Electronic Transactions on Numerical Analysis.
Volume 29, pp. 193-211, 2008.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

PRECONDITIONING OF NONSYMMETRIC SADDLE POINT SYSTEMS AS ARISING IN MODELLING OF VISCOELASTIC PROBLEMS*

MAYA NEYTCHEVA† AND ERIK BÄNGTSSON‡

Abstract. In this paper we consider numerical simulations of the so-called glacial rebound phenomenon and the use of efficient preconditioned iterative solution methods in that context. The problem originates from modeling the response of the solid earth to large scale glacial advance and recession which may have provoked very large earthquakes in Northern Scandinavia. The need for such numerical simulations is due to ongoing investigations on safety assessment of radioactive waste repositories. The continuous setting of the problem is to solve an integro-differential equation in a large time-space domain. This problem is then discretized using a finite element method in space and a suitable discretization in time, and gives rise to the solution of a large number of linear systems with nonsymmetric matrices of saddle point form. We outline the properties of the corresponding linear systems of equations, discuss possible preconditioning strategies, and present some numerical experiments.

Key words. viscoelasticity, (in)compressibility, nonsymmetric saddle-point system, preconditioning, Schur complement approximation, algebraic multilevel techniques

AMS subject classifications. 65F10, 74D05, 45D05

1. Introduction. The behaviour of most materials used in structural engineering is described by Hooke's law of linear elasticity, namely, for small strains, stress σ is proportional to strain ε, $\sigma = C\varepsilon$. In reality the behaviour of virtually all materials deviates from Hooke's law in various ways, exhibiting both elastic and viscous characteristics. A vast variety of materials, such as human tissues, wood, metals at high temperature, synthetic polymers, rocks, ice, concrete, display viscoelastic effects of different magnitude. Since even a small viscoelastic response can have a significant impact on the behaviour of the material, the analysis of material response to load must in many cases incorporate viscoelasticity. Viscoelastic materials obey a constitutive relation between stress and strain, which depends on time.

Knowledge of the viscoelastic response of a material can be based on measurements. The timescale, however, varies widely from 10^{-10} sec to 10^5 years or more. While the short timescales are practically inaccessible to experiments and exact measurements, observations over long times are limited by patience and lifetime. Furthermore, complicated viscoelastic flow phenomena even in relatively simple geometries are mostly not tractable by analytical methods and can be predicted by numerical methods only. Therefore numerical simulations become a necessity and in some cases the ultimate way to go.

The target physical phenomenon in this paper is glacial advance and recession, and the resulting crustal stress state. Numerical simulations are performed in order to check the hypothesis that the stress induced by postglacial readjustment processes has triggered large earthquakes (of magnitude 8) in some regions in the Northern Hemisphere. The analysis of the outcome, in turn would influence the positioning of nuclear waste repositories in Scandinavia, the safety of which has to be guaranteed for a time period of about 200 000 years.

For the numerical simulations we use the so-called *isostatic model*, based on the concept that the elevation of earth's surface seeks a balance between the weight of lithospheric rocks and the buoyancy of asthenospheric "fluid" (nearly-molten rock). The model expresses the

*Received November 2, 2006. Accepted for publication March 31, 2008. Published online on October 6, 2008. Recommended by M. Hochbruck.

†Uppsala University, Department of Information Technology, Box 337, SE-751 05 Uppsala, Sweden (Maya.Neytcheva@it.uu.se).

‡Uppsala University, Department of Information Technology, Box 337, SE-751 05 Uppsala, Sweden (Erik.Bangtsson@it.uu.se). Presently at RaySearch Laboratories AB, Sveavägen 25, 11135 Stockholm, Sweden.

ETNA
Kent State University
etna@mcs.kent.edu

geophysical problem in terms of a system of partial differential equations which describe the equilibrium state of a pre-stressed viscoelastic material body, subject to surface and body forces. It includes a first-order term representing the so-called advection of pre-stress, the incorporation of which has proven to be crucial for the successful modeling of the underlying processes.

The computationally heaviest part of many numerical simulations of complex problems, as in viscoelasticity, is the repeated solution of large linear systems of equations, solved during each time step. Currently, direct solution methods are mostly used, which are known to have very high demands for computer resources and could therefore severely limit the size and complexity of the discrete models, in particular in 3D. With this work we show that a robust and efficient preconditioned iterative solution method is a viable and competitive alternative to the direct solution methods.

The paper is organized as follows. We describe the problem in Section 2. The corresponding variational formulation and the finite element space discretization are presented in Section 3, and details on the discretization in time are given in Section 4. Section 5 contains a description of the preconditioner used in the numerical simulations, which are illustrated in Section 6. In contrast to to [12], in this work, as a part of the preconditioning strategy, we use (nearly) optimal order algebraic multilevel iteration type methods, applied in a nonsymmetric setting with discontinuous problem coefficients. In addition, we apply the proposed solution method to the viscoelastic problem in its full complexity.

2. Problem description. Consider a post-glacial rebound model, based on the isostasy concept, namely, that the elevation of earth's surface (over tens of millions of years) seeks a balance between the weight of lithospheric rocks and the buoyancy of the more viscous asthenosphere. The governing equation describing the equilibrium state of a pre-stressed viscoelastic non-selfgravitating material body, subject to surface and body forces; cf. [15] is of the form

$$-\underbrace{\underline{\nabla}\cdot\sigma}_{(A)} - \underbrace{\nabla(\mathbf{u}\cdot\nabla p)}_{(B)} + \underbrace{\nabla p\nabla\cdot\mathbf{u}}_{(C)} = \mathbf{0} \quad \text{in } \Omega\subset\mathbb{R}^d, d=2,3. \tag{2.1}$$

Here σ, $\mathbf{u} = [u_i]_{i=1}^d$, and p, are the stress tensor, the displacement vector and the hydrostatic pressure, respectively. The pressure is of the form $p = \rho g\,\mathbf{e}$, where ρ and g are density and gravitational acceleration. The term (A) in (2.1) describes the force from spatial gradients in stress. The term (B) represents the so-called *advection of pre-stress* and describes how the hydrostatic background (initial) stress is carried by the moving material. Incorporating (B) has proven to be crucial for the successful modeling of the underlying processes; see [15]. We point out that pre-stress is not included in the available general purpose finite element (FE) packages. Further we assume small deformations, i.e. strain ε and displacements $\mathbf{u}$ are related as $\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$, $1 \le i, j \le d$.

Equation (2.1) can be solved with two different constitutive relations which allow simulations of purely elastic as well as of viscoelastic medium, as is done in various geophysical studies.

Case 1: Stress and strain are related as for linear elastic anisotropic materials, $\sigma(\mathbf{x}) = C(\mathbf{x})\varepsilon(\mathbf{x})$, where ε is the strain tensor and $C(\mathbf{x})$ is the elasticity tensor.

Case 2: The material is assumed to be viscoelastic, obeying a constitutive relation of the

ETNA
Kent State University
etna@mcs.kent.edu

form

$$\begin{aligned}
&\sigma(\mathbf{x},t) = C(\mathbf{x},0)\varepsilon(\mathbf{x}) - \int_0^t \frac{\partial C(t-\tau)}{\partial \tau}\varepsilon(\mathbf{x},\tau)\,d\tau \\
&\text{or with other notations} \\
&\sigma(\mathbf{x},t) = \sigma_E(\mathbf{x},t) - \int_0^t \sigma_V(\mathbf{x},t,\tau)\,d\tau,
\end{aligned} \tag{2.2}$$

which is referred to as Hooke's law with memory; see, for instance [18]. The tensors $C(\mathbf{x},0)$ (here $C(\mathbf{x},0) = C(t-\tau)$ at $\tau = t$) and $\widetilde{C}(\mathbf{x},t-\tau) = \frac{\partial C(t-\tau)}{\partial \tau}$ measure 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.

The memory term in equation (2.2) arises from the fact that a (previously) loaded viscoelastic solid attempts to relax its stress state through *creep* and *relaxation* processes. This behaviour is described by a relaxation function, which depends on the material model used for the solid (Maxwell, Kelvin, Burger, etc.). For different viscoelastic models of earth's lithosphere we refer, for example, to [24].

We consider a slightly more general form of the isostatic moment balance equation (2.1), namely,

$$-\nabla\cdot\sigma(\mathbf{x},t) - \nabla(\mathbf{b}(\mathbf{x})\cdot\mathbf{u}(\mathbf{x},t)) + (\nabla\cdot\mathbf{u}(\mathbf{x},t))\,\mathbf{c}(\mathbf{x}) = \mathbf{f}(\mathbf{x},t), \tag{2.3}$$

where $\mathbf{b}$ and $\mathbf{c}$ are general vector fields and the function $\mathbf{f}$ represents body force. In what follows the dependence on the space variable $\mathbf{x}$ is omitted, unless it is of specific importance in some particular relations.

