Numerical methods for basin-scale poroelastic modelling

by

Carl Joachim Berdal Haga

Thesis submitted for the degree of Philosophiae Doctor
Department of Mathematics
Faculty of Natural Sciences
University of Oslo
May 2011

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 1088

ISSN 1501-7710

Cover: Inger Sandved Anfinsen.
Printed in Norway: AIT Oslo AS.

Produced in co-operation with Unipub.
The thesis is produced by Unipub merely in connection with the
thesis defence. Kindly direct all inquiries regarding the thesis to the copyright
holder or the unit which grants the doctorate.

Preface

This thesis is the product of almost four years of work at Simula Research Laboratory. In the spring of 2006, while finishing a Master's Degree at the University of Oslo, I was made aware of an opening for two PhD positions in the Computational Geosciences group of Simula, which seemed to be an easy way forward.

Wrong, of course!

These years have been rewarding, but have also seen setbacks and frustration. I am therefore grateful to those who didn't let me give up, even when there was good reason to.

First, my supervisors: Harald Osnes and Hans Petter Langtangen, who can apparently see the way out of every problem. Without their encouragement, this thesis wouldn't be.

During my PhD period, I have also been lucky enough to gain a small family: Benedikte, and later our son Hans August. They haven't seen me as much as they deserve. Thank you for your patience!

I have enjoyed my stay at Simula. That I owe to — among many others! — my perennial office-mate and friend Omar Al-Khayat, and to Are Magnus Bruaset, former leader of the CG group, current SSRI director.

Finally I am grateful to my parents, who have been supportive throughout.

Contents

List of Papers

- **Paper I**

  On the performance of an algebraic multigrid preconditioner for the pressure equation with highly discontinuous media
  J. B. Haga, H. P. Langtangen, B. F. Nielsen, H. Osnes
  Presented at the MekIT conference in Trondheim, May 2009

- **Paper II**

  Efficient block preconditioners for the coupled equations of pressure and deformation in highly discontinuous media
  J. B. Haga, H. Osnes, H. P. Langtangen
  Accepted for publication in *International Journal of Analytical and Numerical Methods in Geomechanics* (2010)

- **Paper III**

  On the causes of pressure oscillations in low-permeable and low-compressible porous media
  J. B. Haga, H. Osnes, H. P. Langtangen
  Accepted for publication in *International Journal of Analytical and Numerical Methods in Geomechanics* (2010)

- **Paper IV**

  A parallel block preconditioner for large scale poroelasticity with highly heterogeneous material parameters
  J. B. Haga, H. P. Langtangen, H. Osnes
  Submitted to *Computational Geosciences* (2010)

Introduction

1 Overview and Goals

The principal theme of this thesis is the investigation of techniques for solving the coupled poroelastic equations, used for analysing the stress and flow dynamics of sedimentary basins, in large-scale applications. As the papers presented here demonstrate, this theme straddles a number of fields, from continuum mechanics, through linear algebra and mathematical analysis, to parallel computation.

The geological history of sedimentary basins undergoing changes due to compaction, stresses and fluid flow is of fundamental interest, both to scientists trying to understand the effects of geological processes and to the oil and gas exploration community. For reasons of performance, tools for calculating this history often neglect important couplings in the underlying physics. These couplings, along with the large number of unknowns that are typically needed in order to obtain sufficient accuracy, makes the problem extremely computationally challenging to solve.

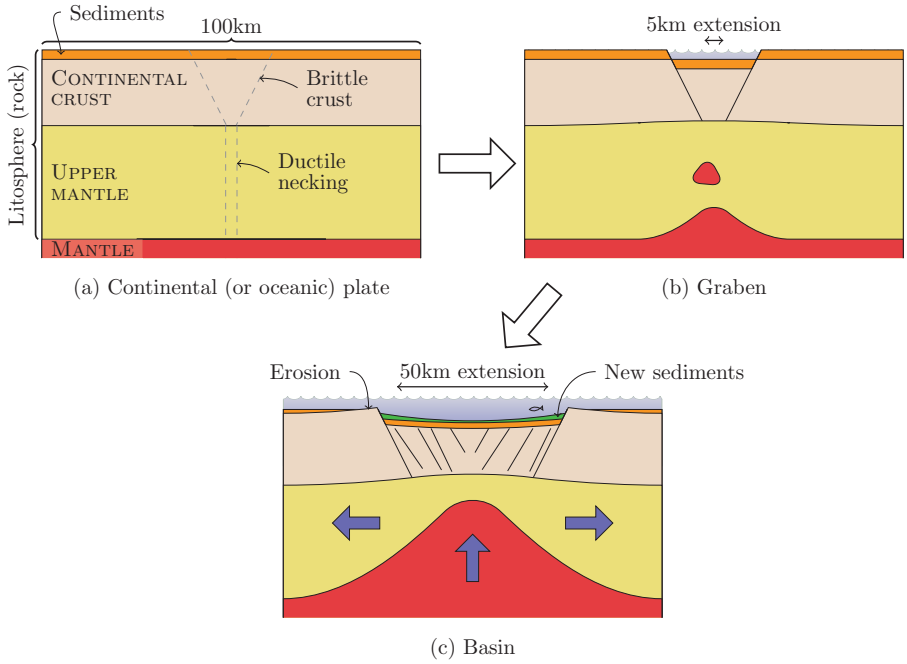
Our aim is to be able to perform simulation of such coupled systems on the scale of real basins. Realistically, these large-scale problems require parallel solution methods. The methods must furthermore be general and robust enough to handle complex geometries with discontinuous and highly contrasting material parameters.

The thesis consists of two parts: An introduction (this one) and a collection of papers. The introductory part aims to give an overview of the problem space, and to summarise the research work undertaken as fulfillment of the PhD requirements, while the collection of papers documents the research work itself.

2 Sedimentary Basins

> Geology, n. The science of the earth's crust — to which, doubtless, will be added that of its interior whenever a man shall come up garrulous out of a well.
>
> *A. Bierce*

Sedimentary basins are depressions in the crust of the Earth, into which debris and organic materials gather and settle. Eventually the sediments harden into porous rock. Over time, millions of tonnes of weight from above and heat from below conspire to transform rock into, well, different rock, and organic material into coal, oil and gas.



(a) Continental (or oceanic) plate

(b) Graben

(c) Basin

Figure 1: Diagram of the formation of an extensional basin. Redrawn from [12].

This, then, is what makes sedimentary basins interesting. Not only do they provide a rich history of the changing geography of a region, but they also provide a large part of the Earth's hydrocarbon reserves.

In this section, we will briefly outline how sedimentary basins can form and fill, while a fuller exposition can be found in standard text books such as [3].

2.1 Formation of sedimentary basins

Sedimentary basins can be formed through many different tectonic mechanisms. What they have in common is that movements of the tectonic plates form depressions in the crust into which the sediments can settle. For ease of presentation, we describe here just one such mechanism: extensional basins, which is the main mechanism by which the North Sea basins are formed.

The formation process of an extensional basin is illustrated in fig. 1. Initially, as the continental plate is pulled apart by movements of the molten mantle, brittle faults develop in the crust. The upper mantle stretches and partially melts. As the extension widens, a rift opens up, and parts of the crust sinks down into the mantle (fig. 1b). Owing to the melting and thinning the upper mantle, volcanic activity is a common feature of the extensional phase.

National Park Service

(a) Heavily eroded sediments. The light-coloured sandstone layer near the top is up to 150m thick. Grand Canyon, Arizona, USA.

E. Zimbres, cc-by-sa-3.0, http://commons.wikimedia.org/

(b) Centimeter-scale turbidites. Point Loma Formation, California, USA.

Figure 2: Sedimentary layers can vary widely in scale.

2.2 Deposition of materials

The depression formed by tectonic mechanisms is slowly filled with sediments (fig. 1c). The depositional processes are far from simple: the tectonic movement may continue or reverse; sediment buildup causes undersea avalanches (called turbidite currents); the sea level changes, as does the local flora and fauna. In fact, depositional modelling is a rich and complex field of its own, and the end result is often highly heterogeneous. Fig. 2 illustrates the range of scales that are present. The distinct rock layers can vary in thickness from a few centimeters up to tens of meters in the same basin.

2.3 Diagenesis and metamorphism

Finally, long-term chemical processes within the buried sediments are important for the history. Heat, acting on a rich geochemical environment, causes transformation of the rocks, and fluid circulates trough the porous channels. Owing to the heterogeneity that results from the depositional and crustal processes, the pressures and fluid flows in a basin may be rather complex. An example is shown in fig. 3.

Many processes are acting on sedimentary basins, and not everything is well understood. For example, the effective rheological properties of the rock may be very different on long time scales (where chemical processes are important) than what can be measured in wells or in laboratory samples. Still, we do know something about the processes, and a number of physical models describe the different processes.

In the present thesis, the contributions are in modelling the full three-dimensional coupled deformation and fluid movement in sedimentary basins with large number of unknowns and heterogeneous materials.

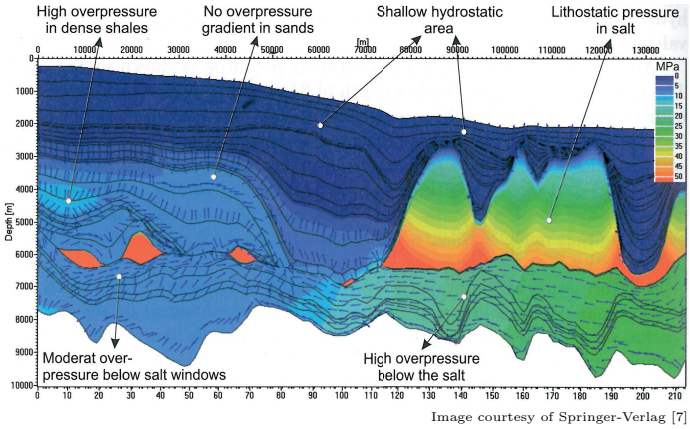


Figure 3: Cross section of a pore pressure simulation of the Santos basin outside São Paulo, Brazil. The pore pressures and fluid flow are complex due to the heterogeneous sedimentary layers and salt deposits.

3 The Mathematical Model

> Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.
>
> *E. P. Box*

As described above, the geological history of sedimentary basins is governed by a number of processes; tectonic, mechanical, chemical and biological. On the short time scales that are the focus of this thesis, however, the main influence on the evolution is in many cases the interaction of the deformation of the porous matrix with the flow of the pore-filling fluid. For a general introduction to other aspects of basin modelling, we refer to [7].

The mechanics of saturated porous media was first described by Terzaghi [13], who developed the original theory of one-dimensional homogeneous soil consolidation, and introduced the ideas of effective stress and the diffusion of fluid pressure by fluid flow. Biot [2] generalised this work to three dimensions and derived the partial differential equations (PDEs) governing coupled three-dimensional fluid flow and deformation in linear elastic porous media. These formulations are based on the continuum hypothesis, i.e., the particle nature or micro-structure of the media are only considered in an average sense. A review of modelling of porous media can be found in [11], while [14] offers a more comprehensive treatment.

Because the fully coupled three-dimensional poroelastic problem is computationally demanding, many practical large-scale analyses simplify the problem, either by assuming purely vertical deformation (Terzaghi consolidation), or by assuming one-way coupling where the fluid pressure influences the deformation but not vice versa. In their comparison of techniques for dealing with the coupling of the poroelastic problem, however, Dean *et al.* [5] found that solving the equations in a fully coupled manner is necessary

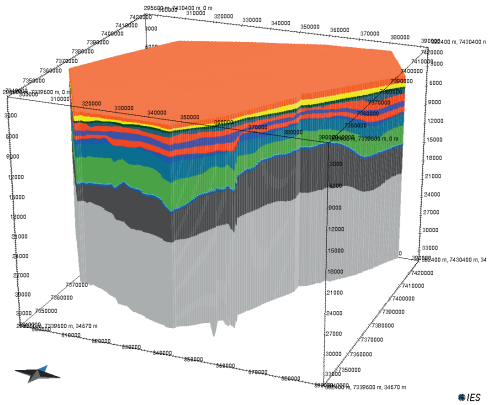
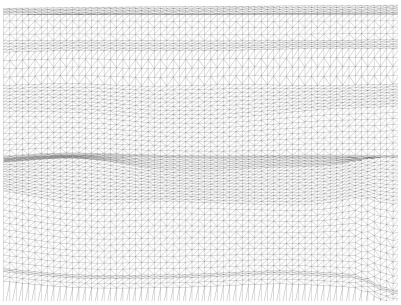


Image courtesy of Statoil and IES [10]

(a) A three-dimensional model of the Vøring basin, off the coast of central Norway.



(b) A cross section of the computational grid, showing about 1/5 of the width and 1/3 of the depth.

Figure 4: The Vøring basin is our main test case for realistic large-scale calculations of the poroelastic equations. It consists of 16 distinct layers of sediments, with about 8.5 million tetrahedral grid cells

in cases where the hydromechanical coupling is sufficiently strong.

In addition to the fundamental hypothesis of overlapping continua for the fluid and solid phase, we make a number of simplifying assumptions. A thorough discussion of the underlying assumptions, and a more general derivation of the equations, may be found for example in [4]. The main assumptions are:

- The solid and fluid phase may be treated as two overlapping continua; i.e., we can ignore micro-structure and treat the material as (piecewise) homogeneous on a representative macroscopic scale.
- The (macroscopic) permeability of the solid is anisotropic, but the elastic parameters are isotropic.
- The deformations and strains are small. Thus, a linearised strain tensor (strain-deformation relation) can be used, and a linear elastic constitutive equation (stress-strain relation) is adopted for the solid deformation. A relatively short time scale is implied.
- Acceleration terms can be ignored (elastostatic or quasistatic conditions).
- Isothermal conditions exist.
- A linear relation between the pressure gradient and the fluid seepage velocity (Darcy's law) is introduced. Experimental evidence suggests this may be inaccurate for low pressure gradients, but these are also less important for the overall dynamic behaviour.

Based on fundamental physical principles, along with the listed assumptions, a system of two coupled PDEs is derived: One for the fluid (pore) pressure p, and one for the solid skeleton displacement field $\boldsymbol{u}$. Additional mathematical equations, e.g., constitutive

relations and boundary conditions, are introduced as needed to close the model. The exposition loosely follows that of Wang [14].

3.1 Equation for the fluid pressure

Following Biot [2], we introduce the increment of fluid content, denoted ξ. This quantity measures the change in the fluid mass (m_f) in a control volume relative to the reference density (ρ_{f_0}); hence, $\xi = \rho_{\mathrm{f}_0}^{-1}(m_\mathrm{f} - m_{\mathrm{f}_0})$ given a reference fluid mass m_{f_0}, and the time derivative is $\partial\xi/\partial t = \rho_{\mathrm{f}_0}^{-1}\partial m_\mathrm{f}/\partial t$. In any control volume, the balance of fluid mass can be expressed as

$$\frac{\partial \xi}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{v}_\mathrm{D} = Q, \tag{1}$$

where $\boldsymbol{v}_\mathrm{D}$ is the fluid flux, or seepage velocity relative to the matrix velocity, and Q is the injection/withdrawal rate of fluid from external sources. The parameters are summarised in table 1.

Assuming, as stated, a linear dependence of the fluid content on the total volumetric stress, and a linear stress-strain relationship, ξ can be written as

$$\xi = \alpha\epsilon_\mathrm{V} + S_\epsilon p, \tag{2}$$

for a given volumetric strain ϵ_V and fluid pressure p, where α is the Biot-Willis coefficient and S_ϵ is the unconstrained specific storage coefficient. These can be determined experimentally,[1] or constructed from the combination of other material properties.

Cauchy's strain tensor, valid for small strains, relates the strain to the deformation through

$$\boldsymbol{\epsilon} = \frac{\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^\mathsf{T}}{2}, \tag{3}$$

from which we can deduce $\epsilon_\mathrm{V} \equiv \operatorname{Tr}\boldsymbol{\epsilon} = \boldsymbol{\nabla}\cdot\boldsymbol{u}$. Here, $\boldsymbol{u}$ is the displacement field of the solid matrix. The substitution of eqs. (2) and (3) into eq. (1) produces

$$S_\epsilon\frac{\partial p}{\partial t} + \alpha\frac{\partial \boldsymbol{\nabla}\cdot\boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{v}_\mathrm{D} = Q. \tag{4}$$

The solid velocity $\partial\boldsymbol{u}/\partial t$ here plays a similar role to the fluid velocity. Somewhat loosely, eq. (4) says that in a representary volume, the change in pressure is determined by the influx of mass, whether from fluid or solid movement or from external sources.

Finally, the fluid seepage velocity is given by Darcy's law. The driving forces are the pressure gradient and the gravity $\boldsymbol{g}$, in a medium with flow mobility $\boldsymbol{\Lambda}$:

$$\boldsymbol{v}_\mathrm{D} = -\boldsymbol{\Lambda}(\boldsymbol{\nabla}p - \rho_\mathrm{f}\boldsymbol{g}). \tag{5}$$

Assuming a near-constant fluid density ρ_f, this gives the final version of the mass balance equation for the fluid pressure, eq. (1), in terms of the primary unknowns p and $\boldsymbol{u}$,

$$S_\epsilon\frac{\partial p}{\partial t} + \alpha\frac{\partial \boldsymbol{\nabla}\cdot\boldsymbol{u}}{\partial t} - \boldsymbol{\nabla}\cdot\boldsymbol{\Lambda}\boldsymbol{\nabla}p = Q. \tag{6}$$

[1] For example, eq. (2) yields $\alpha = \partial\xi/\partial\epsilon_\mathrm{V}|_{\delta p=0}$; that is, α is the change in fluid content when the volumetric strain is increased while the fluid pressure is held constant. This quantity can be measured in a laboratory.

3.2 Equation for the deformation

In the quasistatic (or elastostatic) approximation, we assume that mechanical equilibrium is achieved at any time; thus, inertia (or acceleration) is ignored. The time evolution of such a system is governed by the slow response of the fluid to changes in the porous matrix. Balance-of-forces considerations then dictate that the net force in each space direction is zero, everywhere and at any time. For a control volume positioned at an arbitrary position (x, y, z) with side length Δ, this can be expressed by

$$\bar{F}_x(x,y,z) = \frac{1}{\Delta^3}\int_y^{y+\Delta}\int_z^{z+\Delta} \textstyle\sum_i(\sigma'_{i1}|_{x+\Delta} - \sigma'_{i1}|_x)\ \mathrm{d}z'\,\mathrm{d}y' = 0, \quad (7)$$

$$\bar{F}_y(x,y,z) = \frac{1}{\Delta^3}\int_x^{x+\Delta}\int_z^{z+\Delta} \textstyle\sum_i(\sigma'_{i2}|_{y+\Delta} - \sigma'_{i2}|_y)\ \mathrm{d}z'\,\mathrm{d}x' = 0, \quad (8)$$

$$\bar{F}_z(x,y,z) = \frac{1}{\Delta^3}\int_x^{x+\Delta}\int_y^{y+\Delta}\Big[\textstyle\sum_i(\sigma'_{i3}|_{z+\Delta} - \sigma'_{i3}|_z) - \int_z^{z+\Delta}\rho g\ \mathrm{d}z'\Big]\,\mathrm{d}y'\,\mathrm{d}x' = 0, \quad (9)$$

where $\boldsymbol{\sigma}'$ is the effective stress tensor, including the volumetric stress caused by the fluid pressure (written out below). As Δ approaches zero, eqs. (7)–(9) approaches the derivatives and can be written as the vector equation

$$\boldsymbol{F} = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}' + \rho\boldsymbol{g} = 0. \quad (10)$$

Although the deformations of the sediments are to a large degree plastic, we assume that for small deformations (or small perturbations within a larger deformation process) a more computationally efficient elastic model is adequate [4, 14]. By the assumption of isotropic linear elastic behaviour of the porous medium, the stress components are related to the displacement field through Hooke's law with an additional term for the fluid pressure,

$$\boldsymbol{\sigma}' = (\lambda\boldsymbol{\nabla}\cdot\boldsymbol{u} - \alpha p)\,\mathbf{I} + 2\mu\boldsymbol{\epsilon}. \quad (11)$$

Here, λ and μ are the Lamé material constants, and the strain tensor is given in eq. (3) for small strains. The Biot-Willis coefficient α is the same as in eq. (2), as can be shown by energy considerations [14, p. 19].

The full version of the balance-of-forces equation, in terms of the primary variables $\boldsymbol{u}$ and p, is given by inserting eq. (11) into eq. (10) and reordering the derivatives in the first term, as

$$\boldsymbol{\nabla}(\lambda\boldsymbol{\nabla}\cdot\boldsymbol{u} - \alpha p) + \boldsymbol{\nabla}\cdot\mu(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^{\mathsf{T}}) + \rho\boldsymbol{g} = 0. \quad (12)$$

3.3 Weak form

We rewrite eqs. (4) and (10) in weak form, and use Green's theorem to eliminate second order derivatives. We arrive thence at the requirement that the relations

$$\int_\Omega\left(\pi S_\epsilon\frac{\partial p}{\partial t} + \boldsymbol{\nabla}\pi\cdot\boldsymbol{v}_{\mathrm{D}} + \pi\alpha\frac{\partial}{\partial t}(\boldsymbol{\nabla}\cdot\boldsymbol{u})\right)\mathrm{d}\Omega - \int_\Gamma \pi\boldsymbol{n}\cdot\boldsymbol{v}_{\mathrm{D}}\,\mathrm{d}\Gamma = \int_\Omega \pi Q\,\mathrm{d}\Omega, \quad (13)$$

$$\int_\Omega(\boldsymbol{\nabla}\boldsymbol{\omega}:\boldsymbol{\sigma}')\ \mathrm{d}\Omega - \int_\Gamma \boldsymbol{\omega}\cdot(\boldsymbol{n}\cdot\boldsymbol{\sigma}')\,\mathrm{d}\Gamma = \int_\Omega \boldsymbol{\omega}\cdot\rho\boldsymbol{g}\,\mathrm{d}\Omega, \quad (14)$$

Table 1: The parameters

Parameter	Relation	SI unit	Description
S_ϵ	$= \frac{\phi}{K_\mathrm{f}} + \frac{\alpha-\phi}{K_\mathrm{s}}$	$[\mathrm{Pa}^{-1}]$	Fluid storage coefficient, in terms of the porosity ϕ and the fluid/solid bulk moduli $K_{\mathrm{f/s}}$. A measure of how much more fluid can be stored when the pressure increases.
$\boldsymbol{\Lambda}$	$= \frac{\boldsymbol{\kappa}}{\mu_\mathrm{f}}$	$[\frac{\mathrm{m}^2}{\mathrm{Pa\cdot s}}]$	Flow mobility tensor, in terms of the permeability tensor $\boldsymbol{\kappa}$ and the fluid viscosity μ_f. Measures how fast the fluid moves through the medium at a given pressure gradient.
$\rho_\mathrm{f}, \rho_\mathrm{s}$		$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Fluid and solid density.
ρ	$= \phi\rho_\mathrm{f} + (1-\phi)\rho_\mathrm{s}$	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Total density, in terms of the porosity ϕ and the component densities.
$\boldsymbol{g}$		[N]	Force of gravity.
α	$\approx 1 - K_\mathrm{f}/K_\mathrm{s}$, $\phi \leq \alpha \leq 1$	$[\cdot]$	The Biot-Willis poroelastic coefficient, relating change in fluid content to change in volume. $K_{\mathrm{f/s}}$ are the fluid/solid bulk moduli; ϕ is the porosity.
λ	$= \frac{E\nu}{(1-\nu)(1-2\nu)}$	[Pa]	The Lamé elastic material constants, defined in terms of the undrained Poisson's ratio ν and Young's modulus E.
μ	$= \frac{E}{2(1+\nu)}$	[Pa]	

hold for all test functions π and $\boldsymbol{\omega}$ in their respective spaces, defined on the domain Ω and its boundary Γ. To keep these equations simple, and to make the natural boundary conditions clearer, we do not expand the Darcy velocity $\boldsymbol{v}_\mathrm{D}$ or the stress tensor $\boldsymbol{\sigma}'$ in the relations above. Their definitions in terms of the primary unknowns p and $\boldsymbol{u}$ are found in eqs. (5) and (11), respectively.

The relevant spaces for eqs. (13)–(14) are the spaces of weakly differentiable functions, or Sobolev spaces,

$$p, \pi \in H^1(\Omega), \quad \boldsymbol{u}, \boldsymbol{v} \in [H^1(\Omega)]^d, \tag{15}$$

where d is the number of spatial dimensions. The boundary conditions are of two types: Prescribed values of p or $\boldsymbol{u}$, and prescribed fluxes $\boldsymbol{v}_\mathrm{D} \cdot \boldsymbol{n}$ or tractions $\boldsymbol{\sigma}' \cdot \boldsymbol{n}$ (through the boundary integrals over Γ). When using mixed formulations, other spaces and boundary conditions are used; the details of this are found in Paper III.

The discrete finite element approximation follows from solving eq. (13)–(14) in finite-dimensional subspaces of $H^1(\Omega)$, as we shall presently describe.

4 Numerical Methods

4.1 The Finite Element method

The finite element method, detailed for example in [1], is a well established method for the discrete approximation of PDEs. The basic idea is to solve the weak form of the PDEs in finite dimensional spaces, where the solution can be represented as a weighted sum of trial functions spanning the discrete solution space. Thus, any function f can be approximated in the discrete space V_h as

$$f_h(\boldsymbol{r}) = \sum_i \mathrm{x}_i \phi_i(\boldsymbol{r}), \tag{16}$$

for a vector of weights x, with $\text{span}\{\phi_i\} = \text{span}\, V_h$. The discrete solution is found by using this approximation in the weak equation $a(f, \phi_j) = L(\phi_j)$, where a and L are the (bi-)linear forms defined for example to satisfy eq. (13) or (14). Replacing f with f_h, we can write

$$\sum_i x_i a(\phi_i, \phi_j) = L(\phi_j), \tag{17}$$

and the solution can be found by solving the matrix equation

$$\mathrm{Ax} = \mathrm{b}. \tag{18}$$

The entries in the coefficient matrix A and the load vector b are given as

$$A_{ji} = a(\phi_i, \phi_j), \qquad b_j = L(\phi_j). \tag{19}$$

That the matrix equation in fact solves eq. (17) is readily seen by writing out the matrix product for any single row j: $\sum_i A_{ji} x_i = b_j$.

The trial functions ϕ_i are normally defined such that each is nonzero only on the patch of elements surrounding node i. Hence, the integrals that define the matrix elements in eq. (19) can be computed efficiently by numerical quadrature.

Much more can be said about the solvability of the problem, and about the quality of the solution, but the basic idea is given above. While a general implementation of the finite element method is quite complicated, a number of libraries or framework are available to support this task. Some, like DOLFIN [9], allow the problem to be specified by entering the weak form of the problem (along with the grid and boundary conditions) in a language quite close to the mathematical definition. In a lower-level library like Diffpack [6, 8], which we use, the library handles things like quadrature and matrix assembly automatically, but the developer must manually implement the inner loops over the basis functions of an element. Nonetheless, the translation of the mathematical definitions of sec. 3.3 to working code is quite natural, as the excerpt shown in listing 1 can attest to. This code implements the integrands for the p block-row of all the formulations (two-, three- and four-field) found in Paper III, and the resulting coefficient matrices can be used unchanged for solving the individual decoupled equations (using an iterative method), or for solving the equations partially or fully coupled. In all the papers that follow, the fully coupled solution method is used.

4.2 Linear algebra

Given a matrix equation, constructed for example by the finite element method as above,

$$\mathrm{Ax} = \mathrm{b}, \tag{20}$$

we need to solve it to get the unknown weights x. Since A may contain many million rows, a direct inversion $\mathrm{x} = \mathrm{A}^{-1}\mathrm{b}$ is usually not feasible, and an iterative method is used instead.

