbf{x} = G^{-1}\mathbf{b}, \tag{2.5}$$

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).

P4 The action of G, or G^{-1} on a vector is easily parallelizable.

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

Incorporating a preconditioner transforms the scheme of the iterative method of Equation (2.4) into

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k G^{-1}(\mathbf{b} - A\mathbf{x}_k), \tag{2.6}$$

which can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k G^{-1}A(\mathbf{x} - \mathbf{x}_k), \tag{2.7}$$

where $\mathbf{x}$ is the true solution to Equation (1.1). When 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 (2.5) is called "left preconditioner". It is also possible to use a "right preconditioner", $AG^{-1}\mathbf{y} = \mathbf{b}$, $\mathbf{x} = G^{-1}\mathbf{y}$, or to apply a "symmetric preconditioner", that is, solve $G^{-1}AG^{-1}\mathbf{y} = G^{-1}\mathbf{b}, \mathbf{x} = G^{-1}\mathbf{y}$.

2.3.1 Strategies to construct preconditioners

When strategies to construct preconditioners are discussed, the preconditioning methods are often divided into two main groups, *algebraic* and *problem-oriented*. A preconditioner is said to have an algebraic nature when the information that is passed to it is contained in the matrix A itself, such as the sparsity structure, the graph of the matrix, and the position of the dominating entries. This is in contrast to preconditioners of more problem oriented nature, where one also uses information about the (discretized) underlying problem, such as the PDE, the geometry of the computational domain, the discretization method and/or the mesh.

Since no further information than what is carried by A is used when constructing the algebraic preconditioners, they are more generally applicable than the problem-oriented methods. On the other hand, the latter can in general be more efficient.

2.3.2 Some classes of widely used preconditioners

The classes of methods that are presented in this section are chosen as representatives of widely used classes of preconditioners, but also as a means to explain the building blocks of the more advanced preconditioning methods in Chapter 3.

Incomplete factorization methods

One class of preconditioners of algebraic nature that is widely used is that of the incomplete factorization methods. They are the methods of choice in many applications due to their straightforward implementation and general applicability. For arbitrary matrices the incomplete factorization is based on pointwise incomplete LU factorization (ILU) of A, whereas for symmetric positive definite A, it is based on pointwise incomplete Cholesky factorization (IC). 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 [2] and [55].

Sparse approximate inverse methods

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

A typical SPAI preconditioner is constructed as a matrix $G^{-1} = [g_{ij}]_{i,j=1}^{n}$ with an a priori given sparsity pattern $\mathfrak{S} = \{i, j : g_{ij} \neq 0\}$, e.g. a band matrix. See for example [43] and the references therein.

Transformation based preconditioners

A class of preconditioners with nearly optimal convergence properties is based on Fast Transforms, for instance Fast and Generalized Fourier Transforms. These methods are applicable when A has a structure such that it is (block)-diagonalized by a Fast Transform, or when A can be approximated by such a matrix. Examples of appropriate matrix classes are (block)-circulant or (block)-Toeplitz matrices. See, for example [63], and the references therein.

Domain decomposition methods

The domain decomposition (DD) method, or Schwarz method, 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 "glued" consequently along interior interfaces in some way. This gives the preconditioner attractive parallelization properties. For further information, see for example [61] or [64].

Multigrid methods

The multigrid (MG) method was initially introduced as an efficient iterative solution method for algebraic systems arising form the discretization of elliptical PDEs, for example the Laplace equation. As a basic feature MG possess both optimal convergence rate and optimal computational complexity.

The framework of the MG methods is based on a sequence of grids $T^{(l)}$, $l = 0, \ldots, L$. 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 on the finer level. 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 sufficiently smooth the error.

To demonstrate the MG algorithm, let us consider 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, $\mathbf{r}_1 = \mathbf{b} - A^1\mathbf{x}_1$ 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 $\mathbf{x}_1$.

When the MG method is extended to more that two grid levels, the error equation is recursively solved on coarser and coarser grids, until the exact solution to the error equation is solved on the coarsest one. Depending on how many times each level is visited on the way up and down in the grid hierarchy, V-cycle or W-cycle types of MG methods are obtained.

When $T^{(l-1)}$ 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 [36, 37] and in the work by Bakhvalov (1964) as an efficient iterative solution method for elliptic PDEs. On the other hand, when $T^{(l)}$ 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 [62].

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 [31] and [40].

Multilevel methods

The concept of multilevel (ML) preconditioners is based on some hierarchy of matrices which are not necessarily associated with refinement grids. The ML framework can be viewed as a more general approach than MG or AMG since it does not necessarily involve the MG ingredients (restriction, prolongation, and smoothing). Most commonly, the ML preconditioners utilize some block two-by-two structure of the matrix. For example, the matrix can be split along fine and coarse unknowns according to some mesh hierarchy, or some agglomeration technique.

Representatives for this class are the hierarchical basis functions (HBF) preconditioner [26, 68, 71], the algebraic multilevel iterations (AMLI) method ([16, 17] and follow-up work), and the algebraic recursive multilevel solver (ARMS), see for example [58] and [28]. For further details on multilevel preconditioning methods, see Chapter 3.

3. Block preconditioners

In this chapter, block or block-factorized preconditioners that are based on some 2×2 block form of A are discussed. The exact factorization of a matrix A given in two-by-two block form is

$$
\begin{aligned}
A &= \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} && (3.1)\\
&= \begin{pmatrix} S_1 & A_{12} \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{22}^{-1} A_{21} & I \end{pmatrix} && (3.2)\\
&= \begin{pmatrix} I & 0 \\ A_{21} A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & S_2 \end{pmatrix} && (3.3)\\
&= \begin{pmatrix} I & 0 \\ A_{21} A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & 0 \\ 0 & S_2 \end{pmatrix} \begin{pmatrix} I & A_{12} \\ A_{11}^{-1} & I \end{pmatrix}, && (3.4)
\end{aligned}
$$

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 of the two Schur complements is considered, the subscript is omitted. Utilizing the factorization in Equation (3.3) or (3.4), a preconditioner to A is then sought on block-diagonal, block-triangular, or block-factorized form.

A block 2×2 structure of A can be due to various reasons.

(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) The block structure can be obtained by a 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.

(iii) The matrix structure can correspond to the structure of the underlying PDE. For example, when the matrix arises from a discretization of a system of PDEs (Stokes, Navier–Stokes, Oseen), or from a constrained optimization problem, A exhibits a block structure of saddle point type. More details preconditioners for saddle point matrices are found in Section 3.4.

Unless stated otherwise, throughout this chapter, the system matrix A is assumed to be symmetric and positive definite. The reason for the latter assumption is that the theory for the block preconditioners and multilevel methods discussed here relies on the constant in the strengthened Cauchy-Bunyakowski-Schwarz (CBS) inequality (see Section 3.2), which is defined in the case of spd matrices. For general nonsymmetric matrices, up to the knowledge of the author, no analogous quantity is available.

3.1 The HBF framework

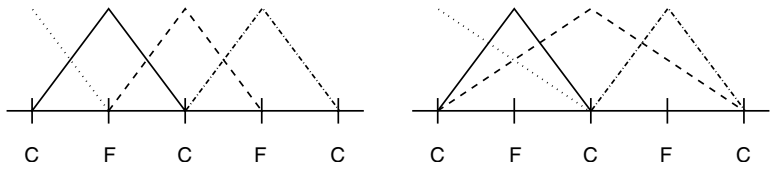
In a series of papers [26, 68, 69, 70], the hierarchical basis functions (HBF) was introduced as a means to construct a preconditioner to a matrix A, arising from the standard nodal basis functions (NBF) version of the finite element method. In the HBF framework the approximate solution $\widehat{u}_h$ is written as

$$\widehat{u}_h = \sum_{i=1}^{n} \widehat{u}_i \widehat{\phi}_i(x),$$

and the hierarchical basis functions $\widehat{\phi}$ are defined as

$$\widehat{\phi}_i = \begin{cases} \phi_i & i \in F \\ \phi_i + \sum_{j \in F_i} c_j \phi_j & i \in C, \end{cases} \tag{3.5}$$

where the sets F and C correspond to a fine-coarse splitting of the underlying mesh. In Equation (3.5), F_i denotes the set of fine nodes that are adjacent to the ith coarse node. In Figure 3.1 a few nodal basis functions (NBF) are shown together with a few hierarchical basis functions on a basis a few nodal bases on a one dimensional grid.



(a) Nodal (b) Hierarchical

Figure 3.1: One dimensional piecewise linear basis functions.

The linear system of equations that corresponds to the HBF formulation of the FE problem reads,

$$\widehat{A}\widehat{\mathbf{u}} = \widehat{\mathbf{f}},$$

where $\widehat{\mathbf{u}} = [\widehat{u}_1, \widehat{u}_2, \ldots, \widehat{u}_n]$.

The linear relation between the hierarchical basis functions $\widehat{\phi}$ and the nodal basis functions ϕ in Equation (3.5) provides a simple transformation

between A and $\widehat{A}$, $\mathbf{u}$ and $\widehat{\mathbf{u}}$, and $\mathbf{f}$ and $\widehat{\mathbf{f}}$ such that

$$\widehat{\mathbf{u}} = J\mathbf{u}, \qquad \widehat{\mathbf{f}} = J\mathbf{f}, \qquad \text{and} \qquad \widehat{A} = J^T AJ.$$

After a reordering of the NBF stiffness matrix along fine and coarse unknowns, the NBF system matrix takes the form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{matrix} \} \text{ fine} \\ \} \text{ coarse,} \end{matrix}$$

and when the transformation matrix J is ordered along the same fine-coarse splitting as in Equation (3.1) it admits the form

$$J = \begin{pmatrix} I & \\ J_{21} & I \end{pmatrix}.$$

The off-diagonal block J_{21} is a sparse matrix with a simple structure. For example, the J_{21} matrix for the example shown in Figure 3.1(b) is as follows

$$J_{21} = \begin{pmatrix} \frac{1}{2} & \\ \frac{1}{2} & \frac{1}{2} \\ & \frac{1}{2} \end{pmatrix}.$$

The HBF stiffness matrix $\widehat{A}$ has the same block structure as A,

$$\widehat{A} = \begin{pmatrix} \widehat{A}_{11} & \widehat{A}_{12} \\ \widehat{A}_{21} & \widehat{A}_{22} \end{pmatrix}, \tag{3.6}$$

and straightforward calculations reveal that

$$\begin{aligned} \widehat{A}_{11} &= A_{11}, \\ \widehat{A}_{12} &= A_{12} + A_{11}J_{12}, \\ \widehat{A}_{21} &= A_{21} + J_{21}A_{11}, \text{ and} \\ \widehat{A}_{22} &= A_{22} + J_{21}A_{11}J_{12} + J_{21}A_{12} + A_{21}J_{12}. \end{aligned}$$

Furthermore, the coarse mesh Schur complement of the NBF stiffness matrix,

$$S_A = A_{22} - A_{21}A_{11}^{-1}A_{12},$$

is identical to the coarse mesh Schur complement matrix of the HBF matrix,

$$S_{\widehat{A}} = \widehat{A}_{22} - \widehat{A}_{21}\widehat{A}_{11}^{-1}\widehat{A}_{12}.$$

It can be noted that in the case of a symmetric positive definite matrix A, $J_{21} = J_{12}^T$.

Remark 3.1.1 *The transformation from A to $\widehat{A}$ is a congruence transformation. That is, the number of positive, negative, and zero eigenvalues of A is not changed under the transformation, and the positive definiteness of $\widehat{A}$ is preserved.*

3.2 The CBS constant

In this section, the properties of the constant γ in the strengthened Cauchy-Bunyakowski-Schwarz (CBS) inequality are defined and discussed. This parameter is of paramount importance for the analysis of the block preconditioners that are presented in Section 3.3. For a more extensive discussion about the CBS constant, see for example [2] and the references therein.

For a two-by-two block matrix as in Equation (3.1), the strengthened CBS inequality reads

$$|\mathbf{v}_1^T A_{12}\mathbf{v}_2| \leq \gamma(\mathbf{v}_1^T A_{11}\mathbf{v}_1)^{1/2}(\mathbf{v}_2^T A_{22}\mathbf{v}_2)^{1/2} \quad \forall \mathbf{v}_1 \in \mathbf{V}_1, \mathbf{v}_2 \in \mathbf{V}_2, \tag{3.7}$$

where $\mathbf{V}_1 \subset \mathbb{R}^{n_1}\backslash N(A_{11})$, $\mathbf{V}_2 \subset \mathbb{R}^{n_2}\backslash N(A_{22})$, and $N(A_{ii})$ denotes the null-space of A_{ii}. An equivalent formulation of Equation (3.7) is

$$\gamma = \sup_{\substack{\mathbf{v}_1 \in \mathbf{V}_1 \\ \mathbf{v}_2 \in \mathbf{V}_2}} \frac{|\mathbf{v}_1^T A_{12}\mathbf{v}_2|}{(\mathbf{v}_1^T A_{11}\mathbf{v}_1)^{1/2}(\mathbf{v}_2^T A_{22}\mathbf{v}_2)^{1/2}}.$$

The CBS constant is the cosine of the angle between the spaces $\mathbf{V}_1$ and $\mathbf{V}_2$, and when the two spaces are disjoint, that is $\mathbf{V}_1 \cap \mathbf{V}_2 = \emptyset$, the upper bound of γ is sharp,

$$0 \leq \gamma < 1.$$

The definition in Equation (3.7) is of purely algebraic nature. In the FE setting however, an equivalent definition can be formulated in terms of the bilinear form $a(\cdot,\cdot)$, namely

$$a(u,v) \leq \gamma\sqrt{a(u,u)}\sqrt{a(v,v)} \quad \forall u \in V_1, v \in V_2, \tag{3.8}$$

where $V_1 \subset V_h$, and $V_2 \subset V_h$, such that $V_1 \cap V_2 = \emptyset$. Further, in the FE framework a local CBS constant γ_E can be computed as

$$\gamma_E = \sup_{\substack{u \in V_1 \\ v \in V_2}} \frac{|a_E(u,v)|}{\sqrt{a_E(u,u)}\sqrt{a_E(v,v)}},$$

where $a_E(u,v)$ is the bilinear form $a(u,v)$ restricted to the element E. As is shown in [9] and [35], among others, the value of the global γ can be estimated locally as

$$\gamma = \max_E \gamma_E.$$

The latter property shows that the CBS constant is independent of inhomogeneities in the coefficients in the underlying PDE, as long as they are aligned with the coarse mesh. Neither does the constant depend on the geometry of the computational domain, the number of elements in the discretization, or the number of refinement levels. On the other hand, the CBS constant does depend the bilinear form $a(\cdot,\cdot)$, the shape of the elements, the type of basis functions, and the number of space dimensions.

In the pioneering works, [9] and [46], upper bounds of the CBS constant for the Laplace equation with homogeneous and inhomogeneous coefficients, are computed. In the first paper, the equation is discretized on a mesh of right-angled triangles, whereas in the second, the shape of the triangles is arbitrary. For further work on estimates of the CBS constant, see [7] and the references therein. In that paper, upper bounds for γ are derived for matrices arising from a FE discretization of the the Laplace equation with anisotropic coefficients, and for the moment balance equation for an elastic anisotropic solid. The derived estimates hold for m-fold refined elements (each side (edge) of a triangle (tetrahedron) is refined m times), and they state that γ is bounded by

$$\gamma_{2D} \leq \sqrt{\frac{m^2-1}{m^2}} \qquad \gamma_{3D} \leq \sqrt{1-\frac{2}{m^4-m^2}}.$$

These estimates hold when piecewise linear basis functions are used, and they are independent of the shape of the refined element. Hence, for standard 2-fold refinement γ is bounded by $\sqrt{3/4}$ in 2D, and by $\sqrt{5/6}$ in 3D.

The importance of the CBS constant is two-fold. Firstly it gives a measure of the strength of the off-diagonal block A_{12} in A, and secondly it provides bound of the spectral equivalence between A_{22} and S_2, namely

$$1-\gamma^2 \leq \frac{\mathbf{v}_2^T S_2 \mathbf{v}_2}{\mathbf{v}_2^T A_{22} \mathbf{v}_2} \leq 1 \quad \forall \mathbf{v}_2 \in \mathbf{V}_2. \tag{3.9}$$

As long as γ is independent of the size of A, jumps in the coefficients of PDE, problem or mesh anisotropy, the geometry of the computational domain, and the number of refinement levels in the mesh, A_{22} is a robust and optimal approximation to the Schur complement S_2. The HBF framework provides a setting where γ is independent of the parameters mentioned above, and bounded away from one. However, for instance, when A arises from the NBF formulation of the FE method the CBS constant can be arbitrarily close to one, and the lower bound in Equation (3.9) becomes nearly zero.

In the next section we consider two-level preconditioning methods, and we shall see how the condition number of the preconditioned matrices depend on γ.

3.3 Two-level preconditioners

The two preconditioners presented in this section are based on the block two-by-two splitting of A in Equations (3.3) and (3.4). The condition number estimates can be found in [2].