We note here, that due to the form of the pre-stress advection term, the glacial rebound problem does not reduce to the well-known Navier-Stokes problem; see [12] for more details.

In the derivations we utilize the following standard relations between stress, strain and displacements,

$$\begin{aligned}
\sigma_E(t) &= \lambda_E\nabla\cdot\mathbf{u}(t) + \mu_E\varepsilon(\mathbf{u}(t)),\\
\sigma_V(t,\tau) &= \partial_\tau\lambda(t,\tau)\nabla\cdot\mathbf{u}(\tau) + \partial_\tau\mu(t,\tau)\varepsilon(\mathbf{u}(\tau)),
\end{aligned} \tag{2.4}$$

as well as the assumption that the stress relaxation functions obey the so-called Maxwell model, i.e., the time-dependence in the stress field is due to time-dependent material coefficients only and is of the form

$$\begin{aligned}
\mu(t,\tau) &= \mu_E e^{-\alpha_0(t-\tau)} = \mu_E\chi(\alpha_0,t,\tau),\\
\lambda(t,\tau) &= \lambda_E e^{-\alpha_0(t-\tau)} = \lambda_E\chi(\alpha_0,t,\tau).
\end{aligned} \tag{2.5}$$

Here, 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.

3. Variational form and space discretization. Let $\mathcal{J} = [0,T]$, $T > 0$ be a given time interval and Ω be a time-independent bounded domain in $\mathbb{R}^d$, $d = 2,3$, with a polygonal (respectively polyhedral) boundary $\partial\Omega$. We assume that the viscoelastic compressible or incompressible body occupies Ω, it is acted upon by a body force $\mathbf{f}(\mathbf{x},t)$, it is rigidly fixed in space and time on a part Γ_D of $\partial\Omega$ with a positive measure and some (time-dependent) surface traction ℓ is acting on the body along another part Γ_N of $\partial\Omega$. We are interested in

ETNA
Kent State University
etna@mcs.kent.edu

obtaining the solution of the problem (2.3) in terms of displacements $\mathbf{u}(\mathbf{x}, t)$, $\mathbf{x} \in \Omega$, and $t \in \mathcal{J}$.

We point out that in the isostasy model used by geophysicists, earth is considered fully incompressible. Still, in certain cases it turns out to be of practical interest to study the behaviour of the crust as a compressible body. Therefore we consider both compressible and incompressible materials.

To define the corresponding weak formulation of (2.3) we introduce in the usual way the Sobolev space $V = \left(H_0^1(\Omega)\right)^d$, $d = 2, 3$, equipped with the norm $\|\cdot\|_{\mathbf{V}} = \|\cdot\|_1$.

The weak form of equation (2.3) reads as follows:
Find $\mathbf{u} \in \mathcal{V}(V; \mathcal{J})$ such that

$$-\int_\Omega \nabla \cdot \sigma(t) \cdot \mathbf{v}\, d\Omega - \int_\Omega \left(\nabla(\mathbf{b}\cdot\mathbf{u}) \cdot \mathbf{v} - (\nabla \cdot \mathbf{u})\mathbf{c} \cdot \mathbf{v}\right)\, d\Omega = \int_\Omega \mathbf{f}(t) \cdot \mathbf{v}\, d\Omega, \tag{3.1}$$

holds for all $\mathbf{v} \in V$. After integration by parts, equation (3.1) takes the form

$$\begin{aligned} &\int_\Omega \sigma(t) : \varepsilon(\mathbf{v})\, d\Omega - \int_\Omega (\nabla(\mathbf{b}\cdot\mathbf{u}) \cdot \mathbf{v} - (\nabla \cdot \mathbf{u})\mathbf{c} \cdot \mathbf{v})\, d\Omega = \\ &\qquad\qquad \int_\Omega \mathbf{f}(t) \cdot \mathbf{v}\, d\Omega + \int_\Gamma \mathbf{n} \cdot \sigma(t) \cdot \mathbf{v}\, d\Gamma. \end{aligned} \tag{3.2}$$

Next, incorporating the viscoelastic constitutive relations (2.2) and utilizing that

$$\int_\Gamma \int_0^t \mathbf{n}\cdot\sigma_V(t,\tau)\cdot\mathbf{v}\, d\tau = \int_\Gamma \int_0^t \alpha_0\chi(\alpha_0, t, \tau)\mathbf{n}\cdot\sigma_E(\tau)\cdot\mathbf{v}\, d\tau = \int_\Gamma \int_0^t \alpha_0\chi(\alpha_0, t, \tau)\ell(\tau)\cdot\mathbf{v}\, d\tau,$$

equation (3.2) takes the following form,

$$\begin{aligned} &\int_\Omega \sigma_E(t) : \varepsilon(\mathbf{v})\, d\Omega - \int_\Omega \left(\nabla(\mathbf{b}\cdot\mathbf{u}) \cdot \mathbf{v} - (\nabla \cdot \mathbf{u})\mathbf{c} \cdot \mathbf{v}\right)\, d\Omega \\ &- \int_\Omega \int_0^t \sigma_V(t,\tau) : \varepsilon(\mathbf{v})\, d\tau\, d\Omega \;= \int_\Omega \mathbf{f}(t) \cdot \mathbf{v}\, d\Omega \\ &+ \int_\Gamma \ell(t) \cdot \mathbf{v}\, d\Gamma - \int_\Gamma \int_0^t \alpha_0\chi(\alpha_0, t, \tau)\ell(\tau) \cdot \mathbf{v}\, d\tau\, d\Gamma. \end{aligned} \tag{3.3}$$

REMARK 3.1. We point out, that the time and space integrals are interchangeable only as long as the material parameters E, ν, and η are constant over the spatial domain of integration. Therefore, to handle inhomogeneous material parameters, the space integrals can be written as a sum of integrals over, say, m disjoint subdomains, each with its own set of material parameters $E_i, \nu_i, \eta, \lambda_{E,i}, \mu_{E,i}, \alpha_{0,i}$ for $i = 1, \ldots, m$. Then, for example the term $\int_\Omega \int_0^t \sigma_V(t,\tau) : \varepsilon(v) d\tau d\Omega$ from equation (3.3) can be handled as

$$\int_\Omega \int_0^t \sigma_V(t,\tau) : \varepsilon(v)\, d\tau\, d\Omega = \sum_{i=1}^m \int_0^t \int_{\Omega_i} \sigma_V(t,\tau) : \varepsilon(v)\, d\Omega\, d\tau.$$

With no loss of generality and for notational simplicity, all derivations in this section are presented for homogeneous isotropic material media.

Next, in order to enable simulations with purely incompressible solids (in which case $\nu = 0.5$ and the coefficient λ becomes undefined), we introduce the (scaled) kinematic pressure $p(\mathbf{x}, t)$,

$$\mu_E\, p(\mathbf{x}, t) = \lambda_E \nabla \cdot \mathbf{u}(\mathbf{x}, t), \tag{3.4}$$

and rewrite the original problem as a coupled system of equations – (2.3) and (3.4). We consider a suitable function space for the pressure variable,

$$P = \{p \in \mathcal{L}^2(\Omega) : \int_\Omega p\, d\Omega = 0\}.$$

Then, the corresponding weak formulation reads as follows.