The efficiency of iterative methods, such as the Conjugate Gradient (CG) method, depends intimately on the condition number of the matrix. The condition number is (for a positive definite matrix) the ratio of the largest to the smallest eigenvalue,

Listing 1: The Diffpack/C++ implementation of the integrands of the block-row associated with the fluid pressure p (where both sides are multiplied by Δt). If the pure $\boldsymbol{u}$-p formulation is used, the pp block is defined by `IntegrandPP_pure` and the $p\boldsymbol{u}$ block by `IntegrandPU`; if the mixed $\boldsymbol{u}$-p-$\boldsymbol{v}_{\mathrm{D}}$ formulation is used, the pp block is defined by `IntegrandPP_mixed`, the $p\boldsymbol{u}$ block by `IntegrandPU`, and the $p\boldsymbol{v}_{\mathrm{D}}$ block by `IntegrandPV`. These integrands can be used, unmodified, for both scalar elements and for vector elements such as Crouzeix-Raviart, and with arbitrary coupling on the algebraic level.

```
void IntegrandPP_mixed::integrandsMx(ElmMatVec &elmat,
                                     const MxFiniteElement &fe)
{
  // eq: -S dp/dt

  for (int i=1; i<=nbf(iP); i++)
    for (int j=1; j<=nbf(iP); j++)
      elmat.A(i,j) -= cf.S * NP(i) * NP(j) * detJxW;

  const real P_prev = the_simulator->Pprev->valueFEM(fe(iP));

  for (int i=1; i<=nbf(iP); i++)
    elmat.b(i) -= cf.S * NP(i) * P_prev * detJxW;
}

void IntegrandPP_pure::integrandsMx(ElmMatVec &elmat,
                                    const MxFiniteElement &fe)
{
  IntegrandPP_mixed::integrandsMx(elmat, fe);

  // eq: \nabla \cdot \Lambda \nabla p, integrated by parts

  for (int i=1; i<=nbf(iP); i++)
    for (int j=1; j<=nbf(iP); j++) {
      real nabla2 = SUM(d,1,nsd, dNP(i,d) * dNP(j,d) * cf.Lambda(d));
      elmat.A(i,j) -= nabla2 * cf.dt * detJxW;
    }
}

void IntegrandPP_pure::integrands4sideMx(int side, int boind,
                                         ElmMatVec& elmat,
                                         const MxFiniteElement& fe)
{
  const real flux = the_simulator->geodata->get(FLUX, fe(iP));
  const real detSideJxW = fe.detSideJxW();

  for (int i=1; i<=nbf(iP); i++)
    elmat.b(i) -= flux * cf.dt * NP(i) * detSideJxW;
}

void IntegrandPU::integrandsMx(ElmMatVec &elmat,
                               const MxFiniteElement &fe)
{
  // eq: -\alpha \nabla \cdot du/dt

  for (int i=1; i<=nbf(iP); i++)
    for (int j=1; j<=nbf(iU); j++)
      for (int r=1; r<=nsd; r++)
        elmat.A(i,_(j,r)) -= cf.alpha * dNU(j,r) * NP(i) * detJxW;

  const real divU_prev = the_simulator->Uprev->divergenceFEM(fe(iU));

  for (int i=1; i<=nbf(iP); i++)
    elmat.b(i) -= cf.alpha * divU_prev * NP(i) * detJxW;
}

void IntegrandPV::integrandsMx(ElmMatVec &elmat,
                               const MxFiniteElement &fe)
{
  // eq: -\nabla \cdot v_D

  for (int i=1; i<=nbf(iP); i++)
    for (int j=1; j<=nbf(iV); j++)
      for (int r=1; r<=nsd; r++)
        elmat.A(i,_(j,r)) -= cf.dt * NP(i) * dNV(j,r) * detJxW;
}
```

and unfortunately it tends to grow with the problem size. The number of iterations typically increases proportionally with the number of elements in each space direction. To minimise this growth, we must in practice introduce a preconditioner P, and look for a solution to the equivalent problem,

$$\mathrm{P}^{-1}\mathrm{A}\mathrm{x} = \mathrm{P}^{-1}\mathrm{b}. \tag{21}$$

With a suitable preconditioner, the condition number of $\mathrm{P}^{-1}\mathrm{A}$ is much smaller than that of A alone (approaching 1 as P^{-1} approaches A^{-1}), and the product $\mathrm{P}^{-1}\mathrm{v}$ is fast to compute (for an arbitrary vector v).

General preconditioners, particularly for coupled systems, are an active research area to which we contribute in Papers I, II and IV.

4.3 Parallel calculations

It is commonly expected that the days of ever-faster sequential processors are past, and that in the near-term future, improvements must be gotten mainly through increased parallelism of the calculations. Luckily, the solution of PDEs by the finite element method are quite amenable to parallelisation due to the locality of most operators. A natural approach is to divide the computational grid between the available processors, making each processor responsible for only a small subgrid. The integration and assembly phases are purely local, and the challenge is to limit communication as much as possible in the algebraic solution phase. In a scheme such as the one outlined in Paper IV, each processor can be seen as responsible for the rows in the (virtual) global matrix that are associated with its own subgrid. The basic linear algebra operations are embellished a bit to use data from neighbouring subgrids when necessary (in matrix-vector products), and global data when necessary (vector norms and inner products). Some operations, like matrix-matrix products, are forbidden because of their complexity, but these are usually not required (and too costly even in a sequential calculation). By this procedure, the necessary local operations can be made to produce the same result as if the global operations were performed using the (virtual) global matrix.

The communication is managed explicitly by message passing (Message Passing Interface, or MPI), but for the most part this is performed automatically and transparently in the linear algebra library.

The outlined procedure is well known, and a number of parallel finite element and linear algebra libraries utilise similar methods. The major outstanding problem lies in the parallelisation of the preconditioner, because it is in the nature of an effective preconditioner that it must be a global operator (ideally, it approximates A^{-1} — a dense matrix).

Multigrid preconditioners, in particular algebraic multigrid (AMG) preconditioners, have been developed that perform very well in a parallel setting, scaling up to many thousand processors. Our work in this area is in efficiently combining AMG preconditioners for the decoupled equations (for solid displacement only, or for fluid pressure only) into an effective preconditioner for the coupled system. This work is performed in Paper II (in a sequential setting) and in Paper IV (in a parallel setting).

5 Summary of the Papers

The cycle of papers presented as part of this thesis in reality began with one that was never published.

In March 2009, I was about to finish my first major paper, on parallel techniques and scalability for a fully coupled thermoporoelastic basin model applied to large-scale, sedimentary basins with severe, realistic jumps in material parameters. At the last minute, a closer examination of the results revealed that our chosen solver (BiCGStab, using AMG preconditioning in a Block Jacobi configuration) did not actually converge. That is, the error turned out to remain large in realistic applications although the residual was small and common convergence criteria for iterative solvers were fulfilled. Consequently, these numerical methods faced a more fundamental problem. The paper was promptly submitted to the waste basket, and we turned our focus to two key questions:

- What are the properties of the basin model that make it difficult to solve?
- What are the remedies for handling these difficulties?

These two questions led to the research reported in Papers I–IV.

5.1 Paper I: On the performance of an algebraic multigrid preconditioner for the pressure equation with highly discontinuous media

The first paper, presented at the Mek'IT Conference for Computational Mechanics in Trondheim in May of 2009, looked at solving the decoupled pressure equation with large permeability contrasts.

The continuum decoupled fluid pressure equation is pysically undefined in the limit of vanishing permeability. Since there is no longer any spatial coupling between points, any pressure solution is equally valid inside the impermeable area.

In a numerical approximation the boundaries are more iffy. We looked at what happens when the permeability in parts of the domain approaches (but not reaches) zero. Through analysing the eigenvalues of the coefficient matrix, we discovered that the number of eigenvalues approaching zero is identical to the number of nodes within the low-permeable region; these are the ill-defined values. Interestingly, the AMG preconditioner is "perfect" in a certain sense on this problem: The preconditioned coefficient matrix has mostly eigenvalues of order unity, except for a single eigenvalue for each high-permeable region that is almost isolated inside a low-permeable region. This reflects the fact that the pressure inside each such isolated region is only decided up to an arbitrary constant.

This result makes it possible to solve such nearly indeterminate problems with iterative solvers, although with a large uncertainty in the pressure associated with each nearly isolated region.

5.2 Paper II: Efficient block preconditioners for the coupled equations of pressure and deformation in highly discontinuous media

The uncertainty (or, more generally, the ill-posedness) encountered in Paper I disappears when the fluid pressure is coupled with the displacement of the porous medium. The numerical difficulties, however, do not.

In this paper, which has been accepted for publication in the *International Journal of Analytical and Numerical Methods in Geomechanics*, we consider how to precondition the coupled equations of fluid pressure and solid elastic displacement. Perhaps due to the nearly vanishing eigenvalues of the decoupled preconditioner, we found the need to use a preconditioner which includes the fluid-solid coupling.

The paper discusses and tests a number of block preconditioners which are based on AMG preconditioners of the decoupled blocks (or modifications thereof). We identify two good preconditioners, one symmetric and one asymmetric, both based on an exact block decomposition of the original system by way of the Schur pressure complement. While the ideas for these two preconditioners have been presented elsewhere, we believe that the actual application and comprehensive testing of the symmetric variant on the poroelastic equations is novel. Furthermore, the identification of these two preconditioners as two variants of the same basic family of preconditioners is, as far as we know, original to this paper.

5.3 Paper III: On the causes of pressure oscillations in low-permeable and low-compressible porous media

As a slight detour, we delve into one other artifact of the numerical solution of Biot's equations in the presence of low-permeable materials. It has long been known that pressure oscillations may occur in the discrete solution, oscillations that have no basis in the physical realm. There have been, however, some differences as to why the oscillations occur, and how to avoid them. In this paper, which has been accepted for publication in the *International Journal of Analytical and Numerical Methods in Geomechanics*, we try to understand the situation through a bit of analysis backed up by extensive numerical experiments. For this purpose, we formulate four different versions of Biot's equation.

The main result of this paper is a guideline for the choice of finite elements in the different cases and with different formulations.

5.4 Paper IV: A parallel block preconditioner for large scale poroelasticity with highly heterogeneous material parameters

In Paper II, we developed and tested robust block preconditioners for Biot's equations in "difficult" cases. We did consider and mention the parallel scalability of these preconditioners, but we did not have time or space to expound fully on that subject.

Parallelisation is, as we have argued, a vital component in the solvability of large-scale problems in coupled geomechanics.

Paper IV, which has been submitted to *Computational Geosciences*, rectifies this problem by implementing and testing the symmetric variant of the preconditioner on parallel computers. Thus we finally demonstrate the ability to solve the original large-scale basin simulation that prompted this investigation in the first place.

6 Future Work

The present geophysical model has — as a physical approximation — some limitations. It assumes a linear elastic response in the small strain regime, which may be valid for short time periods and low stresses. In geology, however, one does not have to look far before nonlinear processes, such as plastic or viscoelastic deformation, become important. Plastic processes generate heat, and heat may be of importance in other scenarios involving for example magmatic intrusions. All of these processes are more time-consuming to model, but we believe the numerical work undertaken herein will still prove useful as a foundation.

A natural next step, given the motivation of making simulation of basin-scale models feasible, would be to further strengthen the integration with industry models. Such stronger integration would enable the comparison of simulation methods to better assess the advantages that a fully coupled formulation in various scenarios, and hence to learn more about the importance of different geomechanical mechanisms.

Finally, the link which is made to low-compressible analysis in Paper III may be explored further in the context of analysis of salt movement. Since salt is nearly incompressible, standard Galerkin methods of modelling are insufficient; but a mixed finite element method which includes the solid volumetric pressure as an independent field variable makes such analysis possible.

Bibliography

[1] K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, 1996.

[2] M. A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12(2):155–164, 1941. doi: 10.1063/1.1712886.

[3] S. Boggs, Jr. *Principles of Sedimentology and Stratigraphy*. Prentice Hall, 3rd edition, 2001. ISBN 0-13-099696-3.

[4] O. Coussy. *Poromechanics*. John Wiley and Sons, 2004.

[5] R. H. Dean, X. Gai, C. M. Stone, and S. E. Minkoff. A comparison of techniques for coupling porous flow and geomechanics. *SPE Journal*, 11(1):132–140, Mar. 2006. doi: 10.2118/79709-PA.

[6] Diffpack. URL `http://www.diffpack.com/`. Library for numerical solution of PDEs from inuTech GmbH.

[7] T. Hantschel and A. I. Kauerauf. *Fundamentals of basin and petroleum systems modeling*. Springer-Verlag, 2009. doi: 10.1007/978-3-540-72318-9.

[8] H. P. Langtangen. *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming*. Springer, 2nd edition, 2003.

[9] A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2):20:1–20:28, 2010. doi: 10.1145/1731022.1731030.

[10] PetroMod. URL `http://www.petromod.com/`. Petroleum systems modelling software from Schlumberger Aachen Technology Center.

[11] R. E. Showalter. Diffusion in deforming porous media. *Dyn. Cont. Discr. Impuls. Syst. (Series A: Math. Anal.)*, 10(5):661–678, 2003.

[12] J. Tarney. Plate tectonics: Geological aspects. URL `http://www.le.ac.uk/geology/art/gl209/`.

[13] K. Terzaghi. Die Berechnung der Durchlassigkeitsziffer des Tones aus dem Verlauf der hydrodynamischen Spannungserscheinungen. *Sitzungsberichte der Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse, Abteilung IIa*, 132:125–138, 1923.

[14] H. F. Wang. *Theory of Linear Poroelasticity with Applications to Geomechanics and Hydrogeology*. Princeton University Press, 2000.

Paper I

On the performance of an algebraic multigrid preconditioner for the pressure equation with highly discontinuous media

Paper II

Efficient block preconditioners for the coupled equations of pressure and deformation in highly discontinuous media

Efficient block preconditioners for the coupled equations of pressure and deformation in highly discontinuous media

J. B. Haga[1,3], **H. Osnes**[2,3], **H. P. Langtangen**[2,4]

[1] Computational Geosciences, Simula Research Laboratory
PO Box 134, N-1325 Lysaker, Norway

[2] Center for Biomedical Computing, Simula Research Laboratory
PO Box 134, N-1325 Lysaker, Norway

[3] Department of Mathematics, University of Oslo,
PO Box 1053 Blindern, N-0316 Oslo, Norway

[4] Department of Informatics, University of Oslo,
PO Box 1080 Blindern, N-0316 Oslo, Norway

Abstract

Large-scale simulations of flow in deformable porous media require efficient iterative methods for solving the involved systems of linear algebraic equations. Construction of efficient iterative methods is particularly challenging in problems with large jumps in material properties, which is often the case in geological applications, such as basin evolution at regional scales. The success of iterative methods for this type of problems depends strongly on finding effective preconditioners.

This paper investigates how the block-structured matrix system arising from single-phase flow in elastic porous media should be preconditioned, in particular for highly discontinuous permeability and significant jumps in elastic properties. The most promising preconditioner combines algebraic multigrid with a Schur complement-based exact block decomposition. The paper compares numerous block preconditioners with the aim of providing guidelines on how to formulate efficient preconditioners.

1 Introduction

Common problems of important industrial and scientific interest in coupled geomechanics include basin modelling, reservoir management, and groundwater depletion. Analysis of such models on a regional scale requires the ability to solve coupled equations with a large number of unknowns, complex geometries and significant spatial variation in the material parameters. To meet the challenge of efficient solution of these models, scalable solvers that are robust with respect to the geometry and discontinuities of realistic problems must be developed. This is addressed in the present paper.

The problem of interest couples single-phase fluid flow with deformation in elastic porous media. This problem is described by a pair of partial differential equations (PDEs), one governing the fluid pressure and one describing the deformation of the porous matrix. Terzaghi [33] developed the original theory of uniaxial soil consolidation, and introduced the ideas of effective stress and the diffusion of fluid pressure by fluid flow. Biot [6] generalised this work to three dimensions and derived the PDEs governing coupling of fluid flow and deformation in linear elastic porous media. The necessity of a hydromechanically coupled formulation has been validated in field and laboratory studies [7, 20, 21]; see Neuzil [25] for an overview. A review of modelling of such systems can be found in [31], while [37] offers a comprehensive modern treatment. In this paper, we apply Biot's equations to a series of test cases and study the efficiency of preconditioned iterative solution methods.

In solvers for algebraic systems of equations, such as those arising from discretisations of PDEs, there is a trade-off between robustness and scalability. Direct solvers are generally the most robust with respect to the numerical properties of the equations, and as a result they have become popular in "difficult" finite element applications. However, they suffer from suboptimal scaling in time and space. The memory requirements in particular grow substantially faster than the number of unknowns in the problem [14]. Furthermore, communication requirements limit parallel scalability [12]. Iterative methods are in contrast highly scalable, but less robust. Their convergence is problem-dependent and sensitive to the parameters of the problem. Even so, their efficiency makes them the only choice for truly large-scale problems.

The number of iterations in Krylov space methods, such as the Conjugate Gradient (CG [18]) or Stabilised Bi-Conjugate Grandient (BiCGStab [36]) methods, for solving a system $\mathcal{A}\mathrm{x} = \mathrm{b}$ is typically proportional to $\sqrt{\kappa}$, where κ is the condition number of the coefficient matrix $\mathcal{A}$ [15]. By applying a preconditioner $\mathcal{P}^{-1}$ to the system, i.e., solving $\mathcal{P}^{-1}\mathcal{A}\mathrm{x} = \mathcal{P}^{-1}\mathrm{b}$, one can reduce the condition number and obtain faster convergence. It is in the nature of the finite element method that the condition number of the coefficient matrix increases when the number of unknowns increases — typically, $\kappa \sim \mathcal{O}(h^{-2})$, where h is the characteristic element length [32]. Using a multigrid method as preconditioner, the condition number can in many cases be made independent of the number of unknowns, a property which is referred to as an optimal method because the amount of work per unknown is then independent of the problem size [3].

Multigrid methods have attracted quite some interest as efficient and widely applicable preconditioners [5, 38]. A difficulty with the standard geometric multigrid method is that it needs a hierarchy of coarse grids. This can be difficult to construct in problems

with complicated geometries and many internal layers of materials, which is the typical case in geological applications. Algebraic multigrid (AMG [29]) is then a promising alternative, since it relies only on the algebraic structure of the coefficient matrix. Previous studies [1, 16] indicate that AMG preconditioning can remove the dependence of the number of iterations on the number of unknowns when solving the individual PDEs in Biot's model. How AMG can be used to efficiently precondition the coupled systems of equations studied herein is, however, an open question, which we address in the present paper.

There are basically two main categories of preconditioners for coupled systems. The first category addresses the system of algebraic equations that arises from numbering the displacement and pressure degrees of freedom consecutively in each node. Such numberings may minimize the bandwidth for banded solvers or the fill-in for direct sparse solvers. The other category is aimed at systems where all the displacement degrees of freedom are numbered first, followed by the pressure degrees of freedom. This numbering gives rise to a coefficient matrix with a block structure that more directly corresponds to the original system of PDEs (e.g., the first row of blocks corresponds to the first PDE and so forth). Block preconditioners rely on creating separate preconditioners for the individual decoupled equations, and combining these to precondition the coupled system. While simple blockwise methods such as block diagonal (or block Jacobi) preconditioning work well on some coupled problems [23], saddle-point problems (for example) require the application of Schur complement based methods, owing to non-invertible diagonal blocks. Schur complement based block preconditioners have also been found to work well on the discretisation of Biot's equations [28, 34], although only homogeneous materials were tested. To our knowledge, the efficacy of block preconditioners for Biot's equations with strongly varying material parameters has not been evaluated.

The main physical parameters that influence the evolution of Biot's equations are the elastic parameters and the permeability of the porous matrix. The permeability in particular may exhibit significant jumps of many orders of magnitude in geological applications [4, 24, 37]. This feature may have a severe impact on the performance of numerical methods for solving Biot's equations. Since there is effectively no flow through the low-permeable regions, the use of tailored techniques such as solving for the pressure on only the high-permeable part of the grid is common. In practice, however, this requires either solving for an additional vector variable for the fluid flux in a mixed finite element formulation, or the manipulation of two separate grid solutions for pressure and displacement. Hence, numerical methods that allow the efficient solution of arbitrary permeability differences without special considerations are attractive.

We assume that the governing differential equations are discretised by a Galerkin finite element method using mixed elements, and study the effect that a large jump in the permeability and a moderate jump in the elastic parameters (consistent with typical geological media) has on the preconditioned iterative solvers. The permeability is parameterised by a factor $\epsilon \ll 1$, meaning that we basically consider a domain with two types of geological media: one with flow mobility (which is proportional to the permeability) Λ_0 and one with flow mobility $\epsilon\Lambda_0$. The typical jump in permeability is then described by a factor $1/\epsilon \gg 1$. The investigations are further extended to the case where the two media have different elastic parameters.

The impact of the jump ϵ^{-1} on the accuracy of the finite element discretisation is not critical as long as the discontinuities are aligned with the element boundaries [26], which we assume in the following. The critical numerical impact of the discontinuities is then on the performance and convergence of solution methods for the coupled linear system $[\mathrm{p\ u}]^\mathsf{T} = \mathcal{A}^{-1}\mathrm{b}$ arising from the discretisation, where p and u denote the pressure and displacement solution vectors, respectively.

The present paper studies the numerical convergence of an AMG-preconditioned conjugate gradient-type method applied to the linear system arising from the coupled equations of pressure and displacement in porous media. Our aim is to extend common knowledge from earlier work by investigating a series of test cases and iterative solvers for the coupled problem, with varying degree of discontinuity in the material parameters. We hope that our findings can guide practitioners in how to choose efficient solution methods for large-scale simulations involving coupled geomechanical problems and highly discontinuous media.

2 The mathematical model

The equations describing poroelastic flow and deformation can be derived from the principles of conservation of fluid mass and the balance of forces on the porous matrix. The linear poroelastic can be expressed, in the small-strains regime, as

$$S\dot{p} - \nabla \cdot \mathbf{\Lambda} \nabla p + \alpha \nabla \cdot \dot{\mathbf{u}} = q, \tag{1}$$

$$\nabla(\lambda + \mu)\nabla \cdot \mathbf{u} + \nabla \cdot \mu \nabla \mathbf{u} - \alpha \nabla p = \mathbf{r}. \tag{2}$$

Here, we subsume body forces such as gravitational forces into the right-hand side source terms q and $\mathbf{r}$. The primary variables are p for the fluid pressure and $\mathbf{u}$ for the displacement of the porous medium, S and $\mathbf{\Lambda}$ are the fluid storage coefficient and the flow mobility respectively, α is the Biot-Willis fluid/solid coupling coefficient, and λ and μ are the Lamé elastic parameters.

As pointed out in the introduction, the aim of the present paper is to study the numerical properties of eqs. (1)–(2), and how to solve these efficiently with an iterative solver. To that end, we ignore effects that are not essential to these properties. The fluid-solid coupling coefficient α is treated as a constant (in practice it varies between about 0.5 and 1). The fluid storage coefficient S is insignificant compared to the fluid mobility in high-permeable regions. In low-permeable regions it acts as an effective fluid compressibility term, and makes the problem less numerically stiff for short time steps. By dropping this term we try to ensure that the validity of the testing is not compromised by choosing a too short ("easy") time step. The other time-derivative term, $\nabla \cdot \dot{\mathbf{u}}$ in eq. (1), couples the displacement to the pressure and *is* included.

We employ a first-order backward finite difference method in time. Our simplified model problem is thus

$$-\Delta t \nabla \cdot \mathbf{\Lambda} \nabla p + \nabla \cdot \mathbf{u} = q\Delta t + \nabla \cdot \mathbf{u}_{k-1}, \tag{3}$$

$$\nabla(\lambda + \mu)\nabla \cdot \mathbf{u} + \nabla \cdot \mu \nabla \mathbf{u} - \nabla p = \mathbf{r}, \tag{4}$$

where variables without subscripts are taken to be at the current time step k. Moreover, we restrict $\mathbf{\Lambda}$ to be isotropic, parameterised by $\epsilon \leq 1$, so that $\mathbf{\Lambda} = \Lambda_0 \mathbf{I}$ in the high-permeable region and $\mathbf{\Lambda} = \epsilon \Lambda_0 \mathbf{I}$ in the low-permeable region, with $\mathbf{I}$ being the identity tensor.

2.1 Numerical approximation

We proceed to rewrite eqs. (3) and (4) in weak form, using integration by parts to eliminate second derivatives. The following relations must then be satisfied for all test functions π and $\mathbf{w}$ in the domain Ω:

$$\int_\Omega [\Delta t \nabla \pi \cdot \mathbf{\Lambda} \nabla p + \pi \nabla \cdot \mathbf{u}] \, \mathrm{d}\Omega = \int_\Omega [\pi \nabla \cdot \mathbf{u}_{k-1} + \pi q \Delta t] \, \mathrm{d}\Omega - \int_\Gamma \pi f_\mathbf{n} \Delta t \, \mathrm{d}\Gamma, \tag{5}$$

$$\int_\Omega [(\nabla \cdot \mathbf{w})(\lambda + \mu)(\nabla \cdot \mathbf{u}) + \nabla \mathbf{w} : \mu \nabla \mathbf{u} - (\nabla \cdot \mathbf{w}) p] \, \mathrm{d}\Omega = - \int_\Omega \mathbf{w} \cdot \mathbf{r} \, \mathrm{d}\Omega + \int_\Gamma \mathbf{w} \cdot \mathbf{t}_\mathbf{n} \, \mathrm{d}\Gamma. \tag{6}$$

The fluid flux $f_\mathbf{n}$ and normal stress $\mathbf{t}_\mathbf{n}$ at the boundary Γ appear here as natural boundary conditions.

The discrete finite element approximation follows from solving eqs. (5) and (6) in finite-dimensional spaces. In this paper, a piecewise (triangular) continuous quadratic space is used for the deformation and a piecewise continuous linear space is used for the pressure,

$$p, \pi \in P^1(\Omega), \quad \mathbf{u}, \mathbf{w} \in [P^2(\Omega)]^d, \tag{7}$$

with dimensionality $d = 2$. The reason for this mix of spaces is that spurious pressure oscillations can occur in low-permeable regions when the same spaces are used for pressure and deformation [22, 27].

2.2 The algebraic system

The algebraic system that results from discretising eqs. (5)–(6) is on the form

$$\mathcal{A}\mathrm{x} = \mathrm{b}, \tag{8}$$

where $\mathcal{A}$ is the coefficient matrix derived from the left-hand sides of eqs. (5) and (6), b is the load vector arising from the right-hand sides, and x is the unknown solution vector. Since this is a coupled system of two equations, the coefficient matrix is a 2×2 block matrix

$$\mathcal{A} = \begin{bmatrix} \mathrm{A_{uu}} & \mathrm{A_{up}} \\ \mathrm{A_{pu}} & \mathrm{A_{pp}} \end{bmatrix}, \tag{9}$$

where the subscripts denote the primary variable(s) each block acts upon: $\mathrm{A_{pp}}$ couples pressure to pressure, $\mathrm{A_{pu}}$ couples displacement to pressure, et cetera. The solution and load vectors are given as $\mathrm{x} = [\mathrm{u\ p}]^\mathsf{T}$ and $\mathrm{b} = [\mathrm{b_u\ b_p}]^\mathsf{T}$. The sign of the equations can be chosen so as to make this a symmetric indefinite problem, which we write as

$$\mathcal{A} = \begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^\mathsf{T} & \mathrm{C} \end{bmatrix}, \tag{10}$$

where A is symmetric positive definite and C is symmetric negative definite.

3 Block preconditioning methods

Since the convergence rate of iterative solvers depends on the numerical properties — the condition number in particular, but also the eigenvalue distribution — of the coefficient matrix, a preconditioner is in most cases required to achieve a satisfactory convergence rate. In general, the preconditioner $\mathcal{P}^{-1}$ should be fast to compute and close to $\mathcal{A}^{-1}$, although the latter is not a necessary condition. In fact, a better (although somewhat circular) requirement is that it gives $\mathcal{P}^{-1}\mathcal{A}$ a beneficial eigenvalue distribution. For the Krylov family of iterative solvers, the exact meaning of "beneficial" is somewhat complicated, but having a small number of tight eigenvalue clusters often leads to rapid convergence [30].

We assume for the moment the availability of good preconditioners for the symmetric definite decoupled problems. These can be formed by, e.g., multigrid or incomplete factorisation methods; we shall discuss these in a later section. The question then is: how can these be combined to an effective preconditioner for the *coupled* Biot's equations? We briefly present here the motivation for the block preconditioners that are chosen for the numerical experiments.

Given a nonsingular 2×2 block matrix

$$\mathcal{A} = \begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^{\mathsf{T}} & \mathrm{C} \end{bmatrix}, \tag{11}$$

such as that in eq. (9), we focus on block preconditioners of $\mathcal{A}$, i.e, those that can be written on the form

$$\mathcal{P}^{-1} = \begin{bmatrix} \mathrm{M} & \mathrm{N} \\ \mathrm{P} & \mathrm{Q} \end{bmatrix}. \tag{12}$$

For example, the standard block Jacobi and block Gauß-Seidel preconditioners can be expressed as

$$\mathcal{P}_{\mathrm{sJ}}^{-1} = \begin{bmatrix} \tilde{\mathrm{A}}^{-1} & 0 \\ 0 & \tilde{\mathrm{C}}^{-1} \end{bmatrix} \quad \text{and} \quad \mathcal{P}_{\mathrm{sGS}}^{-1} = \begin{bmatrix} \tilde{\mathrm{A}}^{-1} & 0 \\ 0 & \tilde{\mathrm{C}}^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{I} & 0 \\ -\mathrm{B}^{\mathsf{T}}\tilde{\mathrm{A}}^{-1} & \mathrm{I} \end{bmatrix}, \tag{13}$$

respectively, where $\tilde{\mathrm{A}}^{-1}$ and $\tilde{\mathrm{C}}^{-1}$ are approximations to the inverses of the diagonal blocks in eq. (11), i.e., to the inverses of the decoupled equations.

Furthermore, when A is nonsingular, the associated Schur complement of $\mathcal{A}$ is

$$\mathrm{S} = \mathrm{B}^{\mathsf{T}}\mathrm{A}^{-1}\mathrm{B} - \mathrm{C}. \tag{14}$$

It is then easily verified that the exact inverse of $\mathcal{A}$ can be written as

$$\mathcal{A}^{-1} = \begin{bmatrix} \mathrm{I} & -\mathrm{A}^{-1}\mathrm{B} \\ 0 & \mathrm{I} \end{bmatrix} \begin{bmatrix} \mathrm{A}^{-1} & 0 \\ 0 & -\mathrm{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{I} & 0 \\ -\mathrm{B}^{\mathsf{T}}\mathrm{A}^{-1} & \mathrm{I} \end{bmatrix}, \tag{15}$$

with S defined as in eq. (14). Using this block decomposition as the basis of a preconditioner for symmetric indefinite systems was proposed by Toh *et al.* [34]. Eq. (15) can also be viewed as a *symmetric* block Gauß-Seidel preconditioner, where C^{-1} is replaced by $-\mathrm{S}^{-1}$ as the $(2,2)$ block. This is seen by comparing eqs. (13) and (15). We generalise this observation by defining the preconditioning basis of $\mathcal{A}$ as

$$\mathcal{A}_{\mathrm{prec}} = \begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^{\mathsf{T}} & \mathrm{D} \end{bmatrix}, \tag{16}$$

where the D block may be replaced by, e.g., the original (C), which leads to the standard block preconditioners in eq. (13); or the negative Schur complement $(-\mathrm{S})$, which produces the Schur complement preconditioners based on eq. (15). We have tested preconditioners using both of these bases, as well as one using an ϵ-capped modification of C, in our numerical experiments.