3.3.1 Block-diagonal preconditioner

The block diagonal preconditioner B_D is the simplest block preconditioner and it is constructed as an approximation of the block-diagonal of A. That is,

$$B_D = \begin{pmatrix} P & 0 \\ 0 & B_{22} \end{pmatrix} \qquad B_D^{-1} = \begin{pmatrix} P^{-1} & 0 \\ 0 & B_{22}^{-1} \end{pmatrix}, \tag{3.10}$$

where P and B_{22} are spectrally equivalent approximations to A_{11} and A_{22}, respectively. That is, there exists constants $0 < \underline{\alpha} \leq \overline{\alpha}$ and $0 < \underline{\beta} \leq \overline{\beta}$ such that

$$\begin{aligned} &\underline{\alpha}\mathbf{v}_1^T A_{11}\mathbf{v}_1 \leq \mathbf{v}_1^T P\mathbf{v}_1 \leq \overline{\alpha}\mathbf{v}_1^T A_{11}\mathbf{v}_1^T, \quad \forall \mathbf{v}_1 \in \mathbf{V}_1, \text{ and} \\ &\underline{\beta}\mathbf{v}_2^T A_{22}\mathbf{v}_2 \leq \mathbf{v}_2^T B_{22}\mathbf{v}_2 \leq \overline{\beta}\mathbf{v}_2^T A_{22}\mathbf{v}_2^T, \quad \forall \mathbf{v}_2 \in \mathbf{V}_2. \end{aligned} \tag{3.11}$$

The condition number of the preconditioned matrix $B_D^{-1}A$ is bounded by

$$\begin{aligned} \kappa(B_D^{-1}A) \leq \frac{\overline{\alpha}}{\underline{\alpha}(1-\gamma^2)} &\left\{ \frac{1}{2}\left(1+\frac{\underline{\alpha}}{\underline{\beta}}\right) + \left[\left(\frac{1}{2}\left(1-\frac{\underline{\alpha}}{\underline{\beta}}\right)\right)^2 + \frac{\underline{\alpha}}{\underline{\beta}}\gamma^2\right]^{1/2} \right\} \times \\ &\left\{ \frac{1}{2}\left(1+\frac{\overline{\beta}}{\overline{\alpha}}\right) + \left[\left(\frac{1}{2}\left(1-\frac{\overline{\beta}}{\overline{\alpha}}\right)\right)^2 + \frac{\overline{\beta}}{\overline{\alpha}}\gamma^2\right]^{1/2} \right\}, \end{aligned}$$

which, with $P = A_{11}$ and $B_{22} = A_{22}$, simplifies to

$$\kappa(B_D^{-1}A) \leq \frac{1+\gamma}{1-\gamma}. \tag{3.12}$$

When the action of B_D^{-1} on a vector is computed, it is required to solve once with A_{11} and once with A_{22}, and since these two steps are independent of one another, B_D possesses attractive parallelization properties.

3.3.2 Block-factorized preconditioner

The block factorized preconditioner is constructed based either on Equation (3.2) or on (3.3), and for the ease of the presentation, we consider only the latter case here. That is, the preconditioner B_M is defined as

$$B_M = \begin{pmatrix} I & 0 \\ A_{21}P^{-1} & I \end{pmatrix} \begin{pmatrix} P & A_{12} \\ 0 & Q \end{pmatrix}, \tag{3.13}$$

where Q approximates the Schur complement of A. Let us assume that, as in the previous section, P is spectrally equivalent to A_{11} with the constants $\underline{\alpha}$ and $\overline{\alpha}$, and for Q it holds that

$$\underline{\beta}\mathbf{v}_2^T A_{22}\mathbf{v}_2 \leq \mathbf{v}_2^T Q\mathbf{v}_2 \leq \overline{\beta}\mathbf{v}_2^T A_{22}\mathbf{v}_2^T, \quad \forall \mathbf{v}_2 \in \mathbf{V}_2. \tag{3.14}$$

If we further assume that $\overline{\alpha} \geq 1 \geq \underline{\alpha} > \gamma^2$ and $\overline{\beta} \geq 1 \geq \underline{\beta} > \gamma^2$, it can be shown that the extremal eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ of $B_M^{-1}A$ are bounded as

$$\lambda_{\min}(B_M^{-1}A) \geq \left\{ 1 + \frac{\max(\overline{\alpha},\overline{\beta}) - 1}{1-\gamma^2} \left[\frac{1+\overline{r}}{2} + \sqrt{\left(\frac{1-\overline{r}}{2}\right)^2 + \overline{r}\gamma^2} \right] \right\}^{-1},$$

and (3.15)

$$\lambda_{\max}(B_M^{-1}A) \leq \left\{ 1 - \frac{1 - \min(\underline{\alpha},\underline{\beta})}{1-\gamma^2} \left[\frac{1+\underline{r}}{2} + \sqrt{\left(\frac{1-\underline{r}}{2}\right)^2 + \underline{r}\gamma^2} \right] \right\}^{-1}.$$

In Equation (3.15) $\overline{r} = \min\left(\frac{\overline{\alpha}-1}{\overline{\beta}-1}, \frac{\overline{\beta}-1}{\overline{\alpha}-1}\right)$, for $\overline{\alpha} > 1$ and/or $\overline{\beta} > 1$, and $\underline{r} = \min\left(\frac{1-\underline{\alpha}}{1-\underline{\beta}}, \frac{1-\underline{\beta}}{1-\underline{\alpha}}\right)$, for $\underline{\alpha} < 1$ and/or $\underline{\beta} < 1$. When $P = A_{11}$ and $Q = A_{22}$, the bound of the condition number of $B_M^{-1}A$ simplifies to

$$\kappa(B_M^{-1}A) \leq \frac{1}{\sqrt{1-\gamma^2}}. \tag{3.16}$$

This estimate is better than for the block diagonal method, but it is achieved for the prize of an extra solve with P and two multiplications with A_{12}. This drawback is in practice outweighed by the better condition number of the block-factorized method compared to the block-diagonal method, cf. [49].

As was pointed out in Section 3.2, for arbitrary two-by-two block splittings of A, γ is arbitrarily close to unity, and both $\kappa(B_D^{-1}A)$ and $\kappa(B_M^{-1}A)$ deteriorate. In the HBF framework on the other hand, γ is bounded away from one, and the two condition numbers are bounded.

Remark 3.3.1 *The relations between A_{22} and S_2 in* (3.9)*, and A_{22} and Q in* (3.14) *provide a spectral equivalence relation between Q and S_2.*

Before we go on and discuss how to construct the approximations P to A_{11} and Q to S, let us consider matrices on saddle point form and related block preconditioners.

3.4 Saddle point preconditioners

A linear system of equations is said to be of (generalized) saddle point form when the system matrix admits the two-by-two block structure

$$\mathscr{A} = \begin{pmatrix} M & B_1^T \\ B_2 & -C \end{pmatrix}, \tag{3.17}$$

and one or more of the conditions

SP1 M is symmetric,

SP2 the symmetric part of M, $H = \frac{1}{2}(A + A^T)$, is positive semidefinite,

SP3 $B_1 = B_2 = B$,

SP4 C is symmetric and positive semidefinite, or

SP5 $C = 0$,

are fulfilled. This definition of a saddle point problem is found in [27] and here we focus on the case when **SP3** and **SP4** hold. That is, when the saddle point matrix is of the form

$$\mathscr{A} = \begin{pmatrix} M & B^T \\ B & -C \end{pmatrix}, \tag{3.18}$$

and where M is invertible. Linear systems of this form arise in many different applications, such as computational fluid dynamics, linear elasticity, and constrained optimization, among others. Throughout this chapter the discussion is focused on saddle point matrices arising from the discretization of PDEs describing the flow of a fluid or the displacements in a solid.

There exists a large amount of scientific literature on the topic of saddle point problems and related preconditioners. It is not the intention of this section to give a detailed overview of existing methods, but rather to put the preconditioners used in Papers II – V in some perspective. For a recent and exhaustive survey on the numerical solution of saddle point problems, the reader is referred to [27].

Depending on the underlying problem, the properties of the matrices M, B, and C differ. For example, when (3.18) arises from a discretization of the Stokes Equation, or the equations of linear elasticity on mixed variables form, M is symmetric positive definite, whereas for the Oseen problem (or the linearized Navier–Stokes equations), M is a non-symmetric matrix.

The properties of C depend on the nature of the modeled fluid or solid and on the discretization. For incompressible flow and solids, C is a zero matrix, whereas in the compressible case, C is symmetric positive definite. When the discretization of the underlying PDEs is done by the FE method, certain conditions are to be met to ensure the stability of the discretization (cf. Section 4.2 and the references therein). These conditions can be circumvented by a stabilization of the FE discretization, which typically consists of adding a spd matrix to the C block.

The saddle point matrix in Equation (3.18) is preconditioned by a matrix utilizing the block structure defined by the structure of $\mathscr{A}$ itself. The preconditioner can be of block-diagonal, block-triangular, or block-factorized form. As the block-diagonal preconditioner is not employed in any of the

papers underlying this thesis, the presentation here includes only the two latter methods.

3.4.1 Block-triangular preconditioner

The block lower-triangular preconditioner $\mathscr{D}_t$ is of the form

$$\mathscr{D}_t = \begin{pmatrix} D_1 & 0 \\ B & -D_2 \end{pmatrix}, \tag{3.19}$$

where D_1 approximates M and D_2 approximate the negative Schur complement of $\mathscr{A}$, $S_{\mathscr{A}} = C + BD_1^{-1}B$. The choices of D_1 and D_2 are motivated by the relation

$$\mathscr{D}_t^{-1}\mathscr{A} = \begin{pmatrix} D_1^{-1}M & D_1^{-1}B^T \\ D_2^{-1}B(I - D_1^{-1}M) & D_2^{-1}(C + B^T D_1^{-1}B) \end{pmatrix}. \tag{3.20}$$

When $\mathscr{D}_t$ is constructed using the exact matrix blocks, that is with $D_1 = M$ and $D_2 = S_{\mathscr{A}}$, the spectrum of $\mathscr{D}_t^{-1}\mathscr{A}$ is clustered at unity, as is evident from Equation (3.20). For a more rigorous spectral analysis of the preconditioned matrix, see for example [12] and [60]. The spectral properties of $\mathscr{D}_t^{-1}\mathscr{A}$ are beneficial, but the application of the block-triangular preconditioner turns a previously symmetric problem into a nonsymmetric one, and this nonsymmetricity must be compensated for by a more iterations in the iterative solution method [13].

Using a method that was introduced in [30], and recently discussed in [12, 60], the nonsymmetric preconditioned matrix can be symmetrized, with the advantage that the resulting linear system can be solved by a CG-like iterative solution method. The disadvantages of this approach is that it requires a "particularly accurate" [12] preconditioner for M, and is less robust than the nonsymmetric method when B is nearly rank-deficient. On the other hand, as is pointed out in [27], "[...] symmetrization is seldom necessary in practice: if good preconditioners to M and $S_{\mathscr{A}}$ are available, a method like GMRES will converge quickly [...]".

3.4.2 Block-factorized preconditioner

The block factorized saddle point preconditioner $\mathscr{D}_f$ follows naturally from Equation (3.3), and is of the form

$$\mathscr{D}_f = \begin{pmatrix} I & \\ BD_1^{-1} & I \end{pmatrix}\begin{pmatrix} D_1 & B^T \\ & -D_2 \end{pmatrix}. \tag{3.21}$$

The preconditioned matrix $\mathscr{D}_f^{-1}\mathscr{A}$ is then

$$\mathscr{D}_f^{-1}\mathscr{A} = \begin{pmatrix} D_1^{-1}M + D_1^{-1}B^T D_2^{-1}B(I - D_1^{-1}M) & D_1^{-1}B^T(I - D_2^{-1}S_{\mathscr{A}}) \\ -D_2^{-1}B(I - D_1^{-1}M) & D_2^{-1}S_{\mathscr{A}} \end{pmatrix}$$

As for the block triangular preconditioner, when D_1 and D_2 are taken to be the exact pivot block M and the exact negative Schur complement, respectively, the spectrum of $\mathscr{D}_f^{-1}\mathscr{A}$ is clustered at unity, but in this case the resulting matrix is symmetric. This, on the other hand, is achieved at the price of an extra solve with D_1 in each iteration, and an extra multiplication with B, compared to $\mathscr{D}_t$.

When a good approximation D_1 is used in $\mathscr{D}_t$ and $\mathscr{D}_f$, as is shown in the numerical experiments in [13], the overhead in iteration counts for the iterative solution method is very moderate. In Paper II, a less good preconditioner for M is used, and there the difference in iteration count between the two preconditioning methods are larger than in [13]. This result is aligned with a remark in [13] saying that $\mathscr{D}_t$ is more sensitive to the quality of D_1, since the nonsymmetricity of $\mathscr{D}_t^{-1}\mathscr{A}$ may amplify the error induced in the approximation of M.

The D_1 matrix

When choosing D_1, as noted in [27], 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, see, for example, [54].

3.5 Schur complement approximations

In the general case it is not feasible to form the Schur complement exactly. Not only due to the high computational cost associated with the construction of the inverse of the pivot block, but also because the Schur complement is in general a dense matrix even when the underlying matrix is sparse. In order to fulfil the objectives **P2** and **P3**, Q and D_2 should not only be an accurate approximation to the Schur complement, but also sparse, and they shall be constructed such that they are easily handled in a parallel environment. These tasks are not easily achieved, and especially the sparsity and the accuracy demands may contradict each other.

3.5.1 The Q block

For the two-level preconditioners, different methods to form Q have been suggested. Within the HBF-framework, the classical approach is to replace the Schur complement with the coarse mesh matrix $\widehat{A}_{22}$. The quality of this approximation depends on the CBS-constant γ only, as is shown in Equation (3.9).

In the NBF-framework, the coarse mesh matrix is not used, and different approaches to construct Q are available in the literature. One way is to compute a Schur complement approximation where A_{11}^{-1} is replaced by a

sparse approximate inverse D, that is

$$Q = A_{22} - A_{21} D A_{12}.$$

This approach has for example been investigated for M-matrices in [11] (for symmetric matrices), and in [52] (for non-symmetric matrices).

In a recent paper [44], a strategy to approximate the coarse mesh Schur complement by assembly of local, exactly computed, Schur matrices is proposed. The idea is that after a fine-coarse reordering, the element stiffness matrix A_E corresponding to a macroelement E (macroelements for a triangular and a quadrilateral grid are shown in Figure 2.1), has the same 2×2 block form as the global matrix,

$$A^E = \begin{pmatrix} A^E_{11} & A^E_{12} \\ A^E_{21} & A^E_{22} \end{pmatrix}.$$

As long as A^E_{11} are nonsingular, a local Schur complement S^E can be computed exactly,

$$S^E = A^E_{22} - A^E_{21} (A^E_{11})^{-1} A^E_{12}, \tag{3.22}$$

and the local contributions can be assembled to a global approximation

$$Q = \sum_E P_E S^E P_E^T. \tag{3.23}$$

In Equation (3.23) P_E is a prolongation operator from the global node ordering to the ordering on the macroelement E. This approach is further analyzed in [8], and it is shown there that

$$\kappa(Q^{-1}S) \leq \frac{1}{\sqrt{1-\gamma^2}}.$$

That is, the so-assembled Schur complement approximation is spectrally equivalent with the true Schur matrix, and it is of the same quality as the classical coarse mesh matrix approximation $\widehat{A}_{22}$ from the HBF-framework.

3.5.2 The D_2 block

When the preconditioners $\mathscr{D}_t$ and $\mathscr{D}_f$ to the saddle point matrix $\mathscr{A}$ are constructed, it is in some cases known how to obtain a good quality approximation for the Schur complement. In some applications it is enough to approximate M by its diagonal or by some sparse approximate inverse of M. For other problems S can be approximated on a differential operator level, as is possible for the Oseen's problem (see [41]). For the Stokes problem, and for the equations of linear elasticity on mixed variables form, it is known that a good approximation of $BM^{-1}B^T$ is the pressure mass matrix (cf. [27] and the references therein).

In Paper II, we propose a strategy to construct D_2 which is based on the assembly of local, exactly computed Schur complements, in a fashion similar to what is done in Equations (3.22) and (3.23) for the two-level preconditioner. Up to our knowledge, this is a novel approach when applied to saddle point preconditioners, and in this framework it can be explained as follows.

Let $\mathscr{A}^e$ be the element stiffness matrix corresponding to a saddle point problem (for example, the Stokes problem). We observe that $\mathscr{A}^e$ exhibit the same 2×2 block structure as the global stiffness matrix $\mathscr{A}$,

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

and compute the local negative Schur complements $S^e = C^e + B^e M^{e-1} B^{eT}$ [1] exactly on each element e. The matrix D_2 is then assembled from the local Schur matrices, $D_2 = \sum_e S^e$. Numerical experiments in Paper II reveal that this approximation is of the same quality as the well-known Schur complement approximation $D_2 = C + M_p$, where M_p is the pressure mass matrix.

In many cases it is not necessary to explicitely form the Schur complement matrix. For example, a saddle point problem, such as Stokes problem or the elasticity equations on mixed form, can be solved by reducing it to the pressure variable, solve the Schur complement system, and retrieve the velocity (displacements) by solving the pivot block. The Schur complement matrix is solved by some iterative solution method, and hence it suffices that one is able to perform matrix-vector multiplications with S,

$$S\mathbf{v} = C\mathbf{v} + B[M]^{-1}B^T\mathbf{v},$$

where $[M]^{-1}$ denotes an iterative solve with M. The drawback of this approach is that systems with the M block must be solved very accurately in order not to loose the positive definiteness of S, and the high accuracy of the inner solver makes this method costly. However, in for example [6] and [14], it is found that the strategy provides a robust iterative solution method of optimal order for the considered saddle point problems.

3.6 The pivot block P

In the block preconditioners for the two-level block matrix A the need arises to solve, once or twice, with the pivot block A_{11}. The computational complexity for doing this exactly is of the same magnitude as the cost to solve

[1] To form S^e it is required that M^e is invertible. This is ensured on elements away from a Dirichlet boundary by adding a regularization term to the diagonal of M^e, $M^e \leftarrow M^e + \epsilon I$, where $\epsilon = \mathcal{O}(h^2)$.

with A, or $\mathscr{A}$, itself, unless the pivot block has a beneficial structure, such as (tri)diagonal or narrowbanded.