For each $t \in \mathcal{J} = [0, T]$, find $\mathbf{u}(\mathbf{x}, t) \in \mathcal{V}(V; \mathcal{J})$ and $p(\mathbf{x}, t) \in \mathcal{P}(P; \mathcal{J})$ such that for any $\mathbf{v}(\mathbf{x}) \in V$ and $q(\mathbf{x}) \in P$ there holds

$$\begin{aligned} a(\mathbf{u}(t), v) - \int_0^t \widetilde{a}(\mathbf{u}(\tau), \mathbf{v}; t)\, d\tau \;+\; b(\mathbf{v}, p(t)) - \int_0^t \widetilde{b}(\mathbf{v}, p(\tau); t)\, d\tau \;&=\; \widetilde{g}(\mathbf{v}; t), \\ b(\mathbf{u}(t), q) - \int_0^t \widetilde{b}(\mathbf{u}(\tau), q; t) d\tau \;-\; c(p(t), q) + \int_0^t \widetilde{c}(p(t), q; t)\, d\tau \;&=\; 0, \end{aligned} \tag{3.5}$$

where $\widetilde{g}(\mathbf{v}; t) = g(\mathbf{v}; t) + \ell(\mathbf{v}; t) - \int_0^t \widetilde{\ell}(\mathbf{v}; t, \tau)\, d\tau$. The bilinear forms are specified in the following.

$$\begin{aligned} a(\mathbf{u}(t), \mathbf{v}) &= \widehat{a}(\mathbf{u}(t), \mathbf{v}) + \bar{a}(\mathbf{u}(t), \mathbf{v}), \\ \widehat{a}(\mathbf{u}(t), \mathbf{v}) &= \int_\Omega 2\mu_E \varepsilon(\mathbf{u}(t)) : \varepsilon(\mathbf{v})\, d\Omega, \\ \bar{a}(\mathbf{u}(t), \mathbf{v}) &= \int_\Omega -\nabla(\mathbf{b} \cdot \mathbf{u}(t)) \cdot \mathbf{v} + (\nabla \cdot \mathbf{u}(t))\mathbf{c} \cdot \mathbf{v}\, d\Omega, \\ \widetilde{a}(\mathbf{u}(\tau), \mathbf{v}; t, \tau) &= \int_\Omega 2\alpha_0 \chi(\alpha_0, t, \tau)\mu_E\, \varepsilon(\mathbf{u}(\tau)) : \varepsilon(\mathbf{v})\, d\Omega, \\ b(\mathbf{v}, p(t)) &= \int_\Omega \mu_E p(t) \nabla \cdot \mathbf{v}\, d\Omega, \\ \widetilde{b}(\mathbf{v}, p(\tau); t, \tau) &= \int_\Omega \alpha_0 \chi(\alpha_0, t, \tau)\mu_E p(\tau) \nabla \cdot \mathbf{v}\, d\Omega, \\ c(p(t), q) &= \int_\Omega \frac{\mu_E^2}{\lambda_E} p(t) q\, d\Omega, \\ \widetilde{c}(p(t), q; t, \tau) &= \int_\Omega \alpha_0 \chi(\alpha_0, t, \tau) \frac{\mu_E^2}{\lambda_E} p(t) q\, d\Omega, \\ \ell(\mathbf{v}; t) &= \int_\Gamma \ell(t) \cdot \mathbf{v}\, d\Gamma, \\ g(\mathbf{v}; t) &= \int_\Omega \mathbf{f}(t) \cdot \mathbf{v}\, d\Omega, \\ \widetilde{\ell}(\mathbf{v}; t, \tau) &= \int_\Gamma \alpha_{0,m} \chi(\alpha_0, t, \tau) \ell(\tau) \cdot \mathbf{v}\, d\Gamma. \end{aligned} \tag{3.6}$$

The particular choice of the function spaces for $\mathbf{u}$, $\mathbf{v}$, p, and q is specified in Section 6.

REMARK 3.2. As we have shown in [12], due to the presence of the first order terms in $\bar{a}(\mathbf{u},\mathbf{v})$, the bilinear form $a(\mathbf{u},\mathbf{v})$ is in general not coercive. However, in the numerical tests and for the particular form of the vectors $\mathbf{b}=\mathbf{c}=[0,-\rho g]^T$ as in the considered isostatic glacial rebound problem, we have never observed any difficulties, which could indicate nonuniqueness of the solution. Without the time-dependent terms, the saddle-point matrix, corresponding to the discrete form of equation (3.5) is easily recognized. We show next that the matrix structure remain the same in the viscoelastic case also.

4. Time discretization and solution algorithm. To handle the viscoelastic terms, we apply some numerical quadrature, which implies that $\mathcal{J}$ is discretized into a number of intervals, such that $t_0=0, t_1=t_0+\Delta t_0,\dots,t_{j+1}=t_j+\Delta t_j,\dots,t_n=T$, and the jth interval is then of length $\Delta t_j=t_j-t_{j-1}$. In general, for any numerical integration, when computing $\mathbf{u}_j$ we would need to store the complete solution history, i.e., all solutions $\mathbf{u}_s, s=0,\dots,j-1$, which is a very memory demanding task, in particular for large time intervals, as for the target problem here. Fortunately, for relaxation functions as in the Maxwell model given by (2.5), the computations simplify significantly. The corresponding derivations can be found elsewhere in the related literature, for example, in [21]. For completeness of the presentation, we illustrate here the essence of the simplification on one of the integral terms in equation (3.5).

$$
\begin{aligned}
\mathcal{I}_0^j &= \int_0^{t_j} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t_j,\tau)\,d\tau \\
&= \int_0^{t_{j-1}} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t_j,\tau)\,d\tau + \int_{t_{j-1}}^{t_j} \widetilde{a}(\mathbf{u}(\tau),\mathbf{v};t_j,\tau)\,d\tau \\
&= \widetilde{\mathcal{I}}_0^{j-1} + \mathcal{I}_{j-1}^j .
\end{aligned}
\tag{4.1}
$$

Then, using (3.6) and the particular form of the Lamé coefficients from (2.5), we obtain

$$
\begin{aligned}
\widetilde{\mathcal{I}}_0^{j-1} &= \int_{t_{j-1}}^{t_j} \int_\Omega 2\alpha_0\chi(\alpha_0,t_j,\tau)\mu_E\,\varepsilon(\mathbf{u}(\tau)):\varepsilon(\mathbf{v})\,d\Omega\,d\tau \\
&= \int_{t_{j-1}}^{t_j} \int_\Omega e^{-\alpha(t_j-t_{j-1})}\left(2\alpha_0\chi(\alpha_0,t_{j-1},\tau)\mu_E\,\varepsilon(\mathbf{u}(\tau)):\varepsilon(\mathbf{v})\right)\,d\Omega\,d\tau \\
&= e^{-\alpha\Delta t_j}\mathcal{I}_0^{j-1},
\end{aligned}
\tag{4.2}
$$

provided that the material coefficients are not space-dependent. The latter shows that $\widetilde{\mathcal{I}}_0^{j-1}$ can be computed by scaling $\mathcal{I}_0^{j-1}$ (globally or per subdomain, for homogeneous or inhomogeneous material coefficients, respectively), which has been already computed at the previous time-instance.

We choose next to approximate $\mathcal{I}_{j-1}^j$ by the trapezoidal rule, that is,

$$
\int_{t_{j-1}}^{t_j} f(t) \approx \frac{\Delta t_j}{2}\left(f(t_j)+f(t_{j-1})\right). \tag{4.3}
$$

ETNA
Kent State University
etna@mcs.kent.edu

Then the first equation in (3.5) becomes

$$
\begin{aligned}
&a(u_j,v) - \frac{\Delta t_j}{2}\widetilde{a}(u_j), v; t_j, t_j) + b(v,p_j) - \frac{\Delta t_j}{2}\widetilde{b}(v,p_j;t_j,t_j) = g(v;t_j)\\
&+ \ell(v;t_j) - \frac{\Delta t_j}{2}\widetilde{\ell}(v;t_j,t_j) - \frac{\Delta t_j}{2}\widetilde{\ell}(v;t_j,t_{j-1}) - \int_0^{t_{j-1}} \widetilde{\ell}(v;t_j,\tau)\,d\tau\\
&+ \frac{\Delta t_j}{2}\widetilde{a}(u_{j-1},v;t_j,t_{j-1}) + \int_0^{t_{j-1}} \widetilde{a}(u(\tau),v;t_j,\tau)\,d\tau\\
&+ \frac{\Delta t_j}{2}\widetilde{b}(v,p_{j-1};t_j,t_{j-1}) + \int_0^{t_{j-1}} \widetilde{b}(v,p(\tau);t_j,\tau)\,d\tau.
\end{aligned}
\tag{4.4}
$$

Similar expressions are derived for the second equation in (3.5).

We next perform a finite element space discretization of Ω, namely, consider a discretized domain Ω_h and some finite dimensional subspaces $V_h \subset V$ and $P_h \subset P$. We obtain then a matrix-vector form of the problem: at time t_j, find the displacements $\mathbf{u}_j$ and the pressure $\mathbf{p}_j$ by solving

$$
\mathcal{A}_j \begin{bmatrix} \mathbf{u}_j \\ \mathbf{p}_j \end{bmatrix} = \begin{bmatrix} \mathbf{r}_j^{(1)} \\ \mathbf{r}_j^{(2)} \end{bmatrix}, \tag{4.5}
$$

where

$$
\mathcal{A}_j = \mathcal{A} - \frac{\Delta t_j}{2}\mathcal{A}_0,\ \mathcal{A} = \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix}, \text{ and } \mathcal{A}_0 = \begin{bmatrix} M_0 & B_0^T \\ B_0 & -C_0 \end{bmatrix}.
$$

The detailed forms of $\mathbf{r}_j^{(1)}$ and $\mathbf{r}_j^{(2)}$ are given later in this section and in the Appendix. Here the matrix blocks M_0, B_0, C_0, correspond to the bilinear forms $\widetilde{a}$, $\widetilde{b}$, $\widetilde{c}$, in (3.6), evaluated at $\tau = t$ and therefore do not explicitly depend on t. The blocks in $\mathcal{A}_0$ depend on α_0, which itself depends on the problem coefficients. These coefficients might be discontinuous in space.