Another Schur complement based preconditioner was evaluated in a homogeneous context by Phoon *et al.* [28], where the Generalised Jacobi preconditioner was defined (in un-inverted form) as

$$\mathcal{P}_{\mathrm{gJ}(\alpha)} = \begin{bmatrix} \tilde{\mathrm{A}} & 0 \\ 0 & \alpha\tilde{\mathrm{S}} \end{bmatrix}, \tag{17}$$

where $\tilde{\mathrm{A}}$ and $\tilde{\mathrm{S}}$ are approximations to the exact $(1,1)$ block and the Schur complement, respectively. The Generalised Jacobi preconditioner is equivalent to a block Jacobi preconditioner with $D = \alpha S$. Phoon *et al.* argue that while the choice of α is not significant when the exact $(1,1)$ block $\tilde{\mathrm{A}} = \mathrm{A}$ is used, a negative value for α performs better when a cruder approximation is used. It was shown that this preconditioner leads to an attractive eigenvalue distribution, with three distinct eigenvalue clusters around 1 and $(1 \pm \sqrt{1 + 4/\alpha})/2$, each with diameter of order $\|\mathrm{S}^{-1}\mathrm{C}\|$. Although this theoretical result depends on the exact inversion of eq. (17), the practical applicability of a diagonal approximation with $\alpha = -4$ was demonstrated.

An interesting question, when utilising a symmetric preconditioner such as one based on eq. (15), is whether the preconditioned coefficient matrix is positive definite. If it is, then the Conjugate Gradient method can be used instead of indefinite methods such as BiCGStab. We can define the "approximate identities" generated by $\tilde{\mathrm{A}}^{-1}$ and $\tilde{\mathrm{S}}^{-1}$ as

$$\tilde{\mathrm{I}}_{\mathrm{A}} = \tilde{\mathrm{A}}^{-1}\mathrm{A}, \tag{18}$$

$$\tilde{\mathrm{I}}_{\mathrm{S}} = \tilde{\mathrm{S}}^{-1}(\mathrm{B}^{\mathsf{T}}\tilde{\mathrm{A}}^{-1}\mathrm{B} - \mathrm{C}). \tag{19}$$

Both approach the identity matrix I (of the appropriate dimension) as $\tilde{\mathrm{A}}^{-1}$ and $\tilde{\mathrm{S}}^{-1}$ approach the real inverses, and both are symmetric positive definite as long as the single-block preconditioners are. The preconditioned coefficient matrix, which can be written as

$$\mathcal{P}_{\mathrm{gSGS}}^{-1}\mathcal{A} = \begin{bmatrix} \tilde{\mathrm{I}}_{\mathrm{A}} & 0 \\ 0 & \tilde{\mathrm{I}}_{\mathrm{S}} \end{bmatrix} + \begin{bmatrix} \tilde{\mathrm{A}}^{-1}\mathrm{B}\tilde{\mathrm{S}}^{-1}\mathrm{B}^{\mathsf{T}} & \tilde{\mathrm{A}}^{-1}\mathrm{B} \\ -\tilde{\mathrm{S}}^{-1}\mathrm{B}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{I} - \tilde{\mathrm{I}}_{\mathrm{A}} & 0 \\ 0 & \mathrm{I} - \tilde{\mathrm{I}}_{\mathrm{S}} \end{bmatrix}, \tag{20}$$

then also approaches the identity, and the problem is trivially solved. Of more practical interest is under what circumstances eqs. (18) and (19) are close enough to the identity

Table 1: Number of applications of the single-block operations for one application of the block preconditioner.

	$\tilde{\mathrm{A}}^{-1}\mathrm{x}$	$\tilde{\mathrm{D}}^{-1}\mathrm{x}$	Bx	$\mathrm{x}+a\mathrm{y}$
Block Jacobi	1	1	0	0
Block Gauß-Seidel	1	1	1	1
Symmetric Block Gauss-Seidel	2	1	2	2

such that eq. (20) is ensured to be positive definite. Since the preconditioned matrix is symmetric,[4] its eigenvalues are on the real axis. The question is whether they are positive. The eigenvalue distribution of eq. (20) (as well as the non-symmetric Gauß-Seidel variant of the same) was analysed in [34][5]. In particular, it was found that the eigenvalues are not guaranteed to be positive unless all eigenvalues of $\tilde{\mathrm{A}}^{-1}\mathrm{A}$ are greater or equal to one, which is typically not the case for efficient single-block preconditioners. Hence, $\mathcal{P}_{\mathrm{gSGS}}^{-1}\mathcal{A}$ is not necessarily positive definite; but since all eigenvalues approach unity in the limit of exact single-block preconditioners, it clearly is if these are sufficiently accurate. The utility of transforming a symmetric indefinite system into a positive definite one was demonstrated in [8], wherein a preconditioner was explicitly designed to transform the system of equations into a positive definite one, solvable by Conjugated Gradients.

3.1 Computational cost

The computational cost of the preconditioner can be divided in two parts. First, the construction of the preconditioner involves, in addition to the cost of constructing the single-block preconditioners, the creation of the D block of the preconditioning basis in eq. (16). If this involves a modified version of the model equations, the cost is that of an extra finite element assembly. The Schur complement can be very costly to construct, but a reasonable approximation (as we shall see, the one used in this paper) can be created at roughly the cost of three single-block matrix-vector products. This is cheaper than a single iteration of the BiCGStab iterative method.

Second, each application of the block preconditioner results in a number of single-block operations, which is listed in table 1. This cost is incurred twice for each iteration in BiCGStab, or once per iteration with CG. For comparison, the 2×2 block BiCGStab iteration also involves eight matrix-vector products (two for each block), twelve vector additions, and eight inner products.

[4]Strictly speaking, it is the spectrally equivalent matrix $\mathcal{E}^{-1}\mathcal{A}\mathcal{E}^{-\mathsf{T}}$ with $\mathcal{P} = \mathcal{E}^{\mathsf{T}}\mathcal{E}$ that is symmetric.
[5]In the reference, these are called the "constrained" and "block triangular" preconditioners.

4 Numerical investigations

4.1 Block preconditioners

In our numerical investigations we compare the performance of ten block preconditioners in combination with the BiCGStab method and one with the CG method. These are selected from the combinations of five different preconditioning bases with three different blocking schemes.

We define the lower-triangular coupling matrix as

$$\mathcal{G} = \begin{bmatrix} \mathrm{I} & 0 \\ -\mathrm{B}^{\mathsf{T}}\tilde{\mathrm{A}}^{-1} & \mathrm{I} \end{bmatrix}. \tag{21}$$

The blocking schemes are then, with reference to the definition of $\mathcal{A}_{\text{prec}}$ in eq. (16), the block Jacobi preconditioning scheme,

$$\mathcal{P}_1^{-1} = \begin{bmatrix} \tilde{\mathrm{A}}^{-1} & 0 \\ 0 & \tilde{\mathrm{D}}^{-1} \end{bmatrix}, \tag{22}$$

where $\tilde{\mathrm{A}}^{-1}$ and $\tilde{\mathrm{D}}^{-1}$ are (in some sense) close to the real inverses; the block Gauß-Seidel preconditioning scheme

$$\mathcal{P}_2^{-1} = \mathcal{P}_1^{-1}\mathcal{G}; \tag{23}$$

and the symmetric block Gauß-Seidel variant

$$\mathcal{P}_3^{-1} = \mathcal{G}^{\mathsf{T}}\mathcal{P}_1^{-1}\mathcal{G}. \tag{24}$$

Note that when $\mathrm{D} = -\mathrm{S}$, eq. (15) is approximated by $\mathcal{P}_3^{-1}$.

The $(2,2)$ block in the preconditioning bases are $\mathrm{D} = \mathrm{C}$ (the "standard" basis), $\mathrm{D} = \alpha\tilde{\mathrm{S}}$ (approximate Schur complement, or "generalised", basis), and $\mathrm{D} = \mathrm{C}_{\epsilon\geq 10^{-4}}$ (capped-ϵ basis). In the latter, the coefficient matrix of a more regular problem, with ϵ capped to nowhere be smaller than 10^{-4}, is used in the basis. This particular value of ϵ was chosen after some experimentation.

The selected combinations are then: The standard basis combined with all three blocking schemes; the Schur complement (generalised) basis with $\alpha = -1$, combined with all three blocking schemes; the Schur complement (generalised) basis with $\alpha = 1$ and $\alpha = \pm 4$, combined with block Jacobi; and the capped-ϵ basis combined with block Jacobi (symmetric block Gauß-Seidel was also tested, but it was not observed to bring any advantages over the Jacobi variant). Finally, the $\alpha = -1$ generalised basis with the symmetric Gauß-Seidel scheme is tested in combination with the Conjugate Gradient method. These combinations are summarised in table 2 along with their abbreviations.

4.2 The single-block preconditioners

The block preconditioners in the previous section depend on the availability of efficient single-block preconditioners $\tilde{\mathrm{A}}^{-1}$ and $\tilde{\mathrm{D}}^{-1}$. We restrict our attention to preconditioners which have the property of being efficient on massively parallel computers. This rules

Table 2: Abbreviations used for the tested preconditioners. These have a three-part structure: The block basis (standard, generalised or capped) in lower case, followed by the preconditioning scheme (Jacobi, Gauß-Seidel or Symmetric Gauß-Seidel), and optionally followed by the variant (the value of α in the generalised Jacobi preconditioners, or the "cg" postfix where the Conjugate Gradient method is used). With the exception of gSGS/cg, all preconditioners are used with the BiCGStab iterative solver.

	Standard	Capped	Generalised	
	$\mathrm{D} = \mathrm{C}$	$\mathrm{D} = \mathrm{C}_{\epsilon \geq c}$	$\mathrm{D} = -\tilde{\mathrm{S}}$	$\mathrm{D} = \alpha\tilde{\mathrm{S}}$
Block Jacobi	sJ	cJ	gJ(−1)	gJ(1), gJ(±4)
Block Gauß-Seidel	sGS		gGS	
Symmetric Block Gauß-Seidel	sSGS		gSGS	
(... with Conjugated Gradients)			gSGS/cg	

out incomplete and approximate direct solvers such as the otherwise excellent ILU methods.

Adams [1] found algebraic multigrid (AMG) to behave very well on problems of elastic deformation, even in the presence of strong material discontinuities. In particular, the smoothed aggregation method [9, 35] was considered to be the overall superior AMG method for elasticity problems. The present authors likewise found AMG to be a nearly optimal preconditioner for the discontinuous Poisson pressure problem, as long as the low-permeable regions do not completely isolate any high-permeable regions [16]. In the limit of $\epsilon \to 0$, such isolation would in fact create a physically indeterminate problem. When coupled with deformation of the solid matrix, however, the problem becomes well-posed both physically and — as we shall see — numerically.

In the light of these earlier results, and the fact that AMG has been shown to scale very well in parallel, to at least thousands of processors [2, 10, 19, 38], we have chosen to use AMG for both the pressure and the displacement equation. As for the other preconditioning bases, both $\alpha\mathrm{S}$ and $\mathrm{C}_{\epsilon \geq 10^{-4}}$ are modifications of the single block in the preconditioning basis associated with the pressure equation, and AMG is used also to approximate the inverses of these.

4.3 Approximating the Schur complement

The Schur complement in eq. (14) is a dense matrix, and as such it is neither feasible nor desirable to compute. While a number of sparse approximations to S are possible, one approximation that is very fast to compute[6] is

$$\tilde{\mathrm{S}}_1 = \mathrm{diag}(\mathrm{B}^{\mathsf{T}}(\mathrm{diag}\,\mathrm{A})^{-1}\mathrm{B}) - \mathrm{C}. \tag{25}$$

This is the approximation used in the numerical experiments in this paper. The entries of $\tilde{\mathrm{S}}$ are simply $\tilde{S}_{ij} = \delta_{ij}\sum_k (A_{kk})^{-1}(B_{ki})^2 - C_{ij}$. When the matrices are stored in the CRS (compressed row storage) representation, this makes the calculation extremely

[6] In particular, this matrix can be calculated with minimal or no interprocess communication on a parallel computer.

cheap: a sequential traversal of three matrices plus arbitrary accesses into the diagonal of A.

More accurate approximations to the Schur complement can be calculated. Toh *et al.* [34] evaluated a number of approximations in the context of iterative solution of Biot's equations, and found the simple approximations to be effective. This matches our experience: In addition to the approximation in eq. (25), we also looked at a slightly more accurate variation,

$$\tilde{\mathrm{S}}_2 = \mathrm{B}^\mathsf{T}(\operatorname{diag}\mathrm{A})^{-1}\mathrm{B} - \mathrm{C}, \tag{26}$$

but no improvement was observed (the performance in initial testing was in fact slightly worse). Other variants, such as using a sparse approximate inverse of A in the triple matrix product, are also possible.

The action of $\tilde{\mathrm{S}}^{-1}$ on a vector v can however also be approximated by an inner iterative solution of $\tilde{\mathrm{S}}\mathrm{x} = \mathrm{v}$, in which case $\tilde{\mathrm{S}}$ need not be formed explicitly. For example, the Conjugate Gradient method can be employed with $\tilde{\mathrm{S}}_3 = \mathrm{B}^\mathsf{T}\tilde{\mathrm{A}}^{-1}\mathrm{B} - \mathrm{C}$. We notice from eq. (19) that it is in fact better if $\tilde{\mathrm{S}}$ approximates $\mathrm{B}^\mathsf{T}\tilde{\mathrm{A}}^{-1}\mathrm{B} - \mathrm{C}$ rather than the exact Schur complement. We have not seen the need to include this procedure in our test, so it is mentioned here only for completeness.

4.4 Implementation

We have implemented the finite element discretisation, block preconditioners and linear solvers using the Diffpack C++ framework [11], somewhat modified for our needs. The single-block AMG preconditioners are from the ML package for smoothed aggregation [13], which is part of Trilinos [17].

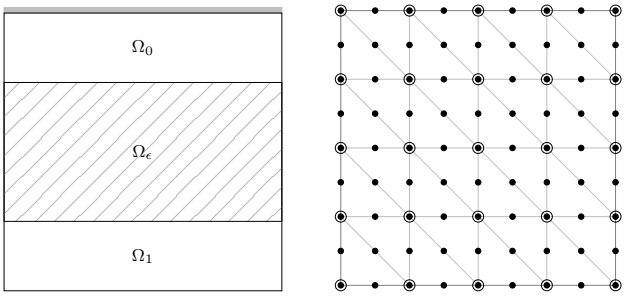
4.5 Test geometry

Fig. 1 shows the two-dimensional domain of the test problems. For the pressure variable, we use essential boundary conditions at the top of the domain (specified pressure) and natural boundary conditions at the bottom and sides (no-flow condition). The displacement boundary conditions are essential at the bottom (fixed position) and natural at the top (specified traction force). At the sides the horizontal displacement components are zero.

It should be noted that when $\epsilon \to 0$, the decoupled pressure equation is ill-posed because Ω_1 in fig. 1a becomes an isolated subdomain with indeterminate pressure because of the pure Neumann conditions. When coupled to deformation, however, the problem is well-posed.

4.6 Convergence criterion

We have in our earlier work observed that a convergence criterion based on the residual in iteration k, $\mathrm{r}_k = \mathrm{b} - \mathcal{A}\mathrm{x}_k$, may be misleading when $\mathcal{A}$ is severely ill-conditioned, owing to some components of x being $\kappa(\mathcal{A})$ times more influential than others [16]. This problem is exacerbated when pushing against the limits of machine precision, as may happen when parameters vary by more than ten orders of magnitude. Hence, in the



(a) The domain. High-permeable regions are Ω_i, low-permeable are Ω_ϵ.

(b) The mesh. Pressure nodes are shown as circles (○), displacement nodes as dots (•).

Figure 1: The domain (a) and the mesh (b). The mesh is the smallest regular P^2–P^1 mesh that aligns the element boundaries with the discontinuities ($N = 9$).

convergence tests in the present paper we exploit an established property of iterative solvers: their rate of convergence is independent of the right-hand side b as long as the initial guess contains all eigenvectors of $\mathcal{A}$ [15, ch. 3.4].

For this reason we have chosen to solve the modified problem $\mathcal{A}\mathrm{x} = 0$, instead of $\mathcal{A}\mathrm{x} = \mathrm{b}$, together with a random initial solution vector x_0. With this choice of right-hand side, the error norm $\|\mathrm{e}_k\|_{\ell_2}$ is trivially available, since $\mathrm{e}_k = \mathrm{x}_k$. The convergence criterion is $\|\mathrm{e}_k\|_{\ell_2} < 10^{-6}\|\mathrm{e}_0\|_{\ell_2}$. We also note that due to this testing procedure, the exact value of any boundary condition is irrelevant, since these values go into the b vector. The only relevant information is whether or not they are essential, since the presence of an essential boundary condition at a node is reflected by a modification to $\mathcal{A}$.

All the reported iteration counts are from at least five runs using different random initial guesses. In the graphs, the mean and range of the results are shown.

4.7 On the order of iterative methods

We often refer to the order of an iterative solution method, or the order of a preconditioner (in combination with an iterative method). As mentioned in the introduction, the number of iterations to solve a linear system to a given accuracy with conjugate gradient-type methods is proportional to $\sqrt{\kappa}$, where $\kappa(\mathcal{P}^{-1}\mathcal{A})$ is the condition number of the preconditioned coefficient matrix. For discretisations of the finite element methods, $\kappa(\mathcal{A}) \sim \mathcal{O}(h^{-2})$, where h is the length scale of the elements. The number of iterations of an iterative method for this unpreconditioned coefficient matrix is then of *order* $\mathcal{O}(h^{-1}) \sim \mathcal{O}(N)$, since $N \sim h^{-1}$ in the present paper denotes the number of nodes in each space direction.

In general, we assume that the number of iterations to reduce the error by a fixed factor can be modelled as

$$n \sim aN^p, \tag{27}$$

where the multiplicative factor a and the exponent p of the order may depend on the geometry and mesh, the heterogeneity of the material parameters, boundary conditions,

and so on; but not on N. By *optimal order (with respect to N)* we mean that $p = 0$, and hence that the number of iterations is independent of N. A method which is *optimal with respect to* ϵ may have $p > 0$, but the number of iterations is independent of ϵ. Finally, a weaker (but still attractive) property is having a *growth rate that is independent of* ϵ; that is, p does not depend on ϵ even if a does.

4.8 Performance of the fully coupled solver with uniform elastic parameters

In the first group of experiments with the fully coupled solver, the elastic parameters are held constant throughout the domain, while the permeability has a discontinuous jump of up to 16 orders of magnitude ($\epsilon = 10^0, \ldots, 10^{-16}$). The time step and fluid mobility are scaled such that $\Lambda_0 \Delta t = 1$, and the elastic parameters are $\lambda = 114$ and $\mu = 455$ (corresponding to Young's modulus $E = 10^3$ and Poisson's ratio $\nu = 0.1$).

Performance with constant permeability

The constant-parameter Biot's equations, with $\epsilon = 10^0$ and uniform elasticity, seem simple to solve. If AMG can solve or precondition the separate equations nearly optimally — which seems to be the case, at least in idealised cases [2, 16] — then one might expect the same to be the case for the fully coupled problem with the application of an equally simple block preconditioner. Yet, as seen in fig. 2, this is not necessarily the case. The (nearly) optimal order, where the number of iterations is independent of problem size, is seen only when the domain is discretised with equal polynomial order quadrilateral (Q^1–Q^1) elements. These elements are however less attractive for other reasons; equal-order elements are susceptible to pressure oscillations in permeability interfaces, and quadrilaterals are less flexible with respect to unstructured geometries than triangular elements. When triangular or mixed elements are used, the order is slightly below $\sqrt{N}$. This is still a major improvement over the expected order N of the unpreconditioned or diagonally scaled finite element method. For two-dimensional problems, it means that the number of unknowns can be increased at least 16 times for a doubling in the number of iterations, whereas using diagonal scaling it can only be increased fourfold.

The figure shows convergence data for the block Jacobi (sJ) preconditioner, but as seen in table 3b similar rates are seen with the other preconditioners for the P^2–P^1 space.

Performance with moderate jumps in permeability.

As long as the jumps in permeability are of moderate size, $\epsilon \geq 10^{-4}$, the problem is numerically well behaved. Fig. 3a shows the convergence behaviour of the different block preconditioners under these conditions. In fact, some of the preconditioned solvers initially have *decreasing* order as ϵ gets smaller (most easily seen by comparing columns one through three in table 3b). This is however a small effect, and not significant compared to the increase in number of iterations observed in fig. 3a.

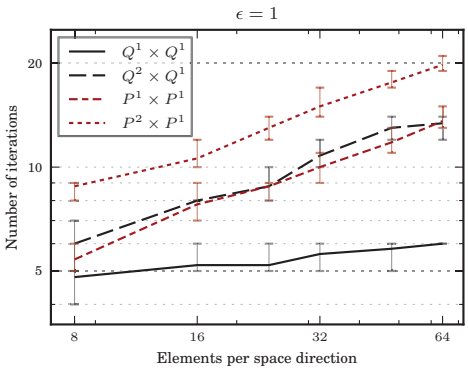


Figure 2: Iteration count for the homogeneous-domain problem. The sJ preconditioner was used. Q denotes quadrilaterals and P denotes simplices of a given polynomial order.

Performance of the fully coupled solver with severe jumps in permeability.

When the discontinuities become more severe, with $\epsilon < 10^{-4}$, several of the preconditioners fail to converge, as shown in fig. 3b. The first to diverge are the standard and generalised Jacobi preconditioners sJ and gJ(-1), which drop out at $\epsilon = 10^{-8}$ (hence these are not plotted in this figure). The Gauß-Seidel preconditioners are better, but when ϵ goes below 10^{-8}, the standard-basis variants sGS and sSGS also fail. At $\epsilon = 10^{-16}$, the gJ(-4) preconditioner does not converge on the finest grid ($N = 65$).

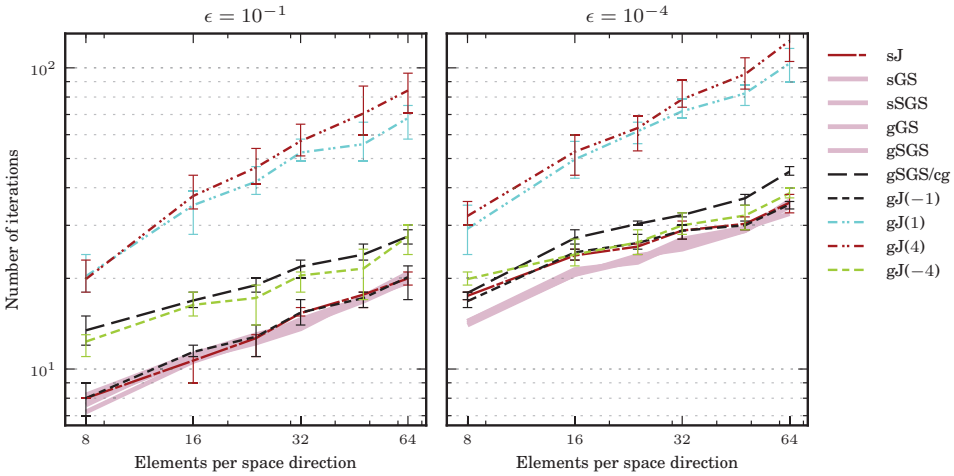
In short, the story told in fig. 3 is that the generalised Gauß-Seidel (gGS and gSGS) preconditioners perform consistently well (the latter also with Conjugated Gradients), with both a low number of iterations and a low growth rate. The gJ(1)/gJ(4) preconditioners also exhibit a low rate of growth, and their higher absolute iteration count is at least partly offset by a lower computational cost per iteration.

4.9 Discontinuities in both permeability and elastic parameters

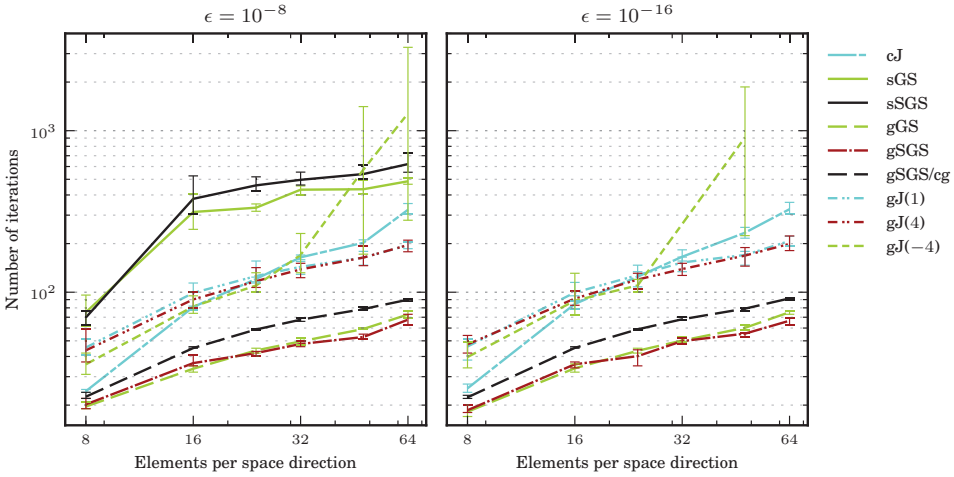
In the experiments we have looked at so far, the elastic parameters have been constant throughout the domain. We now proceed to investigate the effect of discontinuous elastic material parameters. This is a more realistic case of two different geological materials. The parameters of the (soft, high-permeable) surrounding subdomain Ω_0 are the same as in the constant-parameter case. Inside Ω_ϵ, the scaled elastic parameters are $\lambda = 1.43 \cdot 10^5$ and $\mu = 3.57 \cdot 10^4$, corresponding to Young's modulus $E = 10^5$ and Poisson's ratio $\nu = 0.4$.

Performance with moderate jumps in permeability.

Fig. 4a shows the results for a moderate discontinuity in permeability. The general behaviour of the preconditioners is quite similar to the constant-elasticity case, differing mostly by a multiplicative factor (on average, the number of iterations is about doubled). Except for the $\alpha > 0$ generalised Jacobi variants, all preconditioners perform equally



(a) Low to moderate permeability contrast. The Gauß-Seidel methods overlap, and are drawn in gray for legibility.



(b) Severe permeability contrast.

Figure 3: Number of iterations to reach convergence ($|e_k| < 10^{-6}|e_0|$) for the model problem with uniform elastic parameters. In (a), all preconditioners except for gJ($\alpha > 0$) show a growth rate of roughly $N^{0.3}$–$N^{0.4}$, with N being the number of displacement nodes in each space direction. At $\epsilon = 10^{-8}$ (lower left), the sGS and sSGS preconditioners show a surprisingly low growth rate as N increases, but with a large constant factor. When the discontinuities are even stronger (lower right), these variants fail to converge at all. The Schur variants (gGS, gSGS and gJ(1)/gJ(4)) show a growth rate of about $N^{0.5}$ for both values of ϵ, while the cJ preconditioner exhibits linear growth.

well, with a growth rate in the range $N^{0.3}$–$N^{0.4}$ (see table 3b). This demonstrates that heterogeneity in the elastic parameters is not in itself a major difficulty with these block preconditioners.

Performance with severe jumps in permeability.

When the permeability contrasts are strengthened, however, we see some changes relative to the constant-elasticity case. This is shown in fig. 4b (compared with fig. 3b). Four of the preconditioners have the same behaviour as they did with uniform elasticity. These are the sJ and gJ(-1) Jacobi-scheme methods, which fail, and the gGS and gSGS generalised Gauß-Seidel methods, which converge robustly. But the remaining preconditioners behave differently in the problem with discontinuous elastic parameters.