In the HBF framework, when the coarse mesh matrix $\widehat{A}_{22}$ is used to approximate the Schur complement, the approximation P of the A_{11} block does not influence the quality of Q. Then it is enough to ensure that P is spectrally equivalent to A_{11},

$$\underline{\alpha}\mathbf{v}_1^T P\mathbf{v}_1 \le \mathbf{v}_1^T A_{11}\mathbf{v}_1 \le \overline{\alpha}\mathbf{v}_1^T P\mathbf{v}_1, \qquad \forall \mathbf{v}_1 \in \mathbf{V}_1. \tag{3.24}$$

In this way, the positive definiteness of the preconditioner is preserved. See, for example [16], [17], and [9], for details.

An approximation P that satisfies Equation (3.24) can be constructed as a sparse approximate inverse (SPAI), or by some incomplete factorization (ILU, IC, MILU, MIC, etc.). The action of P^{-1} on a vector can also be computed as a few iterations by an inner iterative solution method. This is however an expensive approach, unless the preconditioner for the pivot block is very good.

Next, let us consider the case when the two-by-two splitting of A does not originate from the hierarchical structure of the HBF, for example, when the blocks of A arise from some fine-coarse splitting of the NBF. When P is constructed as an approximate inverse of A_{11}, and Q approximates $S_A = A_{22} - A_{21}A_{11}^{-1}A_{12}$, the Schur complement of A, the following extra conditions on P and Q are introduced in order to ensure the robustness of the two-level preconditioner.

It is shown in [51] that for $\kappa(B_M^{-1}A)$ to be bounded, it is necessary that the following conditions are fulfilled:

(i) P is smaller than A_{11}, in positive definite sense,

$$\mathbf{v}_1^T P\mathbf{v}_1 \le \mathbf{v}_1^T A_{11}\mathbf{v}_1, \qquad \forall \mathbf{v}_1 \in \mathbf{V_1},$$

(ii) Q is spectrally equivalent to the Schur complement,

$$\underline{\beta}\mathbf{v}_2^T S_A\mathbf{v}_2 \le \mathbf{v}_2^T Q\mathbf{v}_2 \le \overline{\beta}\mathbf{v}_2^T S_A\mathbf{v}_2 \qquad \forall \mathbf{v}_2 \in \mathbf{V_2},$$

where $0 < \underline{\beta} \le \overline{\beta}$, and

(iii)

$$\mathbf{v}_2^T A_{21}P^{-1}A_{12}\mathbf{v}_2 \le (1-\xi)\mathbf{v}_2^T A_{22}\mathbf{v}_2 + \xi\mathbf{v}_2^T A_{21}A_{11}^{-1}A_{12}, \qquad \forall \mathbf{v}_1 \in \mathbf{V_1}, \text{ and } \xi < \mathbf{1}, \tag{3.25}$$

or

$$\xi\mathbf{v}_2^T S_A\mathbf{v}_2 \le \mathbf{v}_2^T(A_{22} - A_{21}P^{-1}A_{12})\mathbf{v}_2.$$

The parameter ξ affects the condition number of $B_M^{-1}A$ as

$$\kappa(B_M^{-1}A) \leq \kappa(Q^{-1}S_A)\kappa(P^{-1}A_{11})(2-\xi(1-\beta))(2-\xi),$$

where $\beta = \kappa^{-1}(P^{-1}A_{11})$. Hence, $-\xi$ should be bounded not to destroy the condition number of the preconditioned matrix. One way to ensure this, as was shown in [51], is to chose P such that it acts as a nearly exact inverse of A_{11} for smooth vectors $\mathbf{v}_2$. In [51] it is found also that for symmetric M-matrices, when P is constructed either as a modified ILU factorization, or as a compensated incomplete inverse, the conditions (i) – (iii) above are fulfilled.

In Paper VI, a modification of the block-factorized two-level preconditioner B_M is considered, namely

$$B_M = \begin{pmatrix} P & 0 \\ A_{21} & Q \end{pmatrix} \begin{pmatrix} I & Z_{12} \\ 0 & I \end{pmatrix}, \tag{3.26}$$

where P^{-1} approximates A_{11}^{-1}, and Q is an approximation of S_A, which may not have to utilize the approximation P^{-1} of A_{11}^{-1}. The block Z_{12} is a sparse matrix which approximates the matrix product $A_{11}^{-1}A_{12}$ and it is constructed as a means to reduce the computational complexity of B_M. The numerical experiments in Paper VI reveal that for the two-level FE framework the block preconditioner in Equation (3.26) is optimal and robust with respect to jumps in the problem parameters. This holds for a particular approximations P^{-1} of A_{11}^{-1}, as long as the pivot block is solved accurate enough by an inner iterative solution method which is preconditioned by P^{-1}.

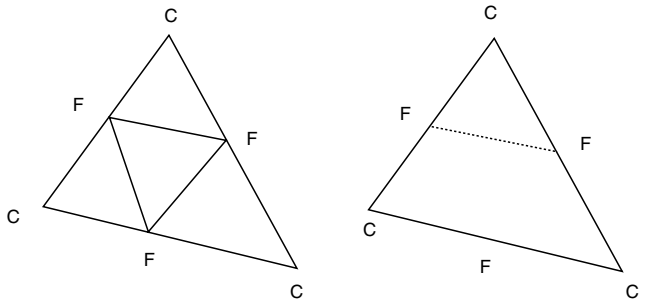


Figure 3.2: The coupling between the fine nodes in two space dimensions. In the right figure, only the strongest coupling remains.

It is in general difficult to construct a good preconditioner for (or approximation of) A_{11} in the case of anisotropic problems, but two examples how to achieve a condition number of the preconditioned matrix, independent of the anisotropy, can be found in [15] and [47].

The approach in [15] fits in the context of block-diagonal multilevel preconditioning for anisotropic elliptic problems. The idea is that the

anisotropy in the problem entails that the couplings between some of the entries in A_{11} are stronger than the couplings between other entries. Figure 3.2 shows a triangular macroelement, which is constructed by agglomeration of four triangles. In the left macroelement, the three internal edges between the midpoints of the outer triangle indicate couplings between the degrees of freedom associated with the midpoint nodes. In the right figure only the strongest of the couplings remains.

When this elimination is done for all macroelements (a local operation), and after a reordering of the degrees of freedom along the strong couplings, the P block becomes a (blockwise) tridiagonal matrix that can be solved exactly with a direct solver at a low cost. This approach is further investigated in [10], where the theory is extended to the framework of multiplicative multilevel preconditioners.

In [47], the anisotropic elliptic problem is discretized on a regular grid which consists of right-angled triangles that are aligned with the x- and y-axis. The fine degrees of freedom are reordered along the dominating direction of the anisotropy such that A_{11} admits a two-by-two block form

$$A_{11} = \begin{pmatrix} D & F \\ F^T & E \end{pmatrix} = \begin{pmatrix} D & 0 \\ F^T & S_{A_{11}} \end{pmatrix} \begin{pmatrix} I & D^{-1}F \\ 0 & I \end{pmatrix},$$

where D is a diagonal matrix, E is block-diagonal with tridiagonal blocks, and $S_{A_{11}} = E - F^T D^{-1} F$. The pivot block approximation is constructed as

$$P = \begin{pmatrix} D & 0 \\ F^T & E \end{pmatrix} \begin{pmatrix} I & D^{-1}F \\ 0 & I \end{pmatrix},$$

and the condition number of $P^{-1}A_{11}$ is independent of the anisotropy of the problem.

3.7 The AMLI method

In this section, the extension of the block-factorized preconditioner in Section 3.3.2 from two to many levels, is discussed. To describe the classical AMLI setting, let us consider a sequence of FE triangulations T_l, where $l = L_0, \ldots, L$ denotes the level. L_0 is the coarsest mesh, and the meshes are nested such that

$$T_{l+1} \supset T_l \quad l = L_0, \ldots, L_{N-1}.$$

That is, each element in the coarser mesh T_l is uniformly refined to form the elements of T_{l+1}.

On each level $B_M^{(l)}$ is recursively defined as

$$B_M^{(l)} = \begin{pmatrix} I & 0 \\ A_{21}^{(l)}(P^{(l)})^{-1} & I \end{pmatrix} \begin{pmatrix} P^{(l)} & A_{12}^{(l)} \\ 0 & Q^{(l)} \end{pmatrix}, \qquad B_M^{(l-1)} = Q^{(l)}, \tag{3.27}$$
$$l = L_0 + 1, \ldots, L_N.$$

The coarsest mesh can be fairly fine, it is sufficient that the coarse mesh matrix $Q^{(L_0+1)}$ is so small that it is not too demanding to solve accurately with it.

Equation (3.27) defines an algorithm of V-cycle type, and from Equation (3.16) it follows that when $P^{(l)} = A_{11}^{(l)}$ and $Q^{(l)} = A_{22}^{(l)}$, the condition number of $B_M^{(L)^{-1}} A^{(L)}$ grows with the number of levels as

$$\kappa\left(B_M^{(L)^{-1}} A^{(L)}\right) \leq \left(\frac{1}{\sqrt{1-\gamma^2}}\right)^{(L-L_0)}.$$

The latter estimate shows clearly an exponential growth with the number of levels $L-L_0$, and as a means to stabilize the condition number of $B_M^{(L)^{-1}} A^{(L)}$, the Algebraic Multilevel Iterations (AMLI) method was introduced in a series of papers, starting with [16] and [17].

There exist various approaches how to stabilize the condition number of the preconditioned matrix. For the sake of completeness, the classical method of polynomial stabilization as presented in the original AMLI papers [16] and [17] is included here. When $B_M^{(l)}$ is stabilized, $Q^{(l)}$ is replaced by

$$\widetilde{Q}^{(l)} = A^{(l-1)}\left(I - \mathscr{P}_\nu\left(B_M^{(l-1)^{-1}} A^{(l-1)}\right)\right)$$

or,

$$\widetilde{Q}^{(l)} = S^{-(l)}\left(I - \mathscr{P}_\nu\left(B_M^{(l-1)^{-1}} S^{(l)}\right)\right)$$

when the exact Schur complement on level l is is known. $\mathscr{P}_\nu(t)$ is a polynomial of degree ν which is chosen such that

$$0 \leq \mathscr{P}_\nu(t) < 1, 0 \leq t < 1, \text{ and } \mathscr{P}(0) = 1. \tag{3.28}$$

Two classes of polynomials that meet the requirements in Equation (3.28) are

$$\mathscr{P}_\nu(t) = (1-t)^\nu,$$

that is, a ν-fold application of the V-cycle algorithm, and the scaled and shifted Chebyshev polynomial

$$\mathscr{P}_\nu^{(l)}(t) = \frac{T_\nu\left(\frac{\beta_l+\alpha_l-2t}{\beta_l-\alpha_l}+1\right)}{T_\nu\left(\frac{\beta_l+\alpha_l}{\beta_l-\alpha_l}\right)},$$

where α_l and β_l are lower and upper bounds of the spectrum of the matrix $B_M^{(l-1)^{-1}} A^{(l-1)}$ ($B_M^{(l-1)^{-1}} S^{(l)}$).

In order to scale and shift the Chebyshev polynomial properly, some spectral information for $B_M^{(l-1)^{-1}} A^{(l-1)}$ is required. For simple test problems accurate estimates of the extreme eigenvalues of $B_M^{(l-1)^{-1}} A^{(l-1)}$ can be inexpensively computed by a few Lanczos iterations. See, for example [4] and the references therein.

One way to avoid eigenvalue computations is to replace the polynomial $\widetilde{Q}$ by a few iterations of an inner iterative solution method for $Q^{(l)}$. The inner solver is recursively preconditioned by $B_M^{(l-1)}$, and the resulting multilevel scheme is of W-cycle type. This idea is introduced in [18], and see also [53] for a similar approach.

It is not necessary to stabilize the multilevel preconditioner on each level. For example, in [65], [18], and [4] the stabilization is performed on certain properly chosen levels. It is shown that for spd matrices arising from a discretization of an elliptic PDE, the resulting preconditioner is robust and has optimal complexity when sufficient, but not too many, inner iterations (degree of $\mathscr{P}_\nu$) are performed.

Two other stabilization techniques for the multilevel methods could also be mentioned. One is introduced in [66] where the recursive splitting of the HBF space is stabilized by the introduction of (approximate) wavelets. Another is found in [50], where the stabilization is performed as pre- and post-smoothing of the vector to which $B_M^{(l-1)^{-1}} Q^{(l)}$ is applied.

Remark 3.7.1 *The condition number $\kappa\left(B_M^{(L)^{-1}} A^{(L)}\right)$ is estimated in terms of the CBS constant γ and the condition number of the (preconditioned) pivot block. Therefore, the AMLI method is considered to be a regularity-free multilevel preconditioning method. The latter is in contrast to the convergence estimates for the classical multigrid methods, see for example [29].*

3.8 The ARMS method

The algebraic recursive multilevel solver (ARMS) is a purely algebraic method, in which the graph of $A^{(l)}$ is divided into independent set. The matrix $A^{(l)}$ is symmetrically permuted into a two-by-two block form

$$P_{(l)}^T A^{(l)} P_{(l)} = \begin{pmatrix} B^{(l)} & F^{(l)} \\ E^{(l)} & C^{(l)} \end{pmatrix},$$

such that the unknowns corresponding to the independent sets are ordered first, and hence, $B^{(l)}$ is (block)diagonal.

The block matrix $P_{(l)}^T A^{(l)} P_{(l)}$ is approximately factorized as

$$P_{(l)}^T A^{(l)} P_{(l)} = \begin{pmatrix} L^{(l)} & 0 \\ G^{(l)} & I \end{pmatrix} \begin{pmatrix} U^{(l)} & W^{(l)} \\ 0 & A^{(l-1)} \end{pmatrix},$$

where $L^{(l)}$ and $U^{(l)}$ are the (incomplete) LU factors of $B^{(l)}$, $G^{(l)}$ approximates $E^{(l)} {U^{(l)}}^{-1}$, and $W^{(l)}$ approximates ${L^{(l)}}^{-1} F^{(l)}$. The "coarse mesh" matrix $A^{(l-1)}$ approximates $C^{(l)} - E^{(l)} {B^{(l)}}^{-1} F^{(l)}$. The block $A^{(l-1)}$ is constructed from $C^{(l)}$, $G^{(l)}$ and $W^{(l)}$, utilizing a dropping strategy such that the result is sparse. For further details on ARMS and related multilevel ILU methods, see for example [28], and [58], and the references therein. A parallel version of ARMS, pARMS, is described in [45].

In [57] a multilevel ILU preconditioner is described where the permutation of $A^{(l)}$ is nonsymmetric,

$$PAQ^T = \begin{pmatrix} B^{(l)} & F^{(l)} \\ E^{(l)} & C^{(l)} \end{pmatrix},$$

and constructed such that the diagonal dominance of $B^{(l)}$ is maximized. In this way, the stability of the preconditioner is increased for general nonsymmetric matrices with nonsymmetric sparsity structure.

4. Summary of Papers

In this chapter the six papers that comprise this thesis are summarized.

4.1 Paper I

Paper I deals with simple algebraic preconditioning for nonsymmetric dense linear systems, that arise from the discontinuous displacement method (DDM) discretization of crack propagation in brittle material, for example rock. In this paper, the finite element method is not used for the discretization, but we use the same type of block-factorized preconditioning technique as in the rest of the papers. Up to the knowledge of the author, 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 [34] for further details on DDM.

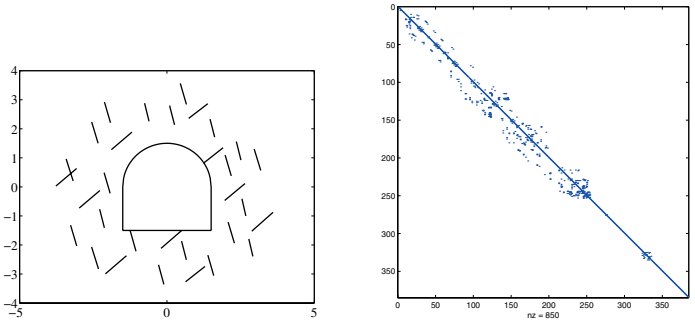
Due to crack singularities, the difference in magnitude between elements of the arising matrix can be enormous, which under a proper ordering of the unknowns leads to a strongly diagonally dominant matrix. However, for fracture networks more complicated than one single crack, it may also contain significant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

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^{-1}. One simple idea is to require that for all indices $i, j \in \mathfrak{S}$ there holds $(G^{-1}A)_{ij} = \delta_{ij}$, 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 [33]. Within the DBAI framework,



(a) A gallery in fractured rock. (b) $A > 0.01$ after a permutation

Figure 4.1: The geometry of Problem 4.1.2 and the structure of A.

G^{-1} is constructed as a matrix with k diagonals which approximates the inverse of the corresponding band part 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 [33] 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 (3.2) or Equation (3.3). We have utilized such a two-by-two block-structure to construct a preconditioner of the form (3.2), 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.

Numerical experiments

In [21], 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 4.1.1 (Borehole with four cracks) A circular borehole in homogeneous infinite media is subjected to uniaxial stress. Four radial cracks are situated in the wall of the hole.

Problem 4.1.2 (Gallery) A model of a gallery in fractured rock at a depth of 500 m.

The geometry of Problem 4.1.2 is shown in Figure 4.1, where also the most significant entries of A are shown (A is scaled to unit diagonal). As is depicted in Figure 4.1, 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 [59].

The system is solved using 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 4.1.1 and Problem 4.1.2, the block-factorized preconditioner is the most numerically efficient (the iterative method converges in few iterations) of the three tested. Furthermore, especially for Problem 4.1.2, BFP is the most robust method with respect to the control of the amount of fill-in in the preconditioner.

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

4.2 Paper II

This paper deals with numerical simulations of a purely elastic model of Earth's 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 some first order terms arising from the so-called advection of pre-stress.