Using the bilinear forms in (3.6), the matrices on the left-hand side of equation (4.5) are computed as

$$
\begin{aligned}
\left(M - \tfrac{\Delta t_j}{2}M_0\right)_{i,k} &= \int_\Omega \mu_E\left(1 - \tfrac{\alpha_0\Delta t_j}{2}\right)\varepsilon(\mathbf{v}_i) : \varepsilon(\mathbf{v}_k)\,d\Omega - \\
&\quad \int_\Omega \nabla(\mathbf{b}\cdot\mathbf{v}_i)\cdot\mathbf{v}_k - (\nabla\cdot\mathbf{v}_i)(\mathbf{c}\cdot\mathbf{v}_k)\,d\Omega,\\
\left(B - \tfrac{\Delta t_j}{2}B_0\right)_{i,k} &= \int_\Omega \mu_E\left(1 - \tfrac{\alpha_0\Delta t_j}{2}\right)(\nabla\cdot\mathbf{v}_i)q_k\,d\Omega,\\
\left(C - \tfrac{\Delta t_j}{2}C_0\right)_{i,k} &= \int_\Omega \left(1 - \tfrac{\alpha_0\Delta t_j}{2}\right)\tfrac{\mu_E^2}{\eta}q_i q_k\,d\Omega.
\end{aligned}
\tag{4.6}
$$

One obtains similar expressions for the matrices involved in the computation of the matrices on the right-hand side of (4.5).

Solution algorithm. Using the matrix-vector notations, the solution of equation (4.5) through time can be formulated as the following time-stepping procedure:

Initialize the vectors $F_0 = 0$, $L_0 = 0$, $U_0 = 0$, $P_0 = 0$ and assemble the matrices M, B, C, M_0, B_0, C_0.
for $j = 1, 2, \cdots$, **do**
  Choose a time-step Δt_j.
  Assemble the matrices $\widetilde{M}_{j-1}^j, \widetilde{B}_{j-1}^j, \widetilde{C}_{j-1}^j$.
  Form

$$\mathcal{A}_j \equiv \begin{bmatrix} \mathcal{M}_j & \mathcal{B}_j^T \\ \mathcal{B}_j & \mathcal{C}_j \end{bmatrix} \equiv \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix} - \frac{\Delta t_j}{2} \begin{bmatrix} M_0 & B_0^T \\ B_0 & -C_0 \end{bmatrix}.$$

  Compute (for each subdomain $\Omega_h^i, i = 1, \ldots, m$)

$$\widetilde{I}_{\ell,0}^{j-1} = \sum_{i=1}^{m} \chi(\alpha_{0,i}, t_j, t_{j-1})\, I_{\ell,0,i}^{j-1},$$

$$\widetilde{I}_{u,0}^{j-1} = \sum_{i=1}^{m} \chi(\alpha_{0,i}, t_j, t_{j-1})\, I_{u,0,i}^{j-1},$$

$$\widetilde{I}_{p,0}^{j-1} = \sum_{i=1}^{m} \chi(\alpha_{0,m}, t_j, t_{j-1})\, I_{p,0,i}^{j-1}.$$

  Compute the contributions from the memory terms

$$\begin{bmatrix} \widetilde{I}_u \\ \widetilde{I}_p \end{bmatrix} = \begin{bmatrix} \widetilde{I}_{u,0}^{j-1} \\ \widetilde{I}_{p,0}^{j-1} \end{bmatrix} + \begin{bmatrix} \widetilde{M}_{j-1}^j & (\widetilde{B}_{j-1}^j)^T \\ \widetilde{B}_{j-1}^j & -\widetilde{C}_{j-1}^j \end{bmatrix} \begin{bmatrix} \mathbf{u}_{j-1} \\ \mathbf{p}_{j-1} \end{bmatrix}.$$

  Compute the loads and body forces at time t_j – $\mathbf{l}j$, $\widetilde{\mathbf{I}}_j^j$, $\widetilde{\mathbf{I}}_j^{j-1}$, $\mathbf{f}_j$, and
  $I_{\ell,0}^j = \widetilde{I}_{\ell,0}^{j-1} + \widetilde{\mathbf{I}}_j^j + \widetilde{\mathbf{I}}_{j-1}^j$,
  $\mathbf{r}_j^{(1)} = \mathbf{l}_j + \mathbf{f}_j + \widetilde{I}_u - I_{\ell,0}^j$,
  $\mathbf{r}_j^{(2)} = \widetilde{I}_p$.
  Solve

$$\mathcal{A}_j \begin{bmatrix} \mathbf{u}_j \\ \mathbf{p}_j \end{bmatrix} = \begin{bmatrix} \mathbf{r}_j^{(1)} \\ \mathbf{r}_j^{(2)} \end{bmatrix}.$$

  Compute

$$\begin{bmatrix} I_{u,0}^j \\ I_{p,0}^j \end{bmatrix} = \begin{bmatrix} \widetilde{I}_u \\ \widetilde{I}_p \end{bmatrix} + \frac{\Delta t_j}{2} \begin{bmatrix} M_0 & B_0^T \\ B_0 & -C_0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_j \\ \mathbf{p}_j \end{bmatrix}.$$

end for

REMARK 4.1. Clearly, the above algorithm depends on the time discretization. We have chosen here the trapezoid rule, which is implicit, stable, and allows for relatively large steps in time, of course related to the size of the space discretization in order to have a balanced total discretization error. In the present context, we see also from (4.5) that Δt_j should not be very large in order to have $1 - \frac{\alpha_0 \Delta t_j}{2} > 0$ satisfied, so that the contribution $\frac{\alpha_0 \Delta t_j}{2} \mathcal{A}_0$ does not dominate that of $\mathcal{A}$. In [20], under certain assumptions, an estimate of the total discretization error is derived, stating that if $\mathbf{u} \in L_\infty(V \cup (H^{1+\alpha}(\Omega))^n; \mathcal{J})$ then for all $t_j \in \mathcal{J}$ there exists a constant C depending on $u(x,t)$ and $u_h(x,t)$ but independent of h and Δt_j, such that

$$\|u(x, t_j) - (u_h(x))_j\|_{1,\Omega} \le C(h^\alpha + \Delta t_j^2),$$

ETNA
Kent State University
etna@mcs.kent.edu

which is of suboptimal order with respect to the space discretization.

From a theoretical point of view, probably the most elegant approach is to consider a time-space FEM discretization of the viscoelastic problem, as is done, for example, in [23] and some other works by the same group of authors. In their analysis they suggest advanced schemes using Discontinuous Galerkin FEM in time and nonmatching spacial discretization meshes from one time level to the next. They also derive optimal order error estimates. However, to our knowledge, so far the latter method has not been fully implemented and tested numerically for viscoelastic problems. We believe that for such very large scale space and time domains as in the target application, the time-space FEM method is not a feasible alternative to the more standard method we use in this work.

5. Preconditioning technique. At each time t_j we have to solve a linear system with the matrix $\mathcal{A}_j$ given in (4.5). We assume that Δt_j is small enough so that it suffices to describe a preconditioner for a general linear system of equations which admits the saddle point form

$$\mathcal{A} = \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix}, \tag{5.1}$$

where $M \in \mathbb{R}^{N_u \times N_u}$ is a sparse nonsymmetric matrix with, in general, indefinite symmetric part, and $C^{N_p \times N_p}$ is positive semidefinite.

There exists a vast amount of literature on how to precondition and solve problems with saddle point matrices. As recent sources which survey such methods we refer, for example, to [9], [5] and [14], and the references therein.

As a general outcome, the major milestones to follow when constructing a preconditioner for a saddle point matrix can be summarized as follows.

(i) The best known preconditioners for saddle point problems utilize the available two-by-two block structure of the system matrix. These can be of block lower-triangular, block upper-triangular or block-factorized form.

(ii) For any of the block preconditioners, listed in (i), in order to ensure favorable spectral properties of the preconditioned matrix one needs to construct a good approximation of the (negative) Schur complements matrix $S_A = C + BM^{-1}B^T$ and to solve accurately systems with the pivot block M, respectively with a good approximation of that block.