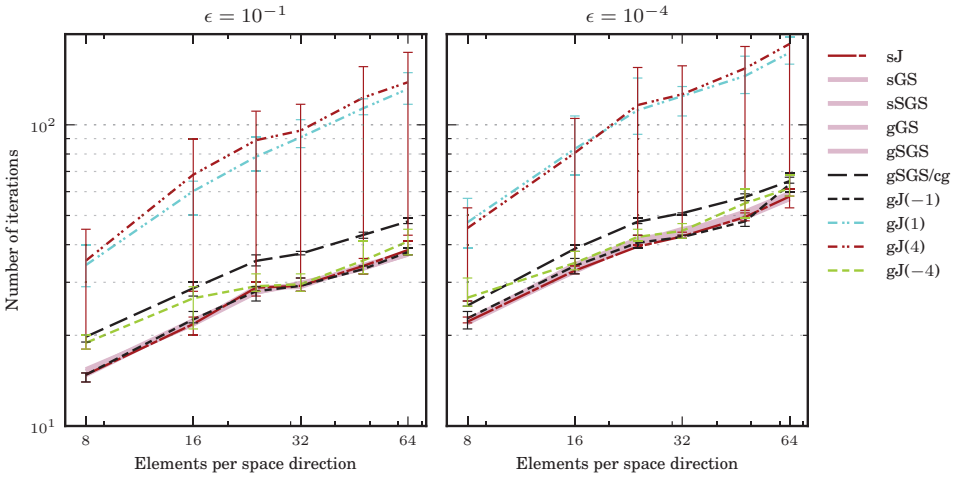
At $\epsilon = 10^{-8}$, the standard Gauß-Seidel preconditioners, sGS and sSGS, perform very well, while the capped-ϵ Jacobi method (cJ) actually converges *faster* as N grows (although the number of iterations is still much higher than for the other methods). All of these methods were among the worst performers with the same value of ϵ and uniform elastic parameters. These anomalies disappear in the most discontinuous case, where $\epsilon = 10^{-16}$; here, the standard basis (sGS, sSGS) methods do not converge at all, and the cJ and gJ(-4) methods fail for large N. The latter result is in line with its performance in the continuous-elasticity case, fig. 3b. While the good result at $\epsilon = 10^{-8}$ is surprising, it has little practical significance since the effect appears to be a result of particular combinations of parameters.

We note that the only four preconditioners that achieve convergence for all values of ϵ are the same that performed best in the constant-elasticity test: The positive-α generalised Jacobi methods gJ(1) and gJ(4), and the generalised Gauß-Seidel methods gGS and gSGS (with either BiCGStab or CG iterations). The high sensitivity of gJ(4) to the initial vector, seen most clearly in fig. 4a as a large variance in the results, can be construed either as a warning flag, or as a sign that it can potentially be more efficient if certain (unidentified) modes are not present in the initial guess.

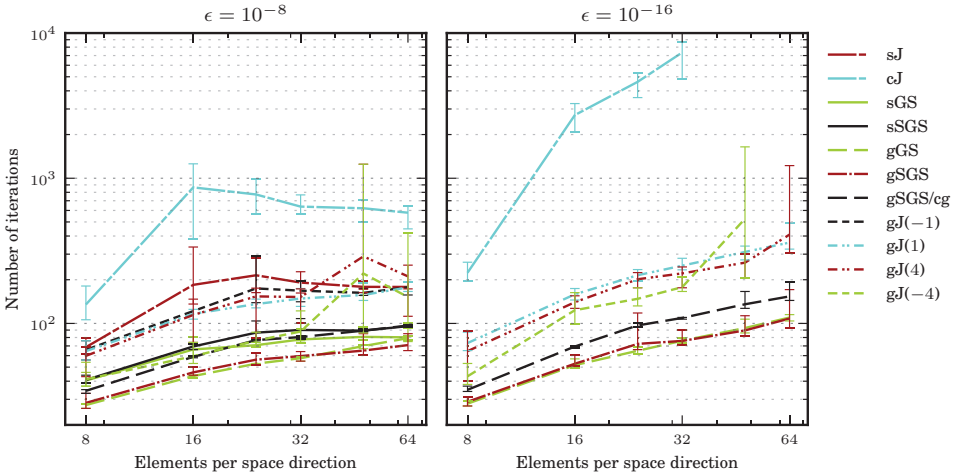
The orders of the different methods, when used with discontinuous elastic material parameters, is given on the right side of table 3b. The gSGS method does not go significantly above $\mathcal{O}(N^{0.5})$ in any of the tests — a remarkably robust result.

4.10 Summary of experimental results

Fig. 5a summarises the performance of the successful preconditioners for the largest problem size, $N = 65$. The ones that fail to converge in one or more of the tests are similarly shown in fig. 5b. It is clear that when $\epsilon \geq 10^{-4}$, it does not matter much which preconditioner is chosen; they all converge, and with the exception of the generalised Jacobi preconditioners gJ(1)/gJ(4) they are equally effective. When the permeability jump becomes larger, however, there are only four preconditioners that converge consistently with every combination of material parameters: the generalised Gauß-Seidel methods gGS/gSGS, and the generalised Jacobi methods gJ(1)/gJ(4), again with gJ(α) being least efficient. Additionally, gSGS/cg (which is solved with Conjugated Gradients) performs well in all cases. Although the number of iterations is higher for this method, the cost per iteration is lower than with BiCGStab, resulting in



(a) Low to moderate permeability contrast. The Gauß-Seidel methods overlap, and are drawn in gray for legibility.



(b) Severe permeability contrast.

Figure 4: Iterations to reach convergence for the model problem with discontinuous elastic parameters. With moderate permeability contrasts, (a), the tested preconditioners show a growth rate of roughly $\sqrt{N}$, with N being the number of divisions in each space direction. When the contrasts are stronger, (b), the picture is more complicated; but the generalised Gauß-Seidel preconditioners remain efficient.

Table 3: Performance of the iterative solvers with the various preconditioners listed in table 2, with uniform and discontinuous elastic parameters. Failure to converge is indicated by "—".

(a) Average number of iterations at $N = 65$

	Uniform elastic parameters					Discontinuous elastic parameters				
$\epsilon \rightarrow$	10^0	10^{-4}	10^{-8}	10^{-12}	10^{-16}	10^0	10^{-4}	10^{-8}	10^{-12}	10^{-16}
sJ	18	35	—	—	—	36	57	178	—	—
sGS	19	34	486	—	—	37	56	80	—	—
sSGS	19	32	621	—	—	37	59	95	—	—
cJ	*same as sJ*		323	313	326	*same as sJ*		579	—	—
gGS	19	36	72	71	75	38	57	79	116	109
gSGS	19	35	67	66	66	37	56	70	104	108
gSGS/cg	26	45	89	91	91	46	65	96	146	153
gJ(−1)	18	35	—	—	—	36	63	177	—	—
gJ(−4)	19	38	1279	1545	—	36	61	152	—	—
gJ(1)	60	103	195	213	207	119	173	177	348	360
gJ(4)	76	122	195	204	199	134	185	210	327	409

(b) Order of convergence (p in eq. (27)) calculated from $N = 33 \ldots 65$

	Uniform elastic parameters					Discontinuous elastic parameters				
$\epsilon \rightarrow$	10^0	10^{-4}	10^{-8}	10^{-12}	10^{-16}	10^0	10^{-4}	10^{-8}	10^{-12}	10^{-16}
sJ	0.41	0.31	—	—	—	0.35	0.45	−0.10	—	—
sGS	0.33	0.48	0.18	—	—	0.41	0.34	0.05	—	—
sSGS	0.43	0.29	0.33	—	—	0.37	0.44	0.09	—	—
cJ	*same as sJ*		1.00	0.86	1.00	*same as sJ*		−0.14	—	—
gGS	0.47	0.47	0.57	0.49	0.60	0.40	0.41	0.47	0.58	0.54
gSGS	0.37	0.46	0.51	0.48	0.43	0.35	0.38	0.26	0.46	0.52
gSGS/cg	0.40	0.49	0.42	0.45	0.44	0.37	0.36	0.27	0.42	0.50
gJ(−1)	0.38	0.30	—	—	—	0.33	0.59	0.08	—	—
gJ(−4)	0.42	0.36	2.98	2.20	—	0.38	0.48	0.81	—	—
gJ(1)	0.52	0.54	0.46	0.50	0.45	0.53	0.49	0.26	0.47	0.54
gJ(4)	0.64	0.66	0.51	0.53	0.53	0.70	0.57	0.48	0.50	0.91

faster overall performance.

We further note that:

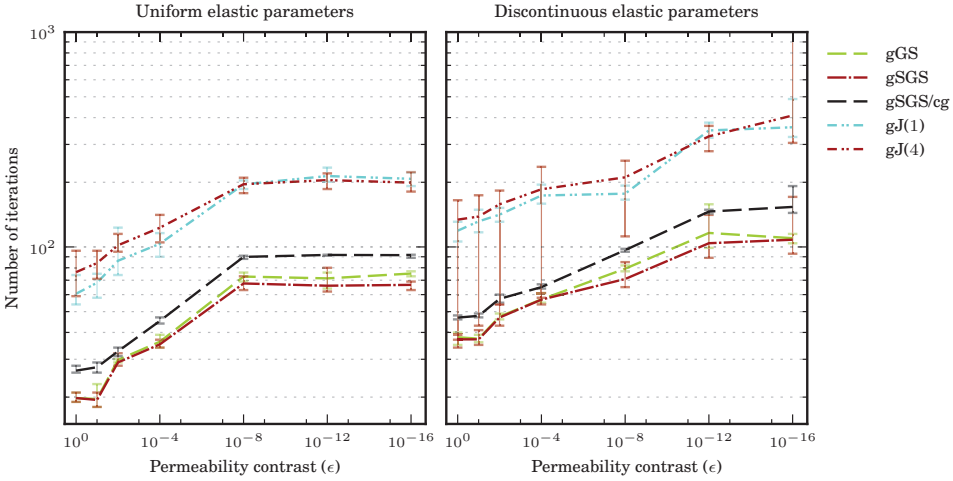
- The standard-basis family of block methods (sJ, sGS and sSGS) does not work well for this problem.
- The Generalised Jacobi family of block preconditioners is unstable with negative α, even though these are more efficient with less severe discontinuities. Positive α is stable, but requires a large number of BiCGStab iterations to converge. The magnitude of α seems to be of less importance, although the variance is much higher with $\alpha = 4$ than with $\alpha = 1$.
- The capped-ϵ cJ preconditioner is stable, although inefficient for large permeability jumps, when the elastic parameters are uniform; but it fails for large jumps in the discontinuous-elasticity cases.
- The generalised symmetric Gauß-Seidel (gSGS) block preconditioner performs well in all cases.
- The gGS block preconditioner, which is a simplified variant of gSGS, performs almost as well (but fails to preserve symmetry, limiting the choice of iterative solver).
- The gSGS block preconditioner with sufficiently accurate single-block preconditioners transforms the problem into a symmetric positive definite one, which can be solved by the Conjugated Gradient method. This combination is denoted as the gSGS/cg method. The AMG method combined with a cheap approximation of the Schur complement is sufficiently accurate for the model problems presented in this paper.

5 Concluding remarks

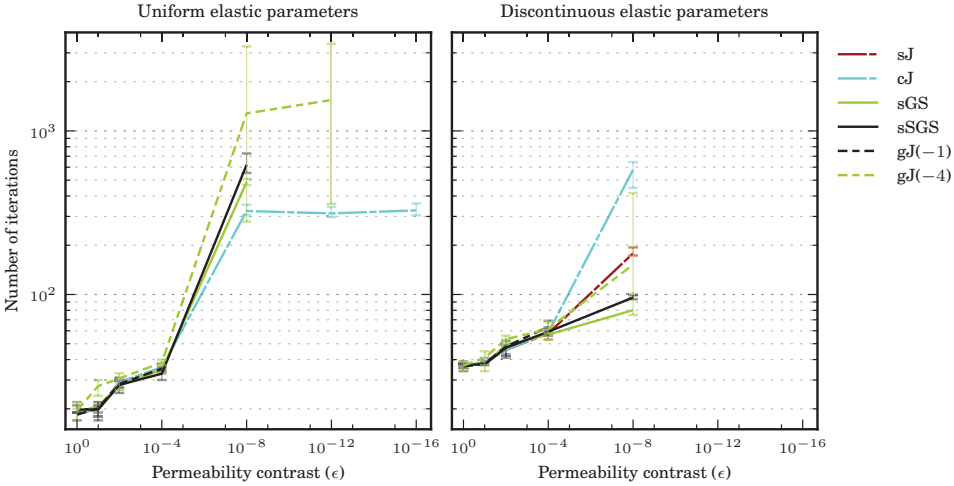
The iterative solution of large-scale problems in geomechanics requires efficient and robust preconditioners. While a number of preconditioners for Biot's equation (and similar symmetric indefinite problems) have been put forth in the literature, their performance with highly discontinuous permeability has to our knowledge not previously been systematically evaluated. This paper evaluates several block preconditioners for this problem in the presence of severe jumps in the material parameters.

Our investigations reveal that discontinuous material parameters, which are present in many realistic geological scenarios, pose a serious challenge for iterative solution methods. Indeed, some seemingly attractive methods converge very slowly, or fail to converge, on a model problem when the heterogeneities are sufficiently strong. These include the standard block Jacobi and block Gauß-Seidel preconditioners [23], as well as the generalised Jacobi block preconditioner [28] with $\alpha < 0$. The generalised Jacobi block preconditioner with $\alpha > 0$ does however converge at an acceptable rate.

Using Algebraic Multigrid as the single-block preconditioners and a cheap approximation to the Schur complement, we identify two block preconditioners that perform



(a) The preconditioning methods that converge for all values of ϵ



(b) The preconditioning methods that fail to converge for small values of ϵ (large jumps in permeability)

Figure 5: The ϵ-dependence of the preconditioners with $N = 65$ displacement nodes in each space direction. The upper plots show the methods that converged in all tests, while those that failed to converge for some combination of parameters are shown at the bottom.

consistently well on Biot's equation with severe jumps in permeability and discontinuous elastic parameters. These two, one symmetric and one non-symmetric variant of the generalised Gauß-Seidel method, are (in our interpretation) based on an exact blockwise inversion of the coupled equations. The performance of these preconditioners is very good, with a number of BiCGStab iterations which is about one third of the generalised Jacobi preconditioner with $\alpha > 0$. Furthermore, the symmetric variant leads (under certain assumptions) to a symmetric positive definite problem which can be solved by the Conjugate Gradient method.

Given that AMG preconditioners have shown themselves to scale to massively parallel computers [2, 19], and that the methods presented herein only have minor parallel communication requirements beyond those of AMG, we anticipate that this combined block preconditioner is equally scalable. This assertion must however be investigated in more detail, which will be performed in a forthcoming paper.

Moreover, owing to its construction from an exact decomposition, we believe that the generalised symmetric Gauß-Seidel preconditioner is widely useful for general difficult coupled problems where the single blocks A and S are individually invertible.

Acknowledgements

The authors would like to thank Kent-Andre Mardal and Xing Cai at Simula Research Laboratory for their advice and support. We also thank Statoil for their support, both financial and in access to their geological models and expertise. This work is supported by a Center of Excellence grant from the Norwegian Research Council to Center for Biomedical Computing at Simula Research Laboratory.

Bibliography

[1] M. Adams. Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics. *International Journal of Numerical Methods in Engineering*, 55:519–534, 2002. doi: 10.1002/nme.506.

[2] M. F. Adams, H. H. Bayraktar, T. M. Keaveny, and P. Papadopoulos. Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC2004)*, page 34. IEEE Computer Society, 2004.

[3] R. E. Bank and T. Dupont. An optimal order process for solving finite element equations. *Mathematics of Computation*, pages 35–51, 1981. doi: 10.2307/2007724.

[4] J. Bear. *Dynamics of fluids in porous media*. Dover Publications, 1988. doi: 10.1097/00010694-197508000-00022.

[5] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182(2):418–477, 2002. doi: 10.1006/jcph.2002.7176.

[6] M. A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12(2):155–164, 1941. doi: 10.1063/1.1712886.

[7] M. Boutéca, D. Bary, J. Piau, N. Kessler, M. Boisson, and D. Fourmaintraux. Contribution of poroelasticity to reservoir engineering: Lab experiments, application to core decompression and implication in HP-HT reservoirs depletion. *Rock Mechanics in Petroleum Engineering*, 1994. doi: 10.2118/28093-MS.

[8] 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. doi: 10.2307/2007912.

[9] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation (αSA). *SIAM Journal on Scientific Computing*, 25 (6):1896–1920, 2004. doi: 10.1137/S1064827502418598.

[10] E. Chow, R. D. Falgout, J. J. Hu, R. Tuminaro, and U. M. Yang. A survey of parallelization techniques for multigrid solvers. In M. A. Heroux, P. Raghavan, and H. D. Simon, editors, *Parallel Processing for Scientific Computing*, pages 179–202. SIAM, 2006.

[11] Diffpack. URL `http://www.diffpack.com/`. Library for numerical solution of PDEs from inuTech GmbH.

[12] S. Doi and T. Washio. Ordering strategies and related techniques to overcome the trade-off between parallelism and convergence in incomplete factorizations. *Parallel Computing*, 25:1995–2014, 1999. doi: 10.1016/S0167-8191(99)00064-2.

[13] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006. URL `http://software.sandia.gov/trilinos/packages/ml/`.

[14] A. George and E. Ng. On the complexity of sparse QR and LU factorization of finite-element matrices. *SIAM Journal on Scientific and Statistical Computing*, 9: 849, 1988. doi: 10.1137/0909057.

[15] W. Hackbush. *Iterative Solution of Large Sparse Systems of Equations*. Springer-Verlag, 1995.

[16] J. B. Haga, H. P. Langtangen, B. F. Nielsen, and H. Osnes. On the performance of an algebraic multigrid preconditioner for the pressure equation with highly discontinuous media. In B. Skallerud and H. I. Andersson, editors, *Proceedings of MekIT'09*, pages 191–204. NTNU, Tapir, 2009. ISBN 978-82-519-2421-4. URL `http://simula.no/research/sc/publications/Simula.SC.568`.

[17] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Transactions on Mathematical Software*, 31(3):397–423, 2005. ISSN 0098-3500. doi: 10.1145/1089014.1089021.

[18] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[19] W. Joubert and J. Cullum. Scalable algebraic multigrid on 3500 processors. *Electronic Transactions on Numerical Analysis*, 23:105–128, 2006.

[20] R. W. Lewis. Coupling of fluid flow and deformation in underground formations. *Journal of Engineering Mathematics*, 128:779, 2002. doi: 10.1061/(ASCE)0733-9399(2002)128:7(779).

[21] R. W. Lewis, A. Makurat, and W. K. S. Pao. Fully coupled modeling of seabed subsidence and reservoir compaction of North Sea oil fields. *Hydrogeology Journal*, 11(1):142–161, 2003.

[22] R. Liu, M. F. Wheeler, and C. N. Dawson. *Discontinuous Galerkin finite element solution for poromechanics*. PhD thesis, University of Texas at Austin, 2006.

[23] K.-A. Mardal, J. Sundnes, H. P. Langtangen, and A. Tveito. Systems of PDEs and block preconditioning. In H. P. Langtangen and A. Tveito, editors, *Advanced Topics in Computational Partial Differential Equations*, pages 199–236. Springer, 2003.

[24] C. E. Neuzil. How permeable are clays and shales? *Water Resources Research*, 30 (2):145–150, 1994. doi: 10.1029/93WR02930.

[25] C. E. Neuzil. Hydromechanical coupling in geologic processes. *Hydrogeology Journal*, pages 11:41–83, 2003.

[26] B. F. Nielsen. Finite element discretizations of elliptic problems in the presence of arbitrarily small ellipticity: An error analysis. *SIAM Journal on Numerical Analysis*, pages 368–392, 1999. doi: 10.1137/S0036142997319431.

[27] P. J. Phillips and M. F. Wheeler. Overcoming the problem of locking in linear elasticity and poroelasticity: an heuristic approach. *Computational Geosciences*, 13(1):5–12, 2009. doi: 10.1007/s10596-008-9114-x.

[28] K. K. Phoon, K. C. Toh, S. H. Chan, and F. H. Lee. An efficient diagonal preconditioner for finite element solution of Biot's consolidation equations. *International Journal of Numerical Methods in Engineering*, 55:377–400, 2002. doi: 10.1002/nme.500.

[29] J. W. Ruge and K. Stuben. Algebraic multigrid. In S. F. McCormick, editor, *Multigrid methods*, volume 3, pages 73–130. SIAM, Philadelphia, PA, 1987.

[30] J. R. Shewchuk. An introduction to the Conjugate Gradient method without the agonizing pain. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[31] R. E. Showalter. Diffusion in deforming porous media. *Dyn. Cont. Discr. Impuls. Syst. (Series A: Math. Anal.)*, 10(5):661–678, 2003.

[32] W. G. Strang and G. J. Fix. *Analysis of the finite element method*. Prentice-Hall, 1973.

[33] K. Terzaghi. Die Berechnung der Durchlassigkeitsziffer des Tones aus dem Verlauf der hydrodynamischen Spannungserscheinungen. *Sitzungsberichte der Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse, Abteilung IIa*, 132:125–138, 1923.

[34] K. C. Toh, K. K. Phoon, and S. H. Chan. Block preconditioners for symmetric indefinite linear systems. *International Journal of Numerical Methods in Engineering*, 60:1361–1381, 2004. doi: 10.1002/nme.982.

[35] R. S. Tuminaro and C. Tong. Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines. In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2000. doi: 10.1109/SC.2000.10008.

[36] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[37] H. F. Wang. *Theory of Linear Poroelasticity with Applications to Geomechanics and Hydrogeology*. Princeton University Press, 2000.

[38] U. M. Yang. Parallel algebraic multigrid methods—high performance preconditioners. In A. M. Bruaset and A. Tveito, editors, *Numerical Solution of Partial Differential Equations on Parallel Computers*, pages 209–236. Springer, 2006. doi: 10.1007/3-540-31619-1_6.

Paper III

On the causes of pressure oscillations in low-permeable and low-compressible porous media

On the causes of pressure oscillations in low-permeable and low-compressible porous media

J. B. Haga[1,3], **H. Osnes**[3], **H. P. Langtangen**[2,4]

[1] Computational Geosciences, Simula Research Laboratory
PO Box 134, N-1325 Lysaker, Norway

[2] Center for Biomedical Computing, Simula Research Laboratory
PO Box 134, N-1325 Lysaker, Norway

[3] Department of Mathematics, University of Oslo,
PO Box 1053 Blindern, N-0316 Oslo, Norway

[4] Department of Informatics, University of Oslo,
PO Box 1080 Blindern, N-0316 Oslo, Norway

Abstract

Non-physical pressure oscillations are observed in finite element calculations of Biot's poroelastic equations in low-permeable media. These pressure oscillations may be understood as a failure of compatibility between the finite element spaces, rather than elastic locking. We present evidence to support this view by comparing and contrasting the pressure oscillations in low-permeable porous media with those in low-compressible porous media. As a consequence, it is possible to use established families of stable mixed elements as candidates for choosing finite element spaces for Biot's equations.

1 Introduction

The coupled poroelastic equations due to Biot [7] describe the behaviour of fluid-filled porous materials undergoing deformation. It is well known that the finite element solution of these equations may exhibit unphysical oscillations in the fluid pressure under certain conditions — low permeabilities, early times (shocks), and short time steps [14, 22, 25]. For the practitioner it is important to know why non-physical oscillations may occur and how to avoid them. This is the research problem we address in the present paper.

Several methods have been proposed to remove the spurious pressure oscillations. Murad *et al.* [14, 15] considered the displacement/fluid pressure (two-field) form of Biot's equation, and identified the initial state (early times) consolidation problem as an instance of the Stokes saddle-point problem, with an associated inf-sup stability test. They developed short- and long-term error bounds for some continuous pressure elements. In particular, they found that the oscillations decay in time and may be treated by post-processing even with unstable element combinations. Wan [23] employed a stabilised finite element method, based on the Galerkin least-squares method, on the two-field and the displacement/fluid velocity/fluid pressure (three-field) formulations. Wan pointed out that the oscillations do not decay, and may even be amplified, under different assumptions, in particular in heterogeneous materials with low-permeable layers. Another stabilisation method was proposed by Aguilar *et al.*[1], who employ a perturbation term depending only on *a priori* material and grid parameters.

More recently, least-squares mixed finite element methods for the stress tensor/displacement/fluid velocity/fluid pressure four-field formulation have been proposed by Korsawe and Starke [12] and Tchonkova *et al.*[21]. These methods have elliptic variational representations and hence appear to be naturally stable.

Phillips and Wheeler [17] investigated the same three-field variant of Biot's equation as Wan, and identified the oscillation phenomenon for short time steps and early times as related to (in-)elastic locking, observed in linear elasticity [6]: The reduction of effective degrees of freedom (owing to vanishing divergence) "locks" the displacement solution.

In the present paper, we investigate the characteristics of the poroelastic fluid pressure oscillations and compare them to those of elastic locking and inf-sup violation. The similarity with the solid pressure oscillations in elasticity is investigated, in part through a mathematical analogy with the elasticity problem and in part through extending the two- and three-field poroelastic formulations to mixed formulations which includes the solid pressure. The addition of a solid pressure field is known to overcome the locking problem in pure elasticity.

Our idea is to link the fluid pressure oscillations to a violation of the compatibility requirements for the discrete finite element spaces. Careful investigations performed in the paper support the view that these phenomena are related. We can then draw upon a large body of knowledge regarding stable spaces for saddle-point problems. This approach helps us to formulate hypotheses about stable mixed finite elements for two-, three-, and four-field formulations of poroelasticity. We test the validity of the hypotheses through extensive numerical experiments. The results form a body of

evidence for our goal of giving practitioners a range of choices for the robust solution of Biot's equations, whether the requirement is a fast solver (which might use a two-field formulation with the minimal-order stable elements) or higher-order accuracy.

2 The mathematical model

The equations describing poroelastic flow and deformation are derived from the principles of conservation of fluid mass and the balance of forces on the porous matrix. The linear poroelastic equations can, in the small-strains regime, be expressed as

$$S\dot{p}_{\mathrm{f}} - \boldsymbol{\nabla}\cdot\boldsymbol{\Lambda}\boldsymbol{\nabla}p_{\mathrm{f}} + \alpha\boldsymbol{\nabla}\cdot\dot{\boldsymbol{u}} = q, \tag{1}$$

$$\boldsymbol{\nabla}(\lambda+\mu)\boldsymbol{\nabla}\cdot\boldsymbol{u} + \boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla}\boldsymbol{u} - \alpha\boldsymbol{\nabla}p_{\mathrm{f}} = \boldsymbol{r}. \tag{2}$$

Here, $\boldsymbol{r}$ represents the total body forces, and q is a fluid injection rate. The primary variables are p_{f} for the fluid pressure and $\boldsymbol{u}$ for the displacement of the porous medium. Furthermore, S and $\boldsymbol{\Lambda}$ are the fluid storage coefficient and the flow mobility respectively, α is the Biot-Willis fluid/solid coupling coefficient, and λ and μ are the Lamé elastic parameters.

The fluid (Darcy) velocity is often of particular interest in poroelastic calculations. It can be written

$$\boldsymbol{v}_{\mathrm{D}} = -\boldsymbol{\Lambda}(\boldsymbol{\nabla}p_{\mathrm{f}} - \boldsymbol{r}_{\mathrm{f}}), \tag{3}$$

and represents the net macroscopic flux, given body forces $\boldsymbol{r}_{\mathrm{f}}$ acting on the fluid phase. For the displacement equation, the main secondary quantity of interest is the effective stress tensor,

$$\sigma' = \sigma - \alpha p_{\mathrm{f}}\mathbf{I} = (\lambda\,\mathrm{Tr}\,\varepsilon - \alpha p_{\mathrm{f}})\mathbf{I} + 2\mu\varepsilon, \tag{4}$$

which is written in terms of the small-strains tensor

$$\varepsilon = (\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^{\mathsf{T}})/2. \tag{5}$$

In the following, this canonical form of Biot's equation given in eqs. (1)–(2) is referred to as the two-field formulation.

Weak discrete-in-time form.

We employ a first-order backward finite difference method in time, which leads to the discrete-time form of eq. (1)

$$Sp_{\mathrm{f}} - \Delta t\boldsymbol{\nabla}\cdot\boldsymbol{\Lambda}\boldsymbol{\nabla}p_{\mathrm{f}} + \alpha\boldsymbol{\nabla}\cdot\boldsymbol{u} = q\Delta t + S\hat{p}_{\mathrm{f}} + \alpha\boldsymbol{\nabla}\cdot\hat{\boldsymbol{u}}. \tag{6}$$

Hatted variables ($\hat{p}_{\mathrm{f}}$, $\hat{\boldsymbol{u}}$) indicate values from the previous time step, while unmarked variables are taken to be at the current time step.

