Towards a robust Terra

Dissertation

zur Erlangung des akademischen Grades doctor rerum naturalium
(Dr. rer. nat.)

vorgelegt dem Rat der Chemisch-Geowissenschaftlichen Fakultät der
Friedrich-Schiller-Universität Jena
von Markus Müller[1]
geboren am 16.10.1974 in Sondershausen

---

[1] Staatsexamen für das Lehramt an Gymnasien in Mathematik und Physik

Gutachter:

1. …
2. …
3. …

Tag der öffentlichen Verteidigung …

Abstract

In this work mantle convection simulation with Terra is investigated from a numerical point of view, theoretical analysis as well as practical tests are performed. The stability criteria for the numerical formulation of the physical model will be made clear.
For the incompressible case and the Terra specific treatment of the anelastic approximation, two inf-sup stable grid modifications are presented, which are both compatible with hanging nodes.
For the $Q_{1_h} Q_{1_{2h}}$ element pair a simple numeric test is introduced to prove the stability for any given grid. For the $Q_{1_h} P_{1_{2h}}^{disc}$ element pair and 1-regular refinements with hangig nodes an existing general proof can be adopted.
The influence of the slip boundary condition is found to be destabilizing. For the incompressible case a cure can be adopted from the literature.
The necessary conditions for the expansion of the stability results to the anelastic approximation will be pointed out.
A numerical framework is developed in order to measure the effect of different numerical approaches to improve the handling of strongly varying viscosity.
The framework is applied to investigate how block smoothers with different block sizes, combination of different block smoothers, different prolongation schemes and semi coarsening influence the multigrid performance.
A regression-test framework for Terra will be briefly introduced.

Kurzzusammenfassung

In dieser Arbeit wird die Simulation der Mantelkonvektion mit dem Programm Terra aus numerisch mathematischer Perspektive betrachtet. Sowohl theoretische Analysen als auch praktische Tests werden durchgeführt. Mathematische Stabilitätskriterien für die numerische Formulierung des physikalischen Modells werden herausgearbeitet. Für den inkompressiblen Fall und die Terra spezifische Behandlung der anelastischen Nährung werden zwei Gittermodifikationen vorgestellt, die beide mit hängenden Knoten kompatibel sind. Es wird ein einfacher, numerischer Test , mit Hilfe dessen die Stabilität des $Q_{1_h}Q_{1_{2h}}$ Elementpaares für jedes gegebene Gitter nachgewiesen werden kann, entwickelt. Eine Anpassung des allgemeinen Beweises für das $Q_2P_1^{disc}$ Paar auf die spezifische Situation in Terra wird vorgestellt und an das $Q_{1h}P_{1_{2h}}^{disc}$ Paar angepasst. Damit können 1-reguläre Verfeinerungen mit hängenden Knoten vorgenommen werden. Der Einfluss der “free slip” Bedingung, bzw. Tangentialspannungsfreiheit am Rand wird deutlich gemacht und eine in der Literatur beschriebene Möglichkeit zur Stabilisierung kurz besprochen. Für eine Ausdehnung der Stabilitätsergebnisse auf den allgemeineren Fall der anelastischen Näherung werden die nötigen Schritte erläutert.
Eine Testumgebung zur quantitativen Bewertung verschiedener numerischer Lösungsansätze für das Problem räumlich stark veränderlicher Viskosität wird vorgestellt. Damit wird der Einfluss verschiedener Blockglätter mit verschiedenen Blockgrößen, der Kombination verschiedener Blockglätter , verschiedener Prolongationsverfahren und Coarsening Strategien auf die Leistung des Multigridverfahrens untersucht.
Eine Regressionstestumgebung für Terra wird kurz beschrieben.

Danksagung

Ich danke meinen Kollegen für das fröhliche, ehrliche und kameradschaftliche Klima sowie ihre vielen (durchschaubaren) Versuche, mich durch vorsätzliche Überschätzung aufzumuntern.
Ich danke allen, die sich Zeit genommen haben, mir etwas zu erklären, dem Lutherhaus-EDV-Team, durch das ich schon ein bisschen Programmieren gelernt habe und allen, denen ich mit meinen “kleinen mathematischen Fragen” zu Leibe gerückt bin.
Ich danke allen, die mir geholfen haben, indem sie meine Aufgaben übernommen haben, geduldig und nachsichtig waren oder für mich gebetet haben. Das gilt für meine Gemeinde, meine Freunde, meine Kollegen und in besonderem Maße für meine Frau, meine Mutter und meine Schwiegereltern.
Für die Unterstützung bei der Anfertigung des Manuskriptes danke ich Prof. Walzer und John Baumgardner.

Singt und spielt dem Herrn in eurem Herzen und sagt allezeit für alles dem Gott und Vater Dank im Namen unseres Herrn Jesus Christus!
Epheser 5,20

Contents

1 Motivation **11**
- 1.1 Terra 11
- 1.2 Governing equations 12
  - 1.2.1 Conservation of mass 12
  - 1.2.2 Conservation of momentum 13
  - 1.2.3 Conservation of energy 19
  - 1.2.4 Velocity boundary conditions 22
  - 1.2.5 Summary 23
- 1.3 General overview about numerical methods 25
  - 1.3.1 Pressure and velocity discretization 25
  - 1.3.2 Time discretization 26
  - 1.3.3 Solution strategies 27
  - 1.3.4 Algebraic multigrid 35
  - 1.3.5 Terra's algorithm 38

2 Stability of the Discrete Problem **43**
- 2.1 Overview 43
- 2.2 Weak formulation 45
  - 2.2.1 Formulation 45
  - 2.2.2 Conditions for existence and uniqueness of the weak solution 47
  - 2.2.3 Differences to standard saddle point problems 50
- 2.3 (De)motivating 2D examples 51
  - 2.3.1 The example continuous problem 51
  - 2.3.2 Solutions of the discrete version 52
  - 2.3.3 Locking 54
  - 2.3.4 Short summary 55
- 2.4 Checking the LBB 57
  - 2.4.1 Terra's grid 57
  - 2.4.2 A new grid 61
  - 2.4.3 Construction of the Fortin operator 61
  - 2.4.4 Macro elements 62
  - 2.4.5 Extension to the mapped macro 70
  - 2.4.6 From the Fortin operator to the LBB 73

2.4.7 Direct proof of the LBB for an alternative grid 74
2.5 Discussion 77
2.5.1 Stability for the general problem 77
2.5.2 Standard Stokes stability 77

3 Multigrid-test framework 81
3.1 Motivation 81
3.2 Test setup 83
3.2.1 Numerics and implementation 83
3.2.2 Viscosity test fields 84
3.2.3 Blocking schemes 91
3.2.4 Multigrid 95
3.2.5 Refined expectations 95
3.3 Results 96
3.3.1 Mode of presentation 96
3.3.2 Comparison of convergence rates 101
3.3.3 Semi coarsening 117
3.3.4 Numerical cost 118
3.4 Conclusions and outlook 120
3.4.1 Discussion of the proposed numerical improvements . . . 120
3.4.2 Necessary enhancement of the framework 122
3.4.3 Implementational aspects 123

4 Implementation in Terra 125
4.1 Motivation 125
4.1.1 An example 125
4.1.2 Difficulties in the implementation 127
4.2 Test framework 128
4.2.1 Features 128
4.2.2 Development 129
4.3 Outlook 129
4.3.1 Further reasons for a test framework 129
4.3.2 Necessary extensions 129

5 Summary and outlook 131
5.1 Summary 131
5.1.1 Connection of the physical model to numerical stability criteria131
5.1.2 Multigrid-test framework 132
5.1.3 Regression-test framework 132
5.2 Outlook 132
5.2.1 Generalization of stability results 132
5.2.2 Unit tests 133
5.2.3 Physical benchmarks 133
5.2.4 Numerical improvements 133

5.2.5 Enhanced physical models 133

A **135**
A.1 Fréchet derivative of A 136
A.2 Symmetry of $a'(.,.)$ 137
A.3 Fréchet derivative of D 138

B **139**
B.1 Selbständigkeitserklärung 139
B.2 Curriculum vitae 140
B.3 Poster und Vorträge 142

Chapter 1

Motivation

In this thesis the primary challenge in view is how to treat the issue of strong spatial variations in viscosity in numerical models of planetary mantle convection in a robust and efficient manner. In this beginning chapter I describe the equations that must be solved in these models, provide an overview of the numerical methods available for solving these equations in a discrete manner, and survey some general constraints for these methods. I also discuss possible improvements in view of the given timeframe for this work and its starting point.

1.1 Terra

Terra is a finite element based computer code for the numerical modeling of mantle convection. It was initially developed by John Baumgardner in 1983 [6], being the first 3D convection model for the spherical-shell geometry. Due to its outstanding numerical performance, which was unrivaled at the time, it has attracted many authors who subsequently enhanced it in many ways. Some components of the solver are even used in up-to-date weather-forecast simulations and in the oceanographic community. Some major numerical steps towards a more realistic simulation of planetary mantles are the parallelization by Bunge and Baumgardner [14], and the incorporation of an algorithm capable to handle variable viscosity by Yang and Baumgardner [34, 72]. There are, however, countless other improvements, like the integration of chemistry, by means of markers or various physical enhancements like specific equations of state, phase boundaries and sophisticated viscosity profiles [67]. It has been used for so many publications, that it is quite impossible to cite them all. Terra is also the basic numerical tool of our group. This and the complex physical frame work around it have determined it as the starting point of this work.

As we will see, the strong connection to this particular code sharply distinguishes this thesis from an approach starting without any compatibility constraints. If general theory is headed at all, it almost always is the starting point. The numerical implementation is a subsequent step which is adapted to the theory.

Instead we will have to go the other way round, start with the code and try to apply general theory to it. This will make the theoretical part of the work much harder, since only few theoretical results match the needs imposed by the given code. However, the opposite approach, to start from scratch (or theory), would hardly have lead to a code equally elaborate as Terra in the geophysical sense. Thus, although no geophysical application is presented, this work is application oriented.

1.2 Governing equations

The primary equations that govern the dynamics of planetary mantle convection are the conservation equations of mass, momentum and energy. I will completely derive these equations. However tedious this procedure may seem at some points, it nevertheless is essential to provide a global picture of the many assumptions that are usually applied and are appropriate to deformation of silicate rock in a planetary mantle. My objective is not only to clarify what actually can be computed, but also what definitely *cannot* be. As we will see in the sequel, the difficulty of the numerical solution, as well as the physical realism of the numerical model, is highly sensitive to certain model parameters. Because the central numerical issues I address in this thesis involve the simultaneous solution of the conservation equations of mass and momentum, I pay special attention to these equations.

One of the important assumptions, for example, is that the mantle rock can be accurately approximated as viscous *fluid* [1] , since it exhibits plastic deformation by rearranging and adjusting lattice defects, within and on the boundaries of mineral grains. Therefore our point of departure will be continuum mechanics.

1.2.1 Conservation of mass

We shall view the fluid in the Eulerian way, that is, we describe its motion to a fixed coordinate frame that does not move with the fluid. Consider a fluid of density ρ moving in a three-dimensional domain Ω. Suppose we observe a particular small volume or particle at the position $\mathbf{x}$ at time t. At the time $t + \delta t$ the particle is found at the position $\delta\mathbf{x}$. From our point of view, the velocity is then defined as:

$$\mathbf{u} = (u_x, u_y, u_z) =: \lim_{\delta t \to 0} \frac{\delta \mathbf{x}}{\delta t}$$

Next we consider a closed surface ∂D enclosing a volume D where the position of D is fixed and hence D does not move in time. The total mass of fluid inside any such volume is given by $\int_D \rho \, d\Omega$ where $d\Omega$ is the increment of volume. The

---

[1] This "fluid", however , is characterized by an infinite Prandtl number. This follows from the fact that the magnitude of the term arising from the deformation of the rock material is so many orders of magnitude larger than all the inertial terms in the momentum conservation equation, as we will see later.

amount of fluid flowing out of D across ∂D per time is

$$\int\limits_{\partial D} \rho \mathbf{u} \cdot \mathbf{n} \, dS$$

where $\mathbf{n}$ is the unit normal vector to ∂D pointing outwards from D and dS is the increment of surface area. Conservation of mass is expressed by the fact that any change of mass inside D can only be achieved by moving fluid through the surface ∂D. Expressed in terms of the rate of change of mass inside D this means:

$$\frac{d}{dt} \int\limits_{D} \rho \, d\Omega = - \int\limits_{\partial D} \rho \mathbf{u} \cdot \mathbf{n} \, dS \tag{1.1}$$

For further transformation we use the divergence theorem of Gauß that holds for smooth enough vector fields $\mathbf{v}$ and any region R with smooth enough boundary ∂R and says:

$$\int\limits_{\partial R} \mathbf{v} \cdot \mathbf{n} \, dS = \int\limits_{R} \nabla \cdot \mathbf{v} \, d\Omega$$

Application to (1.1) yields:

$$\int\limits_{D} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \, d\Omega = 0$$

Since D can be chosen arbitrarily this is equivalent to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \, d\Omega = 0 \tag{1.2}$$

This is the general form of the conservation of mass. In mantle convection simulations for instance in [54] or in our code the density is replaced by a reference density $\rho_0(r)$. This approximation is obvious since the depth dependence of the pressure is assuredly the most important influence on the density.

1.2.2 Conservation of momentum

To be able to apply Newton's Second Law of Motion we have to compute the time derivative of the momentum of the fluid. Imagine a small dyed volume of fluid. Suppose that its velocity is $\mathbf{u}$ at time t and $\mathbf{u} + \delta \mathbf{u}$ at time $t + \delta t$, respectively. For the volume D the rate of change of momentum is given by:

$$\frac{d}{dt} \int\limits_{D(t)} \rho \mathbf{u} \, d\Omega$$

Note that now the boundary of the small test volume is time dependent $D = D(t)$. Using a transformation to Lagrangian coordinates, Euler's transformation of the Jacobi determinant, and (1.2), one finds the following useful identity.

$$\frac{d}{dt} \int\limits_{D(t)} \rho A \, d\Omega = \int\limits_{D(t)} \rho \frac{dA}{dt} \, d\Omega \tag{1.3}$$

Here A can be a scalar, vector or tensor. Since no additional physical assumptions are necessary for this computation I only state the result. The proof is found in many continuum mechanics textbooks. Applying this result, our integral can be simplified to:

$$\frac{d}{dt} \int\limits_{D(t)} \rho \mathbf{u} \, d\Omega = \int\limits_{D} \rho \frac{\mathbf{du}}{dt} \, d\Omega$$

Remembering that $\mathbf{u}$ is a function of both position and time we get:

$$\frac{\mathbf{du}}{dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}$$

and hence

$$\frac{d}{dt} \int\limits_{D(t)} \rho \mathbf{u} \, d\Omega = \int\limits_{D(t)} \rho \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \, d\Omega$$

According to Newton's Second Law of Motion the rate of change of momentum is equal to the forces acting on this volume or its surface. These include

1. external body forces acting on every particle in the volume, in our case only gravity

   $$\int\limits_{D} \rho \mathbf{f} \, d\Omega = \int\limits_{D} \rho \mathbf{g} \, d\Omega$$

2. forces due to the stress at the surface.

   $$\int\limits_{\partial D} \mathbf{s} \, dS = \int\limits_{\partial D} \sigma \cdot \mathbf{n} \, dS$$

   where σ is the stress tensor.

Thus our balance equation has the general form

$$\int\limits_{D} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) \, d\Omega = \int\limits_{D} \rho \mathbf{g} \, d\Omega + \int\limits_{\partial D} \sigma \cdot \mathbf{n} \, dS \tag{1.4}$$

There are several possibilities for the sources of stress. It could be for instance induced by an elastic deformation of the continuum. This possibility is, however, neglected because we are not interested in seismic waves. From a numerical point of view the treatment of those would cause time steps many orders of magnitude smaller than if they are neglected, increasing the computation time appropriately. The remaining forces acting on the surface differ in the direction relative to the latter.

1. Forces parallel to $\mathbf{n}$ induced by the pressure of the surrounding fluid
$$\int\limits_{\partial D} -p\mathbf{n}\, dS$$
and the resistance of the material against volume changes due to volume or bulk viscosity

2. Forces orthogonal to $\mathbf{n}$ due to shear stresses due to shear viscosity

To treat these forces properly we need to have an expression for the stress tensor in terms of pressure and velocity. Let us begin with a consideration of the viscous stress. Following Newton's postulate that for straight, parallel and uniform flow, the shear stress τ between layers is proportional to the velocity gradient $\frac{\partial u}{\partial y}$ in the direction perpendicular to the layers

$$\tau = \eta \frac{\partial u}{\partial y}$$

we try to find generalizations for the scalar η known as the dynamic viscosity and the term $\frac{\partial u}{\partial y}$. We observe that in the simple case of Newton's postulate $\frac{\partial u}{\partial y}$ measures the velocity difference of neighboring particles in the fluid relative to each other. Our first assumption therefore is that also in the general case stress is a function of this relative velocity difference. The task is to provide a more general measure, since the gradient of velocity turns out to be insufficient. We use tensor index notation for this paragraph.

Consider a point x_i and the dislocation field at this point $u_j(x_i)$. Consider further a neighbor $x_i + dx_i$. After the deformation x_i has moved to w_i and $x_i + dx_i$ to $w_i + dw_i$. The following holds:

$$\begin{aligned} w_i &= x_i + u_i \\ dw_i &= dx_i + \frac{\partial u_i}{\partial x_l} dx_l \\ &= \left(\delta_{il} + \frac{\partial u_i}{\partial x_l} \right) dx_l \end{aligned}$$

The tensor $\left(\delta_{il} + \frac{\partial u_i}{\partial x_l} \right)$ thus maps the neighborhood of x_i to the neighborhood of w_i. To get a measure for the actual deformation, apart from rigid translation or rotation, we look at the change of the distance between two neighboring points through the dislocation.

$$\begin{aligned}
\tilde{ds}^2 - ds^2 &= dw_i dw_i - dx_i dx_i \\
&= \left(\delta_{il} + \frac{\partial u_i}{\partial x_l}\right) dx_l \left(\delta_{ik} + \frac{\partial u_i}{\partial x_k}\right) dx_k - dx_i dx_i \\
&= 2\varepsilon_{lk} dx_l dx_k
\end{aligned}$$

with

$$\varepsilon_{lk} = \frac{1}{2}\left(\frac{\partial u_l}{\partial x_k} + \frac{\partial u_k}{\partial x_l} + \frac{\partial u_i}{\partial x_l}\frac{\partial u_i}{\partial x_k}\right)$$

Only in a linear theory, that is for $\frac{du_l}{dx_i} << 1$, the quadratic terms are neglected yielding

$$\varepsilon_{lk} = \frac{1}{2}\left(\frac{\partial u_l}{\partial x_k} + \frac{\partial u_k}{\partial x_l}\right)$$

This tensor describes the change of distance between two neighboring points. To arrive at the rate of change of the distance we have to compute the time derivative. The result is called the strain-rate tensor.

$$\begin{aligned}
\dot{\varepsilon_{lk}} &= \frac{1}{2}\frac{d}{dt}\left(\frac{\partial u_l}{\partial x_k} + \frac{\partial u_k}{\partial x_l}\right) \\
&= \frac{1}{2}\left(\frac{\partial v_l}{\partial x_k} + \frac{\partial v_k}{\partial x_l}\right)
\end{aligned}$$

Which leads to :

$$\frac{d}{dt}(\tilde{ds}^2 - ds^2) = \frac{1}{2}\left(\frac{\partial v_l}{\partial x_k} + \frac{\partial v_k}{\partial x_l}\right) dx_l dx_k$$

The mapping defined by $\dot{\varepsilon}$ describes not only shearing but also compression. To see this we note that the dilatations in the direction of the coordinate axes under the deformation are given by the diagonal entries of ε. The relative volume change is therefore given by:

$$\begin{aligned}
\theta &= \frac{d\tilde{V} - dV}{dV} \\
&= \frac{dw_1}{dx_1} + \frac{dw_2}{dx_2} + \frac{dw_3}{dx_3} - 1 \\
&= (1+\varepsilon_{11})(1+\varepsilon_{22})(1+\varepsilon_{33}) - 1 \\
&\approx \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33} \\
&= \varepsilon_{ii} \\
&= \nabla \cdot \mathbf{w}
\end{aligned}$$

Accordingly the rate of volume change is given by $\dot{\theta} = \nabla \cdot \mathbf{v}$. Now we decompose the strain-rate tensor into a volume preserving and a shape preserving part. To extract the volume preserving part of $\dot{\varepsilon}$, we have to substract a tensor with the same relative volume change, and thus the same trace, from it. The simplest tensor to serve this purpose is $\frac{1}{3}\delta_{lk}\dot{\varepsilon}_{ii}$ which is called the rate of expansion tensor. Thus we finally arrive at a promising candidate for the computation of the viscous part of the shear stress tensor. It is called the deviatoric strain rate or rate of shear tensor.

$$\dot{\hat{\varepsilon}}_{kl} = \varepsilon_{kl} - \frac{1}{3}\delta_{ik}\dot{\varepsilon}_{ii}$$

Thus the quantity $\dot{\hat{\varepsilon}}_{ik}$ can be considered as a generalization of the term ∇v in Newton's postulate for shear stress. Up to now we only assume that the shear stress is a function of $\dot{\hat{\varepsilon}}$. This generality is used e.g. in [55, 73, 52, 72, 1] and is in fact necessary to describe certain mass transport mechanisms in the mantle, which are supposed to be relevant for stress-softening mechanisms in the regime of high strain rates as they arise for instance at convergent plate boundaries. Examples are different types of thermally activated dislocation motion, which can be described by a power law relation. A tensor formulation for the relation of deviatoric stress and the rate of shear tensor is then given by

$$\tau_{lk} = \frac{1}{B^{\frac{1}{n}}} \exp\left(-\frac{E + pV}{nRT} \right) \dot{\varepsilon}_0^{\frac{n-1}{n}} \dot{\hat{\varepsilon}}_{lk}$$

where $\dot{\varepsilon}_0$ is the square root of the second invariant of $\dot{\hat{\varepsilon}}_{lk}$ and τ is the steady state strain rate and deviatoric stress [2] , E and V are the activation energy and the activation volume for the dominant type of dislocation creep, B is a coefficient depending on the length of the Burgers vector and temperature, etc. but is insensitive to grain size, n is the stress exponent varying between 1 and 6 but is close to 3 for many materials. See for instance [69, 47, 38]. This relation is clearly nonlinear. The generalization of η in Newton's postulate is an effective viscosity given by

$$\mu_{eff} \;=\; \frac{1}{2B^{\frac{1}{n}}} \exp\left(-\frac{E + pV}{nRT} \right) \dot{\varepsilon}_0^{\frac{1-n}{n}}$$

Remarks:
Note that in general also the rate of expansion tensor $\frac{1}{3}\delta_{ik}$ contributes to the viscous stress, even if we do not give an explicit measure here. In fact this contribution is neglected in our model since it is small in comparison to τ.
Note also that beside this power law behavior where the generalized, effective viscosity is at least a scalar it is even possible to consider an anisotropic dependency of stress and rate of shear tensor. This idea is induced by the fact, that the grid used for mantle-convection simulations is coarse, $h \approx 50\ km$, relative to the effective reach of the processes determining the material properties like viscosity. Imagine for instance a grid cell with $50\ km$ edge length. A higher resolution may reveal a

---

[2] The stress tensor also can be decomposed in a trace free part, that is supposed to cause volume invariant distortion, and a part that causes compression or expansion and therefore is a multiple of the unit tensor. The trace-free part is called deviatoric stress and denoted τ in the sequel.

low viscosity plane in the cell which would act as a slip plane. Then the reaction of the whole cell to stresses is clearly anisotropic. This possibility is, however, not implemented in Terra.
We now further constrain the model to a Newtonian fluid. For the latter the viscous stress σ_{visc} is a *linear* function of $\dot{\varepsilon}$. The simplest case is again that the fluid is isotropic. Then the viscous stress is given by:

$$\sigma_{visc} = 2\mu\dot{\hat{\varepsilon}} + \xi\delta_{lk}\varepsilon_{ii}$$

The shear viscosity μ is a scalar depending on pressure and temperature. And the quantity

$$\tau \;=\; 2\mu\dot{\hat{\varepsilon}}$$

or in components:

$$\tau_{lk} \;=\; 2\mu(p,T,\mathbf{x})\left(\frac{\partial u_l}{\partial x_k} + \frac{\partial u_k}{\partial x_l} - \frac{1}{3}\delta_{lk}\frac{\partial u_i}{\partial x_i}\right)$$

is the deviatoric stress or shear stress as previously.