The arising system of linear equations is nonsymmetric and of 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.

Target problem

The target problem of this paper is the moment balance equation for a pre-stressed elastic solid body in equilibrium (cf. [42]), which, with the self-gravitating term omitted, reads

$$\nabla \cdot \sigma + \nabla(\mathbf{u} \cdot \nabla p_0) - (\nabla \cdot \mathbf{u}) \nabla p_0 = \mathbf{0}. \tag{4.1}$$

In Equation (4.1), the *pre-stress* p_0 is assumed to be hydrostatic,

$$p_0 = -\rho g \mathbf{e}_d \cdot \mathbf{x},$$

where ρ is the density of the solid, g is the gravitational acceleration, and $\mathbf{e}_d$ is the unit vector directed downwards.

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

$$\sigma = 2\mu\varepsilon(\mathbf{u}) + \lambda\nabla\cdot\mathbf{u}\mathbf{I}, \tag{4.2}$$

where $\varepsilon(\mathbf{u}) = 0.5(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the strain tensor,

$$\mu = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda = \mu\frac{2\nu}{1-2\nu}$$

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 (4.2) is not well posed in the incompressible limit. Therefore, special care is required when discretizing and solving Equation (4.1) for $\nu \to 0.5$.

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

$$p = \frac{\lambda}{\mu}\nabla\cdot\mathbf{u} \tag{4.3}$$

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

$$\begin{cases} -2\nabla\cdot(\mu\nabla\mathbf{u}) - \nabla\times(\mu\nabla\times\mathbf{u}) \\ \qquad -\rho g\,(\nabla(\mathbf{u}\cdot\mathbf{e}_d) - \mathbf{e}_d\nabla\cdot\mathbf{u}) - \mu\nabla p = \mathbf{0} \\ \qquad\qquad \mu\nabla\cdot\mathbf{u} - \dfrac{\mu^2}{\lambda}p = 0 \end{cases} \tag{4.4}$$

with boundary conditions

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) &= \mathbf{0} \qquad \mathbf{x}\in\Gamma_D, \\ \sigma\cdot n &= \ell \qquad \mathbf{x}\in\Gamma_L, \\ \sigma\cdot n &= \mathbf{0} \qquad \mathbf{x}\in\Gamma_N. \end{aligned}$$

On the boundary segment Γ_D ($\text{meas}(\Gamma_D) > 0$) 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 (4.4), we consider a slightly more general form of the advection term, namely

$$-\nabla(\mathbf{u}\cdot\mathbf{b}) + \mathbf{c}\nabla\cdot\mathbf{u}, \tag{4.5}$$

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

$$\nabla(\mathbf{f}\cdot\mathbf{g}) = \underbrace{(\mathbf{f}\cdot\nabla)\mathbf{g}}_{(a)} + \underbrace{(\mathbf{g}\cdot\nabla)\mathbf{f}}_{(b)} + \underbrace{\mathbf{f}\times(\nabla\times\mathbf{g})}_{(c)} + \underbrace{\mathbf{g}\times(\nabla\times\mathbf{f})}_{(d)}, \tag{4.6}$$

and from Equation (4.6) 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 $\mathbf{f}$ is a constant vector, the terms (a) and (c) in (4.6) vanish.

The target problem now reads

$$\begin{cases} -2\nabla\cdot(\mu\nabla\mathbf{u}) - \nabla\times(\mu\nabla\times\mathbf{u}) - \nabla(\mathbf{u}\cdot\mathbf{b}) + \mathbf{c}\nabla\cdot\mathbf{u} - \mu\underline{\nabla}p = \mathbf{0} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mu\nabla\cdot\mathbf{u} - \dfrac{\mu^2}{\lambda}p = \mathbf{0}. \end{cases} \tag{4.7}$$

Variational formulation

The variational formulation corresponding to Equation (4.7) is defined in terms of the Sobolev spaces $\mathbf{V} = \left(H_0^1(\Omega)\right)^d$, $d = 2,3$, and $P = \left\{p \in L^2(\Omega) : \int_\Omega \mu\, p\, d\Omega = 0\right\}$. It leads to the following mixed variable problem:

$$\begin{aligned} &\text{Find } \mathbf{u}\in\mathbf{V} \text{ and } p\in P \text{ such that} \\ &\begin{cases} a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) = & \langle \ell,\mathbf{v}\rangle, \qquad \forall \mathbf{v}\in\mathbf{V}, \\ b(\mathbf{u},q) - c(p,q) = & 0, \qquad \forall q\in P, \end{cases} \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) &= \int_\Omega \Big[2\mu\sum_{k=1}^{d}(\nabla u_k)\cdot(\nabla v_k) - \mu(\nabla\times\mathbf{u})\cdot(\nabla\times\mathbf{v}) \\ &\qquad - \nabla(\mathbf{u}\cdot\mathbf{b})\cdot\mathbf{v} + (\nabla\cdot\mathbf{u})(\mathbf{c}\cdot\mathbf{v})\Big]\, d\Omega, \\ b(\mathbf{u},p) &= \int_\Omega \mu(\nabla\cdot\mathbf{u})\, p\, d\Omega = -\int_\Omega \mu\nabla(p)\cdot\mathbf{u}\, d\Omega, \\ c(p,q) &= \int_\Omega \frac{\mu^2}{\lambda} p\, q\, d\Omega, \quad \text{and} \quad \langle \ell,\mathbf{v}\rangle = \int_{\Gamma_L} \mathbf{v}\cdot\ell\, d\Gamma. \end{aligned} \tag{4.9}$$

A solution to the variational problem (4.8) exists and is unique if $a(\mathbf{u},\mathbf{v})$, $c(p,q)$ and $b(\mathbf{u},p)$ are bounded,

$$a(\mathbf{u},\mathbf{v}) \le \overline{a}\|\mathbf{u}\|_{\mathbf{V}}\|\mathbf{v}\|_{\mathbf{V}} \quad \forall \mathbf{u},\mathbf{v}\in\mathbf{V} \tag{4.10}$$

$$b(\mathbf{v},p) \le \overline{b}\|\mathbf{v}\|_{\mathbf{V}}\|p\|_P \quad \forall \mathbf{u}\in\mathbf{V},\ p\in P \tag{4.11}$$

$$c(p,q) \le \overline{c}\|p\|_P\|q\|_P \quad \forall p,q\in P, \tag{4.12}$$

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

$$a(\mathbf{u},\mathbf{u}) \ge \underline{a}\|\mathbf{u}\|_{\mathbf{V}}^2, \quad \underline{a}>0 \quad \forall \mathbf{u}\in\mathbf{V} \tag{4.13}$$

$$c(p,p) \ge \underline{c}\|p\|_P^2, \quad \underline{c}>0 \quad \forall p\in P. \tag{4.14}$$

As is clear from Equation (4.9) $c(p,q) = 0$, $\forall p,q\in P$ corresponds to $\nu = 0.5$. In this case, Equation (4.8) is solvable if

- the conditions in (4.10) – (4.12) hold,
- $a(\mathbf{u},\mathbf{u})$ is coercive on the null-space of $b(\mathbf{u},q)$, and
- it holds that, for all $\mathbf{u}\in\mathbf{V}$, when $b(\mathbf{u},q)$ vanishes then q is zero.

Furthermore, Equation (4.8) is stable if the following inf-sup (or Ladyzhenskaya-Babuška-Brezzi or LBB) conditions are fulfilled,

$$\inf_{\mathbf{u}\in\mathbf{V}}\sup_{\mathbf{v}\in\mathbf{V}}\frac{a(\mathbf{u},\mathbf{v})}{\|\mathbf{u}\|_{\mathbf{V}}\|\mathbf{v}\|_{\mathbf{V}}}\geq\underline{a}'>0, \tag{4.15}$$

and

$$\inf_{q\in P}\sup_{\mathbf{v}\in\mathbf{V}}\frac{b(\mathbf{u},q)}{\|\mathbf{v}\|_{\mathbf{V}}\|q\|_{P}}\geq\underline{b}>0. \tag{4.16}$$

Note that when $a(\mathbf{u},\mathbf{v})$ is coercive, Equation (4.15) is automatically satisfied. See, for example, [32] for details.

In Paper II, we show that the bilinear forms in Equation (4.8) are bounded, but that $a(\mathbf{u},\mathbf{v})$, in general is not coercive due to the first order terms. The coercivity of $c(p,p)$ is straightforwardly seen.

Finite element discretization

To discretize Equation (4.8), 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 (4.8) then reads:

$$\begin{aligned}&\text{Find } \mathbf{u}_h\in\mathbf{V}_h \text{ and } p_h\in P_h \text{ such that}\\ &\begin{cases} a(\mathbf{u}_h,\mathbf{v}_h)+b(\mathbf{v}_h,p_h)= & \langle\ell,\mathbf{v}_h\rangle \qquad \forall\mathbf{v}_h\in\mathbf{V}^h,\\ b(\mathbf{u}_h,q_h)-c(p_h,q_h)= & 0, \qquad \forall q_h\in P^h. \end{cases}\end{aligned} \tag{4.17}$$

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 (4.17) needs to be stabilized.

A stable pair of finite element spaces for Equation (4.17) 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 (4.16),

$$\sup_{\mathbf{u}_h\in\mathbf{V}^h}\frac{b(\mathbf{u}_h,p_h)}{\|\mathbf{u}_h\|_{\mathbf{V}^h}}\geq\gamma_0\|p_h\|_{P^h}, \qquad \forall p_h\in P^h, \tag{4.18}$$

is satisfied. See, for example, [29] for details.

One way to circumvent the discrete LBB-condition on the finite element spaces is to use an unstable pair of elements and stabilize Equation (4.17). This gives the freedom to choose $\mathbf{V}^h$ and P^h in a way that is preferable from a computational complexity point of view, compared to when stable finite element pairs are used.

A stabilized and consistent equal order FE discretization of Equation (4.17) can be achieved by adding the equation

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

to the second equation of (4.8), where σ_h is some suitably determined stabilization parameter. A derivation of the optimal choice of σ_h is found in [19], and it is shown that $\sigma_h = \mathcal{O}(h^2)$ gives an optimal stability estimate of the form

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + h\|p - p_h\|_0 \leq h^2 C \|\ell\|_0. \tag{4.19}$$

The finite element discretization of the stabilized version of Equation (4.17) leads to a linear algebraic system

$$\mathscr{A} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} \equiv \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} = \begin{bmatrix} \mathbf{r}_h \\ \mathbf{s}_h \end{bmatrix}. \tag{4.20}$$

The system matrix $\mathscr{A}$ 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 (4.20) is further solved by preconditioned iterative solution methods.

Numerical experiments

We apply the preconditioners $\mathscr{D}_t$ and $\mathscr{D}_f$ from Equation (3.19), page 29, on the following realistic benchmark problem.

Problem 4.2.1 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 $\mathscr{D}_t$ or $\mathscr{D}_f$. 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 3.5.2. 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 [55].

Iteration counts

The numerical results in Paper II reveal that the performance of $\mathscr{D}_f$ and $\mathscr{D}_t$ 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 exactly.

Further, the results show that $\mathscr{D}_f$ is a more robust preconditioner than $\mathscr{D}_t$, 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 Poisson number. The growth is, however, acceptable.

CPU time comparisons

In Paper II we also perform CPU-time comparisons using our code and a commercial FEM package for Problem 4.2.1 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.

4.3 Paper III

Paper III is a continuation and extension of Paper II 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 $\mathscr{D}_t$ in Equation (3.19), page 29. 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

$$B_M^{(l)} = \begin{bmatrix} I & 0 \\ A_{21}^{(l)} P^{(l)-1} & I \end{bmatrix} \begin{bmatrix} P^{(l)} & A_{12}^{(l)} \\ 0 & Q^{(l)} \end{bmatrix} \tag{4.21}$$

$$B_M^{(l-1)} = Q^{(l)}, \quad l = l_0, \ldots, L-1, L.$$

On each level l the matrices $Q^{(l)}$ are obtained from assembly of local, exactly computed, macroelement Schur complement matrices as is described in Section 3.5.1. 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 (cf. Chapter 3). However, from the numerical experiments in Paper III 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 (4.20), 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

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

and

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

In Problem 4.2.1 $\mathbf{b} = \mathbf{c} = \rho g \mathbf{e}_d$, where $\mathbf{e}_d$ is the unit vector directed downwards. In Appendix A it is shown how Equation (4.1) can be scaled to admit a dimensionless form. From the scaled equation it can be seen that for physically reasonable values of ρ, g, μ, and λ, the problem is not dominated by the advective part $\bar{a}(\mathbf{u},\mathbf{v})$. This motivates the choice of the preconditioner $\widehat{\mathscr{D}}$, where the AMLI preconditioner in Equation (4.21) 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

$$\widehat{D}^E = \begin{bmatrix} \widehat{D}^E_{ff} & \widehat{D}^E_{fc} \\ \widehat{D}^E_{cf} & \widehat{D}^E_{cc} \end{bmatrix}, \tag{4.22}$$

and the four blocks in Equation (4.22) are used to assemble the matrices $P^{(l)}$, $A^{(l)}_{12}$, $A^{(l)}_{21}$, 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 on the Lamé coefficients, the inequality

$$K(\Omega)\widetilde{a}(\mathbf{u},\mathbf{u}) \le \widehat{a}(\mathbf{u},\mathbf{u}) \le 2\widetilde{a}(\mathbf{u},\mathbf{u}) \tag{4.23}$$

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, [3].

The relations in (4.23) motivates the choice of the preconditioner $\widetilde{\mathscr{D}}$, 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

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

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

In the sequel of this chapter, the combination of $\mathscr{D}_t$ as a preconditioner for the outer iterative solver and inner iterative solvers preconditioned by AMLI for the diagonal blocks of D_1 and D_2, is denoted BT-AMLI.

Numerical results

The matrix $\mathscr{A}$ is solved using the generalized conjugate gradient-minimized residual (GCG-MR) method, preconditioned with $\mathscr{D}_t$ in Equation (3.19), and solved until a relative stopping criterion 10^{-6} is satisfied.

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 $\mathscr{D}$, D_1 and D_2 are solved with GCG-MR, preconditioned with the block-factorized multilevel preconditioner in Equation 4.21, to some relative stopping criteria τ and 10^{-6}, correspondingly. The block $P^{(l)}$ in Equation (4.21) is approximated by an incomplete LU-factorization (ILUT), see [55].

The numerical tests in Paper III illustrate the performance of the proposed preconditioner $\mathscr{D}_t$, 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 $\mathscr{D}_t$ is of optimal order when the inner iterative solution method for M is preconditioned by $\tilde{D}_1$. The number of inner and outer iterations are constant and the overall computation time scales nearly linearly with the size of the problem.

However, $\mathscr{D}_t$ 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 Paper II.

The results also show that there is a trade-off between the overall computation time to solve with $\mathscr{A}$, 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 4.2.1 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.

4.4 Paper IV

Paper IV is a further extension of the work in Papers II and III. In this paper, we address the same problem of isostatic glacial rebound as in the previous two papers, and we use the same preconditioning technique as in Pa-

per III, BT-AMLI. The novelties in this papers are that (i) we address the fully viscoelastic problem and propose a solution algorithm for the mixed variables problem, (ii) we extend the problem to three space dimensions, (iii) we investigate how inhomogeneities in the material coefficients affect the performance of the iterative solution method, and (iv) the stabilized finite element discretization is replaced by a stable modified Taylor–Hood discretization.

A viscoelastic model

The material is assumed to be viscoelastic, that is, it is obeying a constitutive relation of the form

$$\sigma(\mathbf{x}, t) = \sigma_E(\mathbf{x}, t) - \int_0^t \sigma_V(\mathbf{x}, t, \tau)\, d\tau, \tag{4.24}$$

(see for instance [48]), which is referred to as Hooke's law with memory. The tensors σ_E and σ_V describe the elastic and viscoelastic response, respectively, describing rocks with long memory where the state of stress at time t depends on the deformation at time t but also on the deformations at times prior to t.

We utilize the following standard relations between stress, strain and displacements,

$$\begin{aligned} \sigma_E(t) &= \lambda_E \nabla \cdot \mathbf{u}(t) + 2\mu_E \varepsilon(\mathbf{u}(t)) \\ \sigma_V(t,\tau) &= \partial_\tau \lambda(t,\tau) \nabla \cdot \mathbf{u}(\tau) + 2\partial_\tau \mu(t,\tau) \varepsilon(\mathbf{u}(\tau)) \end{aligned} \tag{4.25}$$

where ∂_τ denotes differentiation with respect to τ, as well as the assumption that the stress relaxation functions obey the so-called Maxwell model. That is, the time-dependence in the stress field is due to time-dependent material coefficients only and is of the form

$$\begin{aligned} \mu(t,\tau) &= \mu_E e^{-\alpha_0(t-\tau)} = \mu_E \chi(\alpha_0, t, \tau) \\ \lambda(t,\tau) &= \lambda_E e^{-\alpha_0(t-\tau)} = \lambda_E \chi(\alpha_0, t, \tau). \end{aligned} \tag{4.26}$$

Above, the coefficients μ_E, λ_E and $\mu(t,\tau)$, $\lambda(t,\tau)$ are Lamé coefficients in the elastic case and the viscoelastic case correspondingly, η is the viscosity parameter, and $\alpha_0 = \mu_E/\eta$ is the inverse of the so-called Maxwell time.