Next, we rewrite eq. (2) and (6) in weak form, using integration by parts to eliminate second derivatives. We define the following (bi-)linear forms on the domain Ω with

boundary Γ,

$$\begin{aligned} a_{\mathrm{f}}^{\mathrm{I}}(\phi_{\mathrm{f}}, p_{\mathrm{f}}) &= -\int_\Omega S\phi_{\mathrm{f}} p_{\mathrm{f}} + \Delta t \boldsymbol{\nabla}\phi_{\mathrm{f}} \cdot \boldsymbol{\Lambda}\boldsymbol{\nabla} p_{\mathrm{f}} \,\mathrm{d}\Omega, \\ b^{\mathrm{I}}(\phi_{\mathrm{f}}, \boldsymbol{u}) &= -\int_\Omega \alpha\phi_{\mathrm{f}} \boldsymbol{\nabla}\cdot\boldsymbol{u} \,\mathrm{d}\Omega, \\ l_{\mathrm{f}}^{\mathrm{I}}(\phi_{\mathrm{f}}) &= -\int_\Omega (q\Delta t + S\hat{p}_{\mathrm{f}} + \boldsymbol{\nabla}\cdot\hat{\boldsymbol{u}})\phi_{\mathrm{f}} \,\mathrm{d}\Omega + \int_\Gamma \phi_{\mathrm{f}}(f_{\boldsymbol{n}} - \boldsymbol{n}\cdot\boldsymbol{\Lambda}\boldsymbol{r}_{\mathrm{f}})\Delta t \,\mathrm{d}\Gamma, \end{aligned} \tag{7}$$

and

$$\begin{aligned} a_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{\phi}_{\mathrm{u}}, \boldsymbol{u}) &= \int_\Omega [\lambda(\boldsymbol{\nabla}\cdot\boldsymbol{\phi}_{\mathrm{u}})(\boldsymbol{\nabla}\cdot\boldsymbol{u}) + \mu\boldsymbol{\nabla}\boldsymbol{\phi}_{\mathrm{u}} : \boldsymbol{\nabla}\boldsymbol{u}] \,\mathrm{d}\Omega, \\ l_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{\phi}_{\mathrm{u}}) &= -\int_\Omega \boldsymbol{\phi}_{\mathrm{u}}\cdot\boldsymbol{r} \,\mathrm{d}\Omega + \int_\Gamma \boldsymbol{\phi}_{\mathrm{u}}\cdot\boldsymbol{t}_{\boldsymbol{n}} \,\mathrm{d}\Gamma. \end{aligned} \tag{8}$$

The problem then becomes: Find $p_{\mathrm{f}} \in V_{\mathrm{f}}$ and $\boldsymbol{u} \in \boldsymbol{V}_{\mathrm{u}}$ that satisfy the following relations:

$$a_{\mathrm{f}}^{\mathrm{I}}(\phi_{\mathrm{f}}, p_{\mathrm{f}}) + b^{\mathrm{I}}(\phi_{\mathrm{f}}, \boldsymbol{u}) = l_{\mathrm{f}}^{\mathrm{I}}(\phi_{\mathrm{f}}) \qquad \forall \phi_{\mathrm{f}} \in V_{\mathrm{f}}, \tag{9}$$

$$a_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{\phi}_{\mathrm{u}}, \boldsymbol{u}) + b^{\mathrm{I}}(p_{\mathrm{f}}, \boldsymbol{\phi}_{\mathrm{u}}) = l_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{\phi}_{\mathrm{u}}) \qquad \forall \boldsymbol{\phi}_{\mathrm{u}} \in \boldsymbol{V}_{\mathrm{u}}. \tag{10}$$

The normal flux $f_{\boldsymbol{n}} = \boldsymbol{v}_{\mathrm{D}}\cdot\boldsymbol{n}$ and normal stresses $\boldsymbol{t}_{\boldsymbol{n}}$ on the boundary Γ appear in these equations as natural boundary conditions. We note that eqs. (9)–(10) form a symmetric, but indefinite, system of equations.[7]

The natural spaces for the continuous problem are $V_{\mathrm{f}} = H^1$ (or L^2 when $\boldsymbol{\Lambda} = 0$) for the pressure and $\boldsymbol{V}_{\mathrm{u}} = \boldsymbol{H}^1$ for the displacement. The discrete finite element approximation follows from solving the equations for the weak form in finite-dimensional spaces. We shall return later to the question of discrete spaces.

2.1 Three-field (fluid velocity) formulation

In many applications of the poroelastic equations, the flow of the fluid through the medium is of primary interest. However, due to the differential operator acting on the pressure p_{f}, the flow is of lower accuracy than the pressure itself. Furthermore, the derivative is not continuous between elements, and hence the fluid mass is not in general conserved. A natural extension is then to introduce $\boldsymbol{v}_{\mathrm{D}}$ as an extra primary variable in a mixed finite element formulation. The order of accuracy is higher, and mass conservation for the fluid phase can be ensured by using continuous elements for $\boldsymbol{v}_{\mathrm{D}}$.

By inserting the relation for fluid flux, eq. (3), into eq. (1), we get a coupled system of three equations (of which two are vector equations). The equations for fluid flux and pressure are

$$S\dot{p}_{\mathrm{f}} + \boldsymbol{\nabla}\cdot\boldsymbol{v}_{\mathrm{D}} + \alpha\boldsymbol{\nabla}\cdot\dot{\boldsymbol{u}} = q, \tag{11}$$

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{v}_{\mathrm{D}} + \boldsymbol{\nabla}p_{\mathrm{f}} = \boldsymbol{r}_{\mathrm{f}}, \tag{12}$$

and these are coupled with the unmodified eq. (2) for solid displacement. We shall call this the fluid velocity three-field formulation.

[7] Symmetric because the trial $(p_{\mathrm{f}}, \boldsymbol{u})$ and test $(\phi_{\mathrm{f}}, \boldsymbol{\phi}_{\mathrm{u}})$ functions are interchangeable; indefinite because $a_{\mathrm{f}}^{\mathrm{I}}$ is negative definite while $a_{\mathrm{u}}^{\mathrm{I}}$ is positive definite.

Weak discrete-in-time form.

We define the following additional forms:

$$\begin{aligned} a_{\mathrm{f}}^{\mathrm{II}}(\phi_{\mathrm{f}}, p_{\mathrm{f}}) &= -\int_\Omega S\phi_{\mathrm{f}} p_{\mathrm{f}}\,\mathrm{d}\Omega, \\ b^{\mathrm{II}}(\phi_{\mathrm{f}}, \boldsymbol{v}_{\mathrm{D}}) &= -\int_\Omega p_{\mathrm{f}}\boldsymbol{\nabla}\cdot\boldsymbol{v}_{\mathrm{D}}\,\mathrm{d}\Omega\Delta t, \\ l_{\mathrm{f}}^{\mathrm{II}}(\phi_{\mathrm{f}}) &= -\int_\Omega (q\Delta t + S\hat{p}_{\mathrm{f}} + \boldsymbol{\nabla}\cdot\hat{\boldsymbol{u}})\phi_{\mathrm{f}}\,\mathrm{d}\Omega, \end{aligned} \tag{13}$$

which are derived from eq. (11), and

$$\begin{aligned} a_{\mathrm{v}}^{\mathrm{II}}(\boldsymbol{\phi}_{\mathrm{v}}, \boldsymbol{v}_{\mathrm{D}}) &= \int_\Omega \boldsymbol{\phi}_{\mathrm{v}}\cdot\boldsymbol{\Lambda}^{-1}\boldsymbol{v}_{\mathrm{D}}\,\mathrm{d}\Omega\Delta t, \\ c^{\mathrm{II}}(\boldsymbol{\phi}_{\mathrm{v}}, p_{\mathrm{f}}) &= \int_\Gamma (\boldsymbol{\phi}_{\mathrm{v}}\cdot\boldsymbol{n}) p_{\mathrm{f}}\,\mathrm{d}\Gamma\Delta t, \\ l_{\mathrm{v}}^{\mathrm{II}}(\boldsymbol{\phi}_{\mathrm{v}}) &= \int_\Omega \boldsymbol{\phi}_{\mathrm{v}}\cdot\boldsymbol{r}_{\mathrm{f}}\,\mathrm{d}\Omega. \end{aligned} \tag{14}$$

from eq. (12). The solution is then given as $(\boldsymbol{u}, p_{\mathrm{f}}, \boldsymbol{v}_{\mathrm{D}}) \in V = \boldsymbol{V}_{\mathrm{u}} \times V_{\mathrm{f}} \times \boldsymbol{V}_{\mathrm{v}}$ satisfying

$$a_{\mathrm{f}}^{\mathrm{II}}(\phi_{\mathrm{f}}, p_{\mathrm{f}}) + b^{\mathrm{I}}(\phi_{\mathrm{f}}, \boldsymbol{u}) + b^{\mathrm{II}}(\phi_{\mathrm{f}}, \boldsymbol{v}_{\mathrm{D}}) = l_{\mathrm{f}}^{\mathrm{II}}(\phi_{\mathrm{f}}) \qquad \forall \phi_{\mathrm{f}} \in V_{\mathrm{f}}, \tag{15}$$

$$a_{\mathrm{v}}^{\mathrm{II}}(\boldsymbol{\phi}_{\mathrm{v}}, \boldsymbol{v}_{\mathrm{D}}) + b^{\mathrm{II}}(p_{\mathrm{f}}, \boldsymbol{\phi}_{\mathrm{v}}) + c^{\mathrm{II}}(\boldsymbol{\phi}_{\mathrm{v}}, p_{\mathrm{f}}) = l_{\mathrm{v}}^{\mathrm{II}}(\boldsymbol{\phi}_{\mathrm{v}}) \qquad \forall \boldsymbol{\phi}_{\mathrm{v}} \in \boldsymbol{V}_{\mathrm{v}}, \tag{16}$$

along with eq. (10) for the displacement.

The displacement space is the same as in the two-field formulation, while the pressure space is always L^2 (in the two-field formulation, this is the case only when $\boldsymbol{\Lambda} = 0$). Additionally, we define the fluid velocity space as $\boldsymbol{V}_{\mathrm{v}} = \boldsymbol{H}(\mathrm{div})$[8]. We note that the system is symmetric only when $c^{\mathrm{II}} = 0$; this is achieved when the whole boundary has either zero fluid pressure or zero fluid flux conditions (and the spaces V_{f} and $\boldsymbol{V}_{\mathrm{v}}$ are restricted accordingly).

2.2 Three-field (solid pressure) formulation

In the field of (pure) elasticity, it is well understood that a low-compressible material (with Poisson's ratio close to 0.5) leads to unphysical oscillations in the solid pressure field, and in some cases also to a wrong solution for the calculated displacement. This can be explained by λ becoming very large in eq. (2), leading to the requirement that $\boldsymbol{\nabla}\cdot\boldsymbol{u} \to 0$. When this requirement is applied to standard finite elements, several degrees of freedom become "locked", leaving too few degrees of freedom to represent the correct solution.

One way to overcome this obstacle is to introduce a new primary variable for the solid pressure. We define the (incomplete) solid pressure as

$$\overline{p}_{\mathrm{s}} = -\lambda\boldsymbol{\nabla}\cdot\boldsymbol{u}, \tag{17}$$

whereby eq. (2) can be rewritten as the coupled equations,

$$\boldsymbol{\nabla}\mu\boldsymbol{\nabla}\cdot\boldsymbol{u} + \boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}\overline{p}_{\mathrm{s}} - \alpha\boldsymbol{\nabla}p_{\mathrm{f}} = \boldsymbol{r}, \tag{18}$$

$$\lambda^{-1}\overline{p}_{\mathrm{s}} + \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{19}$$

[8] $\boldsymbol{L}^2 \supset \boldsymbol{H}(\mathrm{div}) = \{\boldsymbol{v} \in \boldsymbol{L}^2 \mid \boldsymbol{\nabla}\cdot\boldsymbol{v} \in L^2\} \supset \boldsymbol{H}^1 = \{\boldsymbol{v} \in \boldsymbol{L}^2 \mid \boldsymbol{\nabla}\boldsymbol{v} \in \mathbf{L}^2\}$

and combined with eq. (1) for the fluid pressure. This definition of the solid pressure makes the equation simpler than when using the volumetric solid stress, $p_\mathrm{s} = -\sigma_\mathrm{vol} = -(\lambda + \frac{2}{3}\mu)\boldsymbol{\nabla}\cdot\boldsymbol{u}$, while still including the "difficult" part of the pressure.

The three-field (solid pressure) formulation can be used with low-compressible or even incompressible materials.

Weak discrete-in-time form.

The additional variational forms associated with eqs. (18)–(19) are

$$\begin{aligned} a_\mathrm{u}^\mathrm{III}(\boldsymbol{\phi}_\mathrm{u}, \boldsymbol{u}) &= \textstyle\int_\Omega \mu \boldsymbol{\nabla}\boldsymbol{\phi}_\mathrm{u} : \boldsymbol{\nabla}\boldsymbol{u}\,\mathrm{d}\Omega, \\ a_\mathrm{s}^\mathrm{III}(\phi_\mathrm{s}, \overline{p}_\mathrm{s}) &= \textstyle\int_\Omega \lambda^{-1}\phi_\mathrm{s}\overline{p}_\mathrm{s}\,\mathrm{d}\Omega, \\ b^\mathrm{III}(\phi_\mathrm{s}, \boldsymbol{u}) &= \textstyle\int_\Omega \phi_\mathrm{s}\boldsymbol{\nabla}\cdot\boldsymbol{u}\,\mathrm{d}\Omega, \end{aligned} \tag{20}$$

and the solution is given as $(\boldsymbol{u}, \overline{p}_\mathrm{s}, p_\mathrm{f}) \in \boldsymbol{V}_\mathrm{u} \times V_\mathrm{s} \times V_\mathrm{f}$ satisfying

$$a_\mathrm{u}^\mathrm{III}(\boldsymbol{\phi}_\mathrm{u}, \boldsymbol{u}) + b^\mathrm{I}(p_\mathrm{f}, \boldsymbol{\phi}_\mathrm{u}) + b^\mathrm{III}(\overline{p}_\mathrm{s}, \boldsymbol{\phi}_\mathrm{u}) = l_\mathrm{u}^\mathrm{I}(\boldsymbol{\phi}_\mathrm{u}) \qquad \forall \boldsymbol{\phi}_\mathrm{u} \in \boldsymbol{V}_\mathrm{u}, \tag{21}$$

$$a_\mathrm{s}^\mathrm{III}(\phi_\mathrm{s}, \overline{p}_\mathrm{s}) + b^\mathrm{III}(\phi_\mathrm{s}, \boldsymbol{u}) = 0 \qquad \forall \phi_\mathrm{s} \in V_\mathrm{s}, \tag{22}$$

along with the original equation for the fluid pressure, eq. (9). The continuous spaces are as in the two-field formulation, with the addition of the solid pressure space $V_\mathrm{s} = L^2$.

2.3 Four-field formulation

Combining the three-field formulations of fluid velocity and solid pressure, we get a formulation of two scalar and two vector fields which attains accurate fluid velocities, and which is stable in the presence of low-compressible materials. The formulation is obtained as the coupled system of eqs. (11)–(12) and (18)–(19), as

$$S\dot{p}_\mathrm{f} + \boldsymbol{\nabla}\cdot\boldsymbol{v}_\mathrm{D} + \alpha\boldsymbol{\nabla}\cdot\dot{\boldsymbol{u}} = q, \tag{23}$$

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{v}_\mathrm{D} + \boldsymbol{\nabla}p_\mathrm{f} = \boldsymbol{r}_\mathrm{f}, \tag{24}$$

$$\boldsymbol{\nabla}\mu\boldsymbol{\nabla}\cdot\boldsymbol{u} + \boldsymbol{\nabla}\cdot\mu\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}\overline{p}_\mathrm{s} - \alpha\boldsymbol{\nabla}p_\mathrm{f} = \boldsymbol{r}, \tag{25}$$

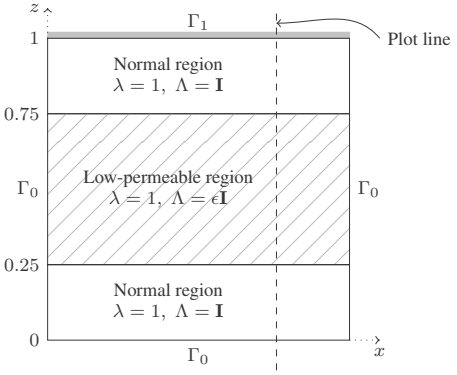
$$\lambda^{-1}\overline{p}_\mathrm{s} + \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{26}$$

Weak discrete-in-time form.

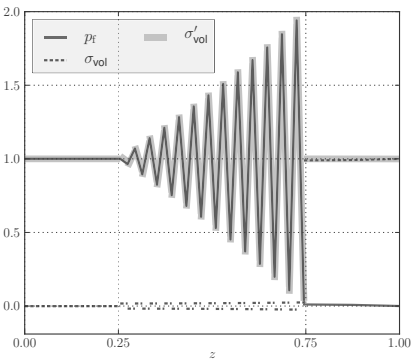
The weak form of the four-field formulation is: Find $(\boldsymbol{u}, \boldsymbol{v}_\mathrm{D}, p_\mathrm{f}, \overline{p}_\mathrm{s}) \in \boldsymbol{V}_\mathrm{u} \times \boldsymbol{V}_\mathrm{v} \times V_\mathrm{f} \times V_\mathrm{s}$ such that eqs. (15)–(16) and eqs. (21)–(22) are satisfied.

3 On the causes of pressure oscillations

It is well known that spurious fluid pressure oscillations may occur in low-permeable regions in finite element calculations of the poroelastic equations [13, 17, 18]. To illustrate this phenomenon, we use a simple test case where a low-permeable layer is placed inside a "normal" material, shown in fig. 1a. The low-permeable layer uses $\boldsymbol{\Lambda} = \epsilon\mathbf{I}$



(a) The two-material problem



(b) Oscillatory solution to the two-material problem with uniform load; $F = 1$, $\epsilon = 10^{-8}$, $\Delta t = 1$.

Figure 1: Domain for illustrating pressure oscillations. On the sides and bottom, no-flux conditions are imposed so that no fluid or solid movement is allowed in the normal direction. The top is drained with fluid pressure $p_f = 0$ and an applied normal stress. Spurious pressure oscillations are clearly present in (b) — the analytical solution is constant $\sigma'_{vol} = 1$.

for some $\epsilon \ll 1$, while the "normal" layer has unit permeability. In all three layers, the elastic parameters are set to $\lambda = \mu = 1$. The boundary conditions at the sides and bottom are no-flux for both the fluid and the solid,

$$f_{\boldsymbol{n}}|_{\Gamma_0} = 0, \qquad \boldsymbol{u} \cdot \boldsymbol{n}|_{\Gamma_0} = 0, \tag{27}$$

while the top boundary is drained, with an applied normal force

$$p_f|_{\Gamma_1} = 0, \qquad \boldsymbol{t}_{\boldsymbol{n}}|_{\Gamma_1} = F(x)\boldsymbol{n}, \tag{28}$$

where $F(x)$ is constant 1 for the present. No body forces are present, and the initial conditions are $\boldsymbol{u} = 0$ and $p_f = 0$ with $\Delta t = 1$.

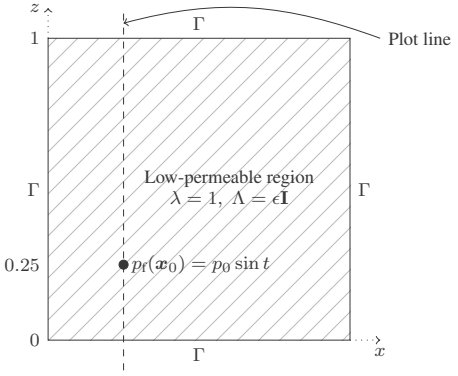
fig. 1b shows the naïve numerical solution to the two-material test case when ϵ is very small, computed with the two-field formulation using first order quadrilateral elements (Q_1/Q_1)[9]. The pressure oscillations in the middle layer clearly have no physical basis, nor are they present in the analytical solution to the problem.

Studying the fluid velocity three-field formulation, Phillips and Wheeler [18] argue that such pressure oscillations have the same cause as the phenomenon known as locking in pure elasticity. To see why, we consider that the basic linear elastic equation is just eq. (2) without the fluid pressure term,

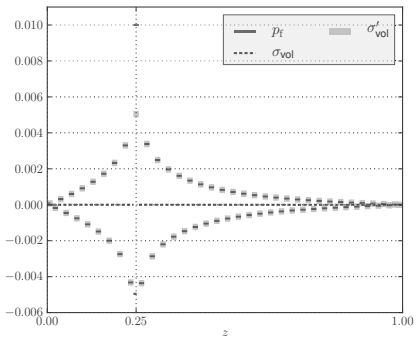
$$\boldsymbol{\nabla}(\lambda + \mu)\boldsymbol{\nabla} \cdot \boldsymbol{u} + \boldsymbol{\nabla} \cdot \mu\boldsymbol{\nabla}\boldsymbol{u} = \boldsymbol{r}. \tag{29}$$

Elastic locking occurs when finite elements are asked to reproduce a displacement field that is nearly divergence free, as $\lambda \to \infty$ corresponds to $\boldsymbol{\nabla} \cdot \boldsymbol{u} \to 0$. Satisfying this with

[9] Elements are listed in the order $\boldsymbol{u}\overline{p}_s/\boldsymbol{v}_D p_f$, where any unused position for a particular formulation is skipped. Hence, the two-field formulation uses $\boldsymbol{u}/p_f$, fluid velocity three-field uses $\boldsymbol{u}/\boldsymbol{v}_D p_f$, and solid pressure three-field uses $\boldsymbol{u}\overline{p}_s/p_f$.



(a) The Barry-Mercer problem.



(b) Oscillatory solution to the Barry-Mercer problem, with $\epsilon = 10^{-8}$, $\Delta t = 0.01$.

Figure 2: The Barry-Mercer problem consists of a pulsating pressure point source embedded in a uniform porous material which is drained on all sides, with zero tangential displacement. Pressure oscillations are clearly visible when using Q_1/RT_1Q_0 elements.

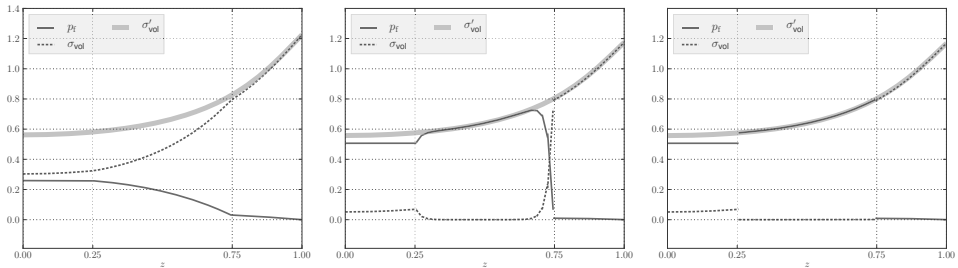
standard (low-order piecewise polynomial) finite elements locks out many of the degrees of freedom, to the extent that only constant displacement fields can be represented. More commonly, the error in displacement is seen to cause nonphysical oscillations in the solid pressure ($p_\mathrm{s} \to \overline{p}_\mathrm{s} = -\lambda \boldsymbol{\nabla} \cdot \boldsymbol{u}$). This is because the errors in $\boldsymbol{\nabla} \cdot \boldsymbol{u} \approx 0$ are magnified by a very large factor λ in the post-process calculation of the volumetric stress.

The argument by Phillips and Wheeler is that under certain conditions the same happens in poroelasticity. Consider eq. (1) with uniform permeability, discretised in time with time step Δt and with $S = q = 0$. Assume furthermore that we take one time step from a divergence-free initial state, which is quite normal at the start of a simulation (when $\boldsymbol{u} = 0$). Then, eq. (1) reduces to

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = \Delta t \boldsymbol{\nabla} \cdot \boldsymbol{\Lambda} \boldsymbol{\nabla} p_\mathrm{f} / \alpha, \tag{30}$$

The right-hand side becomes very small for short time steps and low permeabilities. Again, the requirement for a nearly divergence-free solution for the displacement $\boldsymbol{u}$ appears. Fluid pressure oscillations are demonstrated in (among others) the Barry-Mercer problem (shown in fig. 2a), using the three-field formulation with lowest-order Raviart-Thomas elements for the fluid and linear elements for the displacement (Q_1/RT_1Q_0). This problem [5] consists of a pulsating pressure point source embedded in a uniform porous material, with boundary conditions chosen to permit an analytical solution: $p_\mathrm{f}|_\Gamma = 0$, $\boldsymbol{u} \times \boldsymbol{n}|_\Gamma = 0$, and initial conditions $\boldsymbol{u} = 0$ and $p_\mathrm{f} = 0$. The pressure oscillations disappear when the displacement is instead calculated with a discontinuous Galerkin method, and the optimality of the pressure solution is proven for this method.

As regards the test case shown in fig. 1, we remark that elastic locking can not appear in this test case which is one-dimensional, because in one dimension $\boldsymbol{\nabla} \cdot \boldsymbol{u} = \partial \boldsymbol{u}_x / \partial x \to 0$ implies constant displacement — a trivial solution which can be represented by any element. Hence, the oscillations shown in this figure are *not* caused by elastic locking.



(a) $\epsilon = 10^{-1}$ **(b)** $\epsilon = 10^{-4}$ **(c)** $\epsilon = 10^{-8}$

Figure 3: Plausible (smooth) solution for the three-material problem with a low-permeable layer and non-uniform load. As ϵ decreases, the fluid pressure becomes dominant in the middle layer, and each of the pressure components approach a discontinuous solution.

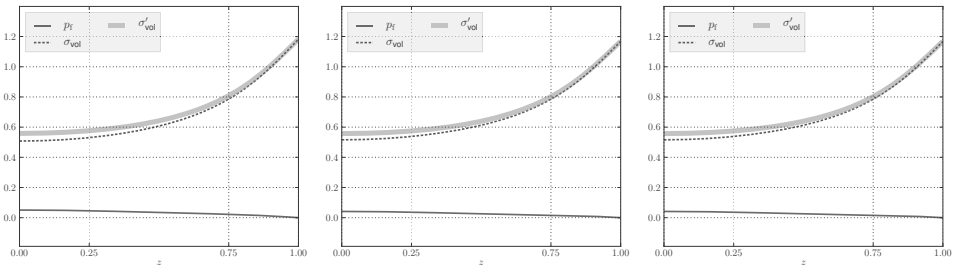
We therefore introduce asymmetry through a load on just the right half of the top boundary, $F(x) = \{0 \text{ when } x < 0.5,\ 1 \text{ otherwise}\}$, in the three-layer problem (fig. 1a). With asymmetric loading we do not have an analytical solution, unlike in the uniform-load case. Instead, we use the fact that the volumetric effective stress,

$$\sigma'_{\text{vol}} = \frac{\operatorname{Tr}\sigma'}{3} = \overline{p}_{\text{s}} + \frac{2}{3}\mu\boldsymbol{\nabla}\cdot\boldsymbol{u} + \alpha p_{\text{f}}, \tag{31}$$

should be continuous and smooth (away from the externally applied discontinuity on the surface at $x = 0.5$). The solution is illustrated in fig. 3, where the thick gray line shows that σ'_{vol} is continuous even when each of its three components is discontinuous. The smoothness of σ'_{vol} does not prove that the numerical solution converges, but it makes it easy to identify many of the wrong solutions with oscillating pressure.[10] In the text, we refer to these apparently correct solutions as "plausible", since they are not compared to a known analytical solution.

We now compare the behaviour of the non-uniform load problem with a low-permeable layer to that of a low-compressible layer. In the latter case, the middle layer of fig. 1a is replaced with a layer with unit permeability but low compressibility; $\lambda = \epsilon^{-1}$, $\boldsymbol{\Lambda} = \mathbf{I}$. The plausible (smooth) solution to this problem is shown in fig. 4. The total pressure profile is similar to the low-permeable problem, but the load in the middle layer is here mainly supported by the volumetric stress, instead of the fluid pressure. Furthermore, we know that this problem is susceptible to elastic locking. fig. 5 compares these two cases using equal-order Q_1/Q_1 elements. As expected, the low-permeable problem has difficulty with the fluid pressure, while the low-compressible problem has difficulty with the volumetric stress. There is, however, a major difference in the effect that this has

[10] We also note that the existence of analytical solutions is no panacea. As noted in, e.g., [16], geologically relevant solutions are often not realisable on a reasonably sized computational mesh. For example, the fluid pressure solution in fig. 1b should have a very sharp gradient between the two top layers, a feature that is not possible to realise with continuous elements unless an extremely fine grid is used. Similarly, the Barry-Mercer problem requires a point pressure source, while discrete analogues have source areas on the order of the element size. These inaccuracies in the discrete model may mask any "real" convergence difficulties for all but very fine meshes.



(a) $\epsilon = 10^{-1}$ **(b)** $\epsilon = 10^{-4}$ **(c)** $\epsilon = 10^{-8}$

Figure 4: Plausible (smooth) solution for the three-material problem with a low-compressible layer and non-uniform load. As opposed to the low-permeable case in fig. 3, the pressure components are continuous.

on the displacement. fig. 6 compares the locking behaviour of the low-permeable and the low-compressible problems. In the low-compressible problem, the faulty pressure is associated with elastic locking, i.e., the displacement is pulled toward a constant in the middle region, fig. 6b. This restriction of the displacement is not seen in the low-permeable problem, fig. 6a.

It appears that elastic locking is not in general a sufficient explanation for the fluid pressure oscillations in low-permeable regions.