The rate of expansion tensor $\frac{1}{3}\delta_{lk}\varepsilon_{ii}$ also influences the viscous stress tensor, but with another constant ξ, which is called bulk or volume viscosity. Absorbing the term $\frac{1}{3}$ in the constant we can write

$$\tau_{exp} = \xi\delta_{lk}\varepsilon_{ii}$$

In our models the effect of bulk viscosity is neglected. Accordingly we can express the stress tensor by:

$$\sigma_{lk} = -p\delta_{lk} + \tau_{lk}$$

Such a relation describes, for instance, the thermally activated diffusion of vacancies through grains (Nabarro-Herring creep) or grain boundaries (Coble creep) This is a sufficient model for the earth's mantle in the regime of low deviatoric stress, small grain size or both [38]. When one of the processes is dominant in material of a single species this yields Newtonian behavior of the fluid. The viscosity is then given by

$$\mu = \frac{k_B T d^2}{A\Omega}\exp\left(\frac{E + pV}{RT}\right)$$

where k_B, T, Ω, d are Boltzmann constant, absolute temperature, atomic or molecular volume and grain size. E and , V are activation energy and activation volume for the relevant diffusion process. p denotes the pressure.
Backed by e.g. on [37, 38], this model [3] has been successfully used for various

---

[3] To avoid misunderstanding, please note, that this short summary of different rheologies is by no means intended as a geophysical discussion, whether or not a certain rheology should be used, or

simulations with Terra. cf. [26, 66, 65, 64, 63, 62, 67].
We can now refine our momentum-balance equation (1.4).

$$\begin{aligned}
\int\limits_D \rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) d\Omega &= \int\limits_D \rho\mathbf{g}\, d\Omega + \int\limits_{\partial D} \sigma\cdot\mathbf{n}\, dS \\
&= \int\limits_D \rho\mathbf{g}\, d\Omega + \int\limits_{\partial D} \tau\cdot\mathbf{n} - p\mathbf{n}\, dS \\
&= \int\limits_D (\rho\mathbf{g} - \nabla p)\, d\Omega + \int\limits_{\partial D} \tau\cdot\mathbf{n}\, dS \\
&= \int\limits_D (\rho\mathbf{g} - \nabla p)\, d\Omega + \int\limits_{\partial D} 2\mu\dot{\hat{\varepsilon}}_{lk}\cdot\mathbf{n}\, dS \\
&= \int\limits_D \Big(\rho\mathbf{g} - \nabla p + \nabla\cdot(2\mu\dot{\hat{\varepsilon}})\Big)\, d\Omega
\end{aligned}$$

Viscosity of the earth's mantle is estimated to be in the range of 10^{18} to 10^{22} Pa s which results in Prandtl numbers, a measure of relative importance of viscous forces to inertial forces, of $10^{21} - 10^{25}$. Thus we can neglect the entire left hand side of the last equation and get.

$$0 = \int\limits_D \Big(\rho\mathbf{g} - \nabla p + \nabla\cdot(2\mu\dot{\hat{\varepsilon}})\Big)\, d\Omega$$

Since the part D of the fluid is arbitrary the integrand must vanish. The conservation of momentum is therefore expressed by

$$\nabla\cdot 2\mu\dot{\hat{\varepsilon}} - \nabla p + \rho\mathbf{g} = 0 \tag{1.5}$$

We turn to the conservation of energy.

1.2.3 Conservation of energy

The energy balance can be expressed in the following form,

$$\frac{d}{dt}E = P = P_W + P_{Qe}$$

---

which model represents the best approximation. This would be far beyond the scope of this work and my own expertise. It is rather clear, even from the few references, that different approaches are in use for different tasks, performed with numerical models of mantle convection. They have been presented in the order, induced by the process of refining the most general assumptions to concrete models appropriate to answer specific questions. The intention was to compile the physical facts, that will control, which numerical methods must be used to solve the arising discrete equations.

where E is the total energy of the system, P_W is the mechanical work per time and P_{Qe} is the heat transferred per time. The mechanical work can be split into a part originating from stress $\mathbf{s}$ at the surface and a part stemming from volume forces $\mathbf{f}$.

$$P_W = \int\limits_{\partial D(t)} \mathbf{s} \cdot \mathbf{v} + \int\limits_{D(t)} \rho \mathbf{f} \cdot \mathbf{v} \, d\Omega$$

Using the conservation of momentum and several transformations this can be shown to be equivalent to

$$P_W = \frac{d}{dt} \int\limits_{D(t)} \rho \left(\frac{\mathbf{v}^2}{2} + \sigma : \nabla \mathbf{v} \right) \, d\Omega$$

Assuming the symmetry of σ which is a consequence of the conservation of angular momentum, as long as no spin is assumed for the particles, [4] we can express the second term by means of the strain rate tensor and get:

$$P_W = \frac{d}{dt} \int\limits_{D(t)} \rho \left(\frac{\mathbf{v}^2}{2} + \sigma : \dot{\varepsilon} \right) \, d\Omega$$

The heat flux can be decomposed into heat conduction through the surface of D and heat produced inside D.

$$P_{Qe} = - \int\limits_{\partial D(t)} \mathbf{q} \cdot \mathbf{n} \, dS + \int\limits_{D(t)} Q \, d\Omega$$

where $\mathbf{q} = k \nabla T$ with the thermal conductivity k and the heat production rate per volume Q which in our case is radiogenic. The total energy is assumed to be the sum of kinetic energy and internal energy so that we get.

$$\begin{aligned} \frac{d}{dt} \int\limits_{D(t)} \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} + \rho \mathbf{u} \, d\Omega &= \frac{d}{dt} \int\limits_{D(t)} \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \, d\Omega + \int\limits_{D(t)} \sigma : \dot{\varepsilon} \, d\Omega \\ &- \int\limits_{\partial D(t)} \mathbf{q} \cdot \mathbf{n} \, dS + \int\limits_{D(t)} Q \, d\Omega \end{aligned}$$

where u is the specific internal energy (energy per mass). Remembering the fact that the last equation holds for any volume D and using the divergence theorem it can be transformed to:

$$\rho \frac{du}{dt} = Q + \nabla \cdot \mathbf{q} + \sigma : \dot{\dot{\varepsilon}} \tag{1.6}$$

---

[4] The assumption of microscopic spins leads to the Cosserat formulation of continuum mechanics.

As we did previously for the conservation of momentum we split the strain-rate tensor in its volume preserving part $\dot{\hat{\varepsilon}}$ and its shape preserving part. Accordingly we can decompose the stress tensor

$$\begin{aligned}\sigma_{ik} &= -p\delta_{ik} + \xi\dot{\varepsilon}_{jj}\delta_{ik} + 2\mu\dot{\hat{\varepsilon}}_{ik} \\ &\approx -p\delta_{ik} + 2\mu\dot{\hat{\varepsilon}}_{ik}\end{aligned}$$

If we again neglect the volume or bulk viscosity consistently with the procedure for the momentum balance, we can transform the last term of (1.6).

$$\sigma_{ik}\dot{\hat{\varepsilon}}_{ik} = -p\varepsilon_{jj} + 2\mu\dot{\hat{\varepsilon}}_{ik}\dot{\hat{\varepsilon}}_{ik}$$

or

$$\begin{aligned}\sigma : \dot{\hat{\varepsilon}} &= -p\nabla\cdot\mathbf{v} + 2\mu\dot{\hat{\varepsilon}} : \dot{\hat{\varepsilon}} \\ &= -p\nabla\cdot\mathbf{v} + \tau_{ik} : \dot{\hat{\varepsilon}}\end{aligned}$$

and can write (1.6) in the form:

$$\begin{aligned}\nabla\cdot\mathbf{q} + Q + \tau : \dot{\hat{\varepsilon}} &= \rho\frac{du}{dt} - p\nabla\cdot\mathbf{v} \\ &= \rho\frac{du}{dt} - p\frac{dV}{dt} \\ &= \rho\left(\frac{du}{dt} - p\frac{dv_s}{dt}\right) \\ &= \rho\frac{ds}{dt} \qquad (1.7)\end{aligned}$$

where $v_s = \frac{1}{\rho}$ is the specific volume (volume per mass). We used the second law of thermodynamics for the pVT system

$$du = T\,ds - P\,dv_s$$

This and all further transformations are taken from [68]. Classical thermodynamic gives us:

$$ds = \left(\frac{\partial s}{\partial T}\right)_v dT + \left(\frac{\partial s}{\partial v_S}\right)_T dv$$

and

$$\left(\frac{\partial s}{\partial T}\right)_v = \frac{c_v}{T}, \qquad \left(\frac{\partial s}{\partial v_S}\right)_T = \alpha K_T$$

This implies

$$Tds = c_v dT + \alpha K_T T d\left(v_S\right)$$

or

$$Tds = c_v dT - \frac{c_v\gamma T}{\rho}d\rho$$

where

$$\gamma_{th} = \frac{\alpha K_T}{c_v \rho} \tag{1.8}$$

stands for the thermodynamic Grüneisen parameter.

Inserting (1.8) into (1.7) we obtain

$$\rho c_v \frac{dT}{dt} - c_v \gamma T \frac{d\rho}{dt} = \tau_{ik} \frac{\partial v_i}{\partial x_k} + \frac{\partial}{\partial x_j} \left(k \frac{\partial}{\partial x_j} T \right) + Q$$

Using (1.2) we can substitute the total time derivatives. and get

$$\frac{\partial T}{\partial t} = -v_j \frac{\partial}{\partial x_j} T - \gamma T \frac{\partial v_j}{\partial x_j} + \frac{1}{\rho c_v} \left[\tau_{ik} \frac{\partial v_i}{\partial x_k} + \frac{\partial}{\partial x_j} \left(k \frac{\partial}{\partial x_j} T \right) + Q \right]$$

or

$$\frac{\partial T}{\partial t} = -\mathbf{v} \cdot \nabla T - \gamma T \nabla \cdot \mathbf{v} + \frac{1}{\rho c_v} \left(\tau : \dot{\varepsilon} + \nabla \cdot (k \nabla T) + Q \right)$$

Note that this becomes a convection diffusion equation in the case of an incompressible fluid since $\nabla \cdot \mathbf{v}$ vanishes.

1.2.4 Velocity boundary conditions

The boundary conditions for the velocity have an important impact on the numerical formulation.

We first note, that there is no noteworthy friction at the surfaces. At the upper boundary the viscous "fluid", the solid rock, is surrounded by air and at the lower boundary by melted material. The viscosity of both materials is many orders of magnitude lower than the viscosity of the mantle. This boundary condition is called "slip" or "*free* slip" and expressed mathematically by:

$$\sigma \mathbf{n} \cdot \mathbf{t_k} = 0 \text{ on } \partial\Omega \; 1 \leq k \leq d-1$$

Where σ is the stress tensor previously defined as $\sigma_{ik} = \tau_{ik} - \delta_{ik} p$ and the $\mathbf{t_k}$ form an orthogonal set of tangent vectors to the surface.

However, from a physical point of view also the two surfaces of the earth's mantle are *free* themselves. The nearly spherical geometry is not a constraint but a *consequence* of the conservation equations, with a dominating influence of gravity. To model the earth's mantle consistently with this generality would include an equal model of the inner and outer core, because the lower surface of the mantle is, as a free surface, of course dependent on the volume of the outer core, which is in turn dependent of thermal properties of the outer core. The same argument applies recursively to the inner core boundary. The position of the upper surface is in general also a function of the temperature and distribution of the different materials, the mantle consists of.

Numerical models would not only have to compute the flow inside given boundaries, but also the position of the boundaries, which are not perfectly spherical, due to the temperature differences. Although such models exist, an application to the whole mantle has not been considered as reasonable up to now. [5] Instead the boundaries are fixed, and convection in the earth's mantle is treated as an enclosed flow problem with.

$$\mathbf{n} \cdot \mathbf{v} = 0 \text{ on } \partial\Omega$$

This is backed up by precise estimations of the overall volume change of the mantle over time. However, for computations ranging over billions of years, some adjustment may be necessary, to avoid the accidental (numerical) construction of an either imploding or exploding bomb, depending on the heating.

1.2.5 Summary

---

[5] With respect to the currently available grid sizes of $50km$, mount Everest is negligible.

For convenience we state all derived conservation equations and boundary conditions together

$$\begin{aligned}
\nabla\cdot\tau-\nabla p+\rho\mathbf{g} &= 0 && (1.9)\\
\nabla\cdot(\rho_0\mathbf{v}) &= 0 && (1.10)\\
-\mathbf{v}\cdot\nabla T-\gamma T\nabla\cdot\mathbf{v}+\frac{1}{\rho c_v}\Big(\tau:\dot{\hat{\varepsilon}}+\nabla\cdot(k\nabla T)+Q\Big) &= \frac{\partial T}{\partial t} && (1.11)\\
\text{for } 1\le k\le d-1 \text{ on } \partial\Omega \quad \sigma\mathbf{n}\cdot\mathbf{t_k} &= 0 && (1.12)\\
\mathbf{n}\cdot\mathbf{v} &= 0 && (1.13)
\end{aligned}$$

with:

ρ = $\rho_0(r)\,(1-\alpha(T-T_0(r)))$

Q = ρH

τ_{lm} = $2\mu\dot{\hat{\varepsilon}}_{lm}$

with either

μ = $\frac{k_B T d^2}{A\Omega}\exp\left(\frac{E+pV}{RT}\right)$

or

μ = $\frac{1}{B^{1/n}}\exp\left(\frac{E+pV}{nRT}\right)\dot{\hat{\varepsilon}}_0^{(1-n)/n}$

$\dot{\hat{\varepsilon}}_{lm}$ = $\left(\dot{\varepsilon}_{lm}-\frac{1}{3}\delta_{lm}\dot{\varepsilon}_{kk}\right)$

= $\frac{1}{2}\left(\frac{\partial v_l}{\partial x_m}+\frac{\partial v_m}{\partial x_l}-\frac{1}{3}\delta_{lm}\frac{\partial v_k}{\partial x_k}\right)$

$\dot{\hat{\varepsilon}}_0$ = $\sqrt{\dot{\hat{\varepsilon}}:\dot{\hat{\varepsilon}}}$

α	coefficient of thermal expansion
ρ	density
$\rho_0(r)$	radial reference density
T	temperature
$T_0(r)$	radial reference temperature
$\mathbf{g}$	gravitational acceleration
$\mathbf{v}$	velocity
γ	thermodynamic Grüneisen parameter
c_v	specific heat at constant volume
k	thermal conductivity
H	specific radiogenic heat production
$\mathbf{t_k}$	orthogonal set of tangent vectors to the surface

Note that μ is strongly dependent on pressure and temperature in either case. It varies over several orders of magnitude.

1.3 General overview about numerical methods

This section provides a point of view that allows classification of the confusing variety of methods in use. Mantle convection has been treated in many different ways. Some early models were based on spectral methods, that try to approximate the solution by a series of spherical harmonics. The latter suffered from the inability to handle lateral variations of parameters. However, even before these codes emerged, Terra used finite elements to discretize the differential equations. [6] In contrast to the spectral methods, the solution is sought as a superposition of functions with small, *local* support. This made it possible to handle laterally varying parameters. Other 2D and 3D models, using other local discretization techniques like finite differences and finite volumes, followed. An up to date overview over the different approaches over time can be found in [54]. State of the art are, local, 3D models in spherical geometry. The very number of degrees of freedom in these models requires the use of multi-grid techniques on parallel computers. In this chapter I will group the different approaches, I am aware of. The distinctive feature will be the part of the algorithm where multi-grid is applied. Before we can do this we have to explain how the coupled system of (1.9) (1.10) and (1.11) can be solved at all, that is which linear systems arise from the discretization.

1.3.1 Pressure and velocity discretization

From the abstract point of view, we take here, the type of discretization can be chosen arbitrarily. As already mentioned, codes using finite differences, finite volumes or finite elements exist. We merely try to give a compact notation to be used in the sequel. After discretization (1.9) and (1.10) can be written in abstract block-matrix notation, where A is the matrix arising from the discretization of $\nabla \cdot \tau$, B^t is the matrix operating on the discrete pressure p_h approximating ∇p and C is the matrix operating on the discrete velocity $\mathbf{v_h}$ approximating $\nabla \cdot \mathbf{v}$. The vectors $\mathbf{v_h}$ and p_h contain the dof representing the discrete velocity and pressure fields. Since, in this section, only these vectors occur we drop the h and write $\mathbf{v}$ and p also for the discrete versions. For a compressible [7] fluid we get

$$\begin{pmatrix} A(T, p, \mathbf{v}, \mathbf{x}) & -B^t \\ C & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \tag{1.14}$$

The gradient operator is the adjoint of the divergence operator.

$$(B^t)^t = B$$

Hence we get for an incompressible fluid.

$$\begin{pmatrix} A(T, p, \mathbf{v}, \mathbf{x}) & -B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \tag{1.15}$$

---

[6] In this context discretization means the translation of the infinite dimensional problem to find the solution of the differential equations into a finite dimensional system of linear equations, that can be solved on a computer.

[7] in our restricted sense

Because the dynamic viscosity μ depends strongly on temperature and also on the gradient of velocity in the case of non-Newtonian rheology, the matrix A does as well. Whether or not an additional linear system has to be solved depends on the time discretization.

1.3.2 Time discretization

Suppose we have estimates for the fields $\mathbf{v}, p, T$. Then the representation (1.11) chosen for the conservation of energy corresponds to an initial value problem of n ordinary differential equations (ode) in time where n is the number of grid points. To obtain the solution of an ode or an ode system respectively one can of course proceed in different ways. But all those procedures fall in one of the following classes.

1. Explicit methods, that use only values already known at the present time to compute the value for the next time step.
2. Implicit or semi implicit methods that have to solve an implicitly stated equation to proceed to the next time step.

For our purpose to decide which linear system will arise, it is sufficient to use the simplest representative of each class. As an example for an explicit scheme we present the Euler forward method.

$$y_{n+1} = y_n + h\, y'(t_n, y_n)$$

Where h is the time step size and where we have used the ode $y' = y'(t, y(t))$ as a placeholder for our actual problem. Application to the system of (1.9) (1.10) and (1.11) yields the following procedure (1.16) with the right hand side of (1.11) playing the part of y'.

```
guess $\mathbf{v}_0, p_0, T_0$
do while $t < t_{end}$
    Choose a reasonable $h$ [a]
    Compute new $T_{n+1} = T_n + hT'(\mathbf{v}_n, \mathbf{x}, T_n, t_n)$.
    Compute new $\mathbf{v}_{n+1}(T_{n+1})$ and $p_{n+1}(T_{n+1})$
    by solving the (still coupled) system (1.14) or (1.15)
    $n = n + 1$
end
```

(1.16)

---

[a] If the solution of the nonlinear system is not shifted off to a pseudo time-marching procedure the reasonable time-step size $h = \Delta t$ is given by the CFL (Courant–Friedrichs–Lewy) condition (in one dimension given by $\frac{\mathbf{u}\Delta t}{\Delta x} < C$) see [45] or [46]

As an example for an implicit method we choose the Euler backward algorithm.

$$y_{n+1} = y_n + h\, y'(t_{n+1}, y_{n+1})$$

which is named after the backward finite difference approximation of the derivative:

$$y'(t) \approx \frac{y(t) - y(t-h)}{h}$$

that leads to the method. Application to our problem makes clear that in each time step now additionally an equation of the form

$$T_{n+1} = T_n h \left[-\mathbf{v} \cdot \nabla T_{n+1} - \gamma T_{n+1} \nabla \cdot \mathbf{v} + \frac{1}{\rho c_v} \Big(\tau : \dot{\hat{\varepsilon}} + \nabla \cdot (k \nabla T_{n+1}) + Q \Big) \right]$$

must be solved for T_{n+1}. This is also true if a stationary solution of (1.11) is sought. In block-matrix form we have to solve for every time step:

$$\begin{pmatrix} A(T,p,\mathbf{v},\mathbf{x}) & -B^t & E \\ C(\mathbf{x}) & 0 & 0 \\ D(T,p,\mathbf{v},\mathbf{x}) & 0 & S(\mathbf{v},\mathbf{x},t) \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \mathbf{f_M} \\ f_p \\ f_E \end{pmatrix} \tag{1.17}$$

Where D describes the viscosity-dependent dissipation that depends on $T, p, \mathbf{x}$ and in the case of a non-Newtonian rheology also on $\mathbf{v}$ since μ does. $S(\mathbf{v}, \mathbf{x}, t)$ is a combination of a convection-diffusion operator that naturally depends on $\mathbf{v}$, and a time- and, due to the inhomogeneity of the mantle, space-dependent heating $Q = Q(t, \mathbf{x})$. Since $C\mathbf{v} = \nabla \cdot \rho_0 \mathbf{v}$ it is at least space-dependent. If the full density was used, an additional pressure dependence would occur due to the equation of state (eos). If the pressure dependency of μ is also restricted to a depth dependency, A and D do not depend on p. E stands for the temperature dependent buoyancy that was formally included in the right hand side of (1.14). and $f_p = 0$. It is clear that (1.17) is nonlinear in many respects. We now describe different solution strategies.

1.3.3 Solution strategies

I point out that some basic understanding of the multigrid technique is necessary to understand the following paragraphs. On the other hand there is no room in this thesis for a detailed description. Therefore the reader is referred to the numerously available literature concerned with this subject. See for instance the inevitable [29] or [13, 57] and the references therein. The aim of this subsection is to provide a path from the abstract problem to the actual numerical procedure and its subtle problems. We start at the abstract formulation (1.17) and therefore with the outermost part of the algorithm We then gradually dive into the subproblems arising inside this outer loop. As we will see, these subproblems again have their subproblems, one of them being the main subject of this thesis. The following presentation is meant to point out its vitality for the whole solution process. This will hopefully be clear at the end of the chapter, but not much earlier. In the process of unwrapping the central part of the algorithm we will use the furthest reaching possible applicability of multigrid methods as a guiding principle, since we expect great benefit from this. This is clear since the system (1.17) is huge in terms of the number of dof. It is therefore of vital importance to have an optimal solver, that means a solver with a linear dependency of the numerical cost and the number of dof. Multi-grid is such a solver and is often applicable for linear systems stemming from the discretization of partial differential equations (pde) which are often sparse with an regular sparsity pattern. We now enumerate the different strategies in the order induced by this criterion.