Combining Equation (4.24), (4.25), and (4.26) with the moment balance equation

$$-\nabla \cdot \sigma(\mathbf{x}, t) - \nabla(\mathbf{b} \cdot \mathbf{u}(t)) + \mathbf{c} \nabla \cdot \mathbf{u}(t) = \mathbf{f}(t),$$

introducing the kinematic pressure

$$\mu_E p = \lambda_E \nabla \cdot \mathbf{u},$$

we obtain, after some manipulations, the following variational problem:

For each $t \in \mathscr{J} = [0,T]$, find $\mathbf{u}(\mathbf{x},t) \in \mathscr{V}(V;\mathscr{J})$ and $p(\mathbf{x},t) \in \mathscr{P}(P;\mathscr{J})$ such that

$$\begin{aligned} a(\mathbf{u}(t),\mathbf{v}) - \int_0^t \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t)\,d\tau \quad &+ b(\mathbf{v},p(t)) - \int_0^t \widetilde{b}(\mathbf{v},p(\tau);t)\,d\tau \quad &= \widetilde{g}(\mathbf{v};t) \\ b(\mathbf{u}(t),q) - \int_0^t \widetilde{b}(\mathbf{u}(\tau),q;t)d\tau \quad &- c(p(t),q) + \int_0^t \widetilde{c}(p(t),q;t)\,d\tau \quad &= 0 \end{aligned} \tag{4.27}$$

holds for any $\mathbf{v}(\mathbf{x}) \in V$ and $q(\mathbf{x}) \in P$. In Equation (4.27),

$$\widetilde{g}(\mathbf{v};t) = g(\mathbf{v};t) + \ell(\mathbf{v};t) - \int_0^t \widetilde{\ell}(\mathbf{v};t,\tau)\,d\tau.$$

Further, the bilinear forms $a(\mathbf{u},\mathbf{v})$, $b(\mathbf{u},p)$, and $c(p,q)$ are defined as in Equation (4.9), whereas

$$\begin{aligned} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t,\tau) &= \int_\Omega 2\alpha_0\chi(\alpha_0,t,\tau)\mu_E\,\varepsilon(\mathbf{u}(\tau)) : \varepsilon(\mathbf{v})\,d\Omega, \\ \widetilde{b}(\mathbf{v},p(\tau);t,\tau) &= \int_\Omega \alpha_0\chi(\alpha_0,t,\tau)\mu_E p(\tau)\nabla\cdot\mathbf{v}\,d\Omega, \\ \widetilde{c}(p(t),q;t,\tau) &= \int_\Omega \alpha_0\chi(\alpha_0,t,\tau)\frac{\mu_E^2}{\lambda_E}p(t)q\,d\Omega, \\ \widetilde{\ell}(\mathbf{v};t,\tau) &= \int_\Gamma \alpha_{0,m}\chi(\alpha_0,t,\tau)\ell(\tau)\cdot\mathbf{v}\,d\Gamma, \\ \ell(\mathbf{v};t) &= \int_\Gamma \ell(t)\cdot\mathbf{v}\,d\Gamma, \text{ and } g(\mathbf{v};t) = \int_\Omega \mathbf{f}(t)\cdot\mathbf{v}\,d\Omega. \end{aligned} \tag{4.28}$$

Discretization in time and space

To handle the viscoelastic terms, i.e. the time integrals in Equation (4.27), we apply some numerical quadrature. This implies that $\mathscr{J}$ is discretized into a number of intervals, such that $t_0 = 0$, $t_1 = t_0 + \Delta t_0$, …, $t_{j+1} = t_j + \Delta t_j$, …, $t_n = T$ and the jth interval is then of length $\Delta t_j = t_j - t_{j-1}$. In general, for any numerical integration, when computing $\mathbf{u}_j$ one would need to store the complete solution history, i.e., all solutions $\mathbf{u}_s$, $s = 0,\cdots,j-1$, which is a very memory demanding task. Fortunately, for relaxation functions as in the Maxwell model, given by Equation (4.26), the computations simplify significantly. For completeness of the presentation, the essence of the simplification is here illustrated on one of the integral terms in Equation (4.27).

Consider the time integral

$$\begin{aligned} \mathscr{I}_0^j &= \int_0^{t_j} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t_j,\tau)\,d\tau = \int_0^{t_{j-1}} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t_j,\tau)\,d\tau + \int_{t_{j-1}}^{t_j} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t_j,\tau)\,d\tau \\ &= \widetilde{\mathscr{I}}_0^{j-1} + \mathscr{I}_{j-1}^j. \end{aligned} \tag{4.29}$$

Then, provided that the material coefficients are not space-dependent, using (4.28) and the particular form of the relaxation functions, from Equation (4.26) one obtains

$$\begin{aligned}\widetilde{\mathscr{F}}_0^{j-1} &= \int_0^{t_{j-1}} \int_\Omega 2\alpha_0 \chi(\alpha_0, t_j, \tau) \mu_E \varepsilon(\mathbf{u}(\tau)) : \varepsilon(\mathbf{v})\, d\Omega\, d\tau \\ &= \int_0^{t_{j-1}} \int_\Omega e^{-\alpha(t_j - t_{j-1})} \left(2\alpha_0 \chi(\alpha_0, t_{j-1}, \tau) \mu_E \varepsilon(\mathbf{u}(\tau)) : \varepsilon(\mathbf{v})\right) d\Omega\, d\tau \\ &= e^{-\alpha \Delta t_j} \mathscr{I}_0^{j-1}.\end{aligned} \tag{4.30}$$

Hence, $\widetilde{\mathscr{I}}_0^{j-1}$ can be computed inexpensively by scaling of $\mathscr{I}_0^{j-1}$ which is known from the previous instance in time.

Next, we approximate $\mathscr{I}_{j-1}^{j}$ by the trapezoidal rule, that is

$$\int_{t_{j-1}}^{t_j} f(t) \approx \frac{\Delta t_j}{2} \left(f(t_j) + f(t_{j-1})\right). \tag{4.31}$$

When the operations in Equation (4.29) through (4.31) are applied to the four time integral terms in Equation (4.27), the terms in the variational problem can be reshuffled such that the terms that depend on the unknown solution u_j are collected on the left-hand side of the equation, whereas the terms that depend on the memory terms (i.e. the loading history) are grouped on the right-hand-side. This is illustrated in Paper IV, Equation (4.4).

After a finite element discretization of Equation (4.27) we obtain the following problem:

At time t_j, find the displacements $\mathbf{u}_j$ and the pressure $\mathbf{p}_j$ by solving

$$\left(\mathscr{A} - \frac{\Delta t_j}{2} \mathscr{A}_0\right) \begin{pmatrix} \mathbf{u}_j \\ \mathbf{p}_j \end{pmatrix} = \begin{pmatrix} \mathbf{r}_j^{(1)} \\ \mathbf{r}_j^{(2)} \end{pmatrix}, \tag{4.32}$$

where

$$\mathscr{A} = \begin{pmatrix} M & B^T \\ B & -C \end{pmatrix} \text{ and } \quad \mathscr{A}_0 = \begin{pmatrix} M_0 & B_0^T \\ B_0 & -C_0 \end{pmatrix}.$$

The matrix blocks M_0, B_0, C_0 correspond to the bilinear forms $\widetilde{a}(\cdot,\cdot)$, $\widetilde{b}(\cdot,\cdot)$, and $\widetilde{c}(\cdot,\cdot)$ in (4.28), evaluated at $\tau = t$ and do therefore not explicitly depend on t nor Δt. This means that the preconditioner for $\mathscr{A} - \frac{\Delta t_j}{2}\mathscr{A}_0$ must not be completely reconstructed in every time step, since it suffices to scale the blocks M_0, B_0, C_0 by Δt_j.

The time-stepping algorithm for the solution of the viscoelastic problem described in Paper IV consists of a pre-solve step in which the right-hand side $\mathbf{r}^{(j)}$ is computed, by the solution step where the linear system in Equation (4.32), $\mathscr{A}_j\mathbf{u} = \mathbf{b}$, is solved, and a post-solve step where the memory terms are updated. The computationally most expensive part of the scheme is the solution step, and it is of utmost importance for the overall efficiency of the solution scheme that this solver is efficient and robust.

Numerical Experiments

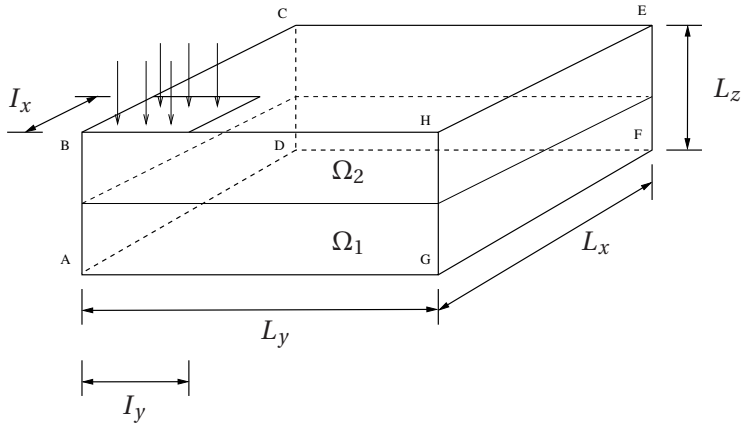


Figure 4.2: The geometry of the problem in 3D.

Figure 4.2 shows the geometry of the problem in 3D. The dimensions of the domain Ω are: $L_x = L_y = 10000$ km, $L_z = 4000$ km, and $I_x = I_y = 1000$ km. The problem is symmetric with respect to the sides $ABCD$ and $ABHG$, homogeneous Dirichlet conditions are imposed on the side $ADFG$, and homogeneous Neumann conditions are imposed on the remaining three sides. The domain is loaded by a homogeneous load from a rectangular sheet of ice of 2 km height with density $981 kg m^{-3}$, denoted by the arrows at the corner B. The subdomains Ω_1 and Ω_2 have a thickness of 2000 km. To obtain the geometry in 2D, L_x and I_x are reduced to zero.

The domain $\Omega = \Omega_1 \cup \Omega_2$ is discretized using uniform rectangular (bricks in 3D) finite elements, on which we use a pair of stable, modified Taylor–Hood ($Q1$–iso $Q1$) bilinear basis functions, namely that the displacements $\mathbf{u}$ live on a mesh that is a uniform refinement of the mesh on which the pressure variables p live. This is in contrast to Papers II and III, where stabilized $Q1-Q1$ bilinear basis functions are used. The benefits of the modified Taylor–Hood element are two-fold. Firstly, the FE discretization is stable by itself which makes the stabilization term in described in Paper II unnecessary. Secondly, the size of the system matrix $\mathscr{A}$ is reduced because the size of the pressure space is reduced by roughly a factor four in 2D and a factor eight in 3D.

Iteration counts

The numerical experiments in Paper IV reveal that the iterative GCG-MR solver, preconditioned by the BT-AMLI method, is robust with respect to the problem size and the number of space dimensions.

The difference in Young modulus (E) and (in)compressibility (ν) in the two subdomains causes the condition number of $\mathscr{A}$ to deteriorate somewhat, which is "felt" by the iterative solver as an increase of the number of iterations. This increase is however acceptable, and it is by no means proportional to the size of the jump in the material parameters.

CPU timings

The CPU-timings presented in Paper IV show that the time spent by the preconditioned iterative solution method is proportional to the increase in iteration count with increasing jumps in the material coefficients. Furthermore, the timings in Paper IV reveal that the proposed preconditioned iterative solution method is a feasible alternative to a direct solver provided in the commercial FE package ABAQUS [1]. More detailed comparisons between the solution approach in Paper III and IV, and ABAQUS are provided in Paper V.

4.5 Paper V

In this paper we address the problem of isostatic glacial rebound with the objective to compare the accuracy and efficiency of two approaches to formulate and discretize the governing PDEs, and to solve the arising linear systems of equations. The two approaches (or frameworks) are the following.

(i) One that is based on the features and restrictions of the commercial finite element package ABAQUS, and

(ii) the mixed variables formulation as described in Papers II – IV.

In the first framework the linear system is solved by a direct method provided by ABAQUS, whereas in the second, the iterative GCG-MR method, preconditioned by BT-AMLI, is used.

Problem description

In order to clearly explain the differences in (i) and (ii), we state the moment balance equation for a pre-stressed (visco)elastic body in equilibrium

$$-\nabla\cdot\sigma-\nabla(\mathbf{u}\cdot\nabla p_0)+(\nabla\cdot\mathbf{u})\nabla p_0=\mathbf{f}\text{ in }\Omega\subset\mathbb{R}^d,\ d=2,3, \tag{4.33a}$$

with boundary conditions

$$\sigma(\mathbf{u}) \cdot \mathbf{n} = \ell \text{ on } \Gamma_L \tag{4.33b}$$

$$\sigma(\mathbf{u}) \cdot \mathbf{n} = \mathbf{0} \text{ on } \Gamma_N \tag{4.33c}$$

$$\mathbf{u} = \mathbf{0} \text{ on } \Gamma_D. \tag{4.33d}$$

The hydrostatic pre-stress is given by

$$p_0 = \rho_r g(y_t - y), \tag{4.34}$$

where g is the gravitational acceleration, ρ_r is the density of the material and y_t is a reference surface (for instance the Earth's surface). From Equation (4.34) it follows that

$$\nabla p_0 = -\rho_r g \hat{\mathbf{e}}_v,$$

where $\hat{\mathbf{e}}_v$ is the unit vector in positive vertical direction. That is, positive y-direction in 2D, and positive z-direction in 3D. For further details on the model and the origin of Equation (4.33a), see for example [67]. Equation (4.33a) differs from the standard moment balance equation for elastic solids

$$-\nabla \cdot \sigma(\mathbf{u}) = \mathbf{f}, \tag{4.35}$$

due to the presence of the first order terms $\nabla(\mathbf{u} \cdot \nabla p_0)$ and $(\nabla \cdot \mathbf{u})\nabla p_0$. In order compare the solutions to Equation (4.33a) and Equation (4.35), we introduce the following two models, $C0$ and $C1$, where one or both of the first order terms are omitted. That is,

$$C0: \quad (\nabla \cdot \mathbf{u})\nabla p_0 - \nabla(\mathbf{u} \cdot \nabla p_0) = \mathbf{0}. \tag{4.36a}$$

$$C1: \quad (\nabla \cdot \mathbf{u})\nabla p_0 = \mathbf{0}. \tag{4.36b}$$

Framework I: ABAQUS

ABAQUS is a commercial finite element package which is designed to solve problems of the type in Equation (4.35), rather than the type in Equation (4.33a). A remedy to this problem is to introduce the modified stress tensor (cf.[67])

$$T(\mathbf{u}) = \sigma(\mathbf{u}) + \mathbf{u} \cdot \nabla p_0 I. \tag{4.37}$$

When the resulting boundary value problem (BVP) is solved in ABAQUS, the tensor $T(\mathbf{u})$ is represented internally by a stress tensor $\sigma(\tilde{\mathbf{u}})$, which is given by Hooke's law,

$$\sigma(\tilde{\mathbf{u}}) = 2\mu\varepsilon(\tilde{\mathbf{u}}) + \lambda(\nabla \cdot \tilde{\mathbf{u}})I. \tag{4.38}$$

This is a tensor where the pre-stress advection term in $T(\mathbf{u})$ is not present, and in order to compensate for this the modified displacements $\tilde{\mathbf{u}}$ are introduced in Equation (4.38). These displacements are related to $\mathbf{u}$, and they are implicitly defined by the tensor equality

$$\sigma(\tilde{\mathbf{u}}) = T(\mathbf{u}). \tag{4.39}$$

Combining Equation (4.33), (4.37), (4.38), and (4.39), the following BVP is achieved

$$
\begin{aligned}
-\nabla\cdot\sigma(\widetilde{\mathbf{u}}) &= \mathbf{f} && \text{in } \Omega, && (4.40a)\\
\sigma(\widetilde{\mathbf{u}}) &= T(\mathbf{u}) && \text{in } \Omega && (4.40b)\\
\sigma(\widetilde{\mathbf{u}})\cdot\mathbf{n} &= \ell + \mathbf{u}\cdot\nabla p_0\mathbf{n} && \text{on } \Gamma_L && (4.40c)\\
\sigma(\widetilde{\mathbf{u}})\cdot\mathbf{n} &= \mathbf{u}\cdot\nabla p_0\mathbf{n} && \text{on } \Gamma_N && (4.40d)\\
\mathbf{u} &= \mathbf{0} && \text{on } \Gamma_D. && (4.40e)
\end{aligned}
$$

The displacements variable $\widetilde{\mathbf{u}}$ can be eliminated by the introduction of the transformation ϕ, defined as

$$\widetilde{\mathbf{u}} = \phi(\mathbf{u}).$$

The transformation ϕ is a formal construction and it is difficult (not to say impossible) to construct it in the general case. It is therefore approximated by the identity function, such that $\widetilde{\mathbf{u}} = \mathbf{u}$.

Framework II: Mixed $\mathbf{u}-p$ formulation

This is the formulation of the problem that is defined and used in Papers II - IV. The first order terms due to pre-stress advection and buoyancy are present in the PDE, that is, there is no need to introduce the modified stress tensor. When it is required by the model describing the problem ($C0$ or $C1$), one or both of the first order terms are omitted.

Numerical experiments

The following two problems are solved within both framework I and II:

Problem 4.5.1 (Uniform Load). A 2D (3D) flat Earth model is subjected to a uniform load from a 2.14 km thick ice sheet. The size of the domain is 10 000 km width (side) and 4000 km depth. The boundary conditions are homogeneous Dirichlet conditions on the boundary $y = -4000$ km ($z = -4000$ km) and symmetry conditions on the vertical boundaries $x = 0$ and $x = 10000$ (and $y = 0$ and $y = 10000$).

Problem 4.5.1 has a known analytical solution, in the sequel of this section denoted by $\mathbf{u}$.