4 Spurious pressure oscillations and saddle-point problems

It is instructive to look at the case of total impermeability, $\boldsymbol{\Lambda} = S = 0$. For clarity of presentation, we furthermore set $\alpha = 1$ and let $\tilde{q} = q\Delta t + \boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}$, where $\hat{\boldsymbol{u}}$ is the value of $\boldsymbol{u}$ at the previous time step. In this case, eqs. (1)–(2)[11] take on almost the same form as those of the *mixed* formulation of incompressible linear elasticity (as opposed to the pure displacement formulation mentioned in the previous section). This is evident when we compare the impermeable poroelastic equations

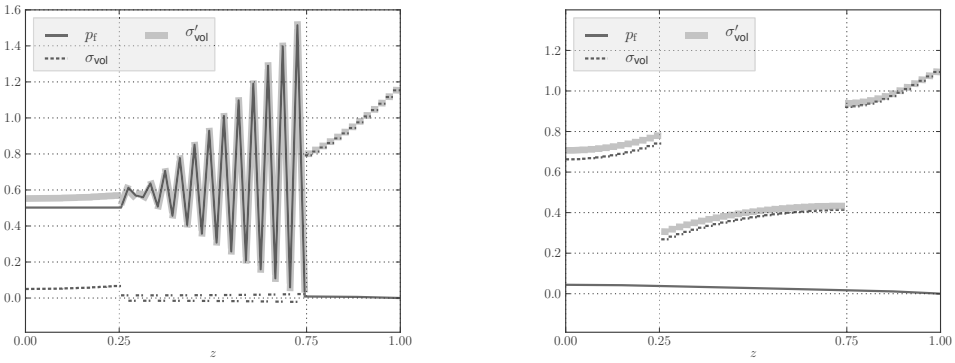
$$\boldsymbol{\nabla}(\lambda + \mu)\boldsymbol{\nabla} \cdot \boldsymbol{u} + \boldsymbol{\nabla} \cdot \mu\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}p_{\mathrm{f}} = \boldsymbol{r}, \qquad \boldsymbol{\nabla} \cdot \boldsymbol{u} = \tilde{q}, \tag{32}$$

with the incompressible elastic equations

$$\boldsymbol{\nabla}\mu\boldsymbol{\nabla} \cdot \boldsymbol{u} + \boldsymbol{\nabla} \cdot \mu\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}\overline{p}_{\mathrm{s}} = \boldsymbol{r}, \qquad \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0. \tag{33}$$

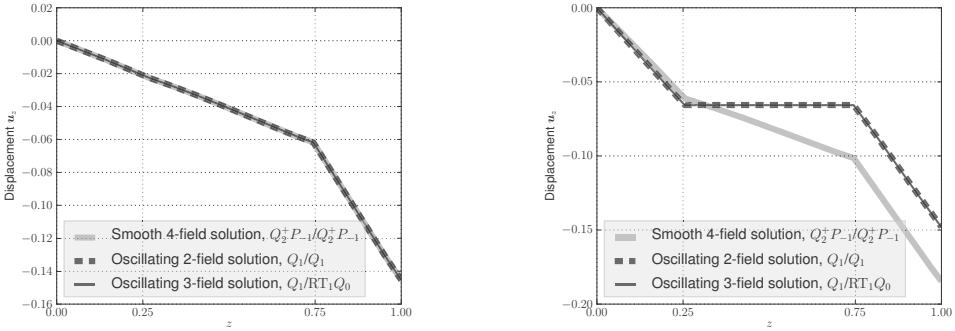
Much of the analysis of eq. (33) is valid also for the present problem. In particular, Bathe [6] notes that the weak form of eq. (33) has two major failure modes: The first is the already mentioned elastic locking, wherein the displacement space is overly constrained by $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$. The second failure mode occurs when the pressure space is too large and contains spurious pressure modes.

[11] Or eqs. (11)–(12) and (2) after eliminating $\boldsymbol{v}_{\mathrm{D}} = 0$.



(a) Low-permeable problem **(b)** Low-compressible problem

Figure 5: Comparison of the two-field (Q_1/Q_1) solutions for a low-permeable and a low-compressible layer. The solutions are erroneous for the fluid pressure in (a) and for the volumetric stress in (b). With this particular choice of elements (and problem geometry), the volumetric stress does not oscillate, but the error is still obvious as an abrupt drop in σ'_{vol}. The corresponding plausible pressure solutions are shown in fig. 3c for (a), and in fig. 4c for (b).



(a) Low-permeable problem **(b)** Low-compressible problem

Figure 6: Comparison of the vertical displacement with non-uniform load. Notice the nearly constant displacement in the low-compressible layer in (b), while the low-permeable layer does not lock the displacement (a). $\epsilon = 10^{-8}$

In linear algebra terms, eq. (33) can be discretised as

$$\begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \mathrm{u} \\ \mathrm{p_s} \end{bmatrix} = \begin{bmatrix} \mathrm{r} \\ 0 \end{bmatrix}, \tag{34}$$

where $B_{ij} = b^{\mathrm{I}}(\phi_{\mathrm{s}}^j, \boldsymbol{\phi}_{\mathrm{u}}^i)$ (and $\boldsymbol{u}$ is approximated as $\boldsymbol{u}_h = \sum_k \mathrm{u}^k \boldsymbol{\phi}_{\mathrm{u}}^k$; similarly for p). Then, locking follows when kernel(B^{T}) does not span the displacement space, while spurious pressure modes are a consequence of a too large kernel(B). The same argument can be used in the poroelastic case, eq. (32), except that the presence of locking is now determined by the space spanned by solutions of $\mathrm{B}^{\mathsf{T}}\mathrm{u} = \tilde{\mathrm{q}}$ instead of the null space.

If the cause of the fluid pressure oscillations lies in the well-posedness of the discrete weak form of the equations, we know from, e.g., [8], that the solvability of the equations and the stability of the solution follows from the Babuška inf-sup condition [3], which should be fulfilled for any mesh size h:

$$\gamma_0 \leq \gamma_h = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{|c(v_h, w_h)|}{\|v_h\|_V \|w_h\|_V}, \tag{35}$$

for some $\gamma_0 > 0$. In the four-field formulation, for example, the discrete space is $V_h = \boldsymbol{V}_{\mathrm{u}} \times V_{\mathrm{f}} \times \boldsymbol{V}_{\mathrm{v}} \times V_{\mathrm{s}}$ and v_h, w_h are functions in this space, e.g., $v_h = (v_{\boldsymbol{u}}, v_{p_{\mathrm{f}}}, v_{\boldsymbol{v}_{\mathrm{D}}}, v_{\overline{p}_{\mathrm{s}}}) \in V_h$. The key insight is that this condition must be fulfilled for the complete coupled system of equations, and not only separately for the fluid velocity/fluid pressure and the solid displacement/solid pressure. Hence, c in eq. (35) is defined as

$$\begin{aligned} c(\boldsymbol{\phi}, \boldsymbol{\psi}) &= a_{\mathrm{f}}^{\mathrm{II}}(\phi_f, \psi_f) + a_{\mathrm{v}}^{\mathrm{II}}(\phi_v, \psi_v) + a_{\mathrm{u}}^{\mathrm{III}}(\phi_u, \psi_u) + a_{\mathrm{s}}^{\mathrm{III}}(\phi_s, \psi_s) + b^{\mathrm{I}}(\phi_f, \psi_u) \\ &+ b^{\mathrm{II}}(\phi_f, \psi_v) + b^{\mathrm{II}}(\psi_f, \phi_v) + c^{\mathrm{II}}(\phi_v, \psi_f) + b^{\mathrm{I}}(\psi_f, \phi_u) + b^{\mathrm{III}}(\psi_s, \phi_u) + b^{\mathrm{III}}(\phi_s, \psi_u). \end{aligned} \tag{36}$$

In the special case of symmetric saddle-point problems, on the canonical form $a(v,u) + b(v,p) + b(u,q) = l((v,q)),\ \forall (v,q) \in V$ and with a coercive, the following Brezzi conditions [8] are equivalent to the Babuška condition. The Brezzi coercivity constant α_h is

$$\alpha_h = \inf_{u \in Z_h} \sup_{v \in Z_h} \frac{a(u,v)}{\|u\|_V \|v\|_V}, \tag{37}$$

with $Z_h = \{v \in V_h \,|\, b(v,q) = 0,\ \forall q \in Q_h\}$, while the Brezzi inf-sup constant[12] β_h is

$$\beta_h = \inf_{q \in Q_h} \sup_{v \in V_h} \frac{b(v,q)}{\|v\|_V \|q\|_Q} \tag{38}$$

Both of these should be bounded from below as $h \to 0$. The Brezzi inf-sup constant is particularly interesting, because zero values for β_h indicate the presence of spurious modes in Q_h (as we stated in terms of the kernel of the matrix B in the previous section).

The two-field formulation approaches a saddle-point problem when $S = 0$ and $\boldsymbol{\Lambda} \to 0$, in which case it is similar to the mixed linear elasticity problem (for finite λ). Spurious pressure modes are then associated with zero values of the Brezzi inf-sup constant,

$$\beta_h = \inf_{q \in V_{\mathrm{f}}} \sup_{\boldsymbol{v} \in \boldsymbol{V}_{\mathrm{u}}} \frac{b^{\mathrm{I}}(q, \boldsymbol{v})}{\|q\|_{V_{\mathrm{f}}} \|\boldsymbol{v}\|_{\boldsymbol{V}_{\mathrm{u}}}}. \tag{39}$$

[12]The Brezzi inf-sup condition is also known as the Ladyzhenskaya-Babuška-Brezzi (LBB) condition.

The three-field (fluid velocity) problem, however, is a true saddle-point problem whenever $S = 0$ (and symmetric when $c^{\mathrm{II}} = 0$). We can define

$$a((\boldsymbol{v}, \boldsymbol{w}), (\boldsymbol{x}, \boldsymbol{y})) = a_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{v}, \boldsymbol{x}) + a_{\mathrm{v}}^{\mathrm{II}}(\boldsymbol{w}, \boldsymbol{y}), \tag{40}$$

$$b(q, (\boldsymbol{v}, \boldsymbol{w})) = b^{\mathrm{I}}(q, \boldsymbol{v}) + b^{\mathrm{II}}(q, \boldsymbol{w}), \tag{41}$$

$$l((p, \boldsymbol{v})) = l_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{v}) + l_{\mathrm{f}}^{\mathrm{II}}(p), \tag{42}$$

and restate eqs. (10) and (15)–(16) in the form of a canonical saddle-point problem: Find the solution $(\boldsymbol{u}, \boldsymbol{v}_{\mathrm{D}}, p_{\mathrm{f}}) \in V$ satisfying

$$a((\boldsymbol{v}, \boldsymbol{w}), (\boldsymbol{u}, \boldsymbol{v}_{\mathrm{D}})) + b(p_{\mathrm{f}}, (\boldsymbol{v}, \boldsymbol{w})) + b(q, (\boldsymbol{u}, \boldsymbol{v}_{\mathrm{D}})) = l((q, \boldsymbol{v})), \quad \forall (\boldsymbol{v}, \boldsymbol{w}, q) \in V, \tag{43}$$

with Brezzi stability constants

$$\alpha_h = \inf_{(\boldsymbol{v},\boldsymbol{w}) \in Z_h} \sup_{(\boldsymbol{x},\boldsymbol{y}) \in Z_h} \frac{a_{\mathrm{u}}^{\mathrm{I}}(\boldsymbol{v}, \boldsymbol{x}) + a_{\mathrm{v}}^{\mathrm{II}}(\boldsymbol{w}, \boldsymbol{y})}{(\|\boldsymbol{v}\|_{\boldsymbol{V}_{\mathrm{u}}} + \|\boldsymbol{w}\|_{\boldsymbol{V}_{\mathrm{v}}})(\|\boldsymbol{x}\|_{\boldsymbol{V}_{\mathrm{u}}} + \|\boldsymbol{y}\|_{\boldsymbol{V}_{\mathrm{v}}})}, \tag{44}$$

$$\beta_h = \inf_{q \in V_{\mathrm{f}}} \sup_{(\boldsymbol{v},\boldsymbol{w}) \in \boldsymbol{V}_{\mathrm{u}} \times \boldsymbol{V}_{\mathrm{v}}} \frac{b^{\mathrm{I}}(q, \boldsymbol{v}) + b^{\mathrm{II}}(q, \boldsymbol{w})}{\|q\|_{V_{\mathrm{f}}}(\|\boldsymbol{v}\|_{\boldsymbol{V}_{\mathrm{u}}} + \|\boldsymbol{w}\|_{\boldsymbol{V}_{\mathrm{v}}})}. \tag{45}$$

The Brezzi inf-sup constant is therefore zero only when the individual terms b^{I} and b^{II} are. These individual couplings between the variables are similar to those of well-known problems, which have known stable choices of finite element spaces:

- The displacement-fluid pressure coupling is similar to the displacement-*solid* pressure coupling in the mixed linear elasticity problem (as shown),
- the displacement-solid pressure coupling is the same as in the mixed linear elasticity problem, and
- the fluid velocity-fluid pressure coupling is same as the Darcy flow problem.

The separation of the coupling terms in the Brezzi inf-sup condition motivates our strategy of choosing combinations of element spaces that satisfy these individual problems. Hence, p_{f} should be chosen to be an element that is usable for mixed formulations of both linear elasticity and fluid flow. For example, if an element combination that is stable for mixed linear elasticity is chosen for $\boldsymbol{u}$ and $\overline{p}_{\mathrm{s}}$, and a combination that is stable for Darcy flow is chosen for $\boldsymbol{v}_{\mathrm{D}}$ and p_{f}, we must then ensure that the resulting combination for $\boldsymbol{u}$ and p_{f} is also stable for mixed linear elasticity. An example of a combination that could work is the lowest order Raviart-Thomas (RT) for $\boldsymbol{v}_{\mathrm{D}}$-$p_{\mathrm{f}}$ and the lowest order Crouzeix-Raviart (CR) or Rannacher-Turek (TR) elements for $\boldsymbol{u}$-$\overline{p}_{\mathrm{s}}$. Both pressure elements (fluid and solid) are then piecewise constant, so the $\boldsymbol{u}$-p_{f} combination is also potentially stable (CR or TR).

With these guidelines, we proceed to examine the stability of a number of combinations of finite elements.

Table 1: Summary of pairwise element combinations. Elements of polynomial order k are classified as P_k or Q_k for Lagrangian elements, while RT_k, CR_k and TR_k are the Raviart-Thomas, Crouzeix-Raviart (triangular) and Rannacher-Turek (quadrilateral) non-conforming elements, respectively. Discontinuous elements are marked as "$_{-k}$" (except $k = 0$, where this is implicit). Enriched (bubble) elements are marked by "$^{+}$".

(a) Triangular elements

Element	Comment
P_1P_1	Equal order Lagrange (lowest order)
P_2P_2	Equal order Lagrange
RT_1P_0	Lowest order Raviart-Thomas (Hdiv) vector element
$P_2^+P_1$	"Good element" (M. Fortin, via [11])
P_2P_1	Lowest order Taylor-Hood
P_2P_0	Only linear convergence in P_2 [10]
CR_1P_0	Lowest order Crouzeix-Raviart non-conforming element
$P_2^+P_{-1}$	From [10]; "optimal" [6], "good element" (M. Fortin, via [11])
$P_1^+P_1$	MINI [2]

(b) Quadrilateral elements

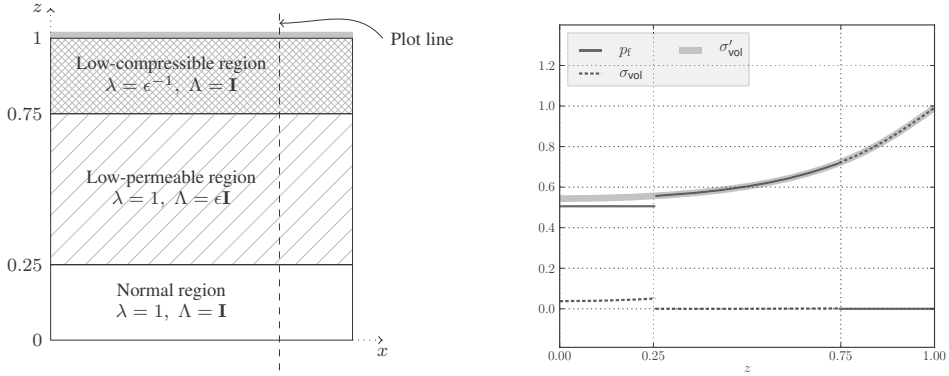
Element	Comment
Q_1Q_1	Equal order Lagrange (lowest order)
Q_2Q_2	Equal order Lagrange
RT_1Q_0	Lowest order Raviart-Thomas (Hdiv) vector element
Q_2Q_1	Lowest order Taylor-Hood
Q_2Q_0	Only linear convergence in Q_2
TR_1Q_0	Lowest order Rannacher-Turek non-conforming
Q_1Q_0	One of the most popular elements in practice [11], LBB unstable (but still usable)
Q_2P_{-1}	Discontinuous, linear (rather than bilinear) pressure; "optimal" [6], "most accurate 2D element" [11]
$Q_1^{++}Q_1$	Quadrilateral MINI analogue [4]

5 Convergence testing

The Babuška or Brezzi conditions require careful work to evaluate analytically, even for a single element family on a two-field problem. For a large number of combinations on three- and four-field problems, it is impractical. As an alternative, the conditions may be tested numerically on a series of meshes, by solving the generalised eigenvalue problems associated with the Babuška or Brezzi conditions [9, 19]. Automated tools are available for this purpose, e.g, ASCoT [20]. The full generality with regards to element definitions and boundary conditions is however not yet achieved. Hence, we have chosen to analyse the element combinations by investigating their real performance on a number of concrete test cases.

We have selected several element pairs, listed in table 1, that are in common use, and tested combinations of these. Using the four-field formulation as an example, we could choose these element pairs: RT_1Q_0 for $\boldsymbol{v}_\mathrm{D}$-$p_\mathrm{f}$ and Q_2P_{-1} for $\boldsymbol{u}$-$\overline{p}_\mathrm{s}$, resulting in Q_2Q_0 for $\boldsymbol{u}$-p_f. This is written as the element combination $Q_2P_{-1}/\mathrm{RT}_1Q_0$.

Two of the test cases are as described earlier: A problem with a low-permeable layer embedded in a normal one from fig. 1a, and the Barry-Mercer problem, with a point pressure source inside a low-permeable material from fig. 2a. For the Barry-Mercer problem, we use elastic parameters $\lambda = \mu = 1$, a time step of $\Delta t = 0.01$, and source



(a) The domain **(b)** A solution

Figure 7: The three-material test case. In the top layer, the load is carried by the solid pressure; in the middle layer by the fluid pressure.

strength $p_0 = 1$.

The third test case is shown in fig. 7a. It is a variation of the earlier embedded-layer case, where the top layer is made low-compressible. Thus, there are three layers: The top one low-compressible; the middle one low-permeable; and the bottom one normal. The two three-layer cases are both tested with uniform load and with load on just the right half of the top boundary.

In either case, we evaluate the solution after a single time step. As reported in, e.g., [23], the pressure oscillations tend to smooth out over time, and hence the first time step is the most revealing one.

We have summarised the results in table 2. Most of the results are as expected based on our previous analysis: The equal interpolation elements, and those which are picked from known-stable pairs in table 1 mostly work. The exceptions are the CR_1P_0 and TR_1Q_0 non-conforming elements for $\boldsymbol{u}$-p_f. The CR_1 or TR_1 element might potentially be useful for $\boldsymbol{u}$ when using RT_1 for $\boldsymbol{v}_\mathrm{D}$, since both are first order and both combine with piecewise constant pressure elements. As noted in the table, we were able to "fix" the TR_1 element by setting extra tangential boundary conditions, but this solution is not very satisfactory in general.

The Q_1/RT_1Q_0 combination for the fluid velocity three-field formulation succeeds with the two- and three-material problems, but fails on the Barry-Mercer problem. The latter failure is shown in figure fig. 2b, which illustrates what is called the "checkerboard" spurious pressure mode (as does fig. 1b). This spurious mode is well known and ubiquitous [6, 11]. It can in many cases be "fixed" by juggling of boundary conditions; in particular, by releasing tangential essential conditions. Furthermore, Gresho and Sani [11] state that in their experience (and analysis) the Q_1Q_0 combination actually has optimal convergence after filtering the spurious pressure modes.

Whenever the domain has large permeability contrasts, the solution may contain steep pressure gradients. Discontinuous elements may then be advantageous to avoid localised oscillations in the fluid pressure. Comparing fig. 8a and fig. 8b, it is evident

Table 2: Summary of the numerical stability results for different elements. The test cases are (in order) Uniform load, Right-Half load for the two- and three-material cases, and the Barry-Mercer problem. The three-material case is used when $\overline{p}_\mathrm{s}$ is present, otherwise the two-material case is used.

(a) Triangular elements

Element				Test case		
$\boldsymbol{u}$	$\overline{p}_\mathrm{s}$	$\boldsymbol{v}_\mathrm{D}$	p_f	U	RH	BM
P_1	—	—	P_1	fail	fail	fail
P_2	—	—	P_2	fail	fail	fail
P_1^+	—	—	P_1	OK[1]	OK[1]	OK[1]
P_2	—	—	P_1	OK[1]	OK[1]	OK[1]
P_2	—	RT_1	P_0	OK	OK	OK
CR_1	—	RT_1	P_0	fail[2]	fail[2]	fail[2]
P_2	—	P_2	P_1	OK[1]	OK[1]	OK[1]
P_2^+	—	P_2^+	P_{-1}	OK	OK	OK
P_1^+	P_1	—	P_1	OK[1]	OK[1]	OK
P_1^+	P_1	RT_1	P_0	fail	fail	fail
P_2^+	P_{-1}	RT_1	P_0	OK	OK	OK
P_2	P_0	RT_1	P_0	OK	OK	OK
P_2^+	P_{-1}	P_2^+	P_{-1}	OK	OK	OK

[1]Continuous pressure elements exhibit local pressure spikes

[2]Singular coefficient matrix

(b) Quadrilateral elements

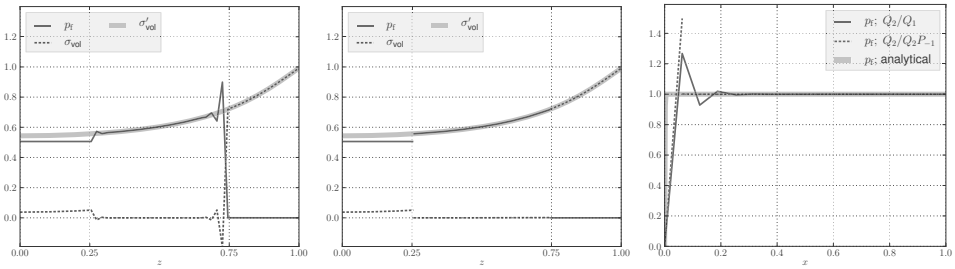
Element				Test case		
$\boldsymbol{u}$	$\overline{p}_\mathrm{s}$	$\boldsymbol{v}_\mathrm{D}$	p_f	U	RH	BM
Q_1	—	—	Q_1	fail	fail	fail
Q_2	—	—	Q_2	fail	fail	fail
Q_1^{++}	—	—	Q_1	OK[1]	OK[1]	OK[1]
Q_2	—	—	Q_1	OK[1]	OK[1]	OK[1]
Q_2	—	RT_1	Q_0	OK	OK	OK
TR_1	—	RT_1	Q_0	fail[3]	fail[3]	OK
Q_1	—	RT_1	Q_0	OK	OK	fail
Q_2	—	Q_2	Q_0	OK	OK	OK
Q_1^{++}	Q_1	—	Q_1	OK[1]	OK[1]	OK
Q_2	P_{-1}	—	Q_1	OK[1]	OK[1]	OK[1]
TR_1	Q_0	RT_1	Q_0	OK	fail	OK
Q_1	Q_0	RT_1	Q_0	OK	OK	fail
Q_2	P_{-1}	Q_2	P_{-1}	OK	OK	OK

[3]Succeeds when tangential displacement BCs are added

that the continuous pressure elements cannot represent the gradient at the interface between the high- and low-permeable region, and the resulting overshoot induces local oscillations in the pressure solution. When using discontinuous elements for the fluid pressure, these oscillations are not present. Discontinuous elements for the fluid pressure can not be used in the two-field formulation, where H^1 regularity is required.

Nevertheless, local pressure oscillations may still occur in certain situations, for example in early times of the Terzaghi consolidation problem. Terzaghi's problem, analysed in for example [24], describes the vertical consolidation of saturated soil. One end of the soil column is drained, and a compressive force of unit magnitude is instantaneously applied. In this case, both continuous and discontinuous elements lead to some overshoot of the fluid pressure, as shown in fig. 8c. In contrast to the earlier case, this problem cannot be well approximated with a small number of elements; arguably, the best approximation to the continous pressure solution at early times is a constant ($p_\mathrm{f} = 1$), but this solution violates the essential boundary condition at the drained end ($p_\mathrm{f} = 0$). Hence, this problem requires additional stabilisation to avoid initial oscillations for short time steps [1, 12, 21].

Depending on the model and on the desired properties of the solution, we list some combinations of element spaces that we find attractive.



(a) Three-layer problem using Q_2/Q_1 and $\epsilon = 10^{-8}$

(b) Three-layer problem; Q_2/Q_2P_{-1} and $\epsilon = 10^{-8}$

(c) The Terzaghi consolidation problem with 32 elements at time $T = 10^{-6}$

Figure 8: Using discontinuous elements for the fluid pressure (b) avoids local oscillation at the edge of the low-permeable material, where the pressure gradient is very steep. It does not, however, smoothly handle the pressure front in early stages of the Terzaghi consolidation problem (c).

- For a fast solver, the two-field formulation may be desirable. The fluid pressure solution must then be a subspace of H^1, (i.e., continuous), and localised pressure oscillations are unavoidable, as remarked above, unless stabilisation is added (such as the flow perturbation proposed by Aguilar *et al.*[1]). The MINI element combination (P_1^+/P_1), or its quadrilateral analogue (Q_1^{++}/Q_1) are attractive choices. The Taylor-Hood element (P_2/P_1 or Q_2/Q_1) is also stable, but the higher accuracy in $\boldsymbol{u}$ may be wasted since $\boldsymbol{v}_\mathrm{D}$ is only piecewise constant.

- If higher accuracy of $\boldsymbol{v}_\mathrm{D}$ is required, the fluid velocity three-field solution is warranted. A popular choice for the fluid velocity is the lowest order Raviart-Thomas elements, with piecewise constant fluid pressure. However, the simplest Stokes-stable element to combine with piecewise constant pressure is Crouzeix-Raviart (or Rannacher-Turek for quadrilaterals), which we found to be problematic. One would then have to use quadratic displacement (P_2/RT_1P_0 or Q_2/TR_1Q_0), which is rather expensive for a method which is still only first order accurate in the velocity. An alternative might be to follow the precept of Phillips and Wheeler [17], and use the Discontinuous Galerkin method for the displacement, or to use a variant which has second order accuracy also for the velocity (such as $P_2^+/P_2^+P_{-1}$ or Q_2/Q_2P_{-1}).

- When low-compressible materials are present, the solid pressure three-field formulation (or even the four-field formulation) may be required. A good choice for the former appears to be the MINI combination $P_1^+P_1/P_1$ or $Q_1^{++}Q_1/Q_1$, although the problem of localised oscillations in both fluid and solid pressure around discontinuities reappears. For the four-field formulation, we recommend $P_2^+P_{-1}/P_2^+P_{-1}$ or the quadrilateral Q_2P_{-1}/Q_2P_{-1}.

6 Concluding remarks

In this paper we have investigated the spurious pressure oscillations that are present in the finite element solution of the poroelastic equations for small time steps and low-permeable materials.

Through comparison with the displacement-solid pressure mixed formulation of linear elasticity, we identify the spurious pressure modes as a specific consequence of a vanishing Brezzi inf-sup constant β_h. Since the Brezzi inf-sup condition for the poroelastic equations takes on a similar form as in, e.g., the mixed linear elasticity or Stokes problem, this identification opens up the field to a plethora of stable element candidates. These can be used directly for the basic solid displacement-fluid pressure two-field formulation of poroelasticity, or in combinations for the various three- and four-field formulations involving solid pressure and/or fluid velocity.

Extensive numerical investigation of the stability of a large set of two-, three- and four-field models have been performed. These investigations provide evidence that most of the element combinations recommended by our theoretical guidelines give oscillation-free solutions for the pressure.

Acknowledgements

The authors would like to thank Dr. Marie Rognes at Simula Research Laboratory for her contributions to the present paper.

Bibliography

[1] G. Aguilar, F. Gaspar, F. Lisbona, and C. Rodrigo. Numerical stabilization of Biot's consolidation model by a perturbation on the flow equation. *International Journal of Numerical Methods in Engineering*, 75(11):1282–1300, 2008. ISSN 1097-0207. doi: 10.1002/nme.2295.

[2] D. N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344, 1984. doi: 10.1007/BF02576171.

[3] I. Babuška. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973. ISSN 0029-599X. doi: 10.1007/BF01436561.

[4] W. Bai. The quadrilateral 'Mini' finite element for the Stokes problem. *Computer Methods in Applied Mechanics and Engineering*, 143:41–47, 1997. doi: 10.1016/S0045-7825(96)01146-2.

[5] S. I. Barry and G. N. Mercer. Flow and deformation in poroelasticity—I unusual exact solutions. *Mathematical and Computer Modelling*, 30(9):23–29, 1999. doi: 10.1016/S0895-7177(99)00177-6.

[6] K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, 1996.

[7] M. A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12(2):155–164, 1941. doi: 10.1063/1.1712886.

[8] F. Brezzi. On the existence, uniqueness and approximation of saddle-point proble arising from Lagrangian multipliers. *RAIRO Analyse Numérique*, R–2:129–151, 1974.

[9] D. Chapelle and K.-J. Bathe. The inf-sup test. *Computers & Structures*, 47(4–5): 537–545, 1993. doi: 10.1016/0045-7949(93)90340-J.

[10] M. Crouzeix and P. A. Raviart. Conforming and nonconforming finite element methods for solving the stationary Stokes equations. *RAIRO Analyse Numérique*, 7:33–75, 1973.

[11] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method. Volume 2: Isothermal Laminar Flow*. John Wiley and sons, 1998.