Multi grid for the whole nonlinear systems

For the mentioned reasons it is natural to attempt to use multigrid for the whole system (1.17), that is in the outermost loop of the solution algorithm. It is however not possible to apply it in a straight forward fashion here. This is due to the nonlinearities of (1.17) which are incompatible with the linear superposition of corrections in an ordinary multigrid procedure. Instead a full approximation storage multigrid must be used which differs from standard multigrid in that respect that it does not restrict and prolongates residual or corrections but always the complete right hand side and solution. (For linear problems both procedures yield the same results.) An example for the application of FAS multigrid to the problem of mantle convection can be found in [1]. Other applications to fluid dynamics can be found in [40] and various works of the same author. It should be pointed out that FAS is not a cure for all problems arising from the solution of system (1.17). This becomes apparent if one considers the choice of the smoother. This must be able to solve still nonlinear, coupled systems on every grid level. To do this one might consider the following procedure. Assume that we want to solve the nonlinear equation.

$$NL(x) = f$$

A nonlinear operator NL applied to a solution vector x can be described as a linear operator L that itself depends on the solution vector x, it is applied to.

$$L = L(x)$$

For every estimate x_n we can therefore define a linearized version $L_n = L(x_n)$. The defect correction algorithm can be described as follows.

```
guess a start estimate $x_0$; $n = 0$
do until $|def| < tol$
    $L_n = L(x_n)$                     #compute new linear operator
    $def_n = L_n x - f$                #compute new defect
    $c = (P(x_n))^{-1} def$            #compute correction
    $x_{n+1} = x_n - \omega c$         #add correction with damping factor
    n+=1;
end
                                                                  (1.18)
```

It is clear that if the defect-correction method converges it converges to the solution, because the defect vanishes at the solution. But we dont discuss here if it does converge at all. A crucial point is the choice of P (for preconditioner). If the Fréchet derivative of NL' of NL is available (not only existent but computable) then $NL'(x_n)$ is a very good choice and $\omega = 1$ yields the Newton method with its known quadratic convergence.

```
guess a start estimate $x_0$;
$n = 0$
do until $|def| < tol$
    $def_n = NL(x) - f$
    $x_{n+1} = x_n - (NL')^{-1} def_n$
    n+=1;
end
                                                                  (1.19)
```

If NL' is not available one could take L_n which is always possible but expected to yield a much slower convergence. This would lead to:

```
guess a start estimate $x_0$; $n = 0$
do until $|def| < tol$
    $L_n = L(x_n)$                     #compute new linear operator
    $def_n = L_n x - f$                #compute new defect
    $x_{n+1} = x_n - \omega L_n^{-1} def$   #add correction with damping factor
    n+=1;
end
                                                                  (1.20)
```

For (1.17) this would lead to:

$$P = \begin{pmatrix} A(T_n, p_n, \mathbf{v_n}, \mathbf{x}) & -B^t & E(T) \\ C(\mathbf{x}) & 0 & 0 \\ D(T_n, p_n, \mathbf{v_n}, \mathbf{x}) & 0 & S(\mathbf{v_n}, \mathbf{x}, t) \end{pmatrix} \tag{1.21}$$

Note that t has no index, expressing the fact that this is not a time-marching scheme but an iteration to obtain a solution for one time step. Note also that the defect correction algorithm is sufficiently general to allow other choices for P. Examples are discrete operators stemming from other discretizations or approximations for the Fréchet derivative. In the case of block-matrix operators like the one in (1.17) an improvement can already be achieved by replacing a single block. The obvious candidate for this is the momentum operator A, since it is nonlinear in itself and could be replaced by its Fréchet derivative F_A . This is also true for D. The iteration matrix P for the defect correction is then given by:

$$P = \begin{pmatrix} F_A(T_n, p_n, \mathbf{v_n}, \mathbf{x}) & -B^t & E(T) \\ C(\mathbf{x}) & 0 & 0 \\ F_D(T_n, p_n, \mathbf{v_n}, \mathbf{x}) & 0 & S(\mathbf{v_n}, \mathbf{x}, t) \end{pmatrix} \tag{1.22}$$

An example for the application of this procedure to the incompressible Navier-Stokes equations can be found in [49]. Also in our case of the momentum operator for non Newtonian rheology, and the dissipation term $\tau : \varepsilon$ the Fréchet derivative can be computed. The computation can be found in the appendix A.
Here $F(\mathbf{u})\delta\mathbf{u}$ means the derivative at position $\mathbf{u}$ (in the function space) applied to $\delta\mathbf{u}$ and the strain rate tensor $\dot{\varepsilon}$ is written as an operator applied to $\mathbf{u}$ It turns out that F_A is similar to A from a numerical point of view. Both are semidefinite and symmetric, as we will see and exploit later.
At this point we have answered the question of smoothing, raised in the last paragraph, to that point that we have shown how the nonlinear systems can be substituted by linearized ones, but we have still not made clear how a smoother for (1.17) works at all. The usual suspects for this position, the splitting based iterative solvers like Gauss Seidel or Jacobi, are out of the question. This can be seen at first glance on the main diagonal of (1.17) which is zero for the two lower parts of the system. The strategy pursued by [1] is to solve the systems, arising at every grid level, with the (generalized) SIMPLER method described in [44]. These methods are also called distributive iterations. Table (1.23) shows a short description for the linearized version which is taken from [1] . Note that the algorithm contains also a slightly changed defect correction iteration of the form (1.21) with a damping parameter $\omega = 1$. This is revealed by the occurrences of different iteration indices in the procedure.

```
guess a start estimate $\mathbf{v}_0, p_0$; $n = 0$
do until$|def| < tol$

    1.) calculate new temperature $T_{n+1}$
        where $\tilde{E}(\mathbf{v_n})$ is an upwind approximation of $E$ from the energy equation
        using $\mathbf{v_n}$ and the defect from the last iteration by solving
        $\tilde{E}(\mathbf{v}_n)T_{n+1} = f_T - (E(\mathbf{v}_n) - \tilde{E}(\mathbf{v}_n))$
        (this is equal to $T_{n+1} = T - \tilde{E}(\mathbf{v}_n)^{-1}def_n$)

    2.) update pressure an velocity
        calculate new pressure using continuity and momentum equation:
        $-C\tilde{A}^{-1}B^t = f_p - C(\mathbf{v} + \tilde{A}^{-1}(\mathbf{f_M} - ET_{n+1} - A\mathbf{v})$
        with $\tilde{A}$ diagonal of $A$
        (Note that for the standard defect correction the old $T_n$ would be used)

        calculate new velocity from momentum equation and $p_{n+1}$

        calculate pressure correction using $-C\tilde{A}^{-1}B^t\delta p = f_p - C\mathbf{v}$

        calculate and apply velocity correction using $\delta\mathbf{v} = -\tilde{A}^{-1}B^t\delta_p$

        n+=1;

end
                                                                        (1.23)
```

The difference to standard defect correction for the whole system is that the defect for $\mathbf{u}, p$ is computed after the temperature is already updated and not in a single step with it. From an abstract point of view one may look at it as a block Gauß-Seidel sweep instead of the single step block Jacobi for the blocks in (1.17).

Closing this paragraph, we point out that we have now provided (from the literature) one example for a smoother for the FAS Multi-grid, which turned out to be a rather complicated thing, consisting of a semi implicit distributive iteration nested in a defect correction. Although this is a working real life example, the complexity may be rather unnerving, when it comes to adaption of the procedure, say, to another discretization. Especially the finite volume discretization used in [1] turns out to be dependent on the geometrical properties of the grid. This is also true for the stabilization procedures for the convection diffusion equation (conservation of energy). Another possible source of trouble is the part of multigrid we have not yet spoken about, namely the transfer of the solution and right hand side from coarser to finer grids or the other way round as well as the construction of the coarse grid operators. Also the (successful) handling of the strongly varying viscosity in [1] by the cell centered multigrid [39] is strongly coupled to the finite volume discretization and therefore to the properties of the grid. Additionally [12] note that SIMPLE type methods are often poor smoothers (not poor solvers) and hence the application in a multigrid framework might be suboptimal. Concluding we find that the desired application of multigrid for the outermost part of the algorithm is possible but the difficulties arising from an application to a slightly different problem (for instance spherical geometry) and the realization of the typical multigrid efficiency might be severe. They might be overcome by splitting the problem in hopefully better behaved subproblems, for which we again try to apply the multigrid method. On our way to the center of the algorithm the next possibility to do so is obvious if one looks at the smoothing step on the finest grid. Inside the defect correction loop linearized versions of (1.17) have to be solved. Why not use multigrid methods to do so?

Defect correction iteration with multigrid for the linearized systems

This procedure is applied in [49] for the compressible Navier-Stokes equations with Boussinesq approximation. The system (1.17) is linearized using the Fréchet derivatives of the block operators, similar to (1.22). The smother used for the linearized system is a generalized block Vanka [58] type smoother. Sometimes this class of solvers is referred to as LMPSC method (Local Multilevel Pressure Schur Complement) [49]. We will not go into details here. The main point for us is that the (block) Vanka solves small subproblems consisting of all variables $\mathbf{v}, p$ and in the generalized form also T, attached to a small part of the grid, together (Local Schur Complement), like a domain decomposition method with very small domains. The consequence is that it is capable of smoothing the whole linear system (1.22) or (1.21) if the Fréchet derivative is not available. This means that even if we do not use multigrid for the whole nonlinear system the only iteration multigrid is embedded in is the defect correction. [8] However the resulting multigrid is still far from robust if strongly varying parameters (like the viscosity in our case) are considered. In [49] many different strategies like viscosity dependent intergrid operators, adaptive blocking and increased block-sizes are considered. The (partly impressive) results still reveal a difficulty that seems common to all multigrid procedures that treat different physical variables together: There sometimes occur contradictions between the prolongated (or restricted) values on the grid they are transfered to. Examples are the problems of pressure interpolation for strongly varying viscosity reported in [49] , [54] and similar applications referenced therein. To avoid this effects one can split (1.17) into subproblems.

---

[8] Remembering that our starting point was the smoothing step on the fine grid of an FAS, it is even imaginable to embed this defect correction in the FAS of the previous paragraph, that is to have a multigrid inside multigrid.

Distinct multi-grid procedures for the pressure-velocity system and temperature

Since we have already used [1] as an example let us look at it again and point out the differences to the presently discussed procedure. This makes sense since this procedure is implemented as an option. There are, however, countless other implementations. In fact it is the standard procedure not only for mantle convection simulations but also for the Stokes system. In the algorithm (1.23) two main steps are marked. We can use separate multigrid procedures for each of them, regardless if the outer procedure is multigrid or defect correction, or both. There exist specialized solvers for the convection diffusion equation, needed for the energy balance, as well as for the Stokes system representing the combination of conservation of momentum and mass. For the convection diffusion equation one would typically use a multigrid preconditioned Krylov subspace method like GMRES BICG ore more sophisticated variants. For the Stokes problem we again try to use multigrid for the whole system, that is, for $\mathbf{v}$ and p simultaneously. For the incompressible case [9] there are at least three possibilities to do so.

1. The above mentioned SIMPLE procedures
2. The above mentioned Vanka-type smoothers
3. The Braess-Sarazin smoother introduced in [12]

We look at these methods with strongly variable viscosity in mind. I will therefore point out the difficulties in order to compare the effort probably needed to get the methods to work for viscosity contrasts of many orders of magnitude. We start with the SIMPLE methods. In [54], which uses such a method, a new pressure transfer procedure is reported that improves the previously applied linear interpolation. The proposed procedure can be derived from quite different points of departure. It is based on a finite volume discretization and hence it is not clear how it can be generalized in order to accommodate the needs arising from other discretizations.
Vanka type smoothers have also been tested by the same author. An other application is found in [49]. This is a FEM (finite element) code. The most difficult problem here was also posed by the transfer operators for $\mathbf{v}$ and p. In case of really big viscosity contrasts convergence could only be achieved by positioning coarse grid edges along the jumps.

I am not yet aware of an application of a Braess-Sarazin smoother to problems with strongly variable viscosity, but this algorithm has some interesting properties. For instance the new iterate $\mathbf{u}_{i+1}, p_{i+1}$ is independent of the last pressure iterate p_i. This way the above mentioned problems of the pressure grid transfer would not

---

[9] The typical treatment of density is to use a depth dependency only, so we have very nearly an (incompressible) Stokes system. It is often possible to find some modification of an incompressible algorithm. We present examples later.

affect the fine grid solution. Another interesting feature is the use of an approximation for A^{-1} the momentum operator, which brings us to the innermost part of the algorithm, where multigrid can be applied.

Multigrid for the momentum operator

This part finally describes the algorithm implemented in Terra which is the subject of this thesis. The solution for $\mathbf{v}$ and p is still obtained simultaneously but by an iteration (CG, MINRES, GMRES ore Uzawa) with a block preconditioner where multigrid is used for the block stemming from the momentum equation. A recent comparison [41] of the two probably fastest of the above described multigrid solvers for both variables (Braess Sarazin, Vanka) and the last mentioned velocity-multigrid preconditioned iterative schemes, showed that the performance is worse by a factor two or three for the latter. This, at first glance, seems to be a reassurance for our guiding principle to use multigrid in the most outward loop of the solver. On the other hand this paper does not take into account the difficulties arising from (spatial) parameter variations. [10] However, at least from a theoretical point of view Krylov subspace methods are extremely robust in this respect, since their convergence can be proofed for a great variety of problems. (CG for instance is convergent for all symmetric positive definite systems.) Accordingly [41] state that from a theoretical point of view the state of affairs is best for the preconditioned MINRES method. Another very interesting and surprising property of preconditioned MINRES is, that it is optimal($O(n)$), see [33] page 286 ff for details. (Roughly speaking the reason is, that its convergence rate depends solely on the approximation of the inverse of the momentum operator A which is obtained via multigrid.) So we really have the assumed benefit of the Vanka and Braess Sarazin solvers also for pMINRES, but without the problems arising from the collective transfer of $\mathbf{v}$ and p. This is due to the fact that the still difficult transfer of $\mathbf{v}$ is solely caused by the viscosity variations. So algebraic anisotropies arising from this can be countered by exploiting the knowledge about the viscosity field. This is a much easier problem than for a combination of $\mathbf{v}$ and p where it is not always clear what variable the problems originate from.

Intermediate discussion

The last subsection contained two quite opposite trends.

1. Because of its optimality, it seems reasonable to apply multigrid as far outwardly as possible. (At first glance) the possible benefit seems to decrease if

---

[10] Although robustness is a main subject in this paper it is not meant as a measure of the capability to handle spatial variability of the viscosity μ but the ratio between μ and ξ in the generalized Stokes equation.

$$\begin{array}{rcl} \xi\mathbf{u} - \mu\Delta\mathbf{u} + \nabla p & = & f \\ \nabla \cdot (\rho_0 \mathbf{u}) & = & 0 \end{array}$$

where μ is free but fixed and *not a function of space.*

one limits the use of multigrid to subproblems.

2. The handling of variable viscosity becomes more difficult in the opposite direction as the use of multigrid is expanded to more variables at once. The problems arise from the transfer operators.

These observations give reason to two possible approaches. The first strategy is to use algebraic multigrid AMG. This is probably obvious from a numerical point of view and has been proposed to me as an option by Arnold Reusken and Irad Yavneh. The second, less ambitious, but finally adopted strategy will be described and justified at the end of the next subsection.

1.3.4 Algebraic multigrid

The optimal solution for the above stated dilemma seems to be algebraic multigrid AMG. Since it is not realized here I feel impelled to explain why. Before doing so I would give some more arguments for its use, since this is interesting for future work.

Arguments for AMG

There are some very simple but profound reasons:

1. From a numerical point of view an AMG is supposed to be able to handle strongly varying and even discontinuous parameters.

2. From a physical point of view regions with different viscosity would coincide with different vigor of the flow, so an accordingly refined mesh would be suitable. A grid generation and refinement software based on a posteriori error estimator could be applied. The resulting possibly unstructured grids could also be handled by AMG.

3. From a software (code reuse) point of view AMG is interesting because it does not need any further knowledge of the pde but solely depends on the operator (matrix) for the construction of the coarse grids and hence the above mentioned often troublesome transfer operators for prolongation and restriction. This makes it possible to use a single algorithm for the whole range of problems described in the previous subsection.

4. The broad possible applicability of AMG makes it worth while to develop fast black-box pde solvers, also parallel variants. Accordingly the parallelisation of algebraic multigrid has been a fast developing field of research for some time. See for instance [31, 28, 51, 17, 27, 3] Complete reusable solvers should be available.

Preliminary drawbacks

A question difficult to answer is why this method has not been widely used for mantle convection yet. At least I want to discuss *some* objections and finally explain what reasons stood against its usage in this thesis.

1. In general algebraic multigrid requires a time consuming aggregation step where the prolongators, restrictors and coarse-grid operators are defined. In [51] an estimate of ten multi-grid v-cycles is given for the AMG method of Ruge and Stüben. This is problematic if the fine-grid operators have to be computed very often since then this seriously affects the overall performance. Such operator updates arise in our problem due to any form of nonlinearity of the whole system of $\mathbf{v}, p$ and T induced for instance by the pressure dependence of the viscosity, the temperature dependence of the viscosity, where temperature is amongst others a function of time, the treatment of nonlinear rheology, where the stress tensor is connected to the strain rate for instance by a power law as suggested in [**?**]. In case of Newtonian rheology and temperature dependent viscosity the momentum operator must be updated only once per time step, in the case of non-linear rheology implicit methods like newton-iterations require the solution of several stokes systems with different viscosity fields for every *single* time-step. Regarding the, in terms of computational cost, still extremely demanding models of mantle convection the temporary repudiation even of methods with a small performance penalty is perhaps understandable. After all a factor of five makes a difference between a day and a week in terms of response time. For some models the latter actually is about five weeks for the fast version.

2. Some existing mantle convection codes, especially Terra, are extremely hardware optimized and fast in terms of relative performance compared to peak performance of the particular machine. [11] This is an advantage a numerically superior method has to make up for.

3. A potential benefit of AMG methods, the ability to handle unstructured grids with varying mesh sizes is hampered by the need of a spatially adaptive time discretization, since the CFL condition would enforce small time-steps in this case for the whole grid. This adds complexity also to other parts of the mantle convection codes (marker transport for instance).

4. The reuse of existent software for grid generation (and operator assembly) and efficient parallel solution of the linear systems is narrowed down by two criteria

   (a) Is the code parallelized for distributed memory computers?

   (b) Is the source open?

---

[11] The author of the original version, John Baumgardner, is in fact a HPC specialist

> The first is obvious if one considers an example problem size of $6.710.886.400$ dof, the second, away from the peculiarities of university funds, because of the flexibility needed for a research instrument. Since the code runs on different supercomputers is must be portable. Here also open source software is clearly an advantage. Today the query of the two criteria yields a few hits, namely UG [12] , ALUGrid [13] , Dune [14] and, in a less general way, petsC [15] The most exiting of these is the Dune framework since it is the most flexible approach and also includes a parallel AMG (algebraic multigrid) solver template. At the beginning of my work the only suitable package I knew of was UG. However, the technical difficulties of porting turned out to be greater than expected. The necessary amount of cooperation could not be achieved due to organizational difficulties, namely funding. This way the reuse of existing software, which would have been preferable, as deduced above, was, at least in my case, rendered preliminarily impossible.

Discussion

I am convinced that in the long run AGM methods will displace the highly optimized but less general codes. If the computer performance is sufficient for reasonable test problems even a pessimistic factor of ten is a small price for the ability to handle variations or jumps of coefficients, unstructured grids and therefore adaptive meshes. For a 3D time-dependent problem this is less than a factor of two in terms of the mesh-size parameter h.
All the same, for the time being, I am not aware of a single application in this field. Up to now all existent parallel tools for the investigation of mantle convection, are based on a domain partitioning of structured grids. Load balancing is achieved by simply using exactly identical domains.

Terra specific conclusions

Since a reuse of a complete solver had not been possible, other possibilities to increase Terra's robustness had to be explored. The terra code is extremely optimized, utilizing properties of the grid and the computers it runs on. This close relation between parallelization and grid extremely amplifies the importance of the latter. To alter the grid of such a code is nearly equivalent to write a new one. This also includes the type of discretization. If even the switch from one structured grid to another is an effort not much smaller than the development of a new code it is clear that something like adaptive mesh refinement is simply out of the question, if existent software cannot be used. This pretty much narrows down possible changes to the code and enforces the exploration of comparatively small but

---

[12] UG stands for unstructured grids see [5]

[13] Adaptive, Load-balanced, and Unstructured Grid Library see [25]

[14] Distributed and Unified Numerics Environment, see [21]

[15] Portable, Extensible Toolkit for Scientific Computation see [4]

efficient changes to the grid. Additionally the code should be refactored to define abstract interfaces. So later available solvers can be used for an increasing number of subproblems. As the first strategy could be described as an top down approach tackling the whole problem from outside this one can be seen as an agile one, changing the code from within. As we will see this has some benefits for the inevitably important testing. To be able to locate those small changes to the code we describe Terra's Algorithm a little bit more detailed.

1.3.5 Terra's algorithm

Time discretization

Terra uses an explicit time marching scheme that is similar to (1.16). The difference is that instead of the Euler-forward method a second order Runge-Kutta scheme is used. That means that in every time step a system of the form (1.14) must be solved. (or (1.15) alternatively for incompressible computations).

Treatment of the nonlinearity of the momentum operator

Up to now the nonlinear momentum operator is not taken into account by an explicit defect correction iteration. Instead the problem is shifted to the time marching. This is common practice and can be seen as an adjournment of the yet unresolved part of the problems arising from the nonlinearity to the next time step. In the limit for time-step size $h = 0$ the normal defect correction iterations ensue. For this kind of procedure h is adapted according to the error of the last iteration, so that for large nonlinearities (or even stiff momentum operators) the time steps become very small. Nevertheless the time line of the problem gets unreliable in the start up phase, so that an engaging phase should be used, before the physical time starts.

Solution of the velocity-pressure system

As already mentioned the system (1.15) is solved using a Krylov-subspace method. In the present version it is CG. Since the compressible version (1.14) is solved using the same method, we describe the procedure for this case and afterwards its generalization to the incompressible case. If one applies the Gaussian elimination blockwise on system (1.15) one gets [16]

$$\begin{pmatrix} A(T,p,\mathbf{v},\mathbf{x}) & -B^t \\ 0 & BA^{-1}B^t \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ -BA^{-1}\mathbf{f} \end{pmatrix} \tag{1.24}$$

The CG method is applied to the second equation to obtain a pressure solution. Obviously $\mathbf{v}$ can be found solving the first one. In fact this is not even necessary, since

---

[16] The same result is obtained if the (abstract) solution $\mathbf{v}$ of the first (momentum) equation is substituted in the second

the summed up $\mathbf{v}$ corrections, necessary to solve for p, equal the $\mathbf{v}$ solution. Of course nobody would like to compute the whole Schur complement $-BA^{-1}B^t$. Instead always when the action of A^{-1} on $\mathbf{v}$ is needed, it is performed by the multi-grid procedure. If one applies the CG algorithm abstractly to the Schur complement and eliminates all occurrences of A^{-1} in this way, the resulting algorithm looks like this.

```
Estimate $p_0$
Solve $A\mathbf{v}_0 - B^t p_0 = \mathbf{f}$
Evaluate residual, $r_0 = B\mathbf{v}_0$
do i=1,N
    if (i=1) then
        $s_1 = r_0$
    else
        $\delta = \frac{\langle r_{i-1}, r_{i-1}\rangle}{\langle r_{i-2}, r_{i-2}\rangle}$
        $s_i = r_{i-1} + \delta s_{i-1}$
    end if
    Solve $A\mathbf{v}_i = B^t s_i$ for $\mathbf{v}$
    $\alpha = \frac{\langle r_{i-1}, r_{i-1}\rangle}{\langle s_i, B\mathbf{v}_i\rangle}$
    $p_i = p_{i-1} + \alpha s_i$
    $\mathbf{v}_i = \mathbf{v}_{i-1} + \alpha \mathbf{v} s_i$
    $r_i = r_{i-1} + \alpha B\mathbf{v}_i$
    if ($||r_i|| < tol$ ) exit loop
end do
                                                        (1.25)
```

Which can be found in [72].