Here we choose to solve the algebraic problem in equation (5.1) with an iterative solution method preconditioned by a block lower-triangular matrix

$$\mathcal{D} = \begin{bmatrix} D_1 & 0 \\ B & -D_2 \end{bmatrix}. \tag{5.2}$$

As already mentioned, this preconditioner can be very efficient, provided that the blocks D_1 and D_2 are properly chosen. Indeed, from the expression

$$\begin{aligned} \mathcal{D}^{-1}\mathcal{A} &= \begin{bmatrix} D_1^{-1} & 0 \\ D_2^{-1}BD_1^{-1} & -D_2^{-1} \end{bmatrix} \begin{bmatrix} M & B^T \\ B & -C \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} D_1^{-1}M - I & D_1^{-1}B^T \\ D_2^{-1}B(D_1^{-1}M - I) & D_2^{-1}(C + BD_1^{-1}B^T) - I \end{bmatrix}. \end{aligned} \tag{5.3}$$

we see that whenever D_1 is a good approximation of M, and D_2 approximates well the approximate negative Schur complement of $\mathcal{A}$, $S = C + BD_1^{-1}B^T$, the eigenvalues of $\mathcal{D}^{-1}\mathcal{A}$ are clustered around unity.

Thus, the two most important means to steer the quality of the preconditioner $\mathcal{D}$ are the choices of D_1 and D_2, and, related to that, how to solve systems of equations with them.

REMARK 5.1. The preconditioner $\mathcal{D}$ is strongly nonsymmetric and one can possibly argue that since the nonsymmetricity in $\mathcal{A}$ is contained only in the block M a more balanced block-factorized form should be used instead. Indeed, the block-lower triangular matrix $\mathcal{D}$ is only one of the possible choices of a preconditioner for $\mathcal{A}$. However, the task to approximate well the top-left block and the corresponding Schur complement matrix remains of utmost importance also for preconditioners with another block-structure. For a detailed discussion on this and other preconditioners for symmetric saddle point matrices, we refer for instance to [5].

As is seen from (5.3), the block D_2, should be chosen as a good approximation of $S_{\mathcal{A}} = C + BM^{-1}B^T$ or rather of $S = C + BD_1^{-1}B^T$. Clearly, except for some special cases, to explicitly form a Schur complement matrix is almost as expensive as it is to solve a system with it. Furthermore, the Schur complement is in general dense even when the matrix is sparse. Good approximations for the Schur complement matrix are known in some cases, for example, for the Stokes problem one can replace $BM^{-1}B^T$ by the pressure mass-matrix. However, in our case this approach would lead to a symmetric D_2, while the true $S_{\mathcal{A}}$ is nonsymmetric.

We use here a simple technique, based on the finite element framework, which enables us to compute D_2 such that it is (i) cheaply constructed, (ii) sparse, (iii) inherits the symmetricity or nonsymmetricity of $\mathcal{A}$, and (iv) is spectrally equivalent to the true Schur complement $S_{\mathcal{A}}$ for symmetric positive definite matrices; see [3].

To this end, we notice that the global stiffness matrix $\mathcal{A}$ is assembled from element stiffness matrices $\mathcal{A}_E$, which are of the same saddle point form as $\mathcal{A}$ itself, namely

$$\mathcal{A} = \sum_E \mathcal{A}_E, \quad \text{where} \quad \mathcal{A}_E = \begin{bmatrix} M_E & B_E^T \\ B_E & -C_E \end{bmatrix}.$$

As long as M_E is nonsingular [1], we can then compute exactly the local Schur complement of each $\mathcal{A}_E$, $S_E = C_E + B_E M_E^{-1} B^T$, and assemble these local contributions to form D_2 as $D_2 = \sum_E S_E$.

5.1. Preconditioning for the blocks D_1 and D_2. Each action of the preconditioner $\mathcal{D}^{-1}$ on a vector, requires one solve with each of the diagonal blocks D_1 and D_2. Instead of trying to approximate the blocks M and D_2, we choose to solve systems with them using an efficient (inner) iterative solution method, equipped with a robust preconditioner.

To construct an efficient preconditioner D_1 for M, we make use of the fact that the problem originates from linear elasticity. We assume that the unknown displacements are ordered in the so-called *separate displacement ordering (SDO)*. In the classical linear elasticity, it is well known that due to Korn's inequality the block-diagonal part of M is a nearly optimal preconditioner to M. Furthermore, it suffices to replace these diagonal blocks by scaled Laplacian matrices $L_i, i = 1, \dots, d$, corresponding to the bilinear form

$$(L_i)_{j,k} = \widetilde{a}(v_{i,j}, v_{i,k}) = \int_\Omega 2\mu \nabla v_{i,j} \cdot \nabla v_{i,k},$$

where $v_{i,j}$ is the j-th basis function for the i-th component of $\mathbf{v}$. Namely, it holds that

$$\alpha(\Omega)\widetilde{a}(\mathbf{u}(t), \mathbf{v}) \le \widehat{a}(\mathbf{u}(t), \mathbf{v}) \le 2\widetilde{a}(\mathbf{u}(t), \mathbf{v}),$$

[1] The nonsingularity of M_E can always be enforced by adding a regularization term to the diagonal, $\widetilde{M}_E = M_E + \epsilon h^2 I$, where $0 < \epsilon \ll 1$ and h is the space discretization parameter.

for certain constant $\alpha(\Omega)$ which depends only on Ω and the boundary conditions. For more details on this, we refer to [2], and the references therein.

Based on the above reasonings, we propose to solve the *nonsymmetric* block M in each time-step, by a *symmetric* block-diagonal preconditioned iterative solver, where the diagonal blocks are Laplacians which obey the same boundary conditions as the original blocks. Finally, we choose the (multiplicative) AMLI framework to solve with each of the diagonal blocks L_i, as well as with the block D_2.

REMARK 5.2. We note that for the glacial rebound problem the advection terms are not negligible but also not dominating the elasticity counterpart. This is clearly seen after scaling the problem and making it dimensionless; cf. [10].

The AMLI preconditioner we use here, belongs to the class of full-block factorization preconditioners of the form

$$P^{(k)} = \begin{bmatrix} B_{11}^{(k)} & 0 \\ A_{21}^{(k)} & I_2^{(k)} \end{bmatrix} \begin{bmatrix} I_1^{(k)} & {B_{11}^{(k)}}^{-1} A_{12}^{(k)} \\ 0 & [P^{(k-1)}] \end{bmatrix}, k = 1, 2, \ldots, l. \tag{5.4}$$

It is recursively defined on a sequence of matrices of decreasing order $A^{(l)}, A^{(l-1)}, \ldots, A^{(0)}$, each split into 2×2 block form, i.e.,

$$A^{(k)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A_{22}^{(k)} \end{bmatrix}.$$

On each level, $B_{11}^{(k)}$ is some approximation of $A_{11}^{(k)}$ and $P^{(k-1)}$ is some approximation of $S^{(k)} = A_{22}^{(k)} - A_{21}^{(k)} {A_{11}^{(k)}}^{-1} A_{12}^{(k)}$. The notation $[P^{(k-1)}]$ indicates that some stabilization is required, at least on some of the levels, in order to prevent the condition number of ${P^{(l)}}^{-1} A^{(l)}$ to grow with the number of levels.

When we apply the AMLI preconditioner for the diagonal blocks of D_1 and for D_2 (for brevity referred to as AMLI(L_i) and AMLI(D_2), correspondingly), we stabilize on every second level by performing three iterations with the Generalized Conjugate Gradient - Minimal Residual method (GCG-MR); cf. [1]. The coarse mesh matrix $P^{(k-1)}$ is the element-wise Schur complement, constructed as is described in [16] and [13], namely by assembly of local, exactly computed, Schur complement matrices. The blocks $B_{11}^{(k)}$ are taken to be incomplete LU-factorizations of the corresponding $A_{11}^{(k)}$.

The theory for the AMLI methods regarding the quality of the 2×2 block matrix splittings, the choices of $B_{11}^{(k)}$ and $P^{(k-1)}$, and how to stabilize them to get optimal order methods, is complete for the case of SPD matrices, see for example [4], [6], [7], [19], [25], to name a few, and the references therein. Furthermore, again for the class of SPD matrices, it is shown that the AMLI preconditioner is robust with respect to arbitrarily large jumps in the problem coefficients, as long as those are aligned with the coarse discretization mesh used in their construction.

The major tool to show convergence in the SPD case is the so-called Cauchy-Bunyakowski-Schwarz constant γ. For the non-SPD case, however, such an analogous parameter is not yet known, and we note here that AMLI(D_2) is performed for the nonsymmetric D_2 without the support of a rigorous theory. The numerical experiments, however, are very promising.