[12] J. Korsawe and G. Starke. A least-squares mixed finite element method for Biot's consolidation problem in porous media. *SIAM Journal on Numerical Analysis*, 43 (1):318–339, 2006. doi: 10.1137/S0036142903432929.

[13] R. Liu, M. F. Wheeler, and C. N. Dawson. *Discontinuous Galerkin finite element solution for poromechanics*. PhD thesis, University of Texas at Austin, 2006.

[14] M. A. Murad and A. F. D. Loula. On stability and convergence of finite element approximations of Biot's consolidation problem. *International Journal of Numerical Methods in Engineering*, 37:645–667, 1994. doi: 10.1002/nme.1620370407.

[15] M. A. Murad, V. Thomée, and A. F. D. Loula. Asymptotic behavior of semidiscrete finite-element approximations of Biot's consolidation problem. *SIAM Journal on Numerical Analysis*, 33(3):1065–1083, 1996. doi: 10.1137/0733052.

[16] C. E. Neuzil. Hydromechanical coupling in geologic processes. *Hydrogeology Journal*, pages 11:41–83, 2003.

[17] P. Phillips and M. Wheeler. A coupling of mixed and discontinuous Galerkin finite-element methods for poroelasticity. *Computational Geosciences*, 12(4):417–435, 2008. doi: 10.1007/s10596-008-9082-1.

[18] P. J. Phillips and M. F. Wheeler. Overcoming the problem of locking in linear elasticity and poroelasticity: an heuristic approach. *Computational Geosciences*, 13(1):5–12, 2009. doi: 10.1007/s10596-008-9114-x.

[19] J. Qin. *On the convergence of some low order mixed finite elements for incompressible fluids*. PhD thesis, The Pennsylvania State University, 1994.

[20] M. E. Rognes. Automated testing of saddle point stability conditions. Unpublished, to appear in Logg A, Mardal KA, Wells GN (eds.) *Automated Scientific Computing*. Springer.

[21] M. Tchonkova, J. Peters, and S. Sture. A new mixed finite element method for poro-elasticity. *International Journal for Numerical and Analytical Methods in Geomechanics*, 32(6):579–606, 2008. doi: 10.1002/nag.630.

[22] P. A. Vermeer and A. Verruijt. An accuracy condition for consolidation by finite elements. *International Journal for Numerical and Analytical Methods in Geomechanics*, 5(1):1–14, 1981. doi: 10.1002/nag.1610050103.

[23] J. Wan. *Stabilized Finite Element Methods for Coupled Geomechanics and Multiphase Flow*. PhD thesis, Stanford University, 2003.

[24] H. F. Wang. *Theory of Linear Poroelasticity with Applications to Geomechanics and Hydrogeology*. Princeton University Press, 2000.

[25] O. Zienkiewicz, A. Chan, M. Pastor, D. Paul, and T. Shiomi. Static and dynamic behaviour of soils: a rational approach to quantitative solutions. I. fully saturated problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 429(1877):285–309, 1990. ISSN 0080-4630.

Paper IV

A parallel block preconditioner for large scale poroelasticity with highly heterogeneous material parameters

A parallel block preconditioner for large scale poroelasticity with highly heterogeneous material parameters

J. B. Haga[1,3], **H. P. Langtangen**[2,4], **H. Osnes**[3]

[1] Computational Geosciences, Simula Research Laboratory
PO Box 134, N-1325 Lysaker, Norway

[2] Center for Biomedical Computing, Simula Research Laboratory
PO Box 134, N-1325 Lysaker, Norway

[3] Department of Mathematics, University of Oslo,
PO Box 1053 Blindern, N-0316 Oslo, Norway

[4] Department of Informatics, University of Oslo,
PO Box 1080 Blindern, N-0316 Oslo, Norway

Abstract

Large-scale simulations of coupled flow in deformable porous media require iterative methods for solving the systems of linear algebraic equations. Construction of efficient iterative methods is particularly challenging in problems with large jumps in material properties, which is often the case in realistic geological applications, such as basin evolution at regional scales. The success of iterative methods for such problems depends strongly on finding effective preconditioners with good parallel scaling properties, which is the topic of the present paper.

We present a parallel preconditioner for Biot's equations of coupled elasticity and fluid flow in porous media. The preconditioner is based on an approximation of the exact inverse of the two-by-two block system arising from a finite element discretisation. The approximation relies on a highly scalable calculation of the global Schur complement of the coefficient matrix, combined with generally available state-of-the-art multilevel preconditioners for the individual blocks. This preconditioner is shown to be robust on problems with highly heterogeneous material parameters. We investigate the weak and strong parallel scaling of this preconditioner on up to 512 processors, and demonstrate its ability on a realistic basin-scale problem in poroelasticity with over 8 million tetrahedral elements.

1 Introduction

Iterative methods have proven to be the most scalable approach for parallel solvers for algebraic systems of equations, such as those arising from discretisations of partial differential equations (PDEs). Nonetheless, the efficiency of iterative solvers is highly problem-dependent and sensitive to the parameters of the system. Biot's equations [4], describing the coupled poroelastic response of fluid-filled materials, have been shown to be a difficult problem for such solvers due to the extreme jumps that some of the material parameters exhibit in realistic problems. As a result, direct solvers are often employed in such situations. Direct solvers, however, suffer from suboptimal scaling in time [9] and in space [14]. Thus, for truly large-scale problems, such as realistic basin-scale models, efficient and robust iterative methods must be found.

In [17], the present authors demonstrated the efficacy of a preconditioner based on the exact block decomposition in the serial case, using the Schur complement of the 2×2 coefficient matrix. For the individual blocks, an algebraic multigrid (AMG) preconditioner is used. The AMG preconditioner has been shown to have good parallel scaling properties for up to thousands of processors [2, 21]. Given scalable preconditioners for the individual blocks, block preconditioners that work on the unmodified blocks of the coefficient matrix are relatively straightforward to parallelise. The Schur complement, however, requires special care. Elman *et al.* [10, 11] studied the parallel scaling of block preconditioners based on the Schur complement for the Navier-Stokes problem. Simpler Schur complement block preconditioners have been employed successfully for Biot's equation [25, 27]. However, it remains to investigate the parallel efficiency of the more advanced block preconditioners from [17], particularly targeting large jumps in material parameters. This is exactly the topic of the present paper.

We perform extensive numerical investigations on model problems in two and three dimensions on a computer cluster using up to 512 processors to verify parallel scalability. Additionally, we perform tests on a realistic basin model exported from a commercial basin simulator. This model is too large to be solved by direct methods, and has so far proven intractable to standard iterative methods due to the strong contrasts in the material parameters (in particular, the permeability).

This paper is organised as follows. In sec. 2 the governing equations of the poroelastic problem are presented, followed by a brief overview of their weak form and the approximation this leads to in the finite element method. An outline of block preconditioners is found in sec. 3, along with algorithms for construction of the distributed Schur complement approximation. Sec. 4 shows how the mathematical model is implemented in software, and details how the parallelism is achieved, while sec. 5 reports the results of the numerical investigations including parallel scaling results. Finally, we give some concluding remarks in sec. 6.

2 Mathematical model

The equations describing poroelastic flow and deformation are derived from the principles of conservation of fluid mass and the balance of forces on the porous matrix. The linear

poroelastic equations can, in the small-strains regime, be expressed as

$$S\dot{p} - \boldsymbol{\nabla} \cdot \boldsymbol{\Lambda} \boldsymbol{\nabla} p + \alpha \boldsymbol{\nabla} \cdot \dot{\boldsymbol{u}} = q, \tag{1}$$

$$\boldsymbol{\nabla}(\lambda + \mu) \boldsymbol{\nabla} \cdot \boldsymbol{u} + \boldsymbol{\nabla} \cdot \mu \boldsymbol{\nabla} \boldsymbol{u} - \alpha \boldsymbol{\nabla} p = r. \tag{2}$$

Here, we subsume body forces such as gravitational forces into the right-hand side source terms q and r. The primary variables are p for the fluid pressure and $\boldsymbol{u}$ for the displacement of the porous medium. Furthermore, S and $\boldsymbol{\Lambda}$ are the fluid storage coefficient and the flow mobility respectively, α is the Biot-Willis fluid/solid coupling coefficient, and λ and μ are the Lamé elastic parameters.

The fluid (Darcy) velocity is often of particular interest in poroelastic calculations. It can be written

$$\boldsymbol{v}_{\mathrm{D}} = -\boldsymbol{\Lambda} \boldsymbol{\nabla} p, \tag{3}$$

and represents the net macroscopic flux. For the displacement equation, the main secondary quantity of interest is the effective stress tensor,

$$\tilde{\sigma} = (\alpha p + p_{\mathrm{s}}) I + \mu(\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^{\mathsf{T}}), \tag{4}$$

which is written here using the solid pressure

$$p_{\mathrm{s}} = -\lambda \boldsymbol{\nabla} \cdot \boldsymbol{u}. \tag{5}$$

2.1 Weak time-discrete form.

We employ a first-order backward finite difference method in time, which leads to the time-discrete form of eq. (1)

$$Sp - \Delta t \boldsymbol{\nabla} \cdot \boldsymbol{\Lambda} \boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{u} = q \Delta t + S\hat{p} + \boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}. \tag{6}$$

Hatted variables ($\hat{p}$, $\hat{\boldsymbol{u}}$) indicate values from the previous time step, while unmarked variables are taken to be at the current time step.

Next, we rewrite eq. (2) and (6) in weak form, using integration by parts to eliminate second derivatives. We define the following (bi-)linear forms on the domain Ω with boundary Γ,

$$\begin{aligned} a^{\mathrm{p}}(\phi^{\mathrm{p}}, p) &= -\textstyle\int_\Omega S\phi^{\mathrm{p}} p + \Delta t \boldsymbol{\nabla} \phi^{\mathrm{p}} \cdot \boldsymbol{\Lambda} \boldsymbol{\nabla} p \,\mathrm{d}\Omega, \\ l^{\mathrm{p}}(\phi^{\mathrm{p}}) &= -\textstyle\int_\Omega (q\Delta t + S\hat{p} + \boldsymbol{\nabla} \cdot \hat{\boldsymbol{u}}) \phi^{\mathrm{p}} \,\mathrm{d}\Omega + \int_\Gamma \phi^{\mathrm{p}} f_{\boldsymbol{n}} \Delta t \,\mathrm{d}\Gamma, \end{aligned} \tag{7}$$

and

$$\begin{aligned} a^{\mathrm{u}}(\boldsymbol{\phi}^{\mathrm{u}}, \boldsymbol{u}) &= \textstyle\int_\Omega [(\lambda + \mu)(\boldsymbol{\nabla} \cdot \boldsymbol{\phi}^{\mathrm{u}})(\boldsymbol{\nabla} \cdot \boldsymbol{u}) + \mu \boldsymbol{\nabla} \boldsymbol{\phi}^{\mathrm{u}} : \boldsymbol{\nabla} \boldsymbol{u}] \,\mathrm{d}\Omega, \\ b(\boldsymbol{\phi}^{\mathrm{u}}, p) &= -\textstyle\int_\Omega \alpha p \boldsymbol{\nabla} \cdot \boldsymbol{\phi}^{\mathrm{u}} \,\mathrm{d}\Omega, \\ l^{\mathrm{u}}(\boldsymbol{\phi}^{\mathrm{u}}) &= -\textstyle\int_\Omega \boldsymbol{\phi}^{\mathrm{u}} \cdot \boldsymbol{r} \,\mathrm{d}\Omega + \int_\Gamma \boldsymbol{\phi}^{\mathrm{u}} \cdot \boldsymbol{t}_{\boldsymbol{n}} \,\mathrm{d}\Gamma. \end{aligned} \tag{8}$$

The problem then becomes: Find $p \in V_{\mathrm{p}}$ and $\boldsymbol{u} \in \boldsymbol{V}_{\mathrm{u}}$ that satisfy the following relations:

$$a^{\mathrm{p}}(\phi^{\mathrm{p}}, p) + b(\boldsymbol{u}, \phi^{\mathrm{p}}) = l^{\mathrm{p}}(\phi^{\mathrm{p}}) \qquad \forall \phi^{\mathrm{p}} \in V_{\mathrm{p}}, \tag{9}$$

$$a^{\mathrm{u}}(\boldsymbol{\phi}^{\mathrm{u}}, \boldsymbol{u}) + b(\boldsymbol{\phi}^{\mathrm{u}}, p) = l^{\mathrm{u}}(\boldsymbol{\phi}^{\mathrm{u}}) \qquad \forall \boldsymbol{\phi}^{\mathrm{u}} \in \boldsymbol{V}_{\mathrm{u}}. \tag{10}$$

The normal flux $f_{\boldsymbol{n}} = \boldsymbol{v}_{\mathrm{D}} \cdot \boldsymbol{n}$ and the normal stress $\boldsymbol{t}_{\boldsymbol{n}}$ on the boundary Γ appear in these equations as natural boundary conditions. The natural spaces for the continuous problem are $V_{\mathrm{p}} = H^1$ for the pressure and $\boldsymbol{V}_{\mathrm{u}} = \boldsymbol{H}^1$ for the displacement.

The discrete approximation follows from solving the equations for the weak form in finite-dimensional subspaces of the continuous spaces: Given finite element basis functions $\boldsymbol{\phi}_i^{\mathrm{u}} \in \boldsymbol{V}_{\mathrm{u}h} \subset \boldsymbol{V}_{\mathrm{u}}$ spanning the discrete displacement space, and basis functions $\phi_i^{\mathrm{p}} \in V_{\mathrm{p}h} \subset V_{\mathrm{p}}$ spanning the discrete fluid pressure space, the unknown functions are approximated as $\boldsymbol{u} \approx \sum_i \mathrm{u}_i \boldsymbol{\phi}_i^{\mathrm{u}}$ and $p \approx \sum_i \mathrm{p}_i \phi_i^{\mathrm{p}}$. The task is to find the vectors u and p that makes these approximations as good as possible (in some sense); this is done by the finite element method.

2.2 The algebraic system

The algebraic system that results from discretising eqs. (9)–(10) is on the form

$$\mathcal{A}\mathrm{x} = \mathrm{b}, \tag{11}$$

where $\mathcal{A}$ is the coefficient matrix derived from the left-hand sides of eqs. (9) and (10), b is the load vector arising from the right-hand sides, and x is the unknown solution vector. As this is a coupled system of two equations, the coefficient matrix can be viewed as a 2×2 block matrix. The signs of the equations have been chosen so as to make this a symmetric indefinite problem, which we write blockwise as

$$\mathcal{A} = \begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^{\mathsf{T}} & \mathrm{C} \end{bmatrix}, \quad \mathrm{x} = \begin{bmatrix} \mathrm{u} \\ \mathrm{p} \end{bmatrix}, \quad \mathrm{b} = \begin{bmatrix} \mathrm{l}^{\mathrm{u}} \\ \mathrm{l}^{\mathrm{p}} \end{bmatrix}, \tag{12}$$

with A symmetric positive definite and C symmetric negative definite. Using the finite element basis functions $\boldsymbol{\phi}_i^{\mathrm{u}}$ and ϕ_i^{p} introduced above, the entries of each block are

$$A_{ij} = a^{\mathrm{u}}(\boldsymbol{\phi}_i^{\mathrm{u}}, \boldsymbol{\phi}_j^{\mathrm{u}}), \tag{13}$$

$$B_{ij} = b(\boldsymbol{\phi}_i^{\mathrm{u}}, \phi_j^{\mathrm{p}}), \tag{14}$$

$$C_{ij} = a^{\mathrm{p}}(\phi_i^{\mathrm{p}}, \phi_j^{\mathrm{p}}). \tag{15}$$

The load vector is defined in a similar way, with $\mathrm{l}_i^{\mathrm{u}} = l^{\mathrm{u}}(\boldsymbol{\phi}_i^{\mathrm{u}})$ and $\mathrm{l}_i^{\mathrm{p}} = l^{\mathrm{p}}(\phi_i^{\mathrm{p}})$.

The solution of algebraic systems of equations like eq. (11), resulting from finite element discretisations, generally shows poor convergence properties when using iterative solvers. To overcome this, suitable preconditioning is crucial.

3 Block preconditioning methods

We seek a preconditioner that exploits the block structure of eq. (12). The simplest example is perhaps the block Jacobi preconditioner,

$$\mathcal{P}_{\mathrm{J}}^{-1} = \begin{bmatrix} \mathrm{A}^{-1} & 0 \\ 0 & \mathrm{C}^{-1} \end{bmatrix}. \tag{16}$$

Algorithm 1: Application of the block Gauß-Seidel preconditioners to a block vector: $[\mathrm{v}\ \mathrm{q}]^\mathsf{T} \leftarrow \mathcal{P}^{-1}_{\mathrm{g(S)GS}}[\mathrm{w}\ \mathrm{r}]^\mathsf{T}$

```
1 v  ← Ã⁻¹w
2 q′ ← Bᵀv − r
3 q  ← S̃⁻¹q′
4 if symmetric then
5     v′ ← Bq
6     v  ← v − Ã⁻¹v′
7 end
```

The single-block inverses are normally too expensive to compute exactly, and will be approximated by single-block preconditioners. In the following, we mark such approximations with a tilde: $\tilde{\mathrm{A}}^{-1}$ and $\tilde{\mathrm{C}}^{-1}$. By further defining the lower-triangular coupling matrix as

$$\mathcal{G} = \begin{bmatrix} \mathrm{I} & 0 \\ -\mathrm{B}^\mathsf{T}\tilde{\mathrm{A}}^{-1} & \mathrm{I} \end{bmatrix}, \tag{17}$$

we can express the block Gauß-Seidel preconditioner as $\mathcal{P}^{-1}_{\mathrm{GS}} = \mathcal{P}^{-1}_{\mathrm{J}}\mathcal{G}$, and its symmetric variation as $\mathcal{P}^{-1}_{\mathrm{SGS}} = \mathcal{G}^\mathsf{T}\mathcal{P}^{-1}_{\mathrm{J}}\mathcal{G}$.

The Schur complement of the block coefficient matrix $\mathcal{A}$ is defined as $\mathrm{S} = \mathrm{B}^\mathsf{T}\mathrm{A}^{-1}\mathrm{B} - \mathrm{C}$. It is symmetric and positive definite. Following [25] we can write the Generalised Jacobi preconditioner as[13]

$$\mathcal{P}^{-1}_{\mathrm{gJ}} = \begin{bmatrix} \tilde{\mathrm{A}}^{-1} & 0 \\ 0 & -\tilde{\mathrm{S}}^{-1} \end{bmatrix}. \tag{18}$$

As the present authors pointed out in [17], the corresponding Generalised Symmetric Gauß-Seidel preconditioner, which we define by analogy with regular Gauß-Seidel as

$$\mathcal{P}^{-1}_{\mathrm{gSGS}} = \mathcal{G}\mathcal{P}^{-1}_{\mathrm{gJ}}\mathcal{G}, \tag{19}$$

is in fact an exact inverse of $\mathcal{A}$, if the single-block inverses are exact. An inexact version of eq. (19), along with its nonsymmetric cousin $\mathcal{P}^{-1}_{\mathrm{gGS}}$, were shown in [17] to be very robust preconditioners for Biot's equations on a problem with extreme contrasts in the material parameters. Algorithm 1 shows the necessary steps to implement this preconditioner. Each assignment requires one global single-block operation, i.e., the processor-local operation followed by an update of the foreign nodes. The application of the $(1,1)$ preconditioner $\tilde{\mathrm{A}}^{-1}$ to a vector is normally by far the most expensive step of this algorithm, and the symmetric variant is therefore about twice as expensive as nonsymmetric generalised Gauß-Seidel. However, the symmetric variant provides the opportunity to use the Conjugate Gradient method instead of more expensive iterative solvers, which justifies this additional cost. In the remainder of this paper, we focus on the symmetric variant.

[13] In the reference, a scalar multiplier α is used for the $(2,2)$ block; here, $\alpha = -1$.

3.1 The distributed Schur complement approximation

First a small note on terminology: The word "node" is traditionally used both in the parallel computing context and in the PDE context, with different meanings. In the following, we reserve *node* to mean a spatially located unknown in the finite element method, while *computational node* is used for a single computer in a cluster. To further clarify the computational hierarchy, *processor* is used interchangeably with *core* to mean a computing unit that runs a single *process*. One or more processors make up a *die*, and one or more dies make up a computational node. Thus, a typical computational node may have two quad-core dies with a total of eight processors.

We shall come back later to the subject of parallel partitioning, but to simplify the discussion we assume the following properties of the partitioning:

(i) Each node is owned exclusively by one processor. This node is then *interior* to the owning processor. The node may also be present on neighbouring processors, where it is a *foreign* node. We also use the term *border node* for those nodes which are interior, but share an element with (and hence couple to, in the coefficient matrix) a foreign node.

(ii) Every interior node has full cover on the owning processor, i.e., all elements that contain the node are present in the local finite element assembly.

While forming the exact Schur complement is infeasible, an approximation that was shown in [17] to perform well for high-contrasting material parameters is

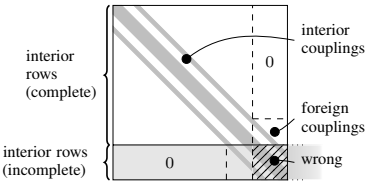
$$\tilde{\mathrm{S}} = \mathrm{Diag}(\mathrm{B}^{\mathsf{T}}(\mathrm{Diag}\,\mathrm{A})^{-1}\mathrm{B}) - \mathrm{C}, \text{[14]} \tag{20}$$

where Diag is an operator that creates a matrix of equal dimension, containing only the diagonal elements.[15] This approximation can be calculated in parallel with overhead equal to that of a single matrix-vector product. To understand how, we look briefly at the behaviour of a parallel matrix-matrix product.

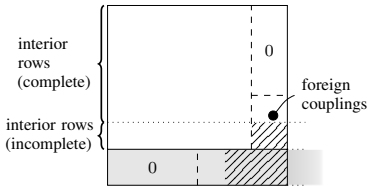
In fig. 1a, we have sketched the structure of a processor-local part of the global coefficient matrix. The salient part is this: All rows *and* columns involving interior nodes are globally correct and complete. Hence, the diagonal of the result of a local matrix-matrix product (shown in fig. 1b) is correct for all entries associated with interior unknowns. Only the entries associated with foreign unknowns are incorrect. This is not a problem, since a matrix-vector product is always followed by an update of the foreign nodes. However, eq. (20) involves a triple matrix product. To ensure that the diagonal of this triple-product is correct for all interior entries, we do need to have globally correct entries for the whole of Diag A; otherwise the product $(\mathrm{Diag}\,\mathrm{A})^{-1}\mathrm{B}$ will *not* have the structure of fig. 1a (the interior columns of the foreign rows will be wrong). The complete algorithm to create the distributed Schur complement of eq. (20) is presented in algorithm 2. The only interprocess communication in this algorithm takes place in step 3, where the diagonal is updated.

[14] We remark that if a crude preconditioner is used for A, the Schur complement should ideally involve an approximation of $\tilde{\mathrm{A}}^{-1}$ rather than A^{-1} [17].

[15] In MATLAB notation, Diag A is written as diag(diag(A)).



(a) Processor-local matrix structure



(b) Structure after local matrix-matrix product

Figure 1: In a processor-local matrix-matrix product, the interior rows with nonzero foreign coupling terms are wrong; only the interior-row part of the diagonal can be trusted

3.2 The single-block preconditioners

The block preconditioners in the previous section depend on the availability of efficient single-block preconditioners $\tilde{A}^{-1}$ and $\tilde{S}^{-1}$. We restrict our attention to preconditioners which are efficient on massively parallel computers. This rules out incomplete and approximate direct solvers such as the otherwise excellent ILU methods.

Adams [1] found AMG to behave very well on problems of elastic deformation, even in the presence of strong material discontinuities. In particular, the smoothed aggregation (SA) method [5, 28] was considered to be the overall superior AMG method for elasticity problems. The present authors likewise found SA to be a nearly optimal preconditioner for the discontinuous Poisson pressure problem (see [16]), and to perform well on the similarly structured Schur complement approximation found in eq. (20) (see [17]).

In the light of these earlier results, and the fact that AMG has been shown to scale very well in parallel, to at least thousands of processors [2, 7, 21, 29], we have chosen to use SA to precondition both the decoupled displacement equation (A) and the Schur complement ($\tilde{S}$).

4 Software framework

We have implemented the finite element discretisation and assembly, the block preconditioners and iterative solvers using the Diffpack C++ framework [8, 23], with extensive modifications in key areas: parallel block systems, parallel partitioning, and mixed finite elements (serial and parallel).

A domain decomposition approach is used for the finite element assembly stage, where each processor works on a subset of the global grid. In the linear algebra stage, message passing (using Message Passing Interface, MPI) is used to formulate globally consistent operations for matrix-vector products, vector inner products, and so on. The main trade-off in this approach is in choosing how to partition the grid between processors. Our choice is mainly motivated by the ease of interfacing with external parallel libraries. Hence, we employ a model wherein each node is owned exclusively by one processor. If we further require that every such interior node is provided with full cover on the owning processor, we gain the desirable property that the matrix rows (and, incidentally, the matrix columns) associated with this node are complete.

Algorithm 2: Construction of the distributed Schur complement approximation $\tilde{\mathrm{S}} \leftarrow \mathrm{Diag}(\mathrm{B}^{\mathsf{T}}\,\mathrm{Diag}(\mathrm{A})^{-1}\mathrm{B}) - \mathrm{C}$

```
1  parallel for each processor P do
2      a^P ← diag(A^P)        ▷ create column vector from diagonal
3      a^P ← update(a^P)      ▷ fetch foreign nodes from neighbours
4      S̃^P ← −C^P
5      for each interior row i do
6          for each nonzero index k in the matrix row B_i^P do
7              S̃_ii^P ← S̃_ii^P + (B_ik^P)^2 (a_k^P)^{-1}
8          end
9      end
10 end
```

The partitioning procedure proceeds in two stages:

(i) Balance the nodes between the processors, while minimising the number of intersected elements,

(ii) Add foreign nodes to each partition until full cover is provided for each interior node.

A hypergraph partitioner, with each hyperedge containing the nodes of one element, should be the ideal way to achieve (i). However, all partitioners use heuristics to achieve acceptable performance, and a graph or even a geometric partitioner may perform equally well on a given problem. We interface with the PHG hypergraph partitioner and a geometric partitioner from Zoltan [6], and with the METIS and ParMETIS [22] graph partitioners.

The single-block AMG preconditioners are from the ML package for Smoothed Aggregation [13], which is part of Trilinos [19]. The ML interface requires the input of complete local rows for the global coefficient matrix, which is greatly aided by the properties of the partitioning listed above.

In addition to the above, we have developed software to import finite element grids, fields and material parameters from Petromod [24], which is one of the leading basin simulation software packages in the oil and gas industry. This allows us to use realistic geometries, initial conditions and material parameters in our tests.

5 Numerical experiments

5.1 Convergence criterion

When using iterative methods for solving algebraic systems of equations, a suitable convergence criterion must be introduced. Different criteria are possible, but the "ideal" criterion which measures the error is generally not available unless the solution is known in advance. More commonly, a convergence criterion based on the residual $\mathrm{r}_k = \mathrm{b} - \mathcal{A}\mathrm{x}_k$ (in the k-th iteration) is used. However, such a criterion may be misleading when

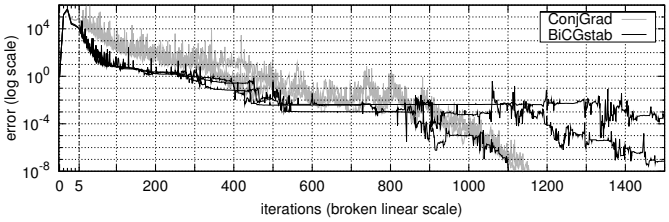


Figure 2: Iterations to reach a given error on the realistic basin case (III)

$\mathcal{A}$ is very ill-conditioned [16], such as with severe jumps in the material parameters. We are less interested in the solution itself than in the convergence properties of the solver, and thus we may exploit a convenient property of iterative solvers: their rate of convergence is independent of the right-hand side b as long as the initial guess contains all eigenvectors of $\mathcal{A}$ [15, ch. 3.4].

Hence, we solve the modified problem $\mathcal{A}\mathrm{x} = 0$ with a randomised initial solution vector x_0, instead of the original $\mathcal{A}\mathrm{x} = \mathrm{b}$. With a zero right-hand side, the error is simply $\mathrm{e}_k = \mathrm{x}_k$. We also note that due to this testing procedure, the exact value of any boundary condition is irrelevant, since these values go into the b vector. The only relevant information in this case is *where* essential boundary conditions are used, since the presence of an essential boundary condition at a node is reflected by a modification to the associated row(s) and column(s) of $\mathcal{A}$.

In the description that follows of the numerical experiments, we use the term *error criterion* (with an associated tolerance ϵ, implying $\|\mathrm{e}_k\| \leq \epsilon$) for the convergence criterion described above. However, in order to measure more narrowly the efficiency of the parallel implementation itself, it is sometimes advantageous to measure the time to complete a fixed number of iterations (convergence criterion $k = k_{\max}$); we shall refer to this as the *iteration criterion*.