Treatment of the inelastic approximation

To be able to treat the inelastic approximation with a depth dependent density the following procedure has been employed. Suppose there exists a variable q connected with the pressure p and the radial reference density ρ_0 by

$$\rho_0 \nabla q = \nabla p$$

which is an estimate. [17] We now substitute ∇p in (1.15) and get:

$$\begin{pmatrix} A(T,p,\mathbf{v},\mathbf{x}) & -RB^t \\ BR & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \tag{1.26}$$

Again using blockwise Gaussian elimination we get the equivalent of (1.24)

$$\begin{pmatrix} A(T,p,\mathbf{v},\mathbf{x}) & -RB^t \\ 0 & BRA^{-1}RB^t \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ -BRA^{-1}\mathbf{f} \end{pmatrix} \tag{1.27}$$

Where R represents a diagonal matrix for the density. Using that $R = R^t$, a look at the new Schur complement makes clear that the ensuing matrix is a similarity transform of the original one and thus has the same eigenvalues and resulting definiteness. So if the use of CG is justified for the original system it is also for the transformed one. [18] We can further simplify the system using the fact that R is diagonal and get.

$$\begin{pmatrix} A(T,p,\mathbf{v},\mathbf{x}) & -RB^t \\ 0 & BA^{-1}B^t \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ -R^{-2}BRA^{-1}\mathbf{f} \end{pmatrix} \tag{1.28}$$

The ensuing algorithm is very similar to the compressible case, all occurrences of B^tp or $B \cdot \mathbf{v}$ are substituted by RB^tp or $R^{-2}BR\mathbf{v}$ respectively:

---

[17] A small calculation makes clear that this cannot be exactly fulfilled but *must* be an estimate

$$\begin{array}{ccccc} & \nabla \times \nabla p & = & \nabla \times (\rho_0(r)\nabla q) \\ \rightarrow & 0 & = & \nabla\rho_0 \times \nabla q + \rho_0 \nabla \times \nabla q \\ \rightarrow & 0 & = & \nabla\rho_0 \times \nabla q \\ \rightarrow & \nabla\rho_0 \parallel \nabla q & \rightarrow q = q(r) & \text{because } \rho_0 = \rho_0(r) \\ \rightarrow & p = p(r) & & \end{array}$$

Such a pressure would of course not be of any interest.

[18] I am aware of the fact that the Schur complement cannot be positive or negative definite but semidefinite because the pressure can be determined only up to a constant, since only its derivative appears in the equation. We will come to this later.

```
compute $\gamma = \frac{\rho_0(r)}{\rho_{ref}}$
guess $p_0$
Solve $A\mathbf{v}_0 - RB^t q_0 = \mathbf{f}$
Evaluate residual, $r_0 = \gamma^{-2} B\gamma\mathbf{v}_0$
do i=1,N
    if (i=1) then
        $s_1 = r_0$
    else
        $\delta = \frac{\langle r_{i-1}, r_{i-1}\rangle}{\langle r_{i-2}, r_{i-2}\rangle}$
        $s_i = r_{i-1} + \delta s_{i-1}$
    end if
    Solve $A\mathbf{v}_i = \gamma B^t s_i$ for $\mathbf{v}$
    $\alpha = \frac{\langle r_{i-1}, r_{i-1}\rangle}{\langle \gamma^{-2} s_i, B\gamma\mathbf{v}_i\rangle}$
    $p_i = p_{i-1} + \alpha s_i$
    $\mathbf{v}_i = \mathbf{v}_{i-1} + \alpha\mathbf{v}s_i$
    $r_i = r_{i-1} + \alpha\gamma^{-2} B\gamma\mathbf{v}_i$
    if ($||r_i|| < tol$ ) exit loop
end do
```

(1.29)

Chapter 2

Stability of the Discrete Problem

2.1 Overview

The chapter deals with the stability of the discretization for pressure and velocity by means of finite elements, that is with the process that results in the systems (1.15) or (1.14) respectively.

This is not at all trivial. In fact, I am not aware that the stability for the general discrete system (1.14) has been shown for the special free slip boundary condition which is used in the models of mantle convection for *any* finite element discretization up to now. Unfortunately this statement remains true even if we restrict ourselves to the linear version of (1.14), where the viscosity is no longer a function of $\mathbf{v}$ and even for constant viscosity. And regrettably, I have not yet been able to change this.

On the other hand we have carried things a good deal further and the achieved results at least lend some value to the special handling of variable density in Terra, which was the primary aim of the analysis. So I will try to present the state of the work and show the connection to the known theoretical results. Since the matter is very complex, this will be done in a way that is an attempted compromise between necessary shortness, accuracy and comprehensibility, which is bound to be disappointing for the expert as well as for the uninitiate.

A kind reader may regard it as a review for both sides, the physicist who cannot be expected to know the numerical background and the mathematician, who is interested in the theoretical challenges of a real world model.

The problem of establishing existence and uniqueness of system (1.14) is connected but not equivalent to the similar problem for the linear version of system (1.15) for a Dirichlet boundary condition. The stability of the latter mainly depends on the LBB (Ladyzhenskaya Babuška Brezzi or inf-sup condition). Although we can check the LBB for our discretization there remain additional problems.

1. The result must be extended to the Dirichlet boundary problem of system (1.14).

2. The FEM discretization for the Dirichlet problem must be stabilized in view of the free slip boundary condition, which otherwise again endangers that the system (1.14) is well posed.

Point (1) has been the subject of [10]. There the proof of existence and uniqueness of the solution of the stokes system has been extended to the case of a space dependent density and space independent kinematic viscosity $\nu = \frac{\mu}{\rho}$ for some standard finite element pairs, including the Taylor-Hood pair which is very similar to the elements used in Terra. One result of this work is that for the velocity-pressure formulation, that is used in our model, some additional inf-sup conditions have to be fulfilled. The proof of those depends, however, mainly on the existence of a Fortin operator for the finite element pair under consideration, which is also the key to check the LBB for our grid. So, if we succeed in the latter, we are in a good position to extend the result to space dependent density.

Another possibility to obtain the desired stability result is to use momentum and pressure as independent variables. Then the system can be formulated as a saddle point problem equivalent to the saddle point formulation of the linear version of system (1.15) and the same finite element pairs can be used, which ensure the stability for the latter. This reformulation, however, destroys some operator properties, e.g. the symmetry of the momentum operator and would enforce the use of different solution techniques in our code.

Point (2) is not clear up to now. In [59] the problem for the Stokes system (1.15) is solved by augmenting the velocity basis functions on the slip boundary by bubble functions to be able to take the normal stresses into account. The LBB is another necessary condition.

In [60] a more general strategy is pursued and a stabilization procedure by means of a penalty method is presented that can be implemented in existing codes more easily. In the case that the LBB holds, only the slip boundary term has to be stabilized, in the case that it does not, also the pressure can (and must) be stabilized. From this it is clear that the LBB alone is in general not sufficient to treat the boundary value problem (1.15) for the slip boundary, but very important. Both treatments rely on a saddle point formulation of the boundary value problem (1.15). The discretization of (1.14) , however, leads to a generalized saddle point form. It is therefore not automatically clear that the arguments of [59, 60] can be applied in this case.

Summarizing I emphasize the importance of the LBB as a key property of the discretization to ensure that our general problem is well posed. This chapter is therefore mainly concerned with it. In the sequel I will present the weak formulation of (1.14). The differences to the standard Stokes system will be made clear. After this we will give a simple example of a discretization that does not fulfill the LBB, to emphasize its importance from a physical point of view. A short discussion of the presently implemented grid will be given and finally we will present a procedure that allows us to test the LBB for a new grid. This can be regarded as something less than a general proof but will be shown to be sufficient for our needs. Additionally I will present a generalization of an existing proof for another grid.

2.2 Weak formulation

The following is neither a comprehensive introduction to the various possibilities to discretize pde's with finite elements which can be found for instance in [33] nor a detailed analysis as in [10]. It is intended to give an idea.

2.2.1 Formulation

Remember the equations for the conservation of momentum and mass (1.9),(1.10)

$$\begin{aligned} \nabla \cdot \tau(\mathbf{u}) - \nabla p + \rho \mathbf{g} &= 0 \\ \nabla \cdot (\rho \mathbf{u}) &= 0 \end{aligned}$$

with the boundary conditions

$$\begin{aligned} \mathbf{n} \cdot \mathbf{u} = 0 & \qquad \text{on } \partial\Omega \\ \sigma \mathbf{n} \cdot \mathbf{t_k} = 0 & \qquad \text{on } \partial\Omega \; 1 \leq k \leq d-1 \end{aligned}$$

Where σ is the stress tensor previously defined as $\sigma_{ik} = \tau_{ik} - \delta_{ik} p$ and the $\mathbf{t_k}$ form an orthogonal set of tangent vectors to the surface. Supposing we have already found a solution $(\mathbf{u}, p)$, then also the following is true for all test functions $\mathbf{v}$ and q inhabiting suitably chosen function spaces to be specified later.

$$\begin{aligned} \int\limits_{\Omega} \mathbf{v} \cdot [\nabla \cdot \tau(\mathbf{u}) - \nabla p + \rho \mathbf{g}] \; d\Omega &= 0 \\ \int\limits_{\Omega} q \nabla \cdot (\rho \mathbf{u}) \, d\Omega &= 0 \end{aligned}$$

The continuity requirements of the weak solution $\mathbf{u}, p$ can be reduced by "shifting" the derivative to the test functions using integration by parts and the divergence theorem. We use also the facts that we deal with an enclosed flow $\mathbf{n} \cdot \mathbf{v} = 0$ on $\partial\Omega$ or $\mathbf{v_n} = \mathbf{0}$ and the tangential stress vanishes $\sigma \mathbf{n} \cdot \mathbf{t_k} = 0$ for $k \in \{1, 2\}$

$$\begin{aligned}
\int_\Omega \mathbf{v}\cdot\nabla p\, d\Omega &= -\int_\Omega p\nabla\cdot\mathbf{v}\, d\Omega + \int_\Omega \nabla\cdot(p\mathbf{v})\, d\Omega \\
&= -\int_\Omega p\nabla\cdot\mathbf{v}\, d\Omega + \int_{\partial\Omega} p\underbrace{\mathbf{n}\cdot\mathbf{v}}_{=0}\, dS \\
&= -\int_\Omega p\nabla\cdot\mathbf{v}\, d\Omega \\
\int_\Omega \mathbf{v}\cdot\nabla\cdot\tau(\mathbf{u})\, d\Omega &= \int_\Omega \nabla\mathbf{v} : \tau(\mathbf{u})\, d\Omega - \int_\Omega \nabla\cdot(\tau(\mathbf{u})\mathbf{v})\, d\Omega \\
&= \int_\Omega \nabla\mathbf{v} : \tau(\mathbf{u})\, d\Omega - \int_{\partial\Omega} \tau(\mathbf{u})\mathbf{n}\cdot\mathbf{v}\, dS \\
&= \int_\Omega \nabla\mathbf{v} : \tau(\mathbf{u})\, d\Omega
\end{aligned}$$

because

$$\begin{aligned}
\tau(\mathbf{u})\mathbf{n}\cdot\mathbf{v} &= (\sigma\mathbf{n} + p\mathbf{n})\cdot(\underbrace{\mathbf{v_n}}_{=0} + \mathbf{v_t}) \\
&= \sigma\mathbf{n}\cdot\mathbf{v_t} + p\mathbf{n}\cdot\mathbf{v_t} \\
&= 0 + 0
\end{aligned}$$

and get

$$\int_\Omega \nabla\mathbf{v} : \tau(\mathbf{u}) - \nabla\cdot\mathbf{v}p + \rho\mathbf{g}\mathbf{v}\, d\Omega = 0 \tag{2.1}$$

$$\int_\Omega q\nabla\cdot(\rho\mathbf{u})\, d\Omega = 0 \tag{2.2}$$

Since no derivatives of q and p occur, the L_2 seems to be sufficient as test and solution space for the pressure p. The occurrence of at most first derivatives suggests a variant of H^1 as test and solution space for $\mathbf{v}$ and $\mathbf{u}$. With the condition $\mathbf{n}\cdot\mathbf{v} = 0$ this leads to the preliminary definition

$$\begin{aligned}
\hat{V} : &= \left\{\mathbf{u}\in H^1(\Omega)^d | \mathbf{u}\cdot\mathbf{n} = 0 \text{ on } \partial\Omega\right\} \\
\hat{M} : &= L_2(\Omega)
\end{aligned}$$

This, however, takes not into account that the solution of the system together with the slip boundary condition is only determined up to rigid body rotations, which form a vector space

$$\mathscr{R} := \text{span}\{\mathbf{u}(\mathbf{x}) = \mathbf{b}\times\mathbf{x} | \mathbf{b}\in\mathbb{R}^3, |\mathbf{b}| = 1, \mathbf{b} \text{ is an axis of symmetry of } \Omega\}.$$

It is useful to exclude the latter from the velocity test and solution spaces, since they destroy the continuity of the mapping $(\mathbf{u}, \mathbf{v}) \to \int_\Omega \nabla\mathbf{v} : \tau(\mathbf{u})\, d\Omega$, which we will need later. We therefore use the quotient vector space $\hat{V}/\mathscr{R}$. Another ambiguity concerns the pressure. Since in the system (1.9),(1.10) the pressure appears only in a derivative and is not specified on the boundary, the pressure solution can be determined only up to a constant function. To avoid this ambiguity we artificially constrain the pressure space to functions with zero mean value.[1] We end up with the following definitions for the spaces.

$$\begin{aligned} V : &= \left\{\mathbf{u} \in H^1(\Omega)^d/\mathscr{R} \mid \mathbf{u}\cdot\mathbf{n} = 0 \text{ on } \partial\Omega\right\} \\ M : &= L_{2,0}(\Omega) := \left\{ p \in L_2(\Omega) \mid \int\limits_\Omega p\, d\Omega = 0 \right\} \end{aligned}$$

If we further assume that $\tau(\mathbf{u})$ is linear, e.g. of the form $\tau = \mu\dot{\varepsilon}$,[2] we can write the problem (2.1) and (2.2) in the following variational form.

Problem 2.1. Find a pair $(\mathbf{u}, p)$ in $V \times M$ such that

$$\begin{array}{lccc} \forall \mathbf{v} \in V & a(\mathbf{u},\mathbf{v}) + b_1(\mathbf{v},p) & = & \langle \mathbf{l}, \mathbf{v}\rangle \\ \forall q \in M & b_2(\mathbf{v},p) & = & 0 \end{array} \tag{2.3}$$

with the bilinear forms

$$\begin{aligned} a(\mathbf{u},\mathbf{v}) &= \int\limits_\Omega \nabla\mathbf{v} : \tau(\mathbf{u})\, d\Omega \\ b_1(\mathbf{v},p) &= -\int\limits_\Omega p\nabla\cdot\mathbf{v}\, d\Omega \\ b_2(\mathbf{v},p) &= -\int\limits_\Omega p\nabla\cdot(\rho\mathbf{v})\, d\Omega \end{aligned}$$

and the right hand side

$$\langle \mathbf{l}, \mathbf{v}\rangle = \int\limits_\Omega \mathbf{l}\cdot\mathbf{v}\, d\Omega = \int\limits_\Omega -\rho\mathbf{g}\cdot\mathbf{v}\, d\Omega$$

2.2.2 Conditions for existence and uniqueness of the weak solution

It is far beyond the scope of this work to provide a complete explanation. I again merely try to emphasize the main points and connect our problem with the general

---

[1] We could of course again take the quotient space L_2/Q_0 with $Q_0 := \{q \in L_2(\Omega), q = const\}$.

[2] This assumption limits this analysis to the defect correction iteration, described in the last chapter. It permits, however, the use of the Fréchet derivative of the nonlinear operator, as used in the Newton method.

theory of [10, 24, 61, 9]. In fact this paragraph is a combination of the results found in these references.
Let us consider the following more general version of problem (2.3)

Problem 2.2. Find a pair $(\mathbf{u}, p)$ in $V \times M$ such that

$$\begin{array}{lccc} \forall \mathbf{v} \in V & a(\mathbf{u}, \mathbf{v}) + b_1(\mathbf{v}, p) & = & \langle \mathbf{l}, \mathbf{v} \rangle \\ \forall q \in M & b_2(\mathbf{v}, q) & = & \langle g, q \rangle \end{array} \tag{2.4}$$

Note that this is not a standard saddle point problem because three different forms are involved $(a(.,.), b_1(.,.), b_2(.,.))$ [3] If we associate linear operators A, B_1, B_2 with the linear forms in the following way

$$\begin{array}{rcll} \langle A\mathbf{u}, \mathbf{v} \rangle & = & a(\mathbf{u}, \mathbf{v}) & \forall \mathbf{u}, \mathbf{v} \in V \\ \langle B_1\mathbf{v}, q \rangle & = & b_1(\mathbf{v}, \mathbf{q}) & \forall \mathbf{v} \in V \text{ and } \forall q \in M \\ \langle B_2\mathbf{v}, q \rangle & = & b_2(\mathbf{v}, \mathbf{q}) & \forall \mathbf{v} \in V \text{ and } \forall q \in M \end{array}$$

The system can be written as follows

Problem 2.3. Find a pair $(\mathbf{u}, p)$ in $V \times M$ such that

$$\begin{array}{rcll} A\mathbf{u} + B_1 p & = & \mathbf{l} & \in V' \\ B_2\mathbf{u} & = & g & \in M' \end{array} \tag{2.5}$$

Now consider the linear mapping $\Phi \in \mathscr{L}(V \times M; V' \times M')$ with

$$\Phi(\mathbf{u}, p) = (A\mathbf{u} + B_1 p, B_2\mathbf{u})$$

which associates a pair $(\mathbf{u}, p)$ with the appropriate right hand sides $(\mathbf{l}, g)$. Suppose this mapping can be proved to be invertible or more precisely to be an isomorphism from $V \times M$ onto $V' \times M'$. Then its inverse $\Phi^{-1} \in \mathscr{L}(V' \times M'; V \times M)$ provides us with the solution for all pairs $(\mathbf{l}, g)$ in $V' \times M'$. To show that Φ is invertible we have to look at the partaking operators. A, B_1, B_2.
It has been shown in [9] and used in [10] that the following inf-sup conditions are necessary and sufficient for Φ to be an isomorphism.
There exist constants α_1 or α_2 such that

$$\forall \mathbf{u} \in K_2, \mathbf{u} \neq \mathbf{0} \sup_{\mathbf{v} \in K_1} \frac{a(\mathbf{u}, \mathbf{v})}{\|\mathbf{u}\|_M \|\mathbf{v}\|_M} \geq \alpha_2 > 0 \tag{2.6}$$

and

$$\forall \mathbf{v} \in K_1, \mathbf{v} \neq \mathbf{0} \sup_{\mathbf{v} \in K_2} a(\mathbf{u}, \mathbf{v}) > 0$$

---

[3] For the moment it is sufficient to note that it was a saddle-point problem if the forms $b_1(.,.)$ and $b_2(.,.)$ were identical. We will explain the usage of the term saddle-point problem when we encounter the first one.

or equivalently

$$\forall \mathbf{u} \in K_1, \mathbf{u} \neq \mathbf{0} \sup_{\mathbf{v} \in K_2} \frac{a(\mathbf{u}, \mathbf{v})}{\|\mathbf{u}\|_M \|\mathbf{v}\|_M} \geq \alpha_1 > 0 \tag{2.7}$$

and

$$\forall \mathbf{v} \in K_2, \mathbf{v} \neq \mathbf{0} \sup_{\mathbf{v} \in K_1} a(\mathbf{u}, \mathbf{v}) > 0$$

where the K_i are the kernels of the forms b_i. Additionally one needs

$$\forall q \in M, \|q\|_M \neq 0 \sup_{\mathbf{w} \in V} \frac{b_1(\mathbf{w}, q)}{\|\mathbf{w}\|_V \|q\|_M} \geq \beta_1 > 0 \tag{2.8}$$

$$\forall q \in M, \|q\|_M \neq 0 \sup_{\mathbf{w} \in V} \frac{b_2(\mathbf{w}, q)}{\|\mathbf{w}\|_V \|q\|_M} \geq \beta_2 > 0 \tag{2.9}$$

These conditions, which ensure the existence and uniqueness of the *continuous* problem have been checked in [10] for an inelastic approximation but with a slight difference to the general model needed to describe mantle convection. In [10] one has after substituting the kinematic viscosity $\nu = \frac{\mu}{\rho}$

$$\tau(\mathbf{u}) = \mu \nabla \mathbf{u}$$

with a viscosity that is strongly coupled to the density $\mu = \nu\rho$. Instead we have

$$\tau(\mathbf{u}) = \mu \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^t - \frac{1}{3} \nabla \cdot \mathbf{u} I \right)$$

with a viscosity, which depends, beside ρ, also on other variables. Note also, that the last two terms of $\dot{\hat{\varepsilon}}$ are missing in the approximation, that is used in [10]. Symmetry of τ is not enforced, and the third term, the second order diffusion, is omitted. Nevertheless I deem it possible to adapt the proofs. To achieve this, some assumptions about the viscosity must be made, which are, however, not subject of this work. If this task can be accomplished, the same inf-sup conditions must be proved for the finite dimensional spaces introduced by the finite element discretization. This also should be possible with only slight differences to the procedure in [10], because fortunately the Taylor Hood element pair is considered there. And especially one feature of this pair is constantly used. This is the possibility to construct a Fortin operator. Since this is needed also in the incompressible case and accordingly I already had to check it, it seems possible to extend the result in the future.

However, as mentioned before the incorporation of the slip boundary condition in the *discrete* system is much more difficult than for the continuous one. The discrete velocity space may be in general not rich enough to handle the additional constraint for the normal stress. A similar problem, as the one for the pressure, which is averted by the above stated conditions, will occur. Either a stabilization is needed, or the velocity space must be augmented by bubble functions on the slip boundary. It is possible that one of the solutions of Verfürth [59, 60] can be easily adapted to the case of the inelastic approximation, but this is still to be proved. This would be very welcome indeed, since Terra's compressibility approximation has the above mentioned side effect that it relies on a substitution that cannot be true exactly.

2.2.3 Differences to standard saddle point problems

Terra's treatment of incompressibility is closely related to the standard Stokes system. This becomes clear if one, looking at (1.28), recognizes that RB^t is the adjoint of BR, since R is diagonal and $R = R^t$. [4] It is worth looking at the differences to the above described general procedure to establish existence and uniqueness of the solution.

1. The forms $b_1(.,.)$ and $b_2(.,.)$ coincide for the standard Stokes system. The operator B_2 is then the adjoint of B_1 thus allowing to interpret it as the saddle point problem [5] and to use the results of [24].

2. Since $b_1(.,.)$ and $b_2(.,.)$ coincide there is only one kernel in (2.6) or (2.7) respectively, and the following conditions for $a(.,.)$ remain

$$\forall \mathbf{v} \in K, \mathbf{v} \neq \mathbf{0} \sup_{\mathbf{u}\in K} \frac{a(\mathbf{u},\mathbf{v})}{\|\mathbf{u}\|_M \|\mathbf{v}\|_M} \geq \alpha > 0 \tag{2.10}$$

$$\forall \mathbf{u} \in K, \mathbf{u} \neq \mathbf{0} \sup_{\mathbf{v}\in K} \frac{a(\mathbf{u},\mathbf{v})}{\|\mathbf{u}\|_M \|\mathbf{v}\|_M} \geq \alpha > 0 \tag{2.11}$$

   which ensure surjectivity and injectivity of the restriction of A to K and thus invertibility of the restriction of A on K. In the case of the Stokes problem the properties (2.11) and (2.10) are a consequence of the the coercivity of $a(.,.)$ on K

3. Since there is only one form $b(.,.)$, there is also only one inf-sup condition

$$\forall q \in M, \|q\|_M \neq 0 \quad \sup_{\mathbf{w}\in V} \frac{b(\mathbf{w},q)}{\|\mathbf{w}\|_V \|q\|_M} \geq \beta > 0 \tag{2.12}$$

   I found it impossible to explain in one sentence why this ensures the unique solvability. The proofs can be found e.g in [61] page 58-60 or [24] page 39-41. But later on I will give a simple example how (2.12) can be used.

---

[4] In the continuous form $C' \in \mathscr{L}(M, V')$ with $C'p = \rho\nabla p$ is the adjoint of $C \in \mathscr{L}(V', M)$ with $C\mathbf{v} = \nabla \cdot \rho\mathbf{v}$

[5] It is the saddle point condition for the Lagrangian functional $\mathscr{L}(\mathbf{v}, q) = J(\mathbf{v}) + b(\mathbf{v}, q) - \langle g, q\rangle$, arising from the incorporation of the constraint $\nabla\cdot\mathbf{v} = 0$ in the minimization of the energy functional $J(\mathbf{v}) = \frac{1}{2}a(\mathbf{v},\mathbf{v}) - \langle \mathbf{l}, \mathbf{v}\rangle$. This is just a generalization of the method of Lagrangian multipliers to Banach spaces. The term saddle point is due to the fact that the stagnation points of the Lagrangian functional are not always its extrema but can be saddle points. This is also true in case that the method of Lagrangian multipliers is applied to find the extrema of a function $f(\mathbf{x}) : R^n \to R$ under the constraints $g_i(\mathbf{x}) = 0$.