To summarize, the preconditioning algorithm for $\mathcal{A}$ is given below, for brevity in the sequel referred to as BT-AMLI:

Solve $\mathcal{A}$ by GCG-MR, preconditioned by

$$\mathcal{D} = \begin{bmatrix} [M] & 0 \\ B & [D_2] \end{bmatrix}$$

ETNA
Kent State University
etna@mcs.kent.edu

- For each action $\mathcal{D}^{-1}\mathbf{v}$,
  - solve M by an (inner) GCG-MR solver, preconditioned by (in 2D)

$$D_M^L = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix},$$

    and systems with L_i are solved by AMLI-preconditioned GCG-MR.
  - solve D_2 by an (inner) GCG-MR solver, preconditioned by AMLI(D_2).

When we solve M and D_2 accurately, the number of outer iterations goes down, which, however does not imply that the overall solution time is reduced due to the extra work spent by the inner solvers. In [13] we investigate how to choose the relative accuracy for M and D_2 optimally, and we find that the smallest overall solution time is achieved for a reduction of the initial residual of order 0.5, for both blocks.

6. Numerical experiments. We illustrate the behaviour of the solution procedure, described in Section 5 on an inhomogeneous model of earth's lithosphere. We consider both the purely elastic and the viscoelastic cases. The elastic model is a simplification of the full viscoelastic problem, but nevertheless important, due to the fact that, under reasonable choice of the timestep, the characteristics of the matrices in the viscoelastic case remain quite close to those of the matrices in the elastic case. Therefore, the overall efficiency and scalability of the method when solving the viscoelastic problem follows that of the iterative solver for the elastic problem, for which we provide a separate study how it scales with problem size, that the preconditioner is robust with respect to the material parameters λ and μ and how its performance compares with a state-of-art sparse direct solution method.

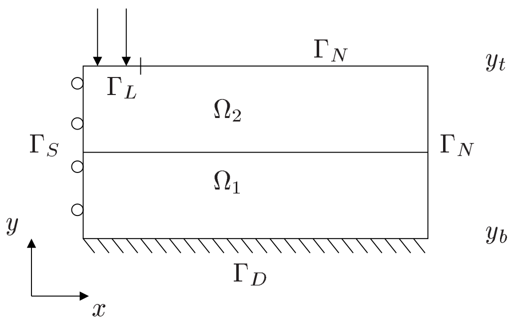

FIG. 6.1. *The geometry of the problem*

We consider a 2D (3D) flat earth model, which is symmetric with respect to $x = 0$ (and $y = 0$), subject to a Heaviside load of a 1000 km wide (side) and 2 km thick sheet of ice with density $\rho_i = 981\ kg\, m^{-3}$.

The size of the 2D domain is 10 000 km width and 4000 km depth. We impose homogeneous Dirichlet boundary conditions on the boundary $y = -4000$ km ($z = -4000$ km) and symmetry conditions on the boundary $x = 0$ (and $y = 0$). Homogeneous Neumann conditions are imposed 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). The geometry of the problem in two space dimensions is shown in Figure 6.1.

The domain Ω is split along the line $y = -2000$ km and both the Young modulus and the Poisson ratio are different in the two subdomains Ω_1 and Ω_2. Material parameters for this problem are given in Table 6.1. The time period for the viscoelastic simulations (which

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 6.1
Material parameters used in experiments.

Parameter		Value	
		Ω_1	Ω_2
Rock density ($kg\,m^{-3}$)	ρ_r	3300	3300
Young's modulus (Pa)	E	$4\,10^{11}$	$4\,10^{11} - 4\,10^{13}$
Poisson number	ν	0.5	$0.2 - 0.5$

TABLE 6.2
Time comparison with the finite element package ABAQUS (in sec) for elastic case

	Direct solver (ABAQUS)		Iterative solver (BT-AMLI)	
N	Assembly	Solver	Assembly	Solver
23603	3.326	4.7178	2.57	4.16(1.95)
93283	13.018	18.055	10.32	18.06(8.96)
370883	50.537	72.984	41.44	83.96(46.58)

influences the scaling of the problem) is $T_{max} = 200000$ years. The actual computations are performed on a dimensionless problem, obtained from the original one after scaling of the space and time domains, as well as of the material coefficient E; cf. [10].

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 variable p lives.

As an outer iterative scheme for equation (5.1), we use the GCG-MR method, both for the outer iterative solver for $\mathcal{A}$ and for the two inner iterative solvers for D_1 and D_2. The outer solver is stopped when the norm of initial residual is reduced by six orders of magnitude.

As outlined in the solution algorithm, the viscoelastic problem requires a sequence of time-steps and during each of those a linear system of equations is solved. The matrix of that linear system changes whenever the time-step changes. It is clear, that within the chosen space-time discretization scheme, savings in the total computational time can be achieved prevailingly on the cost of solving the arising linear systems in a time-efficient manner. Therefore, we address the solution of the linear system in detail. We present two sets of experiments, referred below to as *Set_E* and *Set_V*.

The tests in *Set_E* illustrate the efficiency of the proposed linear system solver alone for a purely elastic model, i.e., $\mathcal{A}_j = \mathcal{A}$; cf. (4.5). There we elaborate on 2D and 3D, homogeneous and inhomogeneous spatial domains with jumps both material coefficients. We present also comparisons with a state-of-art commercial sparse direct solver.

Set_V contains illustrations of the solution of the full viscoelastic problem. Here we restrict ourselves only to 2D.

Set_E (Tables 6.2, 6.3 and 6.4): The software used in the experiments in this set is based on the two open-source packages `deal.II` [8] and `PETSc` [17]. We use `deal.II` to obtain the finite element discretization and `PETSc` for the numerical linear algebra (routines for matrix-vector multiplication, scalar products, etc.). All tests are performed on a 1.5 GHz Intel Itanium 2 computer with 25 GB of RAM.

Table 6.2 shows timings for the solution of a test problem in two space dimensions where a homogeneous compressible elastic solid ($\nu = 0.2$) is loaded by a rectangular sheet of ice. This problem is solved using the sparse direct solver provided in the commercial finite element package ABAQUS, and our software (BT-AMLI). The problem size N shown in

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 6.3
Iteration counts for elastic case

ν_2	0.2	0.4	0.5
		2D	
		$N = 23603$	
$E_2 = 4e11$	$19(1.79, 1)$	$20(1.85, 1)$	$24(1.62, 1)$
$E_2 = 4e12$	$23(2.35, 1)$	$24(2.25, 1)$	$28(2.00, 1)$
$E_2 = 4e13$	$29(3.83, 1)$	$31(3.87, 1)$	$33(3.42, 1)$
		$N = 93283$	
$E_2 = 4e11$	$19(1.79, 1)$	$21(1.76, 1)$	$22(1.73, 1)$
$E_2 = 4e12$	$24(2.29, 1)$	$24(2.29, 1)$	$26(2.19, 1)$
$E_2 = 4e13$	$28(3.93, 1)$	$29(3.69, 1)$	$32(3.62, 1)$
		$N = 370883$	
$E_2 = 4e11$	$18(2.00, 1)$	$20(1.80, 1)$	$21(1.86, 1)$
$E_2 = 4e12$	$24(2.46, 1)$	$25(2.28, 1)$	$25(2.36, 1)$
$E_2 = 4e13$	$30(4.17, 1)$	$32(4.19, 1)$	$34(3.47, 1)$
		3D	
		$N = 12512$	
$E_2 = 4e11$	$19(2.00, 1)$	$20(2.05, 1)$	$21(2.00, 1)$
$E_2 = 4e12$	$25(2.96, 1)$	$25(2.76, 1)$	$27(2.44, 1)$
$E_2 = 4e13$	$28(4.21, 1)$	$31(3.94, 1)$	$33(3.79, 1)$
		$N = 89700$	
$E_2 = 4e11$	$20(2.20, 1)$	$22(2.23, 1)$	$22(2.27, 1)$
$E_2 = 4e12$	$25(3.40, 1)$	$27(2.89, 1)$	$29(2.62, 1)$
$E_2 = 4e13$	$31(5.45, 1)$	$33(4.88, 1)$	$36(4.47, 1)$

the left column refers to the problem size in BT-AMLI which, due to the mixed variable formulation and the choice of the stable finite element discretization, is roughly $1/8$-th larger than the problem size in ABAQUS. The right-most column shows the overall time spent on the solution phase of BT-AMLI, and the figures in parentheses give the time consumed by the iterative solver only (which is included in the overall solution time for BT-AMLI).