Problem 4.5.2 (Footing Problem). A 2D (3D) flat Earth model, which is symmetric with respect to $x = 0$ (and $y = 0$), is subjected to a Heaviside load of a 1000 km wide (side) and 2.14 km thick ice sheet. The size of the domain is 10 000 km width and 4000 km depth. The boundary conditions are as follows: homogeneous Dirichlet conditions on the boundary $y = -4000$ km ($z = -4000$ km) and symmetry conditions on the boundary $x = 0$ (and $y = 0$), homogeneous Neumann conditions on the boundary $x = 10000$ km

(and $y = 10000$ km) and the boundary segment $y = 0$ ($z = 0$), $x > 1000$ km (and $y > 1000$ km).

Problem 4.5.2 is the benchmark problem described in Paper IV with homogeneous material parameters.

Throughout this section, if not stated otherwise, $\mathbf{u}_I$ and $\mathbf{u}_{II}$ denote the numerical solutions obtained from Framework I and II, respectively.

Analytical results for Problem 4.5.1

The $C1$ formulation of Problem 4.5.1 has the analytical solution

$$\begin{cases} u_1 = 0 \\ u_2 = \dfrac{\rho_i h_i}{\rho_r}\left(e^{\frac{-\rho_r g(y_t-y_b)}{2\mu+\lambda}} - e^{\frac{-\rho_r g(y_t-y)}{2\mu+\lambda}}\right), \end{cases} \tag{4.41}$$

when the material is loaded by the surface load $\ell = [0, -\rho_i g h_i]$, the hydrostatic pressure from an homogeneous, rectangular sheet of ice. The density of the ice is denoted by ρ_i, the height of the ice sheet by h_i, and y_b is the y-coordinate for the bottom of the domain. The first two terms in the Taylor expansion of u_2 in Equation (4.41) read,

$$u_2 \approx -\frac{\rho_i g h_i (y - y_b)}{2\mu+\lambda} - \frac{\rho_i \rho_r g^2 h_i}{(2\mu+\lambda)^2}\frac{(y-y_b)(2y_t - y_b - y)}{2}. \tag{4.42}$$

Within Framework I, the introduction of the modified stress tensor changes the problem, and the exact solution to the so-arising BVP reads

$$\begin{cases} \widetilde{u}_1 = 0 \\ \widetilde{u}_2 = -\dfrac{\rho_i g h_i (y - y_b)}{(2\mu+\lambda) + \rho_r g (y_t - y_b)}, \end{cases} \tag{4.43}$$

and assuming that

$$\left|\frac{\rho_r g(y_t - y_b)}{2\mu+\lambda}\right| < 1,$$

the two leading terms in the expansion of $\widetilde{u}_2$ become

$$\widetilde{u}_2 \approx -\frac{\rho_i g h_i (y - y_b)}{2\mu+\lambda} - \frac{\rho_i \rho_r g^2 h_i}{(2\mu+\lambda)^2}(y - y_b)(y_t - y_b). \tag{4.44}$$

Hence, the error in the solution to Problem 4.5.1 produced in Framework I is of the order of the difference between Equation (4.42) and Equation (4.44), that is

$$u_2 - \widetilde{u}_2 = \mathcal{O}\left(E_2(y)\right), \tag{4.45}$$

where

$$E_2(y) = \frac{\rho_i \rho_r g^2 h_i (y - y_b)^2}{2(2\mu+\lambda)^2}.$$

Numerical results for Problem 4.5.1

For the $C0$-formulation of Problem 4.5.1, both $\mathbf{u}_I$ and $\mathbf{u}_{II}$ coincide with the the exact solution

$$\begin{cases} u_1 = 0 \\ u_2 = -\dfrac{\rho_i g h_i (y - y_b)}{2\mu + \lambda}, \end{cases}$$

which is to be expected since the $T(\mathbf{u})$ coincides with $\sigma(\mathbf{u})$ in this case. For the Model $C1$ on the other hand, the error in $\mathbf{u}_I$ is of the same order of magnitude as E_2, whereas the solution $\mathbf{u}_{II}$ coincides with the exact solution up to the convergence criterion of the iterative solver (six orders of magnitude).

For Problem 4.5.1, we also compare the time spent for assembly of the stiffness matrix, and the time used to solve the linear system within the two frameworks. We find that except for the two smallest problems in 3D, the overall time (assembly + solution) is lower for the solver in Framework II, than for that in Framework I, despite the fact that the problem size in Framework II is larger than the problem size in Framework I.

Numerical results for Problem 4.5.2

For the $C0$-formulation of Problem 4.5.2, $\mathbf{u}_I$ and $\mathbf{u}_{II}$ seem to converge to the same solution in both two and three space dimensions. This holds for both the compressible ($\nu = 0.2$) and the nearly incompressible ($\nu = 0.4999999$) case.

As for the $C1$ formulation of Problem 4.5.1, the two solutions to the $C1$ version of Problem 4.5.2 differ. Since no exact solution to the problem is known in this case, it is impossible to tell which of the solutions that is the correct one, but the results from Problem 4.5.1 advocates that it is $\mathbf{u}_{II}$.

4.6 Paper VI

In Paper VI we consider a two-level finite element discretization of elliptic PDEs, and we construct the block-factorized preconditioner B_M based on a fine-coarse splitting of the system matrix A. The novelties of this paper are that we, based on the finite element framework, (i) propose and analyze two methods to construct sparse approximations of the inverse of the pivot block A_{11}, and (ii) introduce a way to construct a sparse matrix Z_{12} that approximates the off-diagonal block $A_{11}^{-1}A_{12}$.

Block-factorized preconditioner

Consider a matrix A which arises from a finite element discretization of an elliptic PDE on a two-level FE mesh, where the fine mesh is a uniform refinement of the coarse one. The corresponding splitting of A into

two-by-two block form reads,

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{matrix} \} fine, \\ \} coarse. \end{matrix}$$

We seek the preconditioner B_M of the form

$$B_M = L_B U_B = \begin{pmatrix} B_{11} & 0 \\ A_{21} & S \end{pmatrix} \begin{pmatrix} I_1 & Z_{12} \\ 0 & I_2 \end{pmatrix}, \tag{4.46}$$

where B_{11}^{-1} approximates A_{11}^{-1} and S is an approximation of the exact Schur complement of A, $S_A = A_{22} - A_{21}A_{11}^{-1}A_{12}$.

The block Z_{12} is a sparse matrix which approximates the matrix product $A_{11}^{-1}A_{12}$, and it is constructed as a means to reduce the computational complexity of B_M. This reduction is due to the fact that when B_M is applied as a preconditioner it suffices to solve once with A_{11}. This is in contrast to the standard version of the block-factorized preconditioner, cf. Equation (3.13), where we have to solve twice with the pivot block. On the other hand, the block Z_{12} makes B_M nonsymmetric even when A is symmetric. However, we do not consider this as a substantial drawback since we aim at handling nonsymmetric matrices, and because B_M turns out to work efficiently within a general iterative method for nonsymmetric matrices.

Finite element based approximation of the pivot block

The approximation of the inverse of the pivot block B_{11}^{-1} is constructed in an element-by-element (EBE) fashion as follows

$$B_{11}^{-1} = \sum_{k=1}^{M} R_k^T A_{11,k}^{-1} R_k, \tag{4.47}$$

where $A_{11,k}$ is the pivot block in the element stiffness matrix on the kth macroelement. The matrix R_k is a restriction matrix from the global numbering to the elementwise numbering of the unknowns.

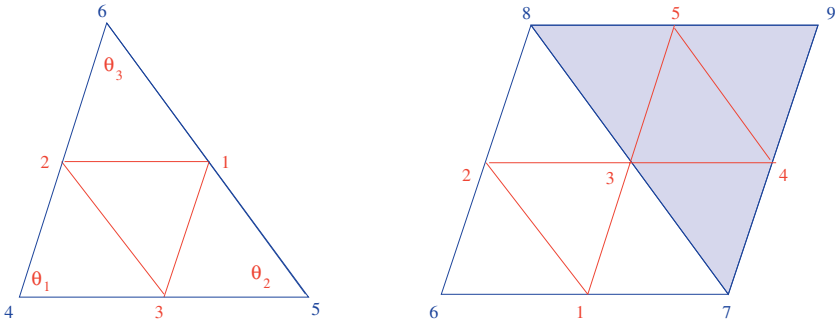
It is shown in [8] that for spd matrices B_{11}^{-1} and A_{11}^{-1} are spectrally equivalent. That is, there exists constants $0 < \alpha_1 \leq \alpha_2$ such that

$$\alpha_1 \mathbf{v}_1^T A_{11}^{-1} \mathbf{v}_1 \leq \mathbf{v}_1^T B_{11}^{-1} \mathbf{v}_1 \leq \alpha_2 \mathbf{v}_1^T A_{11}^{-1} \mathbf{v}_1 \quad \forall \mathbf{v}_1 \in \mathbf{V}_1, \tag{4.48}$$

holds. The bounds α_1 and α_2 depend on the ratio κ_1/κ_2, where

$$\kappa_1 \leq \lambda_{\min}(A_{11,k}) \quad \text{and} \quad \kappa_2 \geq \lambda_{\max}(A_{11,k}) \quad \forall k = 1, \ldots, M.$$

Thus, the bounds in Equation (4.48) are independent of the mesh discretization parameter h, but they are not robust with respect to problem or mesh anisotropy, nor to jumps in the PDE coefficients, since κ_1/κ_2 depend on the latter parameters.



(a) A macroelement (b) A macroelement molecule

Figure 4.3: Macroelements

Therefore, we propose two related methods to construct the approximation of B_{11}^{-1}. The first is the EBE approach, but the macroelement pivot matrix is scaled from one side by a diagonal matrix. In the sequel we refer to this technique as EBES. In the second approach, $A_{11,k}$ is replaced by a matrix which is a restriction of the assembled block A_{11} to the kth macroelement. This method is referred to as EBERS.

As a means to analyze the EBER and EBERS approaches, we introduce the following simple test problem.

Problem 4.6.1 Consider the scalar Poisson equation, discretized on an arbitrary triangular mesh. As pointed out, for example in [5], it is equivalent to consider either a scalar anisotropic Laplace operator on isosceles triangles, or an isotropic Laplace operator on an arbitrary mesh. Here we consider the former case on meshes of the following three types:

$\mathcal{T}_1$: Right-angled isosceles triangles.
$\mathcal{T}_2$: Triangles with one large and two small angles.
$\mathcal{T}_3$: Triangles with two large and one small angle.

To analyze further the EBES and EBERS methods also in the case of discontinuous coefficients, it suffices to consider two neighboring macroelements, as indicated in Figure 4.3(b). Assume that there is a discontinuity of size s aligned with the two macroelements which we incorporate by multiplying the element coefficient matrix in the non-shaded area by s.

Scaled macroelement pivot inverses (EBES)

The EBES approximation of the inverse of the pivot block is denoted as $\widetilde{B}_{11}^{-1}$. It is constructed like

$$\widetilde{B}_{11}^{-1} = \sum_{k=1}^{M} R_k^T A_{11,k}^{-1} D_k R_k, \tag{4.49}$$

where D_k is a diagonal matrix with entries 1 and 1/2 (in 2D). The entries with values 1/2 correspond to the columns of $A_{11,k}^{-1}$ associated with the

nodes that are shared between two neighboring macroelements. A closer look at the product $A_{11}\widetilde{B}_{11}^{-1}$ reveal that (for $k \neq l$)

$$\begin{aligned} A_{11}\widetilde{B}_{11}^{-1} &= I_1 + \sum_{l,k=1}^{M} R_l^T A_{11,l} R_l R_k^T A_{11,k}^{-1} D_k R_k \\ &= I_1 + \sum_{l,k=1}^{M} R_l^T \widetilde{P}_{l,k} R_k = I_1 + \sum_{l,k=1}^{M} \widetilde{W}_{l,k} = I_1 + \widetilde{W}_{11}. \end{aligned} \tag{4.50}$$

Hence, in order to estimate the spectrum of $A_{11}\widetilde{B}_{11}^{-1}$ it suffices to investigate the numerical range of $\widetilde{W}_{11}$, and in Paper VI we show that for appropriate vectors $\mathbf{x}$ and for $k \neq l$ we have

$$\mathbf{x}^T \widetilde{W}_{11} \mathbf{x} \leq 3\sigma_{max}(\widetilde{P}_{l,k}). \tag{4.51}$$

The factor 3 reflects the fact that for 2D triangular elements the global index set for each macroelement pivot matrix intersects that of at most three other pivots from neighboring macroelements.

The matrices $\widetilde{P}_{l,k}$ are of rank one, and we show that these can be expressed as the outer product of two vectors,

$$\widetilde{P}_{l,k} = \frac{s}{2} \widetilde{\mathbf{v}}_{l,k} \widetilde{\mathbf{w}}_{l,k}^T.$$

For Problem 4.6.1, $l,k = 1,2$, these vectors read

$$\widetilde{\mathbf{v}}_{12} = \begin{pmatrix} \frac{a+b+c}{2} \\ -c \\ -b \end{pmatrix} \qquad \widetilde{\mathbf{w}}_{12} = \frac{1}{2} \begin{pmatrix} (a+c)^{-1} \\ (a+b)^{-1} \\ (a+c)^{-1} \end{pmatrix},$$

where $a = \cot\theta_1$, $b = \cot\theta_2$, $c = \cot\theta_3$ and $\theta_1 \geq \theta_2 \geq \theta_3$ are the angles in the triangle shown in Figure 4.3(a). We are able to express the singular value σ of $\widetilde{P}_{12}$ analytically by the expression

$$\sigma = \frac{s}{4} \frac{\sqrt{\left(5(a+b)^2 + 5(a+c)^2 + 2(ab+ac+bc)\right)\left(2(a+b)^2 + (b+c)^2 + 2ac\right)}}{(a+c)(a+b)}, \tag{4.52}$$

and when the formula in Equation (4.52) is evaluated for the triangulations $\mathcal{T}_i$, $i = 1,2,3$, the result is

Triangulation	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$
σ	$2.1213s$	$3.5591s$	$802.505s$.

Clearly, the EBES approximation is robust neither with respect to coefficient jumps nor mesh anisotropy.

Restricted scaled macroelement pivot inverses (EBERS)

We elaborate on the EBES pivot block approximation in order to reduce the dependence of jumps in the problem coefficients, and to this end we propose the following idea. Instead of inverting the local macroelement pivot matrices $A_{11,k}$, we proceed as follows:

(i) restrict the assembled block A_{11} to a macroelement k,

(ii) invert the so-obtained local matrix, scale it with D_k, denote the result $\widehat{A}^{-1}_{11,k}$, and

(iii) let

$$\widehat{B}^{-1}_{11} = \sum_{k=1}^{M} R_k^T \widehat{A}^{-1}_{11,k} R_k, \tag{4.53}$$

In this case we obtain (for $k \neq l$) the product

$$\begin{aligned} A_{11}\widehat{B}^{-1}_{11} &= \widehat{D}_{11} + \sum_{k,l=1}^{M} R_l^T A_{11,l} R_l R_k^T \widehat{A}^{-1}_{11,k} R_k \\ &= \widehat{D}_{11} + \sum_{k,l=1}^{M} R_l^T \widehat{P}_{l,k} R_k, \end{aligned} \tag{4.54}$$

where $\widehat{D}_{11}$ is a sparse matrix. It is not identity since the local contributions in $\widehat{B}^{-1}_{11}$ are based on scaled restrictions of A_{11} to the elements, rather than scaled element matrices.

When we apply the EBERS approach on Problem 4.6.1, the expressions for the vectors $\widehat{\mathbf{v}}_{12}$ and $\widehat{\mathbf{w}}_{12}$ whose product represents $\widehat{P}_{12}$, and for the singular value σ of $\widehat{P}_{12}$ become tedious. Therefore, they are not given here explicitely. Instead the evaluation of the analytical expression of σ for the different triangulations $\mathcal{T}_i$, $i = 1,2,3$, and three representative values of the jump s are presented below.

Triangulation	$\sigma\|_{s=0.001}$	$\sigma\|_{s=1}$	$\sigma\|_{s=10000}$
$\mathcal{T}_1$	3.908e-4	0.183	0.3423
$\mathcal{T}_2$	4.939e-4	0.221	0.4005
$\mathcal{T}_3$	4.939e-4	0.221	0.4005.

It should be remarked here that the EBERS idea is based on intuitive arguments. However, its efficiency is confirmed by numerical experiments, but the approach needs to be fully theoretically justified. We conclude that the singular value of the rank-one matrices $\widehat{P}_{l,k}$ is bounded independently of the coefficient jumps and the mesh anisotropy. The spectrum of $\widehat{D}_{11}$ is not known, and, up to the knowledge of the authors, there is no general theory on how to estimate the spectrum of the sum of two matrices. Therefore, future work on the EBERS approach should include an investigation on how to scale the matrices $\widehat{B}^{-1}_{11,k}$ such that $\widehat{D}_{11}$ becomes the identity matrix.

The Z_{12}-block and the Schur complement approximation

We extend the idea to approximate a block in the block-factorized preconditioner by the sum of its corresponding local finite element counterparts to the construction of Z_{12} as an approximation of $A_{11}^{-1}A_{12}$. The off-diagonal block Z_{12} in B_M is constructed as

$$Z_{12} = \sum_{k=1}^{M} R_k^T \widetilde{A}_{11,k}^{-1} A_{12,k} R_k,$$

where $\widetilde{A}_{11,k}^{-1} = A_{11,k}^{-1} D_k$. If we consider $A_{11}Z_{12}$, then (for $k \neq l$)

$$A_{11}Z_{12} = \widetilde{A}_{12} + \sum_{k,l=1}^{M} W_{12,kl},$$

where

$$\widetilde{A}_{12} = \sum_{k=1}^{M} R_k^T D_k A_{12,k} R_k \quad \text{and} \quad W_{12,kl} = R_k^T A_{11,k} R_k R_l^T A_{11,l}^{-1} D_l A_{12,l} R_l.$$

The blocks Z_{12} are sparse and cheap to compute explicitly, and the results from the numerical experiments in Paper VI convincingly confirm the efficiency of the approach.