Of course these conditions have to be checked for the discrete spaces as well. The V_h ellipticity of a_h which is sufficient for (2.11) and (2.10) is proofed in [53] for piecewise linear elements. The proof becomes simpler for Terra's elements because of their exact approximation of the boundary. In [59] one finds an even more general approach. The author shows that standard approximation properties of finite elements in combination with a Lipschitz boundary are sufficient. The more challenging part is to check the discrete version of (2.12).

Conclusions

Due to the interpretation of Terra's compressibility approximation it nearly fits in the presented framework of the standard Stokes problem. The remaining difference is the slip boundary condition. But due to the results of [59, 60] this problem can be solved by a stabilization. The latter has but not yet been implemented in Terra. Expansion of the results to the inelastic approximation seems possible, but is still to be proved.

2.3 (De)motivating 2D examples

I will demonstrate the effect of ignoring the contents of this chapter by a problem posed in [33]. I solved it with the same code, that was used for the (exact) calculation of the operators used in the multigrid framework of the next chapter. This way it served also as a test case for this computer algebra code. We start with the continuous problem.

2.3.1 The example continuous problem

Consider a incompressible homogeneous Stokes system with constant viscosity, without buoyancy forces

$$\nabla \cdot \tau - \nabla p = 0 \tag{2.13}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.14}$$

For constant μ the viscous stress tensor can be simplified

$$\tau_{lm} = 2\mu \left(\dot{\varepsilon}_{lm} - \frac{1}{3}\delta_{lm}\dot{\varepsilon}_{kk} \right) = \mu \nabla \mathbf{u} \tag{2.15}$$

which gives us

$$\mu \nabla^2 \mathbf{u} - \nabla p = 0 \tag{2.16}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.17}$$

If we impose zero boundary conditions for the velocity we can actually compute the solution by hand.

$$\mathbf{u}(x,y) = 0 \qquad \forall x,y \in \Omega \tag{2.18}$$

and

$$p(x,y) = c \qquad \forall x,y \in \Omega \tag{2.19}$$

That means, that everywhere in the domain velocity is zero and pressure an arbitrary constant. We will have a quick look at our inf-sup conditions and see how they ensure the uniqueness of this solution. We start with the uniqueness of the velocity. Because of the constraint $\nabla \cdot \mathbf{u}$ the velocity solution $\mathbf{u}$ must be in the kernel K of B. That means that $\forall q \in M \quad b(\mathbf{u},q) = 0$. However, that also means that the weak form of the second term in the first equation vanishes for all $\mathbf{u} \in K$.

$$\forall \mathbf{u} \in K \quad \int\limits_{\Omega} \nabla p \mathbf{u} \, d\Omega = 0$$

That means that the first line of the weak Stokes system simplifies to

$$\forall \mathbf{v} \in V \quad a(\mathbf{u},\mathbf{v}) = 0.$$

The invertibility condition of $a(.,.)$ on K (2.11) and (2.10) then ensures that $\mathbf{u} = \mathbf{0}$. It remains to show that the pressure is uniquely determined. This is in fact ensured by (2.12) which is fulfilled for the continuous Stokes system [43]. To see this, observe that (2.12) makes sure that for any non constant $q \in M$ there exists a $\mathbf{u}_q \in V$ such that

$$\frac{\langle q, \nabla \cdot \mathbf{u}_q \rangle}{||\mathbf{u}_q||_V} \geq \beta ||q||_M \tag{2.20}$$

Suppose we had a pressure solution p. If we substitute $\mathbf{u} = \mathbf{0}$ in the weak Stokes system, the first line simplifies to

$$\forall \mathbf{v} \in K \quad \int\limits_{\Omega} \langle p \nabla \cdot \mathbf{u} \rangle = 0$$

That means that the numerator in (2.20) vanishes for all $\mathbf{v}$ so that $||q||_M = 0$. [6]

2.3.2 Solutions of the discrete version

Assuming that we use a 4x4 grid for pressure and velocity, Fig. 2.1 shows a correct pressure solution. If we further assume that we use the bilinear basis functions for pressure and velocity on this grid (Q_1, Q_1 discretization) we find that the arising

---

[6] Since M is the quotient vector space $L_2/Ker(B') = L_2/Q_0$ and accordingly $|| \; ||_M = ||q - \frac{1}{|\Omega|} \int_\Omega q \; d\Omega||$ a quotient-space norm, this means that the pressure is determined up to constant functions in L_2.

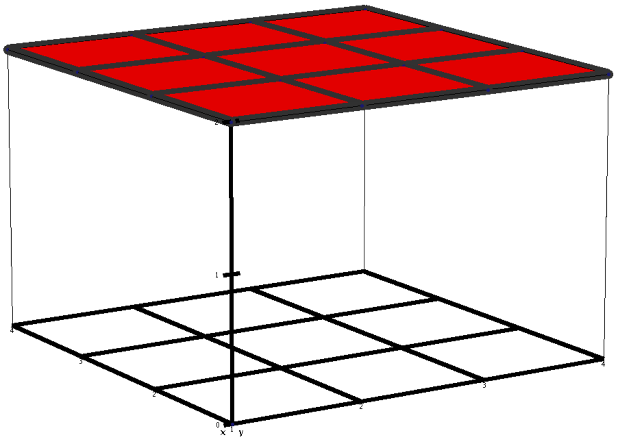


Figure 2.1: The figure shows a rectangular domain which is discretized with a four times four grid. and also the solution for the pressure as a function of x and y, which being a constant function must have the same values on all 16 grid points, this way being perfectly flat.

linear system becomes more ambiguous than the continuous one. It allows not only the correct solutions where pressure is a constant but also the solutions shown in Fig. 2.2. Note that this is not a round-off error. It is a perfectly correct solution of the linear system arising from the discretization, which, however, contains linear dependent equations. Things are actually worse than this. We can find more artificial solutions. Only some of them are shown in Fig. 2.2. The artificial pressure modes actually form a vector space. This means that even the thousandfold of this mode is possible or every linear combination of this modes with amplitudes as huge as one wishes. One shows easily with the exact linear systems that the dimension of this vector space is 8 even for this small 16 node grid. [7] This is still not the worst situation, that can arise if one ignores the LBB. Up to now only the pressure solution is tainted but there are also examples where the velocity is over constrained by the discrete divergence-free request, resulting in locking phenomena. This may even lead to a situation where the only possible solution for the discrete velocity is

$$\mathbf{v} = \mathbf{0} \text{ on } \Omega$$

---

[7] It is also 8 even for the 8x8 or 16x16 mesh.

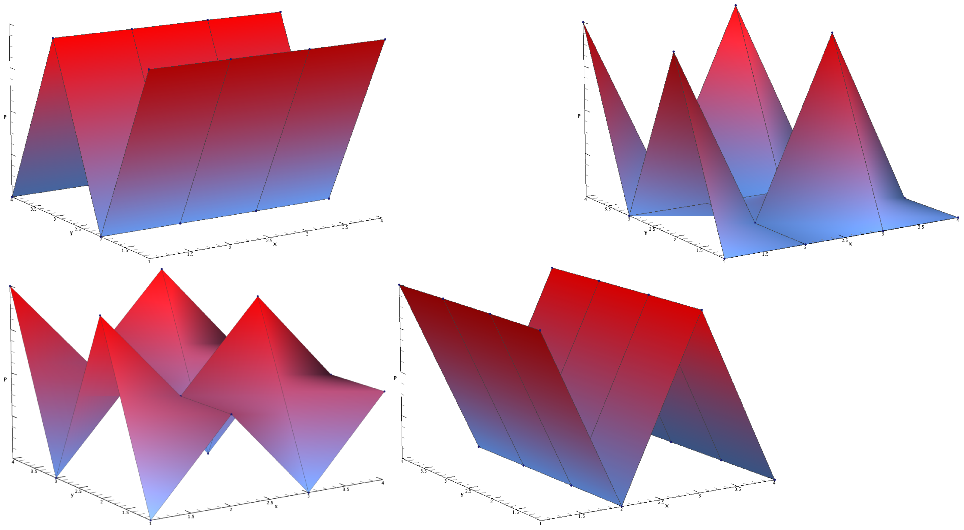


Figure 2.2: This figure shows artificial numerical solutions of the problem which is due to the discretization with bilinear basis functions for pressure and velocity. resulting in an ambiguous linear system. The amplitude is *arbitrary*. These solutions belong to the vector space $S = Ker(B_h^t) = Ker(\nabla_h)$. Since $dim(S) = 8$ these solutions are only a part of a basis of S.

.

It is worth noting that such things frequently occurred for equal-order interpolations [8] and lead to the theory around the discrete LBB. However neither are they restricted to those interpolation nor can they be averted by simply using a less accurate discrete pressure basis, as the following , by no means artificial, example of Brezzi [24] shows.

2.3.3 Locking

Let Ω be a bounded polygon in $\mathbb{R}^2$ with a triangulation $\mathscr{T}_h$. Suppose we use piecewise linear velocity and piecewise constant pressure with zero mean value.

$$W_h = \left\{ \mathbf{w} \in \mathscr{C}^0(\bar{\Omega}); \mathbf{w}|_K \in P_1^2 \qquad \forall K \in \mathscr{T}_h, \right\}$$

$$X_h = W_h \cap H_0^1(\Omega)^2,$$

$$Q_h = \left\{ q \in L^2(\Omega); q|_K \in P_0 \qquad \forall K \in \mathscr{T}_h, \right\}$$

$$M_h = Q_h \cap L_0^2(\Omega)^2,$$

---

[8] as used in Terra

$$L_0^2(\Omega) = \left\{ p \in L^2(\Omega); \int\limits_\Omega p dx = 0 \right\}$$

Now consider the triangulation of domain Ω and let us denote

- t number of triangles
- v_I number of internal vertices,
- v_B number of boundary vertices,

Then we have

- $t - 1$ = number of dof for pressure
- $2v_I$ = number of dof for velocity as $\mathbf{v} = \mathbf{0}$ on $\partial\Omega$
- $t = 2v_I + v_B - 2$ (Euler's equation)
- $\rightarrow (t - 1) > (2v_I - 1)$ number of boundary vertices,

$$\begin{aligned} V_h = \{\mathbf{v_h} \in X_h; \langle \nabla \cdot \mathbf{v_h}, \mu_h \rangle = 0 \qquad \forall \mu_h \in M_h\} \\ = \{0\} \end{aligned}$$

That means, that the velocity is completely locked. The only possible solution that is divergence free in the discrete sense is $\mathbf{v} = \mathbf{0}$. However, a partly locked system is even more dangerous because it is harder to detect. In a numerical algorithm the exact locking will be disguised by round of errors, but influence the condition number disastrously. [9] In the sequel we will have a more detailed look at the threat of over constraining.

2.3.4 Short summary

In the context of saddle point problems the following points must be taken into account.

- Standard finite elements can lead to ambiguous or over constrained linear systems
- The ambiguity leads to vector spaces of artificial solutions. Even if the start estimate for the pressure is clean, over time the amplitudes of this artificial solutions can increase to huge values because they are not detectable by the solver. The over constraining leads to locking phenomena that might be total.

---

[9] This is a reminder that the inf-sup constant β has also a fundamental influence on the error bounds, although this has not been explicitly mentioned up to now.

- *Every* solution is potentially tainted.
- It is *absolutely vital* to ensure the stability of the discretization. since else the problem is not well posed, which is a necessary condition for *every* numerical algorithm.

Sometimes this can be done most elegantly theoretically for all grids built with a special element pair. If this proves to hard a task one can attempt to prove it at least for the grid in use. We describe the procedure in the next subsection.

2.4 Checking the LBB

As already mentioned the LBB is fulfilled for the (continuous) Stokes Problem but implicates a similar condition for the numerical algorithm.

$$\inf_{q_h \in Q_h} \sup_{v_h \in V_h} \frac{b(v_h, q_h)}{\| v_h \|_{V_h} \| q_h \|_{Q_h}} \geq \beta_h;$$

This condition ensures that we can calculate our solution. It depends on the approximation spaces for pressure and velocity, that means, on the grid and the elements used for the approximation. We therefore at first have to describe the latter. Fig. 2.3 shows a cross-section of the grid currently implemented in Terra.

2.4.1 Terra's grid

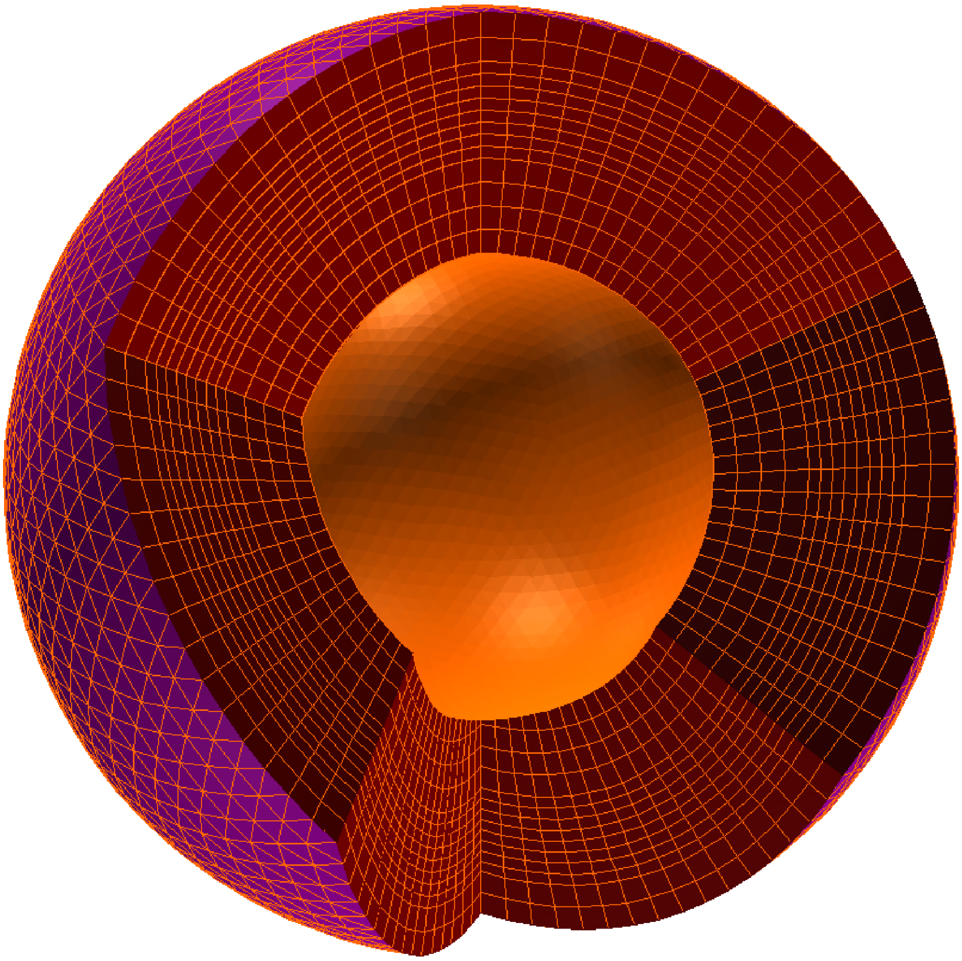


Figure 2.3: A cross-section of Terra's grid

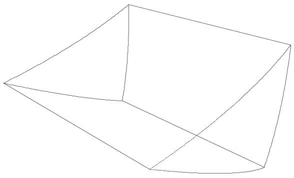


Figure 2.4: A sample grid cell

A grid cell is shown in Fig. 2.4. The basis functions for velocity and pressure are piecewise bilinear on such a grid-cell.
That means continuous equal order interpolation is used. According to the experience obtained for two dimensions for linear and bilinear equal order discretizations (see for instance [61] or [24]) it is very probable that the equal order interpolation can not satisfy the LBB. I do not give an explicit proof. It is not always easy to find one, even for the violation of the LBB, if the rank deficiency of the discrete version of B^t ist not obvious through the comparison of the numbers of dof for pressure and velocity. This is not the case here. [10] What I am driving at, is the non existence of a proof that the LBB holds for this grid. In fact some users of Terra have reported spurious pressure modes that indicate the violation of the LBB. As mentioned before an aim of this work is to implement a local grid refinement. An example grid is given in Fig. 2.6 For the locally refined grid the continuity require-

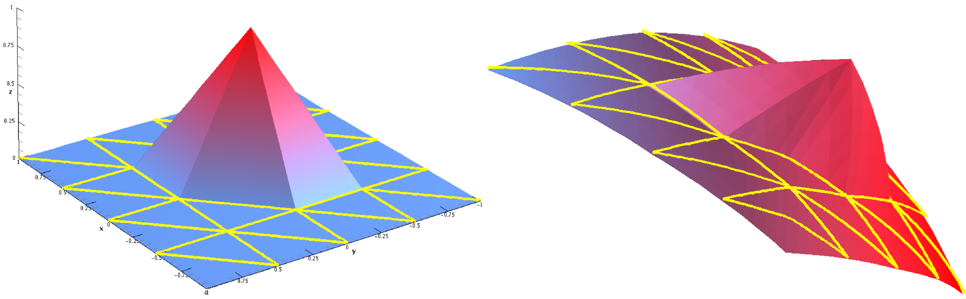


Figure 2.5: A linear basis function defined on the plane and its spherical equivalent. The velocity approximation space is the tensor-product of these basis functions and one dimensional linear basis functions in the radial direction. The lateral basis functions are defined as functions of the spherical barycentric coordinates. Note that the spherical barycentric coordinates are recursively defined by the grid refinement process. There is *no* explicit formulation e.g. as function of ϕ and θ. One can show that the recursive refinement process leads to a bijection, between a reference grid cell and the transformed one. See [7] for the proof. But this bijection is also only given implicitly.

---

[10] I could show that the LBB is not fulfilled locally on a small partition of the grid, but this is only sufficient for the LBB to hold globally. It shows, however, that there is not much hope for the present grid in this respect.

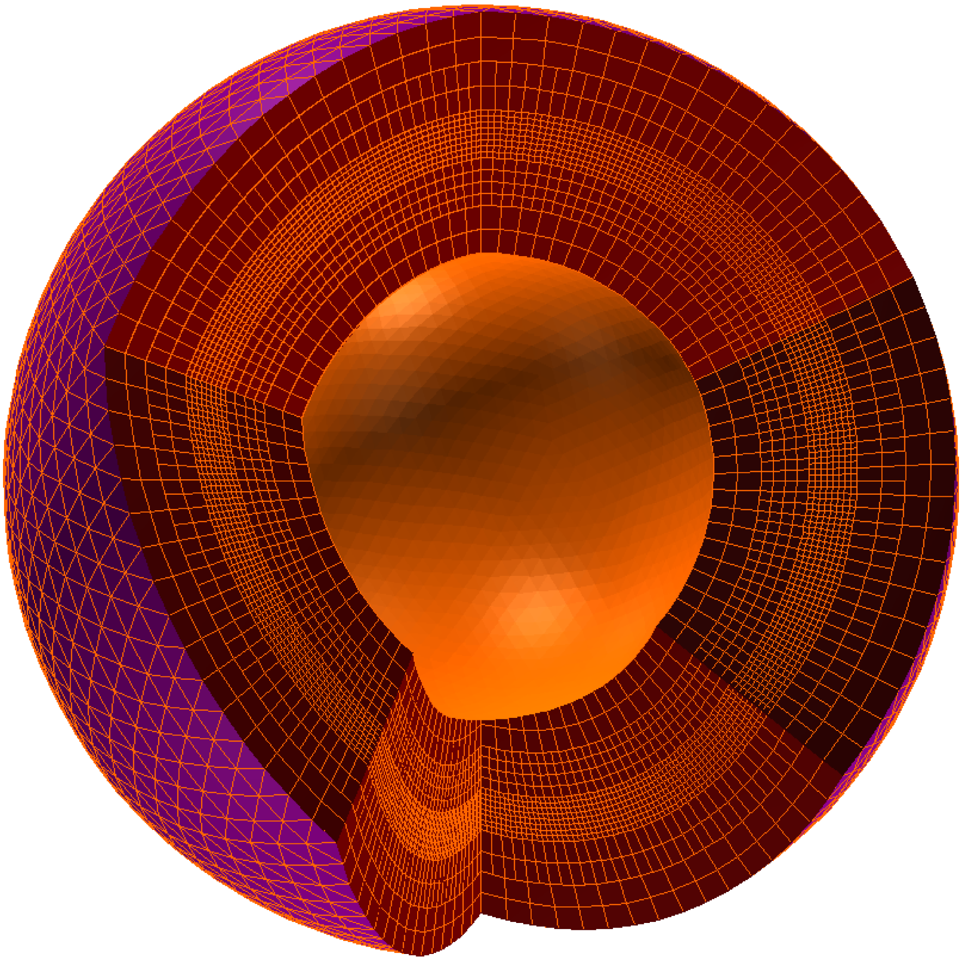


Figure 2.6: A locally refined mesh. Although the picture only shows a radial refinement, lateral refinements must be possible as well without violation of the LBB.

ment for the velocity space enforces the values at hanging nodes to be interpolated, this way erasing degrees of freedom for the velocity, possibly needed to ensure the inf-sup condition. The situation becomes even more difficult. To be able to proof the LBB we will have to change the grid. The one objective we started with is, to achieve this with as few changes in the code as possible.

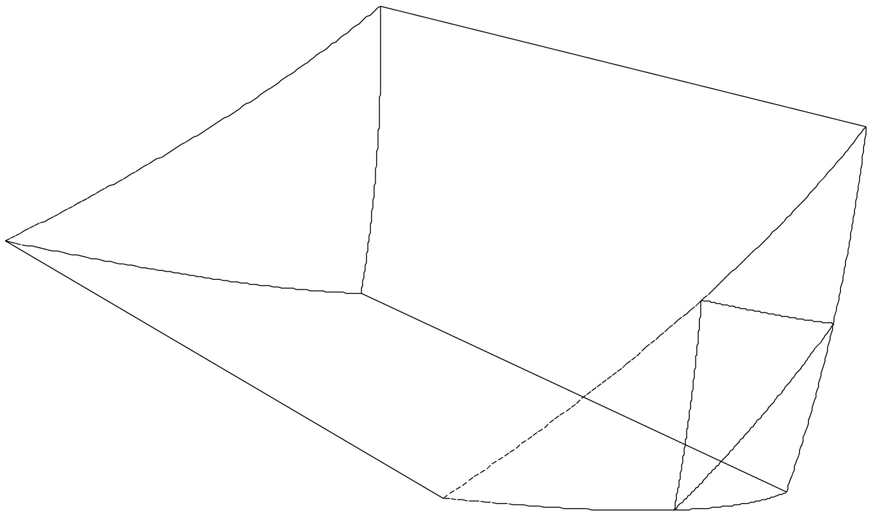


Figure 2.7: An example for a local lateral grid refinement. Note the boundary between a finer and a coarser region of the grid contains hanging nodes.

2.4.2 A new grid

The idea for this new grid was inspired by a variant of the Taylor Hood Element(P_2, P_1) that uses a refined P_1 grid instead of P_2. This element is reported to generate slightly better conditioned matrices than the original Taylor Hood element. But naturally the accuracy is only of first order. Since the Terra code implements a multi-grid, different grid levels already exist. Now the LBB must be proved for this grid.