The results in Table 6.2 deserve a special comment. The observed (almost exactly) linear complexity of the sparse direct solver from ABAQUS is remarkable. Since the software is commercial, we were not able to find out how is this scalability achieved. The same linear scaling of the computational time is achieved for 2D problems of size up to 1.5 million degrees of freedom; see [11]. This advocates that the available highly optimized sparse direct solution methods are feasible for quite large scale applications on nowadays computers with large memories, in particular for 2D problems. However, having in mind that we solve slightly larger problem, the above results also show that our preconditioned iterative solver is a suitable alternative to a direct solution method for the considered class of problems, even in 2D. A real gain of the preconditioned iterative method over the sparse direct solver is achieved for 2D problems of even larger size and for 3D problems of the considered type of size 100000 and above. Further detailed comparisons between the direct and the iterative solution methods, and a discussion about their efficiency and accuracy are presented in [11]. The results in Table 6.2 also indicate that a combination of a preconditioned iterative method which uses a sparse direct method as a block solver could be the alternative to win the fastest overall simulation time.

Table 6.3 shows the number of iterations to reach convergence for different combinations

TABLE 6.4
Time spent to solve the linear system in equation (5.1) for elastic case

ν_2	0.2	0.4	0.5
		2D	
		$N = 23603$	
$E_2 = 4e11$	$6(4)$	$7(4)$	$7(5)$
$E_2 = 4e12$	$9(6)$	$9(6)$	$9(6)$
$E_2 = 4e13$	$14(12)$	$15(13)$	$15(13)$
		$N = 93283$	
$E_2 = 4e11$	$29(18)$	$30(20)$	$31(20)$
$E_2 = 4e12$	$40(29)$	$39(29)$	$41(30)$
$E_2 = 4e13$	$66(56)$	$65(55)$	$70(59)$
		$N = 370883$	
$E_2 = 4e11$	$126(81)$	$128(83)$	$134(89)$
$E_2 = 4e12$	$176(131)$	$173(128)$	$177(132)$
$E_2 = 4e13$	$314(269)$	$334(289)$	$305(260)$
		3D	
		$N = 12512$	
$E_2 = 4e11$	$19(9)$	$19(10)$	$20(10)$
$E_2 = 4e12$	$27(17)$	$26(16)$	$25(16)$
$E_2 = 4e13$	$38(28)$	$39(29)$	$40(30)$
		$N = 89700$	
$E_2 = 4e11$	$223(99)$	$234(110)$	$236(112)$
$E_2 = 4e12$	$312(188)$	$297(173)$	$294(170)$
$E_2 = 4e13$	$493(370)$	$475(351)$	$477(353)$

of Poisson number ν_2, Young modulus E_2 in Ω_2 and refinements of the mesh. The left number in each column is the number of outer iteration, whereas within parentheses, the average number of inner iterations required for D_1 and D_2 per outer iterations are shown. We see that the iterative solver is robust with respect to problem size, jumps in material coefficients, and with respect to (in)compressibility parameter ν.

In Table 6.4 solution times for equation (5.1) in 2D and 3D are shown. The left number in each column is the total solution time, that is, the time spent on assembly of the Schur complement approximation, on construction of the multilevel preconditioners for L_i and D_2, and by the iterative solver. The solution time for the outer iterative method is specified within parentheses. The growth in solution time with increasing problem size and difference in the Young moduli E_1 and E_2 follows the slight growth in the number of inner and outer iterations required for convergence of the inner and outer iterative solvers.

Set_V (Tables 6.5, 6.6 and 6.7): Next we apply the described preconditioned iterative solver to the full viscoelastic problem. All experiments in this set are performed in `Matlab` and therefore timing results are irrelevant.

We recall that in this case, during each time-step j we solve a system with a matrix $\mathcal{A}_j = \mathcal{A} - \frac{\Delta t_j}{2}\mathcal{A}_0$ as denoted in (4.5). We present numerical experiments with two values of the time step Δt_j – 0.87, which corresponds to approximately one years for the unscaled problem and is of the order of the initial (smallest) time-step chosen by ABAQUS and 69.6, which corresponds to 80 years for the unscaled problem and is of the order of the final (largest) time-step in the ABAQUS runs.

We use a block lower-triangular preconditioner as in (5.2), namely, we solve $\mathcal{A}_j$ by

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 6.5
Iteration counts for viscoelastic case and $\Delta_t = 0.87$

ν_2	0.2	0.4	0.5
	$N = 23603$		
$E_2 = 4e11$	13	13	13
$E_2 = 4e12$	15	15	15
$E_2 = 4e13$	15	15	15
	$N = 93283$		
$E_2 = 4e11$	13	13	13
$E_2 = 4e12$	15	16	15
$E_2 = 4e13$	16	17	17

GCG-MR, preconditioned by $\mathcal{D} = \begin{bmatrix} [M] & 0 \\ B & D_2 \end{bmatrix}$. For each action $\mathcal{D}^{-1}\mathbf{v}$, we

– solve M either exactly (Table 6.5) or by an (inner) GCG-MR solver, preconditioned by $D_M^M = \begin{bmatrix} M_{11} & 0 \\ 0 & M_{22} \end{bmatrix}$, where M_{11} and M_{22} are the exact diagonal blocks of M and systems with those are solved with a direct method (Tables 6.6, 6.7).

– solve D_2 by a direct method.

The results in Table 6.5 enable us to judge over the quality of the approximation of the Schur complement alone. It can be seen, that if we take as D_2 the exact Schur complement of $\mathcal{A}_j$ and solve exactly for M and D_2, the outer method will converge in two iterations. The observed here nearly constant number of iterations shows that the element-wise constructed Schur complement is a high quality approximation of the true one even for the nonsymmetric problem, considered here.

In Tables 6.6 and 6.7 we solve the block M using an inner solver with a block-diagonal preconditioner. For clarity, we repeat the differences in the settings for the viscoelastic experiments compared with those from Table 6.3:

- the exact (nonsymmetric) diagonal blocks M_{ii} are used instead of the (symmetric) discrete Laplacians L_i,
- M_{ii} are solved exactly in contrast to AMLI(L_i), $i = 1, 2$

A comparison between Table 6.3 and Tables 6.6 and 6.7 confirms that D_M^L is a robust and efficient alternative to D_M^M as well as that AMLI(L_i) is a competitive replacement of a direct block solver. We also see that for the range of the problem parameters E, ν and η, as well as for the tested values of Δt_j differing by two orders of magnitude, the overall behaviour of the preconditioned iterative solver remains robust. The experiments provide numerical evidence that S approximates well the corresponding Schur complement matrix and that the AMLI framework is applicable also for nonsymmetric problems.

The solution has been compared to that obtained by the ABAQUS solver and found in a good agreement with the latter.

7. Conclusions. In this paper we discuss numerical simulations of viscoelastic problems. The formulation handles compressible as well as fully incompressible materials. The matrices of the so-arising systems of linear equations are nonsymmetric and of saddle-point form. These systems are solved by an inner-outer preconditioned iterative method (GCG-MR). We apply techniques to approximate the Schur complement matrix of the original nonsymmetric indefinite matrix as well as to construct an AMLI-type preconditioner, which is originally developed and fully analysed for symmetric positive definite systems, also in the nonsymmetric case. The presented numerical results show that the resulting preconditioner

TABLE 6.6
Iteration counts for viscoelastic case and $\Delta_t = 0.87$

ν_2	0.2			0.4			0.5		
ε_{inner}	0.5	0.1	0.01	0.5	0.1	0.01	0.5	0.1	0.01
E_2	$N = 23603$								
$4e11$	19(2)	15(3)	14(5)	20(2)	16(3)	14(5)	21(2)	16(3)	14(5)
$4e12$	20(2)	18(4)	16(7)	22(2)	19(4)	16(7)	23(2)	19(4)	17(7)
$4e13$	23(2)	23(6)	17(10)	26(2)	24(6)	19(10)	28(2)	25(5)	20(10)
E_2	$N = 93283$								
$4e11$	18(2)	15(3)	13(4)	19(2)	15(3)	14(5)	20(2)	15(3)	13(5)
$4e12$	20(2)	19(4)	16(6)	22(2)	20(4)	17(6)	23(2)	20(4)	17(6)
$4e13$	23(2)	22(6)	18(9)	26(2)	23(6)	18(9)	27(2)	24(5)	19(9)
E_2	$N = 370883$								
$4e11$	19(2)	14(2)	13(4)	19(2)	14(3)	13(5)	19(2)	14(4)	13(5)
$4e12$	20(2)	19(4)	16(6)	22(2)	20(4)	17(6)	23(2)	19(4)	16(6)
$4e13$	24(2)	24(6)	18(9)	27(2)	23(5)	19(10)	27(2)	25(5)	20(9)

TABLE 6.7
Iteration counts for viscoelastic case and $\Delta_t = 69.6$

ν_2	0.2			0.4			0.5		
ε_{inner}	0.5	0.1	0.01	0.5	0.1	0.01	0.5	0.1	0.01
E_2	$N = 23603$								
$4e11$	19(2)	16(3)	14(5)	20(2)	16(3)	14(5)	19(2)	16(3)	13(5)
$4e13$	27(3)	27(7)	21(12)	29(2)	28(8)	22(12)	30(2)	27(7)	23(12)
E_2	$N = 93283$								
$4e11$	18(2)	15(3)	14(5)	18(2)	16(3)	13(5)	18(2)	15(3)	13(5)
$4e13$	27(3)	27(8)	22(12)	30(3)	29(8)	22(12)	30(2)	30(7)	23(12)
E_2	$N = 370883$								
$4e11$	18(2)	14(3)	13(4)	18(2)	15(3)	13(5)	18(2)	15(3)	12(5)
$4e13$	29(2)	30(8)	22(11)	31(3)	33(8)	22(12)	30(2)	33(9)	22(11)

is nearly optimal, robust with respect to problem parameters, independent on jumps of coefficients, i.e., it inherits the properties of the preconditioning technique in the symmetric case. The method is straightforwardly applicable to 3D. We present comparisons of the execution times of the proposed iterative method and the direct solution method available in the commercial FEM package ABAQUS, and show that the iterative method with the proposed preconditioner is a viable alternative of the direct method.