The Schur complement approximation S is assembled from local contributions which are exactly computed on the macroelements, that is

$$S = \sum_{k=1}^{M} A_{22,k} - A_{21,k}(A_{11,k})^{-1} A_{12,k}.$$

See Section 3.5.1 and the references therein for further details on this approximation.

Numerical experiments

The performance of the preconditioner B_M with the proposed approximations $\widetilde{B}_{11}^{-1}$, $\widehat{B}_{11}^{-1}$, and Z_{12} is investigated on a number of problems, symmetric as well as nonsymmetric. The symmetric ones are characterized by jumps in the PDE coefficients and by mesh anisotropy, whereas the nonsymmetric problem is a convection-diffusion equation with constant wind. When the action of B_M^{-1} on a vector is computed , the pivot block is solved either as one application of $\widetilde{B}_{11}^{-1}$ ($\widehat{B}_{11}^{-1}$), or by an inner iterative solution method (GCG-MR) preconditioned by $\widetilde{B}_{11}^{-1}$ ($\widehat{B}_{11}^{-1}$).

The EBES approximation is applied on the symmetric problems only, and when the solution with the pivot block is accurate enough, the resulting block-factorized preconditioner is robust with respect to coefficient jumps and mesh anisotropy.

The EBERS approach is applied on the symmetric problem with discontinuities in the coefficients of the PDE, and similarly to the EBES case, B_M is

robust when the solution with A_{11} is accurate enough. For the nonsymmetric problem, the EBERS approximation gives a block-factorized preconditioner that is robust with respect to the considered range of the magnitude and the direction of the convection. Finally, we see that the preconditioner is optimal with respect to the problem size.

5. Concluding remarks and future work

In this chapter, I summarize the main contributions in this thesis, and sketch some plans for follow-up work.

For the saddle point problem in Papers II – V, we employ well-known block-triangular and block-factorized preconditioners for which the need arises to approximate the Schur complement of the system matrix. We propose a novel algorithm for the construction of this approximation based on the finite element framework. The global Schur approximation is assembled from elementwise, exactly computed Schur matrices, and for the specific problems considered in Papers II – V, the approach is shown to work well in numerical experiments.

For the future it would be of interest to study whether this Schur complement approximation is good also for other symmetric and nonsymmetric saddle point problems, such as Stokes problem, and Oseen's and Navier – Stokes equations, and compare it to popular approximation techniques used by others. Furthermore, it would be of importance to find a full theoretical justification for this approach.

When it comes to the modeling of isostatic glacial rebound, interesting topics for future work are (i) the choice of the quadrature rule for the time-integration in the solution scheme in Paper IV, (ii) how to actually make beneficial use of the fact that the preconditioner must not be completely reconstructed in every time step, and (iii) the discrepancy between the solutions obtained from Framework I and Framework II in Paper V.

In the solution algorithm in Paper IV, the choice of the trapezoidal rule, which is a closed quadrature method, leads to a constraint on the size of the time-step Δt_j. It is worth to investigate whether the choice of an open quadrature rule could eliminate this problem.

For the third issue, it is of practical interest to find out whether the difference between the solutions to the purely elastic problem obtained from Framework I and II remains also in the viscoelastic case. Some work together with Björn Lund and Volker Klemann has already been initiated, and preliminary results indicate that this is the case.

The contributions in Paper VI are two-fold, where the first one is the construction of the preconditioner P for the pivot block A_{11}. The matrix P is a scaled sparse approximate inverse which is assembled from exactly inverted matrices on each (macro)element of the finite element mesh. When

the local matrix is taken as the restriction of A_{11} to the macroelement, P is numerically found to be independent of the mesh parameter and robust with respect to coefficient jumps in the PDE. The strategy is not fully theoretically justified, and further work is required. One issue that needs further attention is how to scale, if possible, the inverse of the restricted matrix such that the product $A_{11}P^{-1}$ can be expressed as an identity matrix which is perturbed by a sum of low-rank matrices.

The second contribution in Paper VI is that the block $P^{-1}A_{12}$ in the block-factorized two-level preconditioner is replaced by a matrix Z_{12}, which is assembled from exactly computed local elementwise contributions $Z_{12,E} = A_{11,E}^{-1}A_{12,E}$. In this case the local matrices are (scaled) element matrices. Introducing the Z_{12} block in the suggested way, one avoids one application of P^{-1} in the block-factorized two-level method, which reduces the computational complexity of the preconditioner.

6. Sammanfattning på svenska

Inom många vetenskapsområden går det inte att göra klassiska experiment för att få svar på de frågor man ställer sig. Inom till exempel astronomi, astrofysik eller geofysik är experiment omöjliga på grund av de stora avstånden och de långa väntetiderna som är inblandade. I mer jordnära applikationer, som till exempel tillverkningsindustri, är det ekonomiskt fördelaktigt att undvika experiment. Det är mycket billigare att simulera en bilkrock i en dator än att bygga en prototyp för att sedan krascha den mot en betongpelare.

Numeriska simuleringar utförs ofta genom att man skapar en modell av det man är intresserad av i termer av partiella differentialekvationer (PDE) och löser dem numeriskt. PDE utgör grunden för den matematiska fysiken och de kan användas för att beskriva i stort sett alla processer vi ser omkring oss. Till exempel luftrörelser i atmosfären som påverkar vädret och spridningen av miljöfarliga ämnen, luftflöden kring bilar, tåg och flygplan, förbränningen i en bilmotor eller en turbin, utbredningen av radiovågor från mobiltelefoner i luft och mänskliga vävnader, ljud- och bullerutbredning och så vidare.

Lösningsmetoder för linjära ekvationssystem

PDE är sådana att man, förutom i ett begränsat antal specialfall, inte känner till analytiska lösningar till dem. Det betyder att man inte kan skriva ner ett uttryck som i detalj och i varje punkt i rummet beskriver det man är intresserad av, till exempel ett luftflöde eller en förbränningsprocess. Istället väljer man ut ett antal punkter, *noder*, i rummet och gör om problemet så att man kan hitta en ungefärlig lösning till PDEn i dessa noder. Denna process kallas för att diskretisera PDEn och resultatet av diskretiseringen är ett linjärt ekvationssystem av typen

$$A\mathbf{x} = \mathbf{b} \tag{6.1}$$

som måste lösas. I (6.1) är A en gles eller full matris, och $\mathbf{x}$ och $\mathbf{b}$ är vektorer. Att A är gles innebär att en övervägande del av elementen i matrisen är noll, något man kan dra nytta av när den ska lagras i en dator. När A är full är (nästan) alla element skilda från noll.

Den lösning $\mathbf{x}$ man får från (6.1) är en approximation till lösningen till den ursprungliga PDEn och alltså inte exakt i matematisk mening. Men i de allra flesta fall är den tillräckligt noggrann för att kunna

användas i praktiska sammanhang. Om man använder välkända diskretiseringsmetoder för PDEn såsom finita elementmetoden (FEM), finita differensmetoden, eller randelementmetoden (BEM) kan man dessutom teoretiskt förutsäga hur stort felet, skillnaden mellan den exakta och den approximativa lösningen, faktiskt är.

Det kan verka som att gå ur askan in i elden genom att gå från en PDE till ett linjärt ekvationssystem, men fördelen är att när man i det första fallet har *ett* omöjligt problem att lösa, får man i det senare *många* enkla problem. Problem som blir så enkla att lösa att man kan programmera en dator för att göra det åt en.

Det linjära ekvationssystemet man vill lösa kan vara gigantiskt (ett antal miljoner ekvationer är inte ovanligt) och det kan vara svårt att lagra det i minnet även på moderna högpresterande superdatorer. Lagringsproblemet blir dessutom ytterligare förvärrat om man använder en direkt lösningsmetod (till exempel Gausselimination eller Choleskyfaktorisering) för att lösa (6.1). Det beror på att en direkt metod skapar utfyllnadselement som gör en från början gles matris allt fullare och fullare. Dessutom är direkta metoder aritmetiskt sett väldigt krävande eftersom det utförs beräkningar på varje nollskilt element i matrisen i flera omgångar. Det gäller både de ursprungliga elementen och de som fyllts in av lösningsmetoden.

Ett alternativ är att använda en så kallad iterativ lösningsmetod istället. De är gjorda så att man först gissar en lösning till det linjära ekvationssystemet. Sen förbättrar man lösningen steg för steg genom att lägga till och dra ifrån uppdateringar i en upprepad, iterativ, process. Itererandet avbryts när man är nöjd med lösningen, man säger att den har konvergerat. Varje iteration innebär ett visst arbete och man vill av naturliga skäl använda så få iterationer som möjligt. När antalet iterationer som behövs för konvergens är oberoende av parametrarna till den diskretiserade PDEn så säger man att den iterativa metoden är *robust*. När inte heller storleken hos det linjära ekvationssystemet spelar in på antalet iterationer kallas metoden *optimal*.

För att göra en iterativ lösningsmetod mer robust och mer optimal kombineras den med en så kallad för- eller prekonditionerare. Det kan vara en matris som approximerar verkan av As invers på en vektor, $G^{-1}\mathbf{x} \approx A^{-1}\mathbf{x}$. En prekonditionerare kan också vara en matris eller en procedur G som approximerar A själv, men som är lättare att lösa. Genom att applicera prekonditioneraren på (6.1) så transformeras problemet till

$$G^{-1}A\mathbf{x} = G^{-1}\mathbf{b}. \tag{6.2}$$

Om prekonditioneraren väljs på ett bra sätt är (6.2) lättare att lösa iterativt än det ursprungliga problemet (6.1). Det naiva valet av prekonditionerare är naturligtvis A^{-1} självt, men man får då tillbaka alla de problem med minnesåtgång och beräkningskapacitet som är förknippade med de direkta lösningsmetoderna.

Utifrån diskussionen ovan om olika lösningsmetoder kan man få intrycket av att direkta metoder alltid är dåliga och att iterativa alltid är bra, men så är det inte. En direkt lösningsmetod som är väl implementerad är i allmänhet snabbare än en iterativ metod för små och medelstora problem. Men det kräver att ekvationssystemet är tillräckligt litet för att man ska kunna lagra utfyllnadselementen i datorns minne. Dessutom kan det linjära ekvationssystemet vara sådant att en iterativa lösningsmetod inte konvergerar.

Sammanfattning av avhandlingen

I den här avhandlingen studerar jag metoder för att konstruera prekonditionerare till matriser som har en 2×2 blockstruktur,

$$A = \begin{pmatrix} B & C \\ D & E \end{pmatrix}. \tag{6.3}$$

Matrisen kan få en sådan form på olika sätt. Dels kan strukturen hos den diskretiserade PDEn vara sådan att A redan från början har rätt form. Dels kan man ordna om As rader och kolumner så att man får någon önskvärd egenskap hos B eller E. Ett tredje sätt är att göra en uppdelning av As rader och kolumner på ett sätt som motsvaras av fina och grova noder i diskretiseringen.

Det finns välkända och välbeskrivna metoder för att konstruera prekonditionerare till blockmatriser och dessa prekonditionerare kan vara på blockdiagonal, blocktriangulär eller blockfaktoriserad form. I den här avhandlingen fokuserar jag på de båda senare som har gemensamt att de innefattar Schurkomplementet till A, $S_A = E - DB^{-1}C$. Schurkomplementet är en matris som i allmänhet är kostsam att beräkna på grund av B^{-1}. Dessutom är den oftast full, även om de ingående matriserna är glesa. För att undvika de aritmetiska och lagringsmässiga svårigheter som är förknippade med Schurkomplementet måste det approximeras med en matris som är gles och beräkningsmässigt billig att konstruera och hantera.

Uppsats I

I den här avhandlingen studerar jag linjära ekvationssystem som härstammar från tre olika applikationer. I uppsats I kommer ekvationssystemet, som är fullt, från en typ av BEM-diskretisering av ekvationer som beskriver utbredningen av sprickor i spröda material. Blockstrukturen hos A är sådan att ett, eller båda, av de diagonala blocken är diagonaldominanta. Vi visar genom numeriska experiment att enkla algebraiskt konstruerade prekonditionerare resulterar i en iterativ lösningsmetod vars effektivitet är väl jämförbar med en direkt metod.

Uppsats II – V

Den andra typen av linjära ekvationssystem, som studeras i uppsats II, III, IV och V, beror blockstrukturen hos A på den PDE som diskretiseras. PDEn beskriver den (visko)elastiska responsen hos jordskorpan och den övre delen av jordens mantel på en yttre last. Eller på ren svenska; landhöjningen efter en inlandsis framryckning och tillbakadragande. Vi approximerar Schurkomplementet som behövs för prekonditioneraren med en matris som assembleras från elementvis, exakt beräknade, små Schurmatriser. Genom numeriska experiment visar vi att kvaliteten hos den erhållna Schurkomplementapproximationen är bra.

När blockprekonditioneraren för det linjära ekvationssystemet kombineras med en inre iterativ lösare som prekonditioneras med en nästan optimal multilevel-metod så är den resulterande iterativa lösningsmetoden robust med avseende på materialparametrar, antalet rumsdimensioner hos problemet och diskontinuiteter i koefficienterna till PDEn. Dessutom är den resulterande iterativa metoden närapå optimal.

I uppsats V jämför vi två sätt att formulera de PDEr som används för att modellera jordens rent elastiska respons på en inlandsis framryckning och tillbakadragande. Dessutom jämförs två olika lösningsmetoder för de linjära ekvationssystem som uppstår efter en finita-element (FE) diskretisering av ekvationerna. I den första formuleringen av problemet representeras PDEn fullt ut, medan i den andra är representationen begränsad av de möjligheter som erbjuds av den FE mjukvara som används. I det första fallet används den prekonditionerade, iterativa lösningsmetod som beskrivs i de två tidigare styckena, medan i det andra fallet används en direkt lösare. Genom analyser och numeriska experiment visar vi att det första sättet att formulera PDEn och att lösa ekvationssystemet är mer effektivt och noggrant än det senare.

Uppsats VI

I uppsats VI studerar vi linjära ekvationssystem där blockstrukturen tillskrivs en uppdelning av raderna och kolumnerna i A i fina och grova. Även i detta fall approximerar vi Schurkomplementet med en matris som assembleras av lokala, exakta små Schurmatriser. En liknande strategi används för att konstruera en gles approximation av inversen till pivotblocket (B^{-1}) genom assemblering av små, lokala, exakt inverterade matriser. Vår analys av den senare approximationen visar att den är optimal och robust med avseende på diskontinuiteter hos PDEns koefficienter. Genom numeriska experiment bekräftar vi analysen och visar att den iterativa lösaren är optimal för de symmetriska och ickesymmetriska problem som behandlats. Genom att approximera det utomdiagonala blocket $B^{-1}C$ genom assemblering av små lokala, exakt beräknade matriser kan vi minska komplexiteten hos den blockfaktoriserade prekonditioneraren.

Framtida arbete

De viktigaste bidragen i den här avhandlingen är

(i) approximationen av Schurkomplementet i uppsats II–V,

(ii) approximationen av pivotblocket B och det utomdiagonala blocket $B^{-1}C$ i uppsats VI, och

(iii) analysen av modellerna och lösningssätten i uppsats V.

Approximationerna i (i) och (ii) har testats utförligt och visat sig fungera bra i numeriska experiment. Men de är inte fullt ut teoretiskt underbyggda och det är ett ämne för framtida forskning att studera dem vidare. Vad gäller (iii) är det av intresse att analysera om skillnaderna i resultaten från de två angreppssätten består om jorden modelleras som ett viskoelastiskt medium istället för ett elastiskt.

7. Acknowledgements

I would like to thank my supervisor Dr. Maya Neytcheva for her encouragement and the endless hours she has spent on answering my questions and proofreading my reports and papers. Without her this thesis would never have been written.

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.

During my years as a PhD student I have had the pleasure to visit a number of research institutes abroad. I would like to express my gratitude to the people who made those trips possible. They are: Prof. Radim Blaheta, Institute of Geonics, Academy of Sciences of the Czech Republic; Prof. Svetozar Margenov, the Institute for Parallel Processing, the Bulgarian Academy of Sciences; and Prof. Yousef Saad, Department of Computer Science and Engineering, University of Minnesota. Furthermore, I am grateful for the valuable discussions I had with Prof. Raytcho Lazarov, Department of Mathematics, Texas A&M University, during the visit in Sofia in March and April 2006.

Eddie Wadbro and Petra Bängtsson, thank you for proofreading this thesis.

All my colleagues at the Department of Information Technology, thank you for skiing and climbing trips, and endless coffee table discussions over topics ranging from quantum physics to astrology, via railways, alarm clocks, and the care of baby polar bears. It has been a fantastic working place.