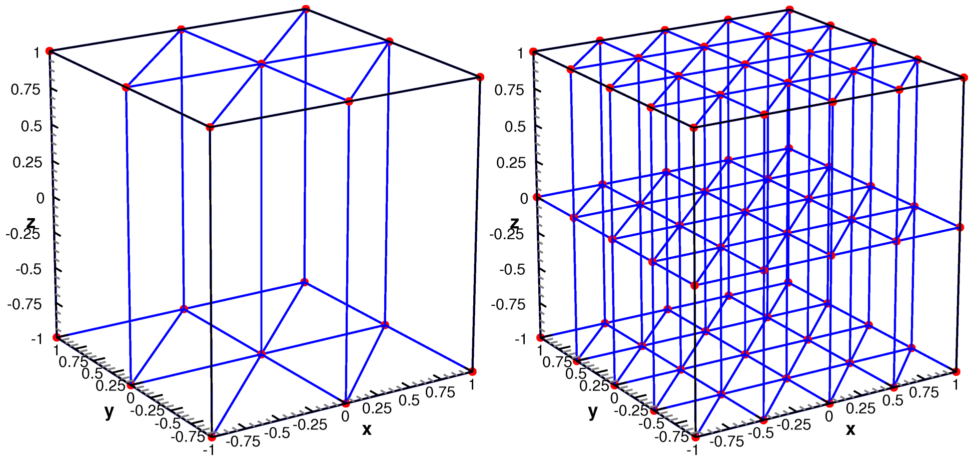


Figure 2.8: Several grid-cells for pressure and velocity. The velocity grid is just the dyadic refined pressure grid. Note that this refinement strategy is already used in the code.

2.4.3 Construction of the Fortin operator

According to [24] this can in general be done as long as the continuous inf-sup condition [11] holds by the construction of a family of uniformly continuous operators Π_h from V_h into V_h satisfying

$$\begin{cases} b(\Pi_h \mathbf{v} - \mathbf{v}, q_h) = 0, \forall q_h \in Q_h, \\ \| \Pi_h \mathbf{v} \|_V \leq c \| \mathbf{v} \|_V \end{cases} \tag{2.21}$$

We will give a short outline of the proof, which can be found in [24] later. But before this we show how this operator is built. The operator Π_h [12] is constructed in two steps.

Theorem 2.1. (Brezzi Fortin) Let $\Pi_1 \in \mathscr{L}(V, V_h)$ and $\Pi_2 \in \mathscr{L}(V, V_h)$ be such

---

[11] which was proofed for the Stokes problem in [43].

[12] which is elsewhere sometimes called a Fortin operator

that

$$\begin{cases} \| \Pi_1 \mathbf{v} \|_V \leq c_q \| \mathbf{v} \|_V, \\ b(\Pi_2 \mathbf{v} - \mathbf{v}, q_h) = 0 \qquad \forall q_h \in Q_h, \\ \| \Pi_2 (I - \Pi_1) \mathbf{v} \|_V \leq c_2 \| \mathbf{v} \|_V, \end{cases} \tag{2.22}$$

then (3) holds, and the inf-sup condition follows.

An approximation operator Π_1 , a Clement operator, can be constructed for many types of finite elements. Its existence follows from the usual regularity assumptions about the grid. The interesting part is to find Π_2.

2.4.4 Macro elements

This is done by a macro-element technique. A macro-element is the union of a fixed number of adjacent elements along a well defined pattern.
Given a partition into macro-elements we can define the following spaces

$$V_{0,M} = \{v_h | \mathbf{v_h} \in V_h, \mathbf{v_h} = \mathbf{0} \text{ in } \Omega \backslash M\}$$

Theorem 2.2. (Brezzi Fortin) Suppose V_h is defined on a mesh of macro-elements and can be written as

$$V_h = \tilde{V_h} \oplus (\oplus_M V_{0,M})$$

and the matrix associated with

$$\int\limits_M \mathbf{v_h} \boldsymbol{\phi}_\mathbf{h} dx, \qquad \forall \mathbf{v_h} \in V_{0,M}, \qquad \forall \boldsymbol{\phi}_\mathbf{h} \in \Phi_M \supset grad\ Q_H | M$$

has full rank. Then a suitable Π_2 can be constructed.

We will proceed as follows

1. We introduce the macro element.
2. We show that the discrete velocity space can be decomposed in such macro elements.
3. We visualize the local function spaces.
4. We show that the local rank condition is fulfilled for a reference macro, even in the presence of hanging nodes.
5. We show how the rank condition can be checked *exactly* by means of computer algebra for every iso parametric (bilinear) transformed macro, if the vertices are known, and that the number of macros, for which this computation is necessary, is small.
6. We propose a possible integration in the grid generation process, that will make sure that the grid fulfills the LBB for Terra's implicitly given mapping.

Grid decomposition

Fig. 2.9 shows a macro with its pressure and velocity grid lines. Fig. 2.10 shows that the grid can be decomposed disjunctively into macros. which is a necessary condition of theorem (2.2).
Remark:
Although this seems to be trivial, it is in fact rather intricate. To see this, assume we would use macros consisting of only 6 pressure-grid cells, which share a common edge. It is in fact possible to show the LBB locally for such macros. [13] But it is not possible to partition the grid with these elements. If we additionally take some five-cell macros we succeed in partitioning the unrefined grid without hanging nodes, and even for a radially refined grid. But it is impossible to do this for a laterally refined grid, because the patches necessary to complete the partition, can be shown to be definitely not inf-sup stable by a simple counting of the functions belonging to $V_{0,M}$ and the pressure dof they have to compensate. So an extension of the proof to the whole grid along the lines of the macro element technique would be impossible. This discussion makes clear that it is much harder to show the LBB for a given grid, than to construct one that is stable.

Mini Stokes on the reference macro

We now proceed to show the definition of the spaces $V_{0,M}$. This is very important because these functions are our only raw material to fulfill the LBB locally on the macro. The local LBB expressed by the rank condition of theorem (2.2) can be seen as a solvability condition for an enclosed flow Stokes problem on the macro. That sounds difficult, but is in fact simple. Without loss of generality let us suppose that we want to solve this problem for homogeneous boundary conditions. Accordingly all velocity values on the boundary have to vanish. To ensure a divergence free solution we can only use the dof belonging to velocity functions whose support is fully inside the macro. The first thing we do, after we have identified these functions, is to count them as well as the pressure basis functions inhabiting the same macro. Since for our mini Stokes on the macro there is no boundary condition for the pressure, the boundary nodes must be taken into account. If the number of pressure functions minus one is greater than the number of (linearly independent) members of $V_{0,M}$ than our errand to fulfill

$$0 = \int\limits_{M} \mathbf{v}\nabla q \, d\Omega \quad \forall q \in Q_h|M$$

is hopeless and we have to try at least another macro definition. However, it does not follow that the LBB will not hold globally. If on the other hand, the number of

---

[13] This was my first idea, because the proof, although not straight forward, could probably make use of some similarities to the Taylor Hood proof. I checked the LBB on such a macro with the rank condition and succeeded.

velocity functions is large enough, this is still not sufficient, but at least we have a chance that the rank condition holds.
Fig. 2.11 shows a member of $V_{0,M}$ and Fig. 2.12 makes clear that not all velocity dof in the macro can be used. In Fig. 2.4.4 the nine nodes with its 27 basis functions can be seen as green dots. since 27 is greater than 17 we have a chance and can proceed to check the rank condition.

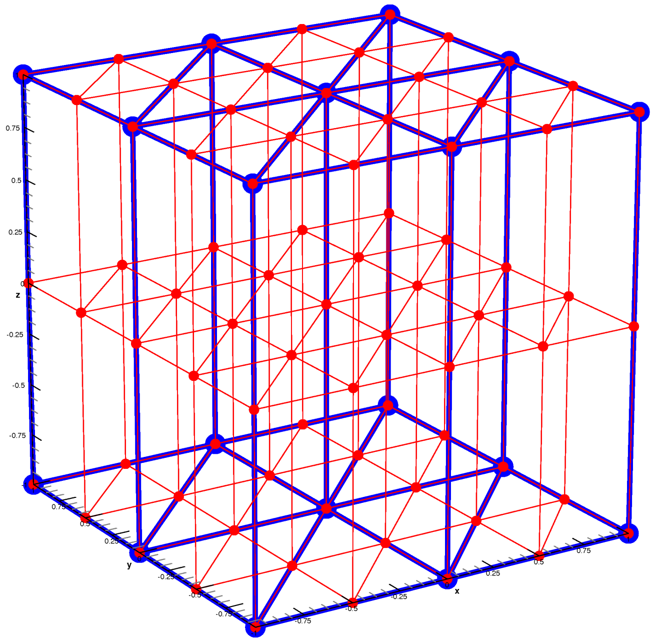


Figure 2.9: One macro with its 3x3x2 dof for pressure (blue points) and its 5x5x3x3 dof for velocity (red points)

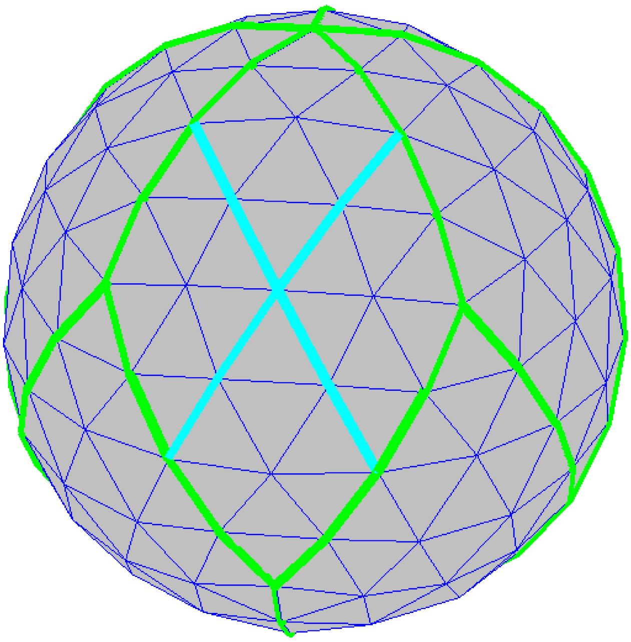


Figure 2.10: This is the coarsest possible grid to show the partition into macros. The blue lines mark the (pressure) grid cells., The green lines mark the edges stemming from the original icosahedron , splitting the sphere into 10 identical diamonds. The cyan lines mark the edges of the macros, splitting every diamond laterally into 4 macros, although the splitting is only shown for one diamond. Since every possible grid in Terra is a recursive dyadic refinement of this grid, it is clear that every grid can be decomposed into macros. Please note, that the impression that the grid is polyhedral is an artifact of the plot. The surface of the *discretized* sphere is perfectly smooth in Terra. However the kinks in the grid lines would remain, even if they were correctly plotted as composition of great circle segments.

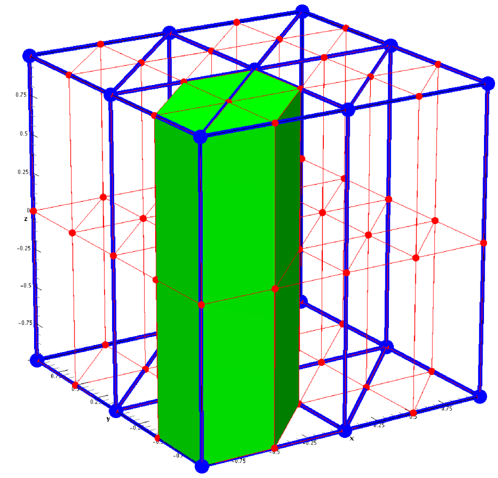


Figure 2.11: Allowed

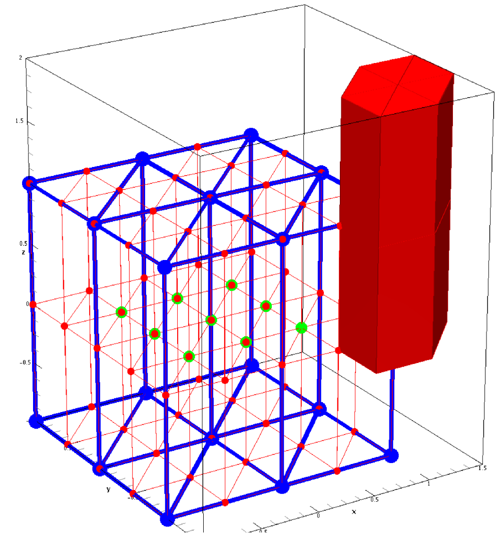


Figure 2.12: Forbidden The support(red) of this basis function is not contained in the macro. $V_{0,M}$ contains only functions with support *completely* inside M. Only the basis functions with maximum on the nine green points belong to $V_{0,M}$.

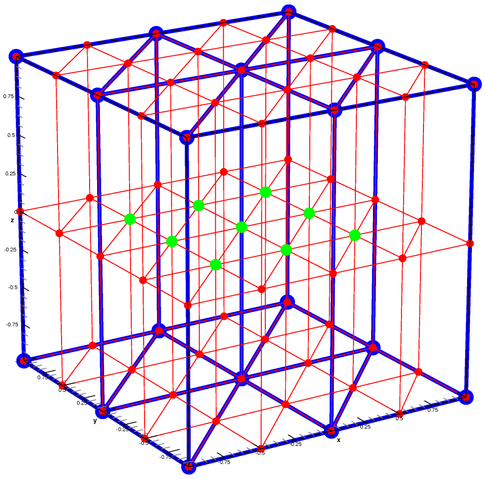


Figure 2.13: The macro with the nodes where the velocities can be defined freely. Only the basis functions with maximum on the nine green points belong to $V_{0,M}$.

$$\begin{pmatrix} \int_M \nabla P_1 \vec{v1}_x dx & \int_M \nabla P_1 \vec{v1}_y dx & \dots & \int_M \nabla P_1 \vec{v9}_z dx \\ \int_M \nabla P_2 \vec{v1}_x dx & & & \vdots \\ \vdots & & & \\ \int_M \nabla P_{18} \vec{v1}_x dx & & \dots & \int_M \nabla P_{18} \vec{v9}_z dx \end{pmatrix} \tag{2.23}$$

I did this with a computer algebra code because this task is even more tedious than the 27 × 18 integrals suggest. Note that the basis functions are only *piecewise* bilinear. This turns one of the above mentioned integrals in a rather longish sum of integrals over the "pieces" of the support intersection. Since $\vec{v}$ is a 3D vector the nine grid points imply 27 basis functions for V_0 hence 27 columns As span for the gradient space of the pressure I used the gradients of the 18 basis functions, which do not form a basis because $dim(gradP_h|_M) = dim(P_h|_M) - 1 = 17$. In fact we could drop an arbitrary ∇P_i because it can be constructed as a linear combination of the remaining 17. Therefore, if everything is well, we expect the matrix to have rank 17. Fig. 2.14 shows the matrix. Fig. 2.15 shows the (symbolically computed) LU decomposition, and thus the desired result of rank 17.

$$\begin{pmatrix}
\frac{1}{16} & \frac{1}{16} & \frac{5}{96} & 0 & \frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{96} & \frac{1}{48} & \frac{1}{48} & \frac{1}{192} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & \frac{1}{16} & -\frac{5}{96} & 0 & \frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 & \frac{1}{16} & 0 & -\frac{1}{96} & \frac{1}{48} & \frac{1}{48} & -\frac{1}{192} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & -\frac{1}{16} & \frac{1}{96} & \frac{1}{8} & -\frac{1}{16} & \frac{5}{96} & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} & 0 & 0 & 0 & \frac{1}{24} & -\frac{1}{48} & \frac{1}{192} & \frac{1}{16} & 0 & \frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{16} & -\frac{1}{16} & -\frac{1}{96} & \frac{1}{8} & -\frac{1}{16} & -\frac{5}{96} & \frac{1}{16} & \frac{1}{16} & -\frac{5}{96} & 0 & 0 & 0 & \frac{1}{24} & -\frac{1}{48} & -\frac{1}{192} & \frac{1}{16} & 0 & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & -\frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & -\frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{16} & \frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{8} & \frac{5}{96} & -\frac{1}{48} & \frac{1}{24} & \frac{1}{192} & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} & 0 & \frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 \\
-\frac{1}{16} & \frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{8} & -\frac{5}{96} & -\frac{1}{48} & \frac{1}{24} & -\frac{1}{192} & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & -\frac{5}{96} & 0 & \frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 \\
-\frac{1}{16} & -\frac{1}{16} & \frac{5}{96} & -\frac{1}{8} & \frac{1}{16} & \frac{5}{96} & -\frac{1}{16} & \frac{1}{16} & \frac{1}{96} & \frac{1}{16} & -\frac{1}{8} & \frac{5}{96} & 0 & 0 & \frac{3}{32} & -\frac{1}{16} & \frac{1}{8} & \frac{5}{96} & \frac{1}{16} & -\frac{1}{16} & \frac{1}{96} & \frac{1}{8} & -\frac{1}{16} & \frac{5}{96} & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} \\
-\frac{1}{16} & -\frac{1}{16} & -\frac{5}{96} & -\frac{1}{8} & \frac{1}{16} & -\frac{5}{96} & -\frac{1}{16} & \frac{1}{16} & -\frac{1}{96} & \frac{1}{16} & -\frac{1}{8} & -\frac{5}{96} & 0 & 0 & -\frac{3}{32} & -\frac{1}{16} & \frac{1}{8} & -\frac{5}{96} & \frac{1}{16} & -\frac{1}{16} & -\frac{1}{96} & \frac{1}{8} & -\frac{1}{16} & -\frac{5}{96} & \frac{1}{16} & \frac{1}{16} & -\frac{5}{96} \\
0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{96} & -\frac{1}{16} & -\frac{1}{16} & \frac{5}{96} & 0 & 0 & 0 & \frac{1}{48} & -\frac{1}{24} & \frac{1}{192} & \frac{1}{16} & -\frac{1}{8} & \frac{5}{96} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & -\frac{1}{16} & \frac{1}{96} \\
0 & 0 & 0 & 0 & -\frac{1}{16} & -\frac{1}{96} & -\frac{1}{16} & -\frac{1}{16} & -\frac{5}{96} & 0 & 0 & 0 & \frac{1}{48} & -\frac{1}{24} & -\frac{1}{192} & \frac{1}{16} & -\frac{1}{8} & -\frac{5}{96} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & -\frac{1}{16} & -\frac{1}{96} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & 0 & \frac{1}{96} & -\frac{1}{24} & \frac{1}{48} & \frac{1}{192} & 0 & 0 & 0 & -\frac{1}{16} & -\frac{1}{16} & \frac{5}{96} & -\frac{1}{8} & \frac{1}{16} & \frac{5}{96} & -\frac{1}{16} & \frac{1}{16} & \frac{1}{96} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & 0 & -\frac{1}{96} & -\frac{1}{24} & \frac{1}{48} & -\frac{1}{192} & 0 & 0 & 0 & -\frac{1}{16} & -\frac{1}{16} & -\frac{5}{96} & -\frac{1}{8} & \frac{1}{16} & -\frac{5}{96} & -\frac{1}{16} & \frac{1}{16} & -\frac{1}{96} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{48} & -\frac{1}{48} & \frac{1}{192} & -\frac{1}{16} & 0 & \frac{1}{96} & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{96} & -\frac{1}{16} & -\frac{1}{16} & \frac{5}{96} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{48} & -\frac{1}{48} & -\frac{1}{192} & -\frac{1}{16} & 0 & -\frac{1}{96} & 0 & 0 & 0 & 0 & -\frac{1}{16} & -\frac{1}{96} & -\frac{1}{16} & -\frac{1}{16} & -\frac{5}{96}
\end{pmatrix}$$

Figure 2.14: The matrix of the discrete divergence Operator of the macro.

$$\begin{pmatrix}
\frac{1}{16} & \frac{1}{16} & \frac{5}{96} & 0 & \frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{96} & \frac{1}{48} & \frac{1}{48} & \frac{1}{192} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{1}{8} & -\frac{1}{24} & \frac{1}{8} & -\frac{1}{8} & \frac{1}{24} & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} & -\frac{1}{16} & 0 & -\frac{1}{96} & \frac{1}{48} & -\frac{1}{24} & 0 & \frac{1}{16} & 0 & \frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{5}{48} & 0 & 0 & -\frac{1}{48} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{48} & 0 & 0 & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{8} & -\frac{1}{16} & \frac{23}{480} & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} & -\frac{1}{16} & \frac{1}{8} & \frac{23}{480} & \frac{1}{48} & \frac{1}{48} & \frac{1}{120} & \frac{1}{16} & 0 & \frac{1}{96} & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} & 0 & \frac{1}{16} & \frac{1}{96} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{16} & \frac{43}{480} & 0 & \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & 0 & \frac{43}{480} & \frac{1}{24} & \frac{1}{24} & \frac{31}{320} & 0 & \frac{1}{8} & \frac{1}{16} & \frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{5}{96} \\
0 & 0 & 0 & 0 & 0 & -\frac{1}{10} & 0 & 0 & -\frac{5}{48} & 0 & 0 & \frac{1}{240} & 0 & 0 & -\frac{1}{120} & 0 & 0 & -\frac{1}{48} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & -\frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{48} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{3}{32} & \frac{1}{16} & 0 & \frac{3}{32} & \frac{1}{16} & 0 & \frac{3}{32} & \frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{8} & 0 & \frac{1}{16} & \frac{1}{8} & 0 & \frac{1}{16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{5}{48} & \frac{1}{48} & \frac{1}{48} & \frac{19}{192} & \frac{1}{16} & 0 & \frac{3}{32} & \frac{1}{16} & -\frac{1}{16} & \frac{11}{96} & 0 & \frac{1}{16} & \frac{11}{96} & \frac{1}{16} & \frac{1}{16} & \frac{7}{96} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{48} & -\frac{1}{48} & \frac{1}{192} & -\frac{1}{16} & 0 & \frac{1}{96} & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{96} & -\frac{1}{16} & -\frac{1}{16} & \frac{5}{96} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{12} & 0 & 0 & 0 & \frac{5}{96} & -\frac{5}{96} & \frac{43}{576} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{48} & \frac{73}{3840} & -\frac{73}{3840} & -\frac{1657}{23040} & 0 & 0 & -\frac{19}{240} & 0 & 0 & \frac{1}{240} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & \frac{1}{16} & -\frac{1}{96} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{48} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{3}{8} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Figure 2.15: LU decomposition of the matrix of the discrete divergence Operator of the macro.

2.4.5 Extension to the mapped macro

Up to now, we have only shown that the grid can be partitioned into macros, and that we can show that the LBB holds on a *reference* macro. It is still to be proved that this remains true for the mapped macros. If possible we would like to show it for a large class of mappings at once so that at least every macro in the present grid can be defined as the image under such a mapping.
The most serious obstacle to be overcome is that the mapping itself is only implicitly given. It is not possible to formulate it as a formula. We only know that it is bijective and does not alter the geometry very much. That is not much to work upon. Of course one can find measures for the geometrical distortion under the mapping, which is small, and try to base the proof on some assumptions about the distortion like e.g. in [42]. But I found it too hard to do this for the rank-condition criterium. The problem is that F will be in most cases defined piecewise, that is "prism wise". To show how things will get complicated let F be the mapping that maps the reference element to a real element.
Let further $PRV(\mathbf{v})$ be the set of all pressure prisms that are inside the support intersection of *one* velocity basis function $\mathbf{v}$ and *one* pressure basis function p. The according shape functions on the reference element are denoted $\mathbf{v_0}$ and p_0.
Then

$$\begin{aligned} \int_M \nabla p \mathbf{v} dx &= \int_M \nabla p_0 \mathbf{v_0} det(F) dx \\ &= \sum_{k=1}^{card(PRV)} \int_{vpr_k} \nabla p_0|_{vpr_k} \mathbf{v_0}|_{vpr_k} det(F)|_{vpr_k} dx \end{aligned}$$

. This is just what happens for one matrix entry of Fig. 2.14. A rather unwelcome consequence is the following:
Even a mapping, which is piecewise affine on the prisms inside the macro, does not multiply whole lines of the matrix with the same number.
From this abstract point of view it is therefore nearly unpredictable what the rank will be under a certain mapping even if the mapping is known.
Its abstractness and therefore wide applicability is the strength of the rank-condition criterium. On the other hand the same abstractness is also its weakness because it is rather hard to predict under a mapping. Consequently it is better used as a test. This is in fact, what we do.

Testing the iso parametric mapping symbolically

From a theoretical point of view it is natural to try to prove the stability for the iso parametric mapping, because asymptotically the achievable accuracy is optimal. If one chooses Terra's grid points as images of the bilinear mapping, it also is a good approximation for Terra's elements, although the top and bottom surfaces are always plane. Since the bilinear mapping is explicitly defined by the images of the vertices of the prisms, the equivalent to matrix Fig. 2.14 can be computed symbolically. I have done this for several mappings. Fig. 2.17 and Fig. 2.16 show two examples. The result was always positive. This suggests the following strategy.