5.2 Choice of iterative solver

The coefficient matrix $\mathcal{A}$ is symmetric indefinite. Since the preconditioner is symmetric, the preconditioned coefficient matrix $\mathcal{P}_{gSGS}^{-1}\mathcal{A}$ is also symmetric, and, given sufficiently accurate single-block preconditioners, it may even be positive definite [17]. With such a system of equations, one would normally prefer an iterative solver which can be used with indefinite systems. However, the Conjugate Gradient method is often considered the best choice for symmetric positive definite matrices, and it is known that it can perform well even when there are a few negative eigenvalues [12]. Fig. 2 compares the Stabilised Bi-Conjugate Gradient (BiCGStab) method, which is designed for general use, with the Conjugate Gradient (ConjGrad) method for the realistic basin model described below. Three experiments are shown for each of BiCGStab and Conjugate Gradients, using random initial vectors with error 10^0 (jumping to $\sim 10^5$ in the first iteration).

This is our most difficult test case for the iterative solver. It appears that the Conjugate Gradient method performs just as well as the BiCGStab method, and furthermore that it has much smaller sensitivity to the (random) initial solution vector.

Consequently, we use the Conjugate Gradient method in our experiments.

We should note, lest the results in fig. 2 make our chosen preconditioner look bad, that this test problem is one which we previously have not been able to solve at all using standard iterative solvers and preconditioners. Thus, even a preconditioner which requires 500+ iterations is a significant step forward.

5.3 Scaling

Before looking further into the experimental data, it may be advantageous to have a rough idea what to expect from the results. We can identify five main causes of imperfect parallel scaling:

(i) increased local problem size due to duplicated nodes and imbalance,

(ii) point-to-point (neighbour) communication,

(iii) collective (global) communication,

(iv) increasing number of iterations in the iterative solver for a given accuracy, and

(v) slowdown due to congestion of shared resources (within or beetween computational nodes).

In general, (iv) depends on the chosen preconditioner/iterative solver combination, and can be controlled by using an iteration criterion instead of an error criterion, while (v) is hardware dependent and must usually be discovered through testing.

We investigate the parallel scalability in two different scaling paradigms. In the weak scaling paradigm, the number of nodes (or work) per processor is fixed. Causes (i) and (ii) should then approach constant overhead (after an initial ramp-up), while the cost of cause (iii) is of order $\log P$ on P processors [26].

We also investigate strong scaling, where the total problem size is fixed as the number of processors increases. Strong scaling has received less attention than weak scaling in the literature, but in practical applications the need to solve a large problem *as fast as possible* is perhaps more common than the need to solve a problem that is *as large as possible* in a given time. In this case, the absolute overhead due to causes (i) and (ii) decreases with increasing P. It does not, however, decrease as fast as the amount of useful work per processor. Hence, the relative overhead increases.

We define the *efficiency* as the ratio of the perfect-scaling runtime to the actual runtime, or, equivalently, the number of unknowns processed per unit aggregate time. In D spatial dimensions, the walltime T and efficiency E in the weak scaling paradigm, with N nodes on each of P processors, can be modelled as

$$T(1) = cN, \tag{21}$$

$$T_{\mathrm{w}}(P) = T(1) + acN^{\frac{D-1}{D}} + bc\log P, \tag{22}$$

$$E_{\mathrm{w}}(P) = \frac{T(1)}{T_{\mathrm{w}}(P)} = \left[1 + aN^{\frac{-1}{D}} + bN^{-1}\log P\right]^{-1}, \tag{23}$$

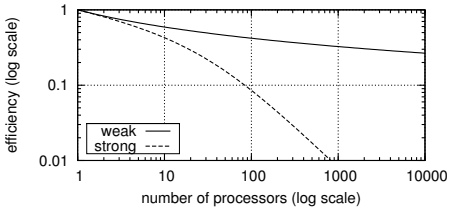


Figure 3: Typical shape of the efficiency curves for strong and weak scaling.

where a, b and c are constant factors depending on the specifics of the problem and of the platform. In the strong scaling paradigm, with $N/P \geq 1$ nodes per processor, these can be modelled as

$$T_{\mathrm{s}}(P) = T(1)/P + ac(N/P)^{\frac{D-1}{D}} + bc\log P, \tag{24}$$

$$E_{\mathrm{s}}(P) = \frac{T(1)}{PT_{\mathrm{s}}(P)} = \left[1 + a(P/N)^{\frac{1}{D}} + b(P/N)\log P\right]^{-1}. \tag{25}$$

This assumes perfectly scalable hardware (no interconnect saturation, etc), and a fixed number of iterations of the iterative solver. The constant a comes from (i)–(ii) above, and b comes from (iii); the quantity $N^{\frac{D-1}{D}}$ is proportional to the number of nodes intersected by a slice through the domain.

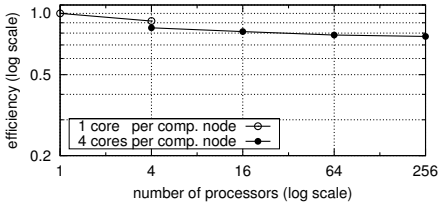
Disregarding the exact value of the various constants, we expect to see efficiency curves of the general shapes shown in fig. 3: A nearly flat, slightly upturned curve on the log-log plot in weak scaling, and a strongly downturned curve in strong scaling.

Comparing this with numerical tests on various hardware is instructive. Two such are shown in fig. 4 for weak scaling, at a fixed number of iterations. On the Cray cluster[16] (fig. 4a), the scaling appears roughly as in the simple model illustrated in fig. 3, except a small ($\sim$ 10%) drop when utilising all four processors on a single computational node instead of one processor on each of four computational nodes. This drop must be caused by contention of a shared resource internally to a computational node, most likely exhaustion of the memory bandwidth. Compare this with a commodity cluster[17] (fig. 4b) when utilising multiple cores per computational node: Four cores on a single computational node, 36% drop in efficiency; eight cores, 64% drop! Clearly, this hardware is not very efficient for such a data intensive workload.

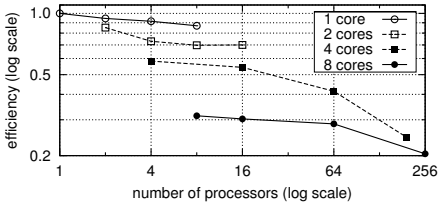
Furthermore, we see a rapid worsening of the efficiency on the commodity cluster when more than a few tens of processors are involved. This does not match our expectation from the weak scaling curve of fig. 3, and we therefore suspect it is caused by congestion of the interconnect between computational nodes. Measuring the time spent in MPI communications shows this to be the major cause, as shown in fig. 4c. On the commodity cluster (which uses a GHz Ethernet interconnect), most of the time is indeed spent doing MPI communication.

The point of this comparison is that the interpretation of parallel scaling experiments must consider the hardware they are performed on, since even a good algorithm may

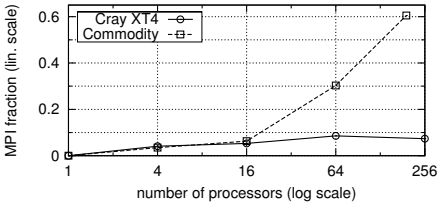
[16]The *hexagon* Cray XT4 cluster located in Bergen, Norway [20].
[17]The *bigblue* computer cluster at Simula Research Laboratory [3].



(a) Cray XT4 cluster



(b) Commodity cluster



(c) Fraction of solution time spent in MPI communication

Figure 4: Weak scaling efficiency on two different hardware platforms

look bad on inadequate hardware. Since we want our algorithm to look good, we perform the remainder of our experiments on the Cray cluster.

5.4 On the number of iterations for the iterative solver

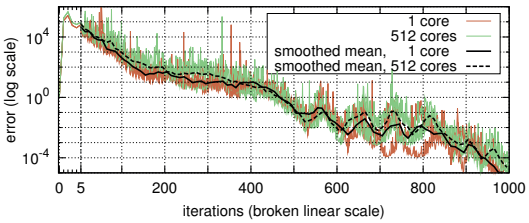
In [17], we estimated the number of iterations of the $\mathcal{P}_{\text{gSGS}}$ preconditioner on the current problem as proportional to $h^{-0.4}$–$h^{-0.5}$, where h is the characteristic element size. It should be remarked that this estimation was performed using only a quite small two-dimensional test problem. There are two questions we need to answer.

(i) Is the number of iterations independent of the number of processors P in the strong scaling paradigm?

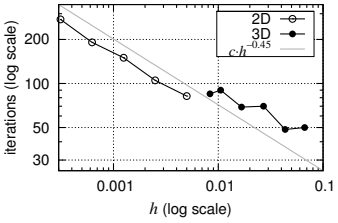
(ii) Does the number of iterations keep growing at about the same rate as previously estimated in the weak scaling paradigm?

Question (i) only makes sense if some of the operations in the iterative solver are not independent of P. Generally, the Conjugate Gradient iteration is independent of P, as is the block preconditioner. However, the Smoothed Aggregation single-block preconditioners behave somewhat differently when P is large. In particular, the high-level aggregates do not cross processor boundaries [29]. To answer this question, we compared the convergence of the basin-scale model from sec. 5.7 when it is run in sequential mode and in parallel using 512 processors. The results, shown in fig. 5a, indicate that the differences in convergence are minimal. The smoothed mean of three experiments is shown for each case, along with the individual experiments.

The answer to question (ii) is found in fig. 5b; $h^{-0.45}$ remains a fair estimate of the order of the solver. The test case used to gather this data is described in sec. 5.5, with



(a) Convergence of test case III on 1 and 512 processors



(b) Iterations of the solver for the error convergence criterion (case II)

Figure 5: The dependence of the convergence on the number of processors (a) and the characteristic grid cell size (b) of the problem

Q^2Q^1 Taylor-Hood elements and a factor 10^{-6} reduction of the error in $L2$-norm as the convergence criterion.

5.5 Test case I: Weak scaling

Layered media with severe jumps in material parameters constitute the normal case in basin modelling. To capture the essence of the numerical difficulties with such media, we have constructed a test problem with three layers as shown in figs. 6a–6b. We have investigated these (and similar) model problems in earlier works. A low-permeable layer with vanishing fluid storage coefficient S creates an ill-defined problem for the decoupled pressure equation [16], which can nevertheless be solved (up to an arbitrary constant) by an AMG-preconditioned iterative solver. The coupling of the fluid pressure with displacement makes the problem well-defined, but when the permeability contrasts are sufficiently strong (starting around $\|\mathbf{\Lambda}_1\|/\|\mathbf{\Lambda}_2\| = 10^{-4}$ with $S = 0$), novel preconditioners such as the one presented herein are required for convergence [17].

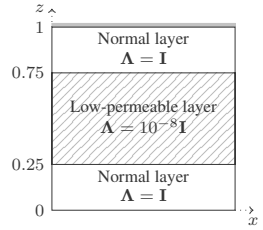
As explained in sec. 5.1, we do not care about the actual boundary conditions, except to note where essential conditions are in use:

- The displacement equation has essential boundary conditions in the normal direction at the sides and the bottom,
- The fluid pressure equation has essential boundary conditions at the top.

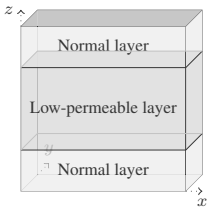
Another difficulty is that of nonphysical oscillations in the fluid pressure, which may occur in models where low-permeable layers are present. Pursuant to the results in [18], we avoid this by using the Taylor-Hood quadrilateral element combination, with second order Lagrange elements for the displacement and first order Lagrange elements for the fluid pressure.

In this test case of weak scaling, each processor is responsible for about 200^2 elements in 2D, or about 16^3 elements in 3D; these are the largest problems that can fit comfortably in the available 1GB of memory per processor. Parallel partitioning is performed using the METIS graph partitioner [22].

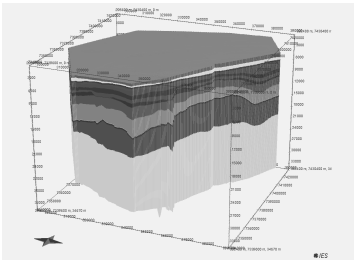
The plots in fig. 7 show the parallel efficiency of the iterative solver phase of a single time step of this model, i.e., of solving eq. (11). When using an iteration convergence



(a) An embedded low-permeable layer in a 2D unit square (I–II)



(b) A low-permeable layer in a 3D unit cube (I–II)



(c) The Vøring basin model, with 16 layers and 1.7 million nodes (III)

Figure 6: The domains of test cases I–III

criterion, the parallel scalability is excellent, with only 10–20% lower efficiency at 512 processors. Since the ratio of foreign to interior nodes is much larger in the three-dimensional case, it is somewhat less efficient than the two-dimensional case. However, once a more practical error criterion is used, this is turned upside down: Since the condition number of the matrix (as a function of the problem size) deteriorates less rapidly in the three-dimensional case, the actual error-reduction efficiency is much better in 3D than in 2D. The 2D efficiency drops below 50% at around 32 processors, while the 3D efficiency still remains above 50% at 512 processors.

5.6 Test case II: Strong scaling

Test case II uses the same model geometry, parameters, elements, and partitioning as test case I. The only difference is that it is fixed in size: 400^2 elements in 2D and 26^3 elements in 3D. Again, the size is determined by memory considerations: These are the largest problems to fit in memory on a single 4GB computational node.

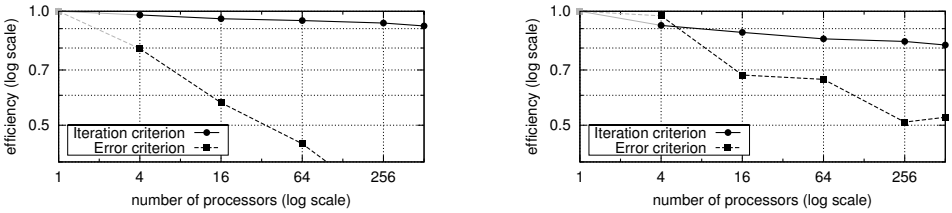
We assume that the rate of error reduction is nearly constant (as discussed in sec. 5.4), and hence that the error and iteration criteria are nearly equivalent; an error criterion with $\epsilon = 10^{-6}$ is used.

The scalability results are shown in fig. 8a. As we may expect from the results of test case I, the 3D test drops off faster in efficiency, but both tests exhibit adequate scalability up to 256 processors.

We remark that in the strong scaling paradigm, the limits of scalability are determined to a large degree by the problem size. A large problem can be subdivided more times before the number of foreign nodes becomes significant. For example, with 256 processors the number of foreign nodes is larger than the number of interior nodes in the 3D test.

5.7 Test case III: Strong scaling on a basin-scale geometry

Our final test case is a realistic model of a sedimentary basin, derived from a real industry model. Shown in fig. 6c, the model consists of 16 distinct layers of sediments, $8.4 \cdot 10^6$ tetrahedral elements, and $1.7 \cdot 10^6$ nodes. No attempt has been made to make



(a) 2D test (see fig. 6a) **(b)** 3D test (see fig. 6b)

Figure 7: Weak scaling in a two- and three-dimensional test case

the computational grid more friendly to finite element calculations, and thus the grid quality is low in some places — outer/inner radius ratios exceed 100 in many elements. The material parameters are also from the real model, and are listed in table 1.

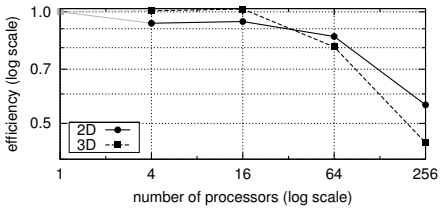
For this test, equal-order Lagrange tetrahedral elements P^1P^1 are used. We believe this to be acceptable, since the fluid storage coefficient S does not vanish anywhere (see discussion in sec. 5.5 and [18]). Even if it were not acceptable, Taylor-Hood elements would simply be too expensive on this grid — the memory requirements would increase almost tenfold, to well over a hundred gigabytes. One possibility would be to use a mixed element with extra internal degrees of freedom, such as the MINI element, and to eliminate the internal degrees of freedom at the element level by static condensation. Such a procedure would reduce the size of the system to that of the P^1P^1 combination used here.

The efficiency results are shown in fig. 9a, with the associated runtime (for both the assembly and the solution phase) in fig. 9b. A peculiarity with these graphs is that the single-processor runtime is only estimated, because the memory requirements for this test case precludes running it on fewer than five computational nodes. This estimate, which is used both to determine the multiplicative factor $T(1)$ in the efficiency and to determine the "perfect scaling" line in fig. 9b, is made by simply subtracting the MPI communication overhead from the five-processor aggregate runtime.

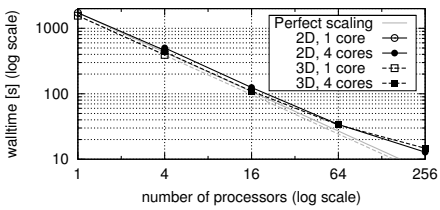
6 Concluding remarks

We have implemented and tested a parallel block preconditioner for the finite element discretisation of a fully coupled 3D problem of fluid flow in elastic porous media. The parallel preconditioner targets especially the challenges of real-world geological problems: unstructured computational grids and heterogeneous material parameters with severe jumps between geological layers. As the numerical results in previous sections show, we achieve strong scaling results for a realistic large-scale basin model that are quite acceptable on up to five hundred processors, thereby making simulations on this scale practical. The performance of this parallel block preconditioner is robust with respect to heterogeneities and severe grid distortion.

The smaller strong scaling case (test case II) shows an earlier drop-off in efficiency. This may be expected from the fact that a smaller problem has a higher ratio of foreign



(a) Solution parallel efficiency.

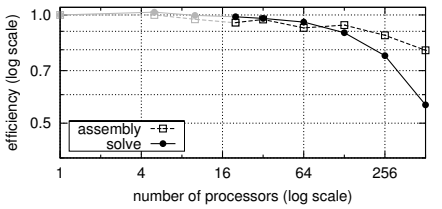


(b) Elapsed time for assembly and 10^{-6} reduction of the error.

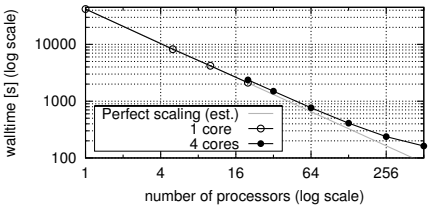
Figure 8: Results for test case II, strong scaling on unit cube

Table 1: Material parameters for test case III.

Layer no.	$S[\mathrm{Pa}^{-1}]$	$\Lambda_x, \Lambda_y[\mathrm{m}^2\mathrm{Pa}^{-1}\mathrm{s}^{-1}]$	$\Lambda_z[\mathrm{m}^2\mathrm{Pa}^{-1}\mathrm{s}^{-1}]$	$\nu[\cdot]$	$G[\mathrm{Pa}]$
1	$1 \cdot 10^{-10} - 2 \cdot 10^{-10}$	$3 \cdot 10^{-1} - 7 \cdot 10^{0}$	$8 \cdot 10^{1} - 2 \cdot 10^{3}$	0.35	$5 \cdot 10^{8}$
2	$1 \cdot 10^{-10} - 2 \cdot 10^{-10}$	$6 \cdot 10^{1} - 3 \cdot 10^{2}$	$2 \cdot 10^{4} - 1 \cdot 10^{5}$	0.35	$5 \cdot 10^{8}$
3	$1 \cdot 10^{-10} - 2 \cdot 10^{-10}$	$3 \cdot 10^{0} - 2 \cdot 10^{1}$	$1 \cdot 10^{2} - 6 \cdot 10^{2}$	0.35	$5 \cdot 10^{8}$
4	$1 \cdot 10^{-10} - 2 \cdot 10^{-10}$	$2 \cdot 10^{-2} - 1 \cdot 10^{-1}$	$8 \cdot 10^{0} - 3 \cdot 10^{1}$	0.35	$5 \cdot 10^{8}$
5	$1 \cdot 10^{-10}$	$5 \cdot 10^{-3} - 7 \cdot 10^{-2}$	$1 \cdot 10^{0} - 2 \cdot 10^{1}$	0.35	$5 \cdot 10^{8}$
6	$1 \cdot 10^{-10}$	$2 \cdot 10^{-6} - 5 \cdot 10^{-2}$	$5 \cdot 10^{-4} - 2 \cdot 10^{1}$	0.35	$5 \cdot 10^{8}$
7	$1 \cdot 10^{-10}$	$1 \cdot 10^{-2} - 3 \cdot 10^{-2}$	$3 \cdot 10^{0} - 5 \cdot 10^{0}$	0.35	$5 \cdot 10^{8}$
8–9	$1 \cdot 10^{-10}$	$2 \cdot 10^{-6} - 1 \cdot 10^{-4}$	$5 \cdot 10^{-4} - 4 \cdot 10^{-2}$	0.35	$5 \cdot 10^{8}$
10	$1 \cdot 10^{-10}$	$2 \cdot 10^{-6} - 4 \cdot 10^{-4}$	$5 \cdot 10^{-4} - 1 \cdot 10^{-1}$	0.35	$5 \cdot 10^{8}$
11	$1 \cdot 10^{-10}$	$2 \cdot 10^{-3} - 5 \cdot 10^{-2}$	$2 \cdot 10^{-1} - 6 \cdot 10^{0}$	0.35	$5 \cdot 10^{8}$
12	$2 \cdot 10^{-10}$	$5 \cdot 10^{-2} - 8 \cdot 10^{0}$	$5 \cdot 10^{0} - 8 \cdot 10^{2}$	0.25	$8 \cdot 10^{8}$
13	$1 \cdot 10^{-10}$	$2 \cdot 10^{-3} - 6 \cdot 10^{-3}$	$4 \cdot 10^{-1} - 1 \cdot 10^{0}$	0.35	$5 \cdot 10^{8}$
14	$6 \cdot 10^{-11}$	$5 \cdot 10^{-14}$	$5 \cdot 10^{-14}$	0.40	$1 \cdot 10^{9}$
15	$2 \cdot 10^{-10}$	$3 \cdot 10^{-1} - 3 \cdot 10^{2}$	$7 \cdot 10^{0} - 6 \cdot 10^{3}$	0.20	$9 \cdot 10^{8}$
16	$1 \cdot 10^{-10}$	$2 \cdot 10^{-2} - 3 \cdot 10^{-2}$	$3 \cdot 10^{0} - 6 \cdot 10^{0}$	0.35	$5 \cdot 10^{8}$



(a) Assembly and solve parallel efficiency.



(b) Elapsed time for assembly and 10^{-5} reduction in the error.

Figure 9: Test case III, a realistic basin model

nodes to interior nodes, which increases relative communication overhead as well as local overhead.

The results for weak scaling can be interpreted in two different ways. On one hand, the parallel scalability for a fixed number of iterations is very good, and should easily scale into thousands of processors (limited mainly by the per-processor problem size, as alluded to in the strong-scaling case). On the other hand, the preconditioner is not optimal, in that its performance degrades with problem size (see fig. 5b). This degradation is rather small, but it still overwhelms the parallel overhead, and thus the weak scalability (particularly in 2D) is less good when using an error criterion for convergence. Further research into improving the size-dependence of the preconditioner may be warranted.

As our main result, we demonstrate that good parallel scaling is achievable on a complex problem in coupled geomechanics, using a standard iterative solver and state-of-the-art general single-block preconditioners, combined in a novel fashion.

Acknowledgements

The authors would like to thank professor Xing Cai at Simula, who implemented the initial parallel simulator, and who has been very forthcoming with ideas and assistance for using and extending the Diffpack parallel framework. We also thank Statoil ASA for their support, both financial and in access to their geological models and expertise. We are grateful to the Norwegian Metacenter for Computational Science (NOTUR) for allowing us the use of the *hexagon* computational cluster. This work is supported by a Center of Excellence grant from the Norwegian Research Council to Center for Biomedical Computing at Simula Research Laboratory.

Bibliography

[1] M. Adams. Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics. *International Journal of Numerical Methods in Engineering*, 55:519–534, 2002. doi: 10.1002/nme.506.

[2] M. F. Adams, H. H. Bayraktar, T. M. Keaveny, and P. Papadopoulos. Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. In *Proceedings of the ACM/IEEE Conference on Supercomputing (SC2004)*, page 34. IEEE Computer Society, 2004.

[3] bigblue. The Simula computer cluster *bigblue*. URL `http://simula.no/research/sc/cbc/events/2008/081105-slides/bigblue-intro.pdf`.

[4] M. A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12(2):155–164, 1941. doi: 10.1063/1.1712886.

[5] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation (αSA). *SIAM Journal on Scientific Computing*, 25 (6):1896–1920, 2004. doi: 10.1137/S1064827502418598.

[6] U. V. Catalyurek, E. G. Boman, K. D. Devine, D. Bozdag, R. T. Heaphy, and L. A. Riesen. Hypergraph-based dynamic load balancing for adaptive scientific computations. In *Proc. of 21st International Parallel and Distributed Processing Symposium (IPDPS'07)*. IEEE, 2007.

[7] E. Chow, R. D. Falgout, J. J. Hu, R. Tuminaro, and U. M. Yang. A survey of parallelization techniques for multigrid solvers. In M. A. Heroux, P. Raghavan, and H. D. Simon, editors, *Parallel Processing for Scientific Computing*, pages 179–202. SIAM, 2006.

[8] Diffpack. URL `http://www.diffpack.com/`. Library for numerical solution of PDEs from inuTech GmbH.

[9] S. Doi and T. Washio. Ordering strategies and related techniques to overcome the trade-off between parallelism and convergence in incomplete factorizations. *Parallel Computing*, 25:1995–2014, 1999. doi: 10.1016/S0167-8191(99)00064-2.

[10] H. C. Elman, V. E. Howle, J. N. Shadid, and R. S. Tuminaro. A parallel block multi-level preconditioner for the 3D incompressible Navier-Stokes equations. *Journal of Computational Physics*, 187(2):504–523, 2003. doi: 10.1016/S0021-9991(03)00121-9.

[11] H. C. Elman, V. E. Howle, J. N. Shadid, R. Shuttleworth, and R. S. Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 227(3): 1790–1808, 2007. doi: 10.1016/j.jcp.2007.09.026.

[12] R. Fletcher. Conjugate gradient methods for indefinite systems. In G. Watson, editor, *Numerical Analysis*, volume 506 of *Lecture Notes in Mathematics*, pages 73–89. Springer, 1976. doi: 10.1007/BFb0080116.

[13] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006. URL `http://software.sandia.gov/trilinos/packages/ml/`.

[14] A. George and E. Ng. On the complexity of sparse QR and LU factorization of finite-element matrices. *SIAM Journal on Scientific and Statistical Computing*, 9: 849, 1988. doi: 10.1137/0909057.

[15] W. Hackbush. *Iterative Solution of Large Sparse Systems of Equations*. Springer-Verlag, 1995.

[16] J. B. Haga, H. P. Langtangen, B. F. Nielsen, and H. Osnes. On the performance of an algebraic multigrid preconditioner for the pressure equation with highly discontinuous media. In B. Skallerud and H. I. Andersson, editors, *Proceedings of MekIT'09*, pages 191–204. NTNU, Tapir, 2009. ISBN 978-82-519-2421-4. URL `http://simula.no/research/sc/publications/Simula.SC.568`.

[17] J. B. Haga, H. Osnes, and H. P. Langtangen. Efficient block preconditioners for the coupled equations of pressure and deformation in highly discontinuous media. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2010. URL `http://simula.no/research/sc/publications/Simula.SC.660`. Accepted for publication.

[18] J. B. Haga, H. P. Langtangen, and H. Osnes. On the causes of pressure oscillations in low-permeable and low-compressible porous media. Submitted to *International Journal for Numerical and Analytical Methods in Geomechanics*, 2011. URL `http://simula.no/publications/Simula.simula.18`.

[19] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Transactions on Mathematical Software*, 31(3):397–423, 2005. ISSN 0098-3500. doi: 10.1145/1089014.1089021.

[20] hexagon. The NOTUR computer cluster *hexagon*. URL `http://www.notur.no/hardware/hexagon`.

[21] W. Joubert and J. Cullum. Scalable algebraic multigrid on 3500 processors. *Electronic Transactions on Numerical Analysis*, 23:105–128, 2006.

[22] G. Karypis, K. Schloegel, and V. Kumar. ParMETIS parallel graph partitioning and sparse matrix ordering library, version 3.1. *University of Minnesota, Minneapolis*, 2003. URL `http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview`.

[23] H. P. Langtangen. *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming*. Springer, 2nd edition, 2003.

[24] PetroMod. URL `http://www.petromod.com/`. Petroleum systems modelling software from Schlumberger Aachen Technology Center.

[25] K. K. Phoon, K. C. Toh, S. H. Chan, and F. H. Lee. An efficient diagonal preconditioner for finite element solution of Biot's consolidation equations. *International Journal of Numerical Methods in Engineering*, 55:377–400, 2002. doi: 10.1002/nme.500.

[26] R. Thakur and W. Gropp. Improving the performance of collective operations in MPICH. *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pages 257–267, 2003.

[27] K. C. Toh, K. K. Phoon, and S. H. Chan. Block preconditioners for symmetric indefinite linear systems. *International Journal of Numerical Methods in Engineering*, 60:1361–1381, 2004. doi: 10.1002/nme.982.

[28] R. S. Tuminaro and C. Tong. Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines. In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2000. doi: 10.1109/SC.2000.10008.

[29] U. M. Yang. Parallel algebraic multigrid methods—high performance preconditioners. In A. M. Bruaset and A. Tveito, editors, *Numerical Solution of Partial Differential Equations on Parallel Computers*, pages 209–236. Springer, 2006. doi: 10.1007/3-540-31619-1_6.