A. On the dimensionless scaling of the equations of glacial rebound

Consider the moment balance equation of a linear pre-stressed (visco)elastic solid that occupies the domain Ω,

$$\nabla_{\mathbf{x}} \cdot \sigma(\mathbf{u},\mathbf{x}) + \nabla_{\mathbf{x}}(\mathbf{u}\cdot\mathbf{b}) - (\nabla_{\mathbf{x}}\cdot\mathbf{u})\mathbf{c} = \mathbf{f}(\mathbf{x}) \quad \text{in } \Omega \tag{A.1}$$

where $\sigma = \sigma(\mathbf{u},\mathbf{x})$ is the stress tensor,$\mathbf{b}$ and $\mathbf{c}$ are coefficient vectors, $\mathbf{u}$ is the displacement field, $\mathbf{f}$ is a body force, and

$$[\nabla_{\mathbf{x}}]_i = \frac{\partial}{\partial x_i}.$$

The size of Ω is of magnitude L, and to start with, let us scale the coordinate system with this typical length,

$$\widetilde{\mathbf{x}} = \left[\widetilde{x},\widetilde{y},\widetilde{z}\right] = \left[\frac{x}{L},\frac{y}{L},\frac{y}{L}\right], \tag{A.2}$$

and the stress with some typical stress S,

$$\widetilde{\sigma}(\mathbf{u},\mathbf{x}) = \frac{\sigma(\mathbf{u},\mathbf{x})}{S}. \tag{A.3}$$

With those two scalings imposed, Equation (A.1) reads

$$\frac{1}{L}\nabla_{\widetilde{\mathbf{x}}}\cdot[S\widetilde{\sigma}(\mathbf{u},\mathbf{x})] + \frac{1}{L}\nabla_{\widetilde{\mathbf{x}}}(\mathbf{u}\cdot\mathbf{b}) - \frac{1}{L}(\nabla_{\widetilde{\mathbf{x}}}\cdot\mathbf{u})\mathbf{c} = \mathbf{f}(\widetilde{\mathbf{x}}L) \quad \text{in } \widetilde{\Omega}. \tag{A.4}$$

Linear isotropic elastic solid

From standard theory of linear isotropic elasticity for small displacements, the stress is related to the displacements according to the constitutive relation

$$\sigma(\mathbf{u},\mathbf{x}) = \lambda(\nabla_{\mathbf{x}}\cdot\mathbf{u})I + 2\mu\varepsilon_{\mathbf{x}}(\mathbf{u}(\mathbf{x})) \tag{A.5}$$

where

$$[\varepsilon_{\mathbf{x}}(\mathbf{u})]_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

is the strain tensor, and

$$\mu = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda = \frac{2\mu\nu}{1-2\nu}$$

are the Lamé coefficients, E is the Young modulus, and ν is the Poisson number.

After a scaling of $\mathbf{u}$ with some typical displacement U,

$$\widetilde{\mathbf{u}} = \frac{\mathbf{u}}{U}, \tag{A.6}$$

we get

$$\begin{aligned} \widetilde{\sigma}(\mathbf{u},\mathbf{x}) &= \frac{U}{L}\widetilde{\sigma}(\widetilde{\mathbf{u}},\widetilde{\mathbf{x}}) \\ \nabla_{\widetilde{\mathbf{x}}}(\mathbf{u}\cdot\mathbf{b}) &= U\nabla_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}\cdot\mathbf{b}) \\ (\nabla_{\widetilde{\mathbf{x}}}\cdot\mathbf{u})\mathbf{c} &= U(\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}})\mathbf{c}. \end{aligned} \tag{A.7}$$

Substitution of the relations in Equation (A.7) into Equation (A.4) yields

$$\frac{U}{L^2}\nabla_{\widetilde{\mathbf{x}}}\cdot[S\widetilde{\sigma}(\widetilde{\mathbf{u}},\widetilde{\mathbf{x}})] + \frac{U}{L}\nabla_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}\cdot\mathbf{b}) - \frac{U}{L}(\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}})\mathbf{c} = \mathbf{f}(L\widetilde{\mathbf{x}}) \qquad \widetilde{\mathbf{x}}\in\widetilde{\Omega} \tag{A.8}$$

or,

$$\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\sigma}(\widetilde{\mathbf{u}},\widetilde{\mathbf{x}}) + \frac{L}{S}\nabla_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}\cdot\mathbf{b}) - \frac{L}{S}(\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}})\mathbf{c} = \frac{L^2}{SU}\mathbf{f}(L\widetilde{\mathbf{x}}) \qquad \widetilde{\mathbf{x}}\in\widetilde{\Omega}. \tag{A.9}$$

After substitution of Equation (A.5) into Equation (A.9), the following dimensionless equation is obtained (with tildes and implicit dependencies omitted) ,

$$\left[\frac{\lambda}{S}+\frac{\mu}{S}\right]\nabla(\nabla\cdot\mathbf{u})I + \frac{\mu}{S}\nabla\mathbf{u} + \frac{L}{S}\nabla(\mathbf{u}\cdot\mathbf{b}) - \frac{L}{S}(\nabla\cdot\mathbf{u})\mathbf{c} = \frac{L^2}{SU}\mathbf{f}(L\mathbf{x}) \qquad \mathbf{x}\in\widetilde{\Omega} \tag{A.10}$$

Remark A.1 *For the problem of interest in Papers II – V, the Young modulus $E = 400$ GPa, the typical length L is 10^7 m, and $\mathbf{b} = \mathbf{c} = \rho g\mathbf{e}_d$. With the density of the rock ρ of the magnitude 10^3 $kg\,m^{-3}$, and the gravitational acceleration $g = \mathcal{O}(10)$, we can conclude that the elliptic and the advective parts of Equation* (A.10) *are of the same magnitude. Hence, the convective terms does not dominate the problem.*

Linear isotropic viscoelastic solid

When the rheology of the solid is viscoelastic, the constitutive relation connecting the stress and the displacements reads,

$$\begin{aligned} \sigma(\mathbf{x},t) =& \lambda(\mathbf{x},t,t)\nabla_{\mathbf{x}}\cdot\mathbf{u}(\mathbf{x},t,t)I + \mu(\mathbf{x},t,t)\varepsilon_{\mathbf{x}}(\mathbf{u}(\mathbf{x},t)) \\ &- \int_0^t \lambda_\tau(\mathbf{x},t,\tau)\nabla_{\mathbf{x}}\cdot\mathbf{u}(\mathbf{x},\tau)I + \mu_\tau(\mathbf{x},t,\tau)\varepsilon_{\mathbf{x}}(\mathbf{u}(\mathbf{x},\tau))d\tau, \end{aligned} \tag{A.11}$$

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, and when the stress-relaxation function obey the so-called Maxwell model, they read

$$\lambda(\mathbf{x},t,\tau)=\lambda_E(\mathbf{x})e^{-\frac{(t-\tau)}{\tau_0}}\qquad \mu(\mathbf{x},t,\tau)=\mu_E(\mathbf{x})e^{-\frac{(t-\tau)}{\tau_0}}, \tag{A.12}$$

where $\tau_0(\mathbf{x})=\frac{\mu_E(\mathbf{x})}{\eta(\mathbf{x})}$ is the Maxwell time, and η is the dynamic viscosity, see [42]. In the sequel the space dependence of the parameters is omitted for notational simplicity.

When the relations (A.2), (A.3) and (A.6) are substituted into Equation (A.11), together with the scalings

$$\widetilde{t}=\frac{t}{T}\qquad \widetilde{\tau}=\frac{\tau}{T} \tag{A.13}$$

of the time variables, the result reads

$$\begin{aligned}\widetilde{\sigma}(\widetilde{t}) =&\frac{U}{LS}\lambda(\widetilde{t},\widetilde{t})\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}}(\widetilde{t},\widetilde{t})I+\mu(\widetilde{t},\widetilde{t})\varepsilon_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}(\widetilde{t}))\\ &-\frac{U}{LS}\int_0^{\widetilde{t}}\lambda_{\widetilde{\tau}}(\widetilde{t},\widetilde{\tau})\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}}(\widetilde{\tau})I+\mu_{\widetilde{\tau}}(\widetilde{t},\widetilde{\tau})\varepsilon_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}(\widetilde{\tau}))d\widetilde{\tau}.\end{aligned} \tag{A.14}$$

The scaled viscoelastic counterparts to the elastic Lamé coefficients are

$$\lambda(\widetilde{t},\widetilde{\tau})=\lambda_E e^{-\frac{\mu_E T}{\eta}(\widetilde{t}-\widetilde{\tau})}\qquad \mu(\widetilde{t},\widetilde{\tau})=\mu_E e^{-\frac{\mu_E T}{\eta}(\widetilde{t}-\widetilde{\tau})}. \tag{A.15}$$

When the differentiations with respect to $\widetilde{\tau}$ in Equation (A.14) are performed, we obtain

$$\begin{aligned}\widetilde{\sigma}(\widetilde{t}) =&\frac{U}{LS}\lambda(\widetilde{t},\widetilde{t})\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}}(\widetilde{t},\widetilde{t})I+\mu(\widetilde{t},\widetilde{t})\varepsilon_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}(\widetilde{t}))\\ &-\frac{U}{LS}\int_0^{\widetilde{t}}\frac{\mu_E T}{\eta}\lambda(\widetilde{t},\widetilde{\tau})\nabla_{\widetilde{\mathbf{x}}}\cdot\widetilde{\mathbf{u}}(\widetilde{\tau})I+\frac{\mu_E T}{\eta}\mu(\widetilde{t},\widetilde{\tau})\varepsilon_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{u}}(\widetilde{\tau}))d\widetilde{\tau}.\end{aligned} \tag{A.16}$$

Bibliography

[1] ABAQUS, Inc. *ABAQUS manuals, version 6.5*, 2004. [November 23 2006] http://www.abaqus.com.

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

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

[4] O. Axelsson. Stabilization of algebraic multilevel iteration methods; additive methods. *Numerical Algorithms*, 21:23 – 47, 1999.

[5] O. Axelsson and V.A. Barker. *Finite Element Solution of Boundary Value Problems. Theory and Computation*. Academic Press, Inc, 1984.

[6] O. Axelsson, V.A. Barker, M. Neytcheva, and B. Polman. Solving the Stokes problem on a massively parallel computer. *Mathematical Modelling and Analysis*, 4:7–27, 2000.

[7] O. Axelsson and R. Blaheta. Two simple derivations of universal bounds for the C.B.S inequality constant. *Applications of Mathematics*, 49(1):57 – 72, 2001.

[8] O. Axelsson, R. Blaheta, and M. Neytcheva. Preconditioning of boundary value problems using elementwise Schur complements. Technical Report 2006-048, Department of Information Technology, Uppsala University, November 2006.

[9] O. Axelsson and I. Gustafsson. Preconditioning and two-level mulitgrid methods of arbitrary degree of approximation. *Mathematics of Computation*, 40(161):219 – 242, 1983.

[10] O. Axelsson and S. Margenov. On multilevel preconditioners which are optimal with respect to both problem and discretization parameters. *Computational Methods in Applied Mathematics*, 3(1):6 – 22, 2003.

[11] O. Axelsson and M. Neytcheva. Algebraic multilevel iteration methods for Stieltjes matrices. *Numerical Linear Algebra with Applications*, 1:213 – 236, 1994.

[12] O. Axelsson and M. Neytcheva. Preconditioning methods for linear systems arising in constrained optimization problems. *Numerical Linear Algebra with Applications*, 10:3–31, 2003.

[13] O. Axelsson and M. Neytcheva. Eigenvalue estimates for preconditioned saddle point matrices. Technical Report 2004-019, Department of Information Technology, Uppsala University, 2004.

[14] O. Axelsson and A. Padiy. On a roubust and scalable linear elasticity solver based on a saddle point formulation. *International Journal for Numerical Methods in Engineering*, 44:801 – 818, 1999.

[15] O. Axelsson and A. Padiy. On the additive version of the algebraic multilevel iteration method for anisotropic elliptic problems. *SIAM Journal on Scientific Computing*, 20(5):1807 – 1830, 1999.

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

[17] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods II. *SIAM Journal on Numerical Analysis*, 27(6):1569 – 1590, 1990.

[18] O. Axelsson and P.S. Vassilevski. Variable-step multilevel preconditioning methods, I: Self-adjoint and positive definite elliptic problems. *Numerical Linear Algebra with Applications*, 1(1):75 – 101, 1994.

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

[20] E. Bängtsson and B. Lund. A comparison between two solution tecnhiques to solve the equations of linear isostasy. Submitted to *International Journal for Numerical Methods in Engineering*, Jan 2007.

[21] 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.

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

[23] E. Bängtsson and M. Neytcheva. An agglomerate multilevel preconditioner for linear isostasy saddle point problems. In I. Lirkov, S. Margenov, and J. Wasniewski, editors, *Proceedings of the 5th International Conference on Large-scale Scientific Computations 2005, LSSC 2005*, volume 3743 of *Lecture Notes in Computer Science*, pages 113–120. Berlin Springer, 2006.

[24] E. Bängtsson and M. Neytcheva. Preconditioning of nonsymmetric saddle point systems as arising in modelling of visco-elastic problems. Submitted to *Electronic Transactions on Numerical Analysis*, Nov 2006.

[25] E. Bängtsson and M. Neytcheva. Finite element block-factorized preconditioners. Technical Report 2007-008, Department of Information Technology, Uppsala University, 2007.

[26] R.E. Bank, T.F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numerische Mathematik*, 52:427 – 458, 1988.

[27] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Mathematica*, 14:1–137, 2005.

[28] E.F.F. Botta and F.W. 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.

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

[30] J.H. Bramble and J.E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation*, 50(181):1 – 17, 1988.

[31] 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(5):1570–1592, 2000.

[32] 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.

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

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

[35] V. Eijkhout and P.S. Vassilevski. The role of the strenghened Cauchy-Buniakowskii-Schwarz inequality in multilevel methods. *SIAM Review*, 33(3):405 – 419, 1991.

[36] R.P. Fedorenko. A relaxation method for solving elliptic difference equations. *Zh. vych. mat.*, 1(5):922 – 927, 1961.

[37] R.P. Fedorenko. The speed of convergence of one iterative process. *Zh. vych. mat.*, 4(3):559 – 564, 1964.

[38] A. George and J.W. Liu. *Computer solution of large sparse positive definite systems*. Prentice – Hall, Inc., 1981.

[39] G.H. Golub and C.F. van Loan. *Matrix computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[40] J.E. Jones and P. S. Vassilevski. AMGE based on element agglomeration. *SIAM Journal on Scientific Computing*, 23(1):109–133, 2001.

[41] 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.

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

[43] L. Kolotilina and A. Yeremin. Factorized sparse approximate inverse preconditionings. i. theory. *SIAM Journal on Matrix Analysis and Applications*, 14(1):45–58, 1993.

[44] J.K. Kraus. Algebraic multilevel preconditioning of finite element matrices using local Schur complements. *Numerical Linear Algebra with Applications*, 13(1):49–70, 2006.

[45] Z. Li, Y. Saad, and M. Sosonkina. pARMS: a parallel version of the algebraic recursive multilevel solver. *Numerical Linear Algebra with Applications*, 10:485 – 509, 2003.

[46] J.F. Maitre and F. Musy. The contraction number of a class of two-level methods; an exact evaluation for some finite element subspaces and model problems. In W. Hackbusch and U. Trottenberg, editors, *Multigrid methods*, volume 960 of *Lecture Notes in Mathematics*, 1982.

[47] S.D. Margenov and P.S. Vassilevski. Algebraic multilevel preconditioning of anisotropic elliptic problems. *SIAM Journal on Scientific Computing*, 15(5):1026 – 1037, 1994.

[48] J. Nedoma. *Numerical Modelling in Applied Geodynamics*. John Wiley & Sons, 1998.

[49] M. Neytcheva. *Arithmetic and Communication Complexity of Preconditioning Methods*. PhD thesis, Katholieke Universiteit Nijmegen, 1995.

[50] Y. Notay. Optimal V-cycle algebraic multilevel preconditioning. *Numerical Linear Algebra with Applications*, 5:441 – 459, 1998.

[51] Y. Notay. Using approximate inverses in algebraic multilevel methods. *Numerische Mathematic*, 80(3):397–417, 1998.

[52] Y. Notay. A robust algebraic multilevel preconditioner for non-symmetric M-matrices. *Numerical Linear Algebra with Applications*, 7(5):243 – 267, 2000.

[53] Y. Notay. Robust parameter-free algebraic multilevel preconditioning. *Numerical Linear Algebra with Applications*, 9:409 – 428, 2002.

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

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

[56] Y. Saad. Iterative methods for sparse linear systems. [27 March 2007] http://www-users.cs.umn.edu/~saad/books.html, January 2000. Second Edition with corrections.

[57] Y. Saad. Multilevel ILU with reorderings for diagonal dominance. *SIAM Journal on Scientific Computing*, 27(3):1032 – 1057, 2005.

[58] 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.

[59] B. Shen. *FRACOD^{2D} Two Dimensional Fracture Propagation Code, version 1.1, User's manual.* [March 27 2007] http://www.fracom.fi.

[60] V. Simoncini. Block triangular preconditioners for symmetric saddle-point problems. *Applied Numerical Mathematics*, 49(1):63 – 80, 2004.

[61] B. Smith, P. Bjørstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, 1996.

[62] K. Stüben. An introduction to algebraic multigrid. In U. Trottenberg, C. Oosterlee, and A. Schüller, editors, *Multigrid*, pages 413–479. Academic Press, 2001.

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

[64] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer-Verlag Berlin Heidelberg, 2005.

[65] P.S. Vassilevski. Hybrid V-cycle algebraic multilevel preconditioners. *Mathematics of Computation*, 58(198):489 – 512, 1992.

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

[67] P. Wu. Using commercial finite element packages for the study of earth deformations, sea levels and the state of stress. *Geophysics Journal International*, 158:401 – 408, 2004.

[68] H. Yserentant. Hierarchical bases of finite-element spaces in the discretization of nonsymmetric elliptic boundary value problems. *Computing*, 35:39 – 49, 1985.

[69] H. Yserentant. On the multi-level splitting of finite element spaces for indefinite elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 23(3):581 – 595, 1986.

[70] H. Yserentant. On the multi-level splitting of finite element squares. *Numerische Mathematik*, 49:379 – 412, 1986.

[71] H. Yserentant. Two preconditioners based on the multi-level splitting of finite element spaces. *Numerische Mathematik*, 58:163 – 184, 1990.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 296

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title "Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology".)

Distribution: publications.uu.se
urn:nbn:se:uu:diva-7828

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2007