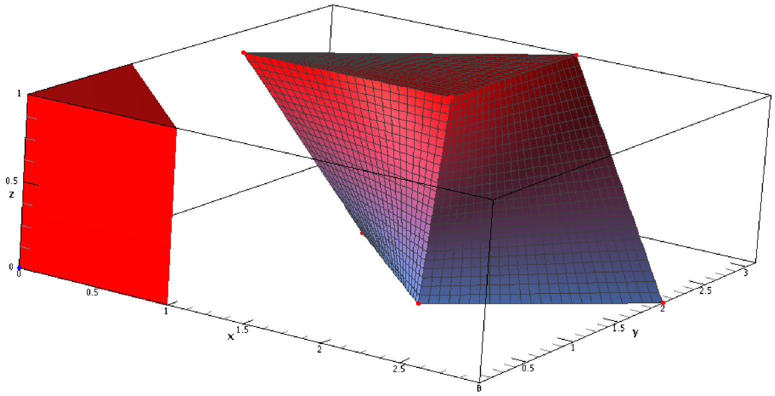


Figure 2.16: The top and bottom of the transformed prism is always plane, since the mapping is only affine in the x-y plane. The sides of the prism, which are mapped bilinearly, can be wound.

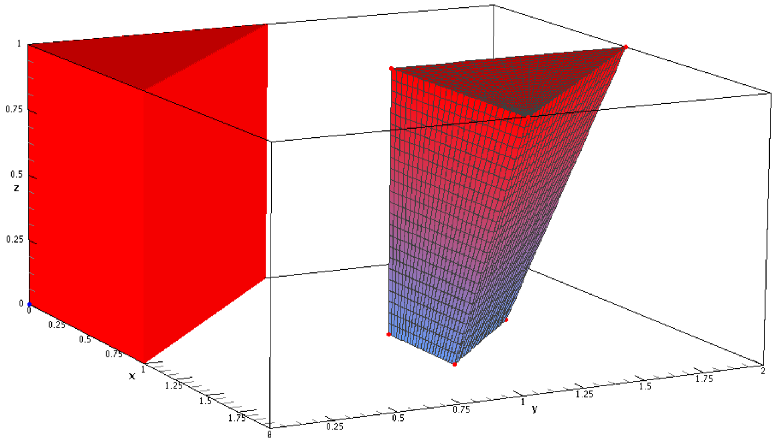


Figure 2.17: A more Terra like element

Testing the mapping numerically

As a consequence of the many successful tests even with exceptional mappings, the success for Terra's implicitly defined elements is *almost* sure. [14] To make it *completely* sure for every grid that is actually generated for a computation in Terra, I propose therefore the following procedure.

1. Instead of the reference macro compute the equivalent of the matrix (2.23) but this time with a basis of $gradQ_H$, that means with only 17 pressure functions.

---

[14] This is what one never ever expects a mathematician to say, but is in fact what many told me when I asked them after I had presented the preliminary results on a conference. Because the similarity to the Taylor-Hood element is so striking.

2. Use Terra's numerically defined operators on its numerically defined macros
3. Compute the singular values of the local discrete divergence operator $B_h|_M$ and check, if it is significantly greater than zero. The smallest singular value is the LBB for the macro.

This could be done for all the macros of the whole grid in a very short time, but due to the following symmetries it is even more simple.

1. The rank of the matrix (2.23) is invariant under a dilatation in the z or r direction respectively. Therefore the test has to be performed only for one grid layer.
2. The 20 faces of the icosahedron imply 20 identical grid patches, Therefore the test must be run only on the 20th part of a grid layer.

Hanging nodes and relation to the Taylor-Hood pair

The problem of hanging nodes has not been discussed completely yet. The reason is, that it does not exist. They only influence the accuracy of the Clement operator Π_1. If the number of hanging nodes per adjacent unrefined edge is bounded, as one would expect, nothing goes wrong. Note that this constraint is weak enough to allow several hanging nodes per edge or face, as long as their number does not increase as the mesh size h decreases. Looking at Fig. 2.11 one recognizes that the local LBB is solely ensured by the velocity basis functions with support inside the macro. The hanging nodes along an outer boundary are not important. The values are set according to the continuity constraints for pressure and velocity, not bothering the velocity of the bigger macro. For the smaller macros at the border of a big one the values at the boundary are equally unimportant.

This is a very useful observation, because it can be extended e.g. to the Taylor-Hood element. For this element the Fortin operator is constructed explicitly, which is in fact the standard procedure for many finite elements, for which the inf-sup stability can be proved. While we are satisfied when the rank condition makes sure that we *could* find the values of the dof assuring the local LBB, the Taylor-Hood proofs found in [61] and [24] actually do it. A proof for the fife and six prism macros probably would look similar [15].

The exiting similarity is that in the Taylor-Hood proofs also macros are used and the local fulfillment of the LBB is achieved by functions with support inside the macro. These functions are not affected by hanging nodes on the boundary of the macro. The only assumption needed is that the refined grid can be decomposed into macros again. This may prove very useful if one ever wants to build a grid

---

[15] Some differences would remain, because these proofs use some properties that our mapping does not provide, e.g. the iso parametric mapping for a Taylor-Hood element is affine as well as the pressure approximation. The constant pressure gradient remains a constant function under the mapping. This is heavily used in the proofs which are therefore not *easyly* extended even to the bilinear case.

with Taylor-Hood elements and hanging nodes.
Since we now have all the facts needed to understand how the Fortin operator, build by the macro-element technique, ensures the LBB we present the promised proof of theorem (2.22).

2.4.6 From the Fortin operator to the LBB

1. Let $(\mathbf{v}, p)$ denote the solution of the Stokes problem. Assume that we have a Clement interpolation operator Π_1. The operator $\Pi_1\mathbf{v}$ fulfills the approximation property

$$\sum_K h_k^{2r-2}|\mathbf{v} - \Pi_1^2\mathbf{v}|_{r,K} \le c\|v\|_{1,\Omega} \quad r = 0, 1 \tag{2.24}$$

   but is not divergence preserving with respect to the discrete pressure basis. The residual vector $\{r_i\}$ of the restriction of $\mathbf{v}$ to a macro is given by

$$r_i = \int\limits_M (\mathbf{v} - \Pi_1\mathbf{v})\nabla p_i dx \qquad\qquad i = 1\ldots \mathrm{dim}(gradQ)$$

2. To ensure the divergence preserving property of the Fortin operator Π we must be able to find a correction $\mathbf{w_h} \in V_{h,0}$ that *exactly* satisfies

$$\int\limits_M \mathbf{w_h}\nabla p_i dx = r_i$$

   That we always can find this, is ensured by the rank condition of the matrix Q with

$$Q_{i,j} = \int\limits_M \mathbf{v}_j\nabla p_i dx,$$

   with $\{\mathbf{v_j}\}$ a basis of $V_{0,M}$, and $\{gradQ_j\}$ a basis of $gradQ_H|M$. In other words we are able to construct an operator Π_2 that satisfies

$$b(\Pi_2\mathbf{v} - \mathbf{v}, q_h) = 0 \qquad \forall q_h \in Q_h$$

3. To establish the continuity of Π we have to estimate Π_2. What we need is this:

$$\|\Pi_2(I - \Pi_1)\mathbf{v})\|_V \le c_2\|\mathbf{v}\|$$

   We can do this using the SVD of Q especially the smallest singular value β_M. This gives us $\|\mathbf{w}\| \le \frac{1}{\beta_M}$. Since $\mathbf{w}$ is already estimated by (2.24) we have a Π_h with

$$\begin{cases} b(\Pi_h v - v, q_h) = 0, \forall q_h \in Q_h, \\ \| \Pi_h v \|_V \le c \| v \|_v \end{cases}$$

A short summary of how this ensures the LBB can be found in [24] page 58 and, simplified for our purpose, reads like this:
We start by substituting a discrete pressure function in the continuous LBB which gives us

$$\sup_{w \in V} \frac{b(w, q_h)}{\|w\|_V} \geq \beta_W \|q_h\|_{M_h}$$

and obtain the discrete LBB by the help of this and Π_h.

$$\begin{aligned}
\sup_{v_h \in V_h} \frac{b(v_h, q_h)}{\|v_h\|_V} &\geq \sup_{w \in V} \frac{b(\Pi_h w, q_h)}{\|\Pi_h w\|_V} && (2.25)\\
&= \sup_{w \in V} \frac{b(w, q_h)}{\|\Pi_h w\|_V} && (2.26)\\
&\geq \sup_{w \in V} \frac{1}{c} \frac{b(w, q_h)}{\|w\|_V} && (2.27)\\
&\geq \frac{\beta_w}{c} \|q_h\|_M && (2.28)\\
&\geq \frac{\beta_w}{c} \|q_h\|_{Q/KerB^t} && (2.29)
\end{aligned}$$

2.4.7 Direct proof of the LBB for an alternative grid

Bubbles, hanging nodes, and discontinuous pressure

If our point of departure had been a grid without a huge code attached to it, and if we had had the freedom to change the grid to be stable in the sense of the LBB, we could have achieved this much more easily. A standard procedure in such a case is to enrich the velocity space by functions that allow to control the flux through the faces of the grid cells [16]. See e.g. [23, 24, 61, 16, 32, 42]. To avoid the influence of these functions on the other dof[17] they are defined in such a way that they vanish at the vertices, so that their support is restricted to that face, they control the flux through. Therefore these functions are called bubble functions. Since for the purpose of flux control only the normal component with respect to the cell face is important, one usually adds only a degree of freedom for this component. However, there is no harm if the grid provides more than these. This is used e.g. in [42] for the proof of the LBB of the Q_k, P_{k-1}^{disc} element and for the low order pair with $k = 2$ by Fortin in [23]. It is also possible that the bubble functions are contained in a V_h created by a grid refinement.
However, since these dof inhabit the faces of the grid cells they are subject to the averaging enforced by the continuity request of the velocity, if hanging nodes are present. The situation is then much more intricate than for the inner-node macro-element technique of Brezzi and Fortin [24], that we used for the bilinear pressure approximation in the last subsection. But I found a remedy in [32]. The trick is

---

[16] or through the edges in the 2D case

[17] and so to preserve the orthogonal control in the sense used in information science [35]

that for a *1-regular* [18] refinement in the sense of [32] the hanging fine nodes are geometrically matched by a dof of the bigger cell. It turns out that the flux through the union of the faces of the fine grid neighbors can be controlled by the faces they share among themselves, which are not subject to the continuity request at the face to their unrefined neighbor. This is exactly the situation that arises if we refine our velocity grid.

Terra's relation to the Q_2, P_1^{disc} element pair

If we refine the velocity grid, combine two prisms, and allow the pressure to be a linear function on this macro, discontinuous at the faces, then the resulting macro is very similar to the Q_2, P_1^{disc} element. In fact the reference macro has the same dof in both cases. The only difference is that we do not use piecewise triquadratic functions but bilinear functions on a refined grid. C.f. Fig. 2.18. The technique of the proof of the LBB for the Q_2, P_1^{disc} element pair relies on the fact that the bubble functions must be contained in $V_{h,M}$. For the kind of bubble functions used in the proofs found in [61, 42] this cannot be achieved with bilinear functions for $\mathbf{v}$ on the spherical prisms, so if we want to use the same technique, we have to use triquadratic functions on the hexahedrons, build from two prisms.
We give an outline of the proof found in [61]. It uses another macro-element technique, namely the one of Boland Nicolaides [11]. Accordingly it is sufficient to show the LBB globally for a subspace of V_h and piecewise constant pressure and locally for the desired pressure approximation on the macros. For piecewise constant pressure the discrete divergence constraint reduces to the mass balance equation per macro. In the incompressible case this means

$$\int\limits_{\partial M} \mathbf{v} \; dS = 0 \tag{2.30}$$

This has been proved for the Q_k, P_{k-1}^{disc} element in [32] for the mentioned *1-regular grid* and can be used without modification. This shows, as a byproduct, that we have stability for the refined double prism (with triquadratic velocity) and piecewise constant pressure on the double prism. [19] To extend the proof to a linear discontinuous pressure a local inf-sup condition must be fulfilled. C.f. H4 in [61] page 130.

Theorem 2.3. There exists a constant $\lambda > 0$, independent of h and r, such that:

$$\sup_{\mathbf{v}\in V_h} \left[\left(\int\limits_M \frac{q\nabla\cdot\mathbf{v}}{|\mathbf{v}|_{1,M}} \right) \right] \geq \lambda \|q_h\|_{0,M} \quad \forall q_h \in Q_h|M \tag{2.31}$$

---

[18] which means that adjacent cells differ only by one level of grid refinement

[19] However it does *not* mean, that an existing Terra version, called Monash Terra, that uses bilinear (unrefined) velocity and piecewise constant pressure on the prism is stable. In fact this is unlikely, because of the instability of the Q_1, P_0 element. See [24, 61] for instance.

It is shown in [61] that one can take

$$\mathbf{v}|_M = \Pi_{i=1..8}\lambda_i \nabla p|_M$$

with the barycentric coordinates λ_i.
That the grid can be decomposed in double prisms is obvious. The extension of the proof for the reference macro to the mapped macro is possible since Terra's mapping fulfills the non-distortion assumption (10) in [42] [20] if the grid is sufficiently fine, which is the case for every grid of practical interest. This shows that a Q_k, P_{k-1}^{disc} discretization build from Terra's combined prisms fulfills the LBB for $1-regular$ grids with hanging nodes. This observation is very interesting if one ever wants to change the velocity discretization in Terra, because it connects Terra's grid with the most popular LBB-stable finite-element pair.

From Q_2, P_1^{disc} to $Q_{1_h}, P_{1_{2h}}^{disc}$

Since we know that there is a stable discretization that uses exactly the same velocity dof as our refined bilinear velocity approximation, it is an interesting question, if we could just take this existing velocity discretization and show that we can fulfill the LBB with it. To do so we summarize the crucial points of the proof for stability of the Q_k, P_{k-1}^{disc} element.

1. The flux through the faces is controlled by functions that are different from zero only on one face.
   This is done by the dof of the mid-face nodes.
2. The local inf-sup condition is ensured by a function that vanishes on the faces and is one at the center of the macro.
3. The functions used to achieve points 1 and 2 are contained in $V_h|M$.
4. Hanging nodes can be coped with by functions controlling the flux trough the faces of the refined macros.

To see, that all this can be done with our refined velocity grid, have a look at Fig. 2.18. The basic idea is to construct the equivalents to the bubbles from the functions contained in $V_h|M$. We use the basis functions, attached to the red points to control the flux through the faces. To fullfill condition (2.31) we do not use the product of the barycentric coordinates but the bilinear basis functions belonging to the center node. At this point we set $\mathbf{v} = \nabla p$ as previously. Let us call this element pair $Q_{1_h}, P_{1_{2h}}^{disc}$.

---

[20] The condition is that the deviation of the mapping from the affine is small enough in the sense

$$\gamma = sup_{\hat{x}\in\hat{M}} \|B_K^{-1} E_K(\hat{x})\| < 1$$

where B_K is the affine part of the transformation and E_K the "rest". $\hat{K}$ is the reference element.

2.5 Discussion

2.5.1 Stability for the general problem

We have shown the extensions necessarry to incorporate either variable density or the slip boundary condition. The conditions ensuring both of them are subject to further investigations and not yet proved, for any FEM discretization.

2.5.2 Standard Stokes stability

We have found three possibilities to reuse the existing velocity grid with a modified pressure approximation. Two of them can be proved to be inf-sup stable. For the third one at least a test for every given grid can be performed.

Q_k, P_{k-1}^{disc}

From a theoretical point of view the Q_k, P_{k-1}^{disc} element is most appealing, since it provides a second order velocity approximation and can be shown to be inf-sup stable for any grid that can be composed of double prisms, in advance. This reamains true even if hanging nodes are present as long as the refinement is $1 - regular$. The approximation properties are also good. We get e.g. conservation of mass at (macro) element level. For an $n \times n \times n$ grid we have $4(n-1)(n-1)(n-1)$ [21] dof. Surprisingly this means that the linear approximation provides more dof for n large enough, and is therefore probably more accurate, than the bilinear one. This of course a consequence of the abolished continuity constraint.
However this discretization does not resemble anything already implemented in Terra. The only things that would remain unchanged are the dof and their position. The operators for velocity as well as for pressure would have to be derived anew, and so would all routines that use them.

$Q_{1_h}, P_{1_2h}^{disc}$

Less interesting from a theoretical point of view but preserving more properties of the original scheme would be the use of the $Q_{1_h}, P_{1_2h}^{disc}$ element pair. The operators containing only velocity basis functions could be reused but modifications would still be necessary for all operators operating on the pressure basis functions. On the other hand we get only first order accuracy for the velocity with the same dof. However the stability properties are unchanged.

---

[21] four dof per double prism, three for the gradient of p and one for the piecewise constant part

$Q1_h, Q1_{2h}$

From a practical point of view, the equal-order different grid-size method $Q1_h, Q1_{2h}$ is the most interesting one, since the necessary changes in the code are much smaller than for the other schemes. This grid has the additional benefit that its stability is not endangered by any number of hanging nodes per unrefined grid cell. The continuity constraint takes care of those automatically. This is a (small) advantage if some parts of the domain must be refined extensively.

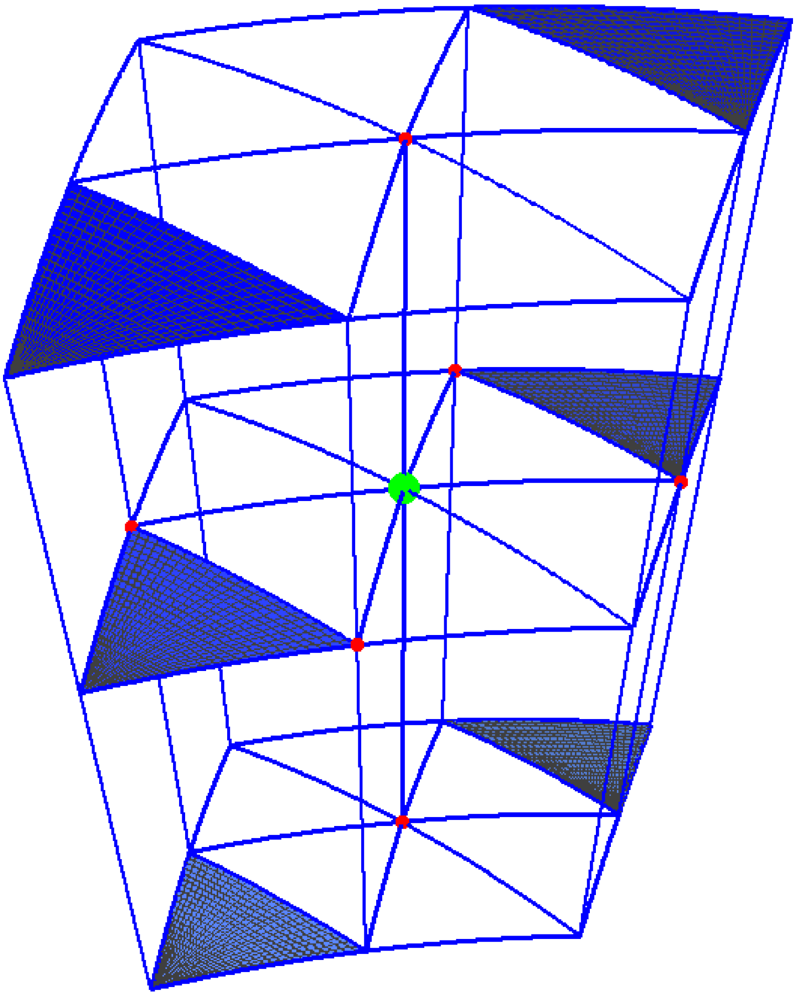


Figure 2.18: Refined double prism. Note that this macro has the same dof as the quadrilateral Q_2, P_1^{disc} element pair. The dof at the red points can be used to control the flux through the faces. This is necessary to fulfill the global LBB for discontinuous piecewise constant pressure. The dof at the green point can be used to fulfill the local LBB for the piecewise linear pressure. Unlike the Q_2, P_1^{disc} element pair it is not possible to define a bubble function which is different from zero on the *whole* top face and zero on all other faces. The same is true for the bottom face. The difficulties stem from the filled triangles. A piecewise linear function that is zero on the boundary of the whole quadrilateral is bound to be zero on the filled triangles. Nevertheless it is possible to control the flux through the whole face by the dof situated at the red point in the center of the face. The “bubble” is just a“tent” which is “flat” on the filled triangles.

Chapter 3

Multigrid-test framework

3.1 Motivation

The subject of this chapter is the robustness of the multigrid solver with respect to strong variation of viscosity. Literature concerned with strongly varying or even discontinuous parameters in a general way has long been available. See for instance: [29, 70, 20, 36, 39]. However, the authors do not focus on our special physical problem and its numerical implications.
On the other hand there is also a number of papers dealing with exactly the same physical facts, as we have seen in the introductory chapter. We note here for instance: [1, 54, 55, 56] and the references therein.
But in this kind of work the numerics are only briefly discussed. Additionally the numerical methods in use differ considerably in many aspects, so that results cannot be transfered directly. It is for instance not easy to find out in which way a special decision influences the overall performance.
The task to improve the robustness of our scheme therefore can be accomplished neither by deduction from general theory nor by copying an already existent technique. Nevertheless it is possible to extract some promising common ideas. But these ideas have to be tested. For this purpose, I present a test framework, which can give approximate answers in a short time. It would have been impossible to implement all the ideas presented in the sequel in Terra.
I found the work of Schmachtel [49] particularly helpful, because it describes many different possible improvements and their influence on the problems at hand. Although used in a totally different solver scheme, at least the discretization with finite elements is similar to our setup. Additionally I have recently found in [50] that the semi coarsening multigrid method, which can be seen as a special case of a recursive domain decomposition method, can be applied successfully for elliptic problems with discontinuous coefficients. Since our problem is elliptic this is very interesting. Summarizing I note the following experiences, which are important for us:

1. Block smoothers can achieve much better performance than their non blocked

counterparts if parameter jumps occur.

2. Boundaries of domains of different viscosity should not be matched by the block boundaries of block oriented smoothers. Therefore adaptive blocking is advised.

3. The grid transfer operators should be dependent of the viscosity contrast of adjacent elements. Especially for large contrast injection is preferred over full interpolation.

4. Semi-coarsening might be a cure for very hard prolongation issues.

These are the guidelines for the new scheme that will be briefly described now. I will neither describe the general concept of smoothers nor multigrid but only refer the reader to [29] or [33].
Since Terra is very efficiently parallelized using a domain decomposition approach we are bound to use block Jacobi smoothers. [1] The numerical cost and therefore the time for the inversion of the block matrices is dependent on the block size. So it is not possible to use adaptive blocking without disastrous influence on the load balancing. On the other hand the implementation of an adaptive load balancing for instance by means of space-filling curves is far too complicated to be entirely new developed for the present code. In this case one would rather use an existent framework and write an entirely new code. A possible remedy is to use different block smoothers with blocks of the same size but shifted or staggered position. The different blocking schemes result in different approximate inverses which are consecutively applied. In pseudocode the smoother for three different blocking schemes looks like this:

---

[1] This is true if the alternative is global (block) Gauss Seidel, because in the latter case the ordering of the blocks is important. It is however possible to apply Gauss Seidel for all blocks of the sub domain and Jacobi for every sub domain-block. I would rather call the resulting smoother improved Jacobi instead of hampered Gauß Seidel.

```
while$|res| > tol$ do
    $res = Ax - b$
    $cor_1 = {A_1}^{-1} res$
    $x = x + cor_1$
    $res = Ax - b$
    $cor_2 = {A_2}^{-1} res$
    $x = x + cor_2$
    $res = Ax - b$
    $cor_3 = {A_3}^{-1} res$
    $x = x + cor_3$
end while
                                                    (3.1)
```

The idea for this procedure was induced by a scheme for the solution of the convection diffusion described in [33] on page 191. There four different Gauss Seidel smoothers are applied with four different numberings of unknowns from right to left, top to bottom, left to right and bottom to top. Every smoother is optimal for certain parts of the grid, where the flow matches the ordering direction of the unknowns, since Gauss Seidel is an exact solver in this case. To avoid the reordering of the unknowns according to the flow direction all four smoothers are applied. This way also circulating flows are handled properly. In our case the effect is not as prominent, because even the best adapted blocking scheme does not lead to a smoother that is an exact solver. But we can nevertheless ensure that the best adapted blocking scheme for a given block size is used. It turns out that the alternating use of different schemes is also beneficial for combinations of non optimal blocking schemes. Although I did not pick up the idea from the literature, I afterwards found one example that fits in this general scheme. This is the criss-cross blocking where two or three line Jacobi methods are combined, for instance implemented in Dendy's Black-Box Multigrid [20]. We, however, not only use blocking along grid lines, which can be expected to be important in the context of multigrid. The test framework presented in the next subsection is constructed in order to check how the new scheme performs in comparison to established alternatives.