The test problem originates from glacial rebound, where some extra advection terms appear in the balance equations. The numerical solution techniques, however, are applicable for other problems which give rise to systems with matrices of saddle point form.

Acknowledgments. The authors are indebted to Dr Björn Lund for the fruitful collaboration and for helping them getting through the labyrinth of the glacial rebound phenomena.

Appendix. For completeness we include the detailed form of all matrices and vectors, which arise from the space and time discretization of equation (3.5).

$$
\begin{aligned}
\{\mathbf{u}_j\}_i &= \mathbf{u}(\mathbf{x}_i, t_j),\\
\{\mathbf{p}_j\}_i &= \mathbf{u}(\mathbf{x}_i, t_j),\\
\{\widehat{M}\}_{i,k} &= \widehat{a}(\mathbf{v}_i, \mathbf{v}_k),\\
\{\bar{M}\}_{i,k} &= \bar{a}(\mathbf{v}_i, \mathbf{v}_k),\\
\{\widetilde{M}_l^m\}_{i,k} &= \tfrac{\Delta t_m}{2}\widetilde{a}(\mathbf{v}_i, \mathbf{v}_k; t_m, t_l),\\
\{B\}_{i,k} &= b(\mathbf{v}_i, q_k),\\
\{\widetilde{B}_l^m\}_{i,k} &= \tfrac{\Delta t_m}{2}\widetilde{b}(\mathbf{v}_i, q_k; t_m, t_l),\\
\{C\}_{i,k} &= c(q_i, q_k),\\
\{\widetilde{C}_l^m\}_{i,k} &= \tfrac{\Delta t_m}{2}\widetilde{c}(q_i, q_k; t_m, t_l),\\
\{I_{u,l}^m\}_i &\approx \int_{t_l}^{t_m} \left[\widetilde{a}(\mathbf{u}(\tau), \mathbf{v}_i; t_m, \tau) + \widetilde{b}(\mathbf{v}_i, p(\tau); t_m, \tau)\right] d\tau,\\
\{I_{p,l}^m\}_i &\approx \int_{t_l}^{t_m} \left[\widetilde{b}(\mathbf{u}(\tau), q_i; t_m, \tau) - \widetilde{c}(p(\tau), q_i; t_m, \tau)\right] d\tau,\\
\{I_{\ell,l}^m\}_i &\approx \int_{t_l}^{t_m} \widetilde{\ell}(v_i; t_m, \tau)\, d\tau,\\
\{\mathbf{f}_j\}_i &= \mathbf{f}(\mathbf{v}_i; t_j),\\
\{\mathbf{l}_j\}_i &= \ell(\mathbf{v}_i; t_j),\\
\{\widetilde{\mathbf{l}}_j^m\}_i &= \tfrac{\Delta t_m}{2}\widetilde{\ell}(v_i; t_m, t_l).
\end{aligned}
$$

REFERENCES

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

[2] O. AXELSSON, *On iterative solvers in structural mechanics; separate displacement orderings and mixed variable methods*, Math. Comput. Simulation, 50 (1999), pp. 11–30.

[3] 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, 2006.
http://www.it.uu.se/research/publications/reports/2006-048/

[4] O. AXELSSON AND M. NEYTCHEVA, *Algebraic multilevel iteration methods for Stieltjes matrices*, Numer. Linear Algebra Appl., 1 (1994), pp. 213–236.

[5] O. AXELSSON AND M. NEYTCHEVA, *Eigenvalue estimates for preconditioned saddle point matrices*, Numer. Linear Algebra Appl., 13 (2006), pp. 339–360.

[6] O. AXELSSON AND A. PADIY, *On the additive version of the algebraic multilevel iteration method for anisotropic elliptic problems*, SIAM J. Sci. Comput., 20 (1999), pp. 1807–1830 .

[7] O. AXELSSON AND P. S. VASSILEVSKI, *A survey of a class of algebraic multilevel iteration methods for positive definite symmetric matrices*, in Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications (ENUMATH 99), T. Tiihonen, P. Tarvainen, P. Neittaanmäki, eds., World Scientific, Singapore, 2000, pp. 16–30.

[8] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, `deal.II` Differential Equations Analysis Library, Technical Reference, IWR. http://www.dealii.org.

[9] M. BENZI, G. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14, 2005, pp. 1–137.

[10] E. BÄNGTSSON, *Robust Preconditioners Based on the Finite Element Framework*, Ph.D. Thesis, Acta Universitatis Upsaliensis, Uppsala, 2007.
http://publications.uu.se/abstract.xsql?dbid=7828

[11] E. BÄNGTSSON AND B. LUND, *A comparison between two solution techniques to solve the equations of glacially induced deformation of an elastic earth*, Internat. J. Numer. Methods Engrg., 75 (2008), pp.479–502.

[12] E. BÄNGTSSON AND M. NEYTCHEVA, *Numerical simulations of glacial rebound using preconditioned iterative solution methods*, Appl. Math., 50 (2005), pp. 183–201.

[13] E. BÄNGTSSON AND M. NEYTCHEVA, *An agglomerate multilevel preconditioner for linear isostasy saddle point problems*, in Proceedings of the 5th International Conference on Large-scale Scientific Compu-

tations, Lecture Notes in Computer Science, 3743, I. Lirkov, S. Margenov, and J. Wasniewski, eds., Springer, Berlin, 2006, pp. 113–120.

[14] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.

[15] V. KLEMANN, P. WU, AND D. WOLF, *Compressible viscoelasticity: stability of solutions for homogeneous plane-earth models*, Geophys. J., 153 (2003), pp. 569–585.

[16] J. KRAUS, *Algebraic multilevel preconditioning of finite element matrices using local Schur complements*, Numer. Linear Algebra Appl., 13 (2006), pp. 49–70.

[17] *Portable, Extensible Toolkit for Scientific computation (PETSc) suite*, Mathematics and Computer Science Division, Argonne National Laboratory. http://www-unix.mcs.anl.gov/petsc/petsc-2/

[18] J. NEDOMA, *Numerical Modelling in Applied Geodynamics*, Wiley, Chichester, 2002.

[19] Y. Notay, *Using approximate inverses in algebraic multilevel methods*, Numer. Math., 80 (1998), pp. 397–417.

[20] S. SHAW, M. K. WARBY, J. R. WHITEMAN, C. DAWSON, AND M. F. WHEELER, *Numerical techniques for the treatment of quasistatic viscoelastic stress problems in linear isotropic solids*, Comput. Methods Appl. Mech. Engrg., 118 (1994), pp. 211–237.

[21] S. SHAW AND J. R. WHITEMAN, *Some partial differential Volterra equation problems arising in viscoelasticity*, Report BICOM 97/9, February 28, 1997.
http://citeseer.ist.psu.edu/shaw98some.html

[22] S. SHAW AND J. R. WHITEMAN, *Numerical solution of linear quasistatic hereditary viscoelasticity problems*, SIAM J. Numer. Anal., 38 (2000), pp. 80–97.

[23] S. SHAW AND J. R. WHITEMAN, *A posteriori error estimates for space-time finite element approximation of quasistatic hereditary linear viscoelasticity problems*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 5551–5572.

[24] D. L. TURCOTTE AND G. SCHUBERT, *Geodynamics, Applications of Continuum Physics to Geological Problems*, Wiley, New York, 1982.

[25] P. VASSILEVSKI, *On two ways to stabilizing the hierarchical basis multilevel methods*, SIAM Rev., 39 (1997), pp. 18–53.