3.2 Test setup

3.2.1 Numerics and implementation

To verify the usefulness of the ideas mentioned above, I used a small test system that is much less sophisticated than the actual Terra code. It implements a small 2D example with bilinear basis functions for the velocity. This is similar to the actual 3D code Terra, that also uses this discretization for velocity. The discretization of the viscosity should fit in easily with the existing framework. So I used the same bilinear basis functions as for the velocity components. Doing this I could reuse all the functions computing the integrals for the matrix assembly. A side effect of this mere technical decision is, that we avoid discontinuities of viscosity by definition, but get very steep gradients instead. As the tests will show this turns out to be very important. This usage of bilinear, node based viscosity is a difference to Terra which uses a cell based piecewise constant approximation. As we will see it could be an important improvement. [2] The test system consists of several parts:

1. A code written in mupad, a computer algebra system,to compute the operator exactly by means of solving the integrals of the basis functions symbolically and this way check matrix properties such as rank, symmetry or definiteness for small test cases.

2. An octave code that implements a numeric solver for the same problems, because exact computation is much to expensive even for the small test problems to be presented later.

3. A small test framework that checks the numeric solver against it's symbolic counterpart and many small test cases.

3.2.2 Viscosity test fields

We have the following objectives for the viscosity fields.

1. Really challenging jumps must be tested. Therefore the test cases show a factor of 10^{14} of viscosity variation between adjacent elements.

2. The frequencies of of the alterations should be variable to test the general suitability of multigrid. Multigrid schemes are supposed to suffer significantly if important properties of the problem cannot be resolved on the coarse grid. The parameter jumps are suspected to be such an important property. We provide test cases with the smallest patch size of isoviscous sub domains possible on the grid.

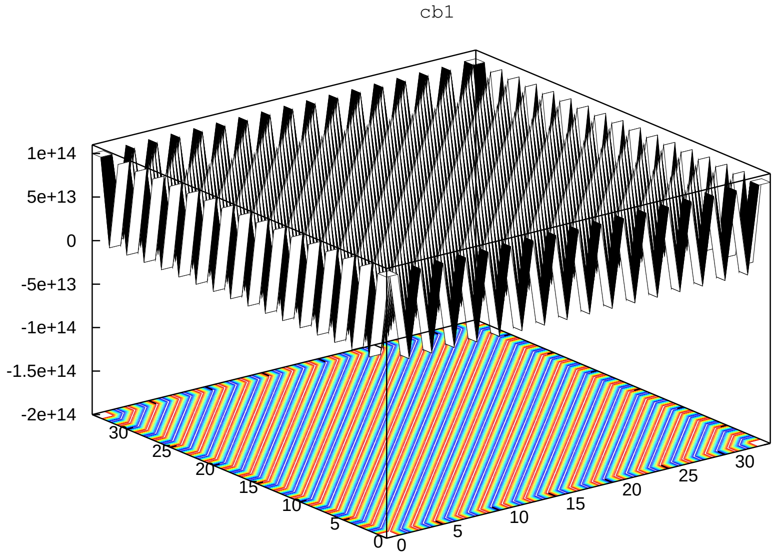
3. Since we want to test the influence of a new combined block smoother we want to provide as many traps for conventional schemes as possible. That those traps are likely to exist is indicated by the dependency of the efficiency of block smoothers on the capability to catch parameter contrast within the blocks. Especially we want to find the configurations for which the already implemented radial line Jacobi smother suffers.

---

[2] One may argue that this constrains the physical model which allows discontinuities of viscosity at phase boundaries. On the other hand the existing discretization *enforces* such discontinuities (at every grid-cell face) where the physics do not. Additionally the approximation error of the new discretization is better for continuous viscosity. So the question of the best viscosity approximation is not that easy to answer in favor of the discontinuous version.

4. Since multigrid is, this time in an unintended sense, aware of grid symmetries, the results may be corrupted if the test cases exhibit such symmetry. Therefore the symmetry is avoided by randomization.

These objectives led to the following viscosity profiles:



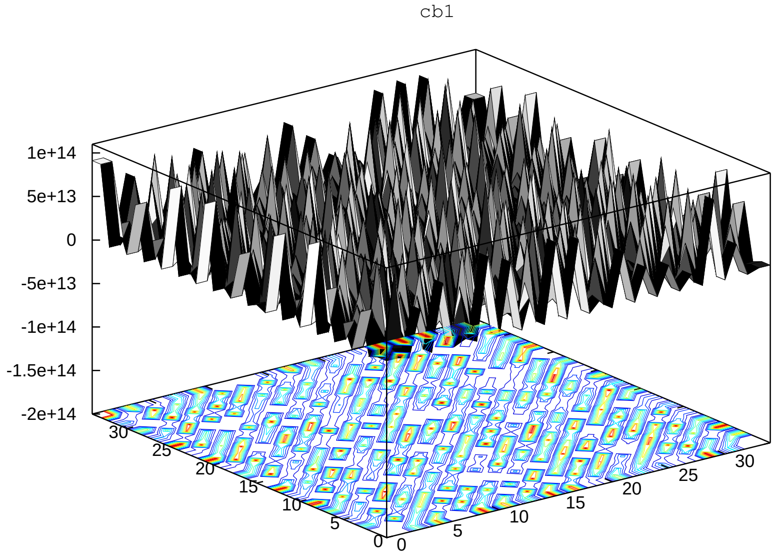
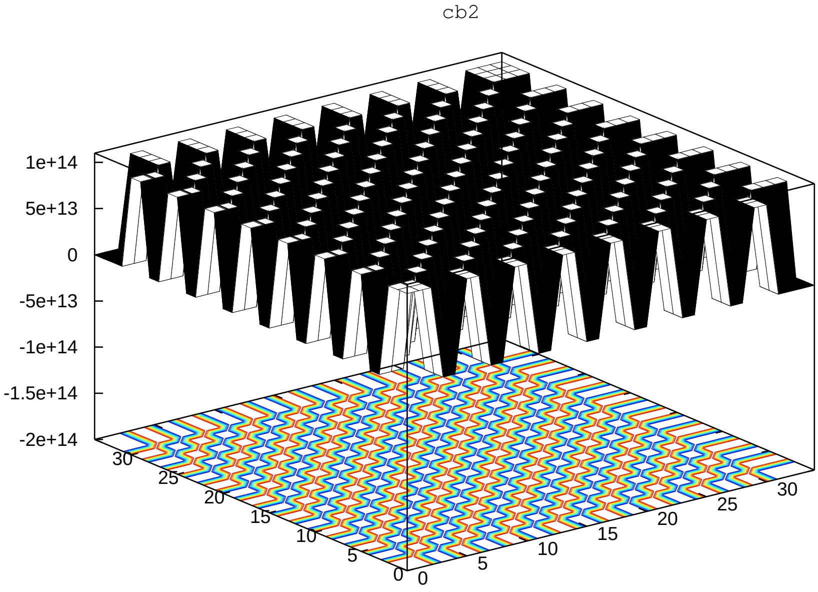


Figure 3.1: cb1 means checkerboard viscosity with patch size 1×1 As in all other cases the viscosity varies from 10^0 to 10^{14} from patch to patch. This profile is supposed to be the most unpleasant one for multigrid because a coarser grid could definitely not sample all jumps. The second plot shows the unsymmetrical version of cb1, the biggest jump is also 10^{14} but the others are randomized. The randomization was introduced because the results had been too good for this scheme. This fact, however, has not been changed by the randomization as we will see and explain later.



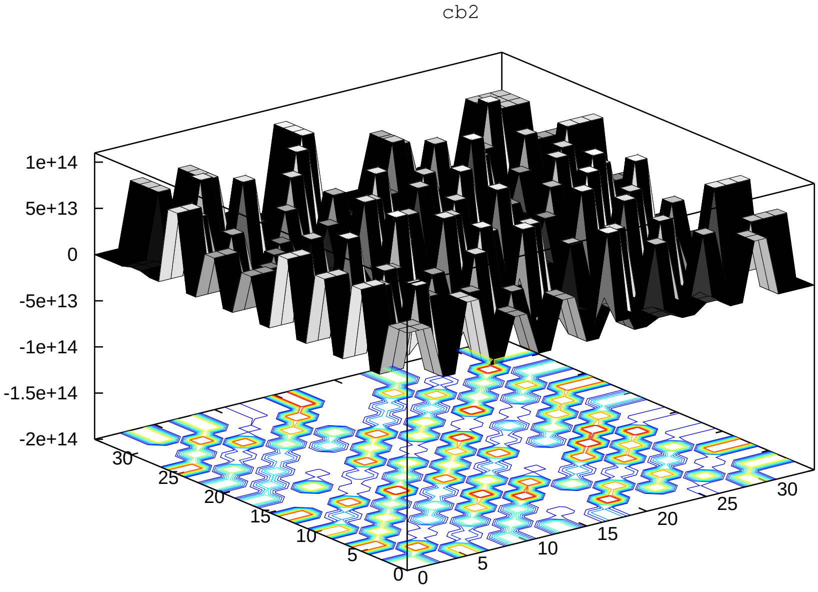


Figure 3.2: slightly increased patch-size to 2 × 2 as the index 2 suggests. Bigger patches occur at the boundary due to the odd grid-size

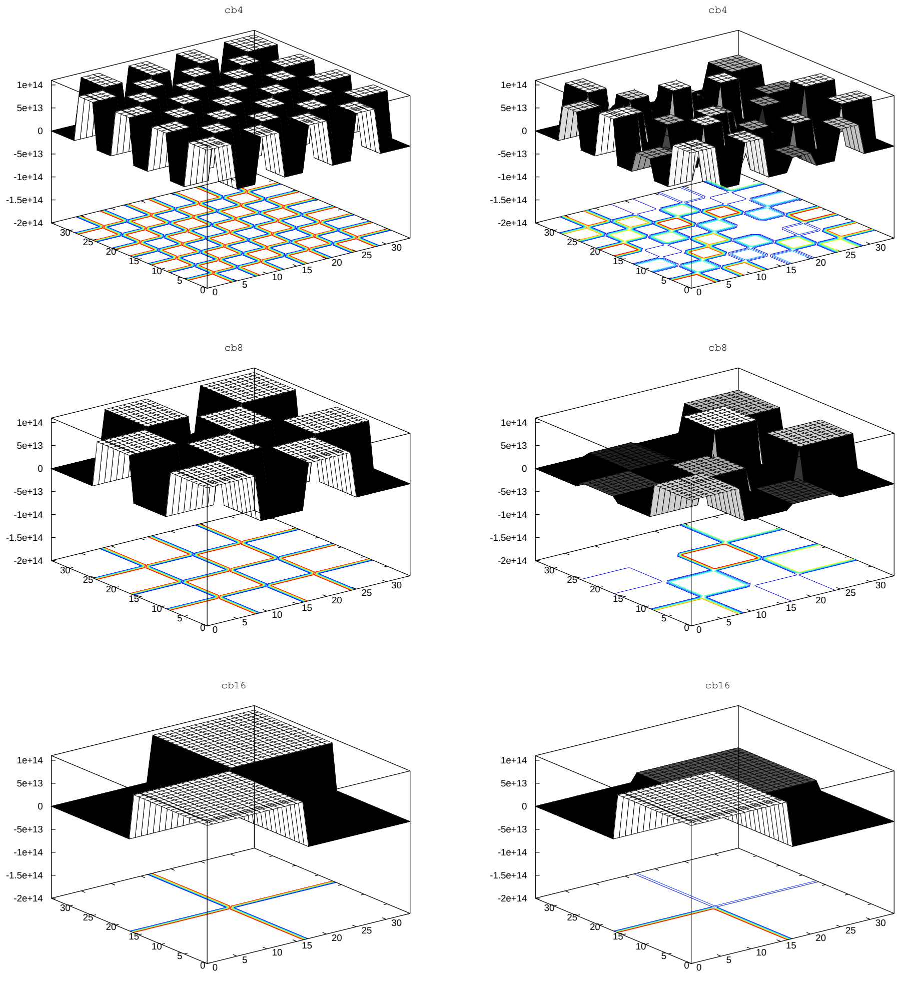


Figure 3.3: further increased patch-size to $4, 8$ and 16 with randomized versions These test cases are intended to distinguish between smoother and multigrid performance. The bigger patches should suit multigrid better.

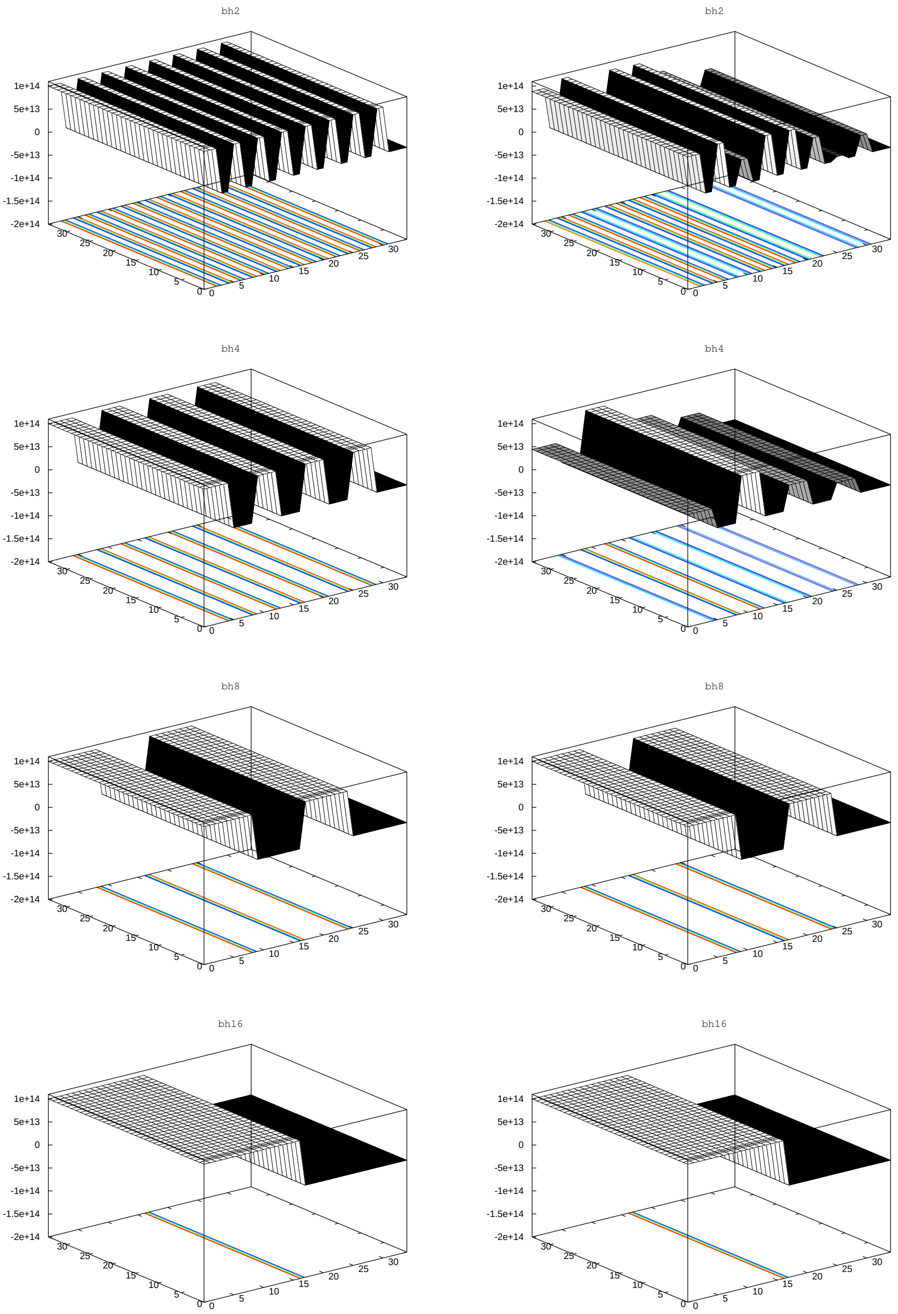


Figure 3.4: bh means banded horizontally This profile is included to have a test-case, where line Jacobi blocking (in the other direction) is supposed to be most suitable. It is also a case of high practical relevance because in the earths mantle viscosity is strongly depth dependent.

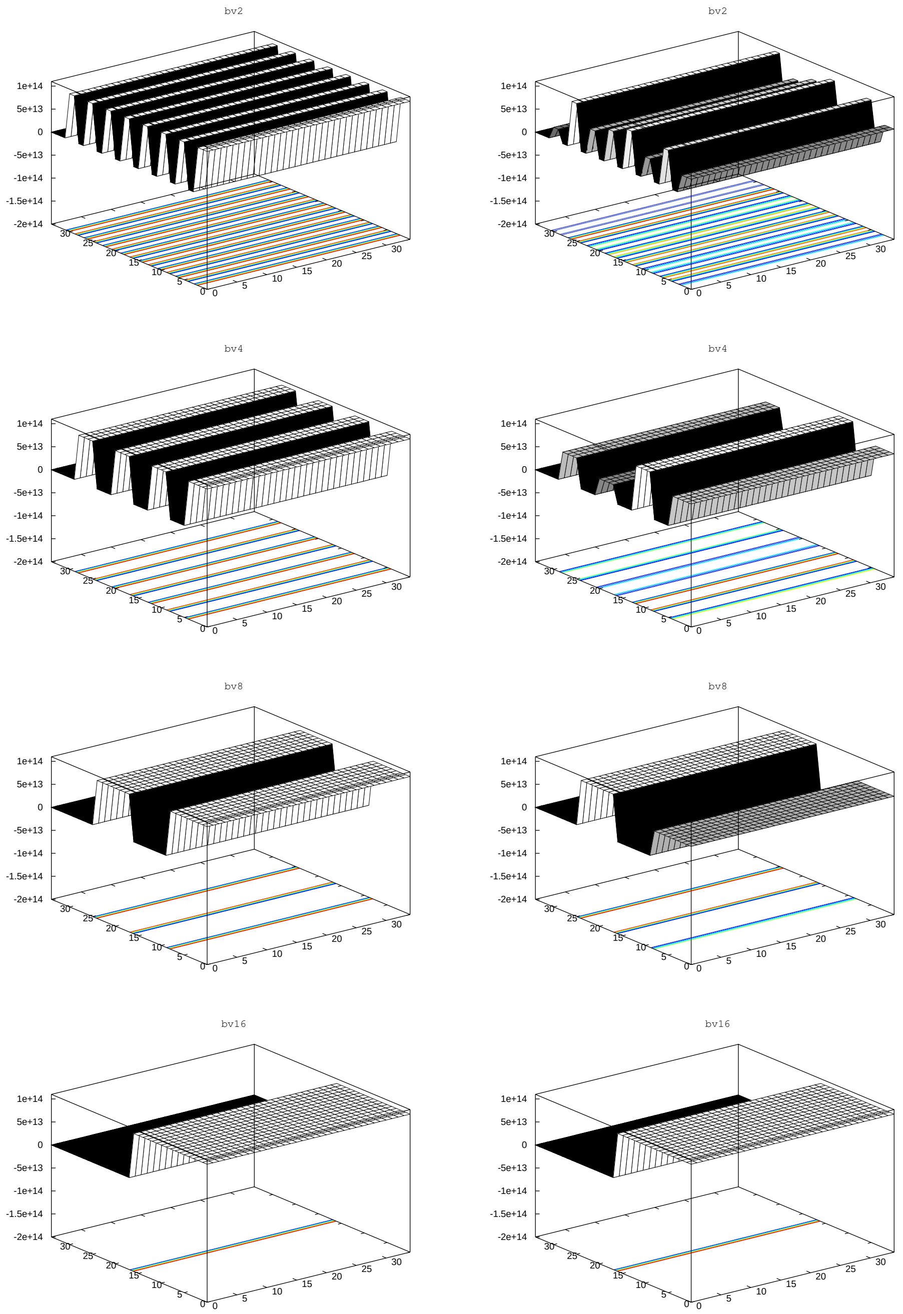


Figure 3.5: The profile bv (banded vertically) is the counterpart for bh and mainly included to be able to compare line Jacobi blocking in the other direction and for testing purposes, since the performance of the two line smoothers is expected to be symmetric.

3.2.3 Blocking schemes

From a matrix centered point of view the dof for all grid-points are successively stored in one vector. To find the position of the dof of one particular grid point in this vector one can imagine a thread running from point to point connecting the grid points in arbitrary order, with the only constraint to meet every grid point only once.

If one marks a certain grid point, straightens out the thread and counts the number n of grid-points before the marked point one can determine the rows in the vector referring to this points dof. If the number of dof per grid-point is nd they start at $n \times nd + 1$ and end at $n \times (nd + 1)$. In our 2-d example bilinear example holds $nd = 2$, representing the two components of the velocity vector at a certain grid node.

One further observes that the dof belonging to the grid point that succeeds the marked point on the thread succeed the dof of the marked point also in the vector. This way succeeding points on the thread refer to an uninterrupted part of the vector of dof. In our example two succeeding points of the thread are represented by four succeeding vector entries.

This is also true for the right hand side of the matrix equation representing the values of the application of the operator at certain grid points. The part of the matrix needed to compute the value of row m of the right hand side is represented by row m of the matrix, for successive points on the thread represented by the vector rows $m_1 \dots m_2$ we therefore need also the matrix rows $m_1 \dots m_2$.

If we only want to take into account the portion stemming from the dof represented by the rows $m_1 \dots m_2$ we actually only have to consider the columns $m_1 \dots m_2$ of the matrix. So in this case only the small diagonal *block* of the matrix is needed, giving the name to a whole class of iterative solvers, which approximate the actual operator by its block diagonal.

These solvers are based on the assumption that the values of the right hand side of a certain group of grid points depend mostly on the values of the solution at these points, in other words that the matrix operates locally in the geometrical sense. This assumption is especially true for operators derived from differential equations, which are, so to speak, infinitesimally local.

Up to now we have not taken into account the actual form of the thread or space-filling curve mentioned above. As already mentioned, it would be possible to connect the grid-points in an arbitrary order, jumping around trough the grid. The block size is no geometrical measure but only a counter of dof. Given a block size, that is a number of dof which form an uninterrupted part of the vector, the crucial point is to include those dof which have the largest influence on each other. (This includes inevitably the influence of the dof at a certain point on the rhs at this very point itself.)

Applying the argument of locality again one would try to make sure that points which are neighbors in the geometrical sense should also be neighbors in the vector description.

Beside locality there are other considerations that lead to refined blocking strategies. According to the above mentioned observation in [49] blocks should be chosen to catch viscosity patch boundaries. This means, in an algebraic sense, that the influence of neighbors with extremely different viscosity is larger than the influence of the neighbors with the same viscosity. It is therefore advisable to take this into account when choosing the members of a block or, more globally, the course of the space-filling curve.

Both arguments, locality and inclusion of viscosity boundaries imply an advantage of combined block smoothers. If you look at the pictures following this introduction it will be almost obvious that this is true for the latter. We therefore postpone the discussion.

The fact that combined blocking is also beneficial if viscosity is constant, can also be derived from the following pictures. In the figures adjacent points of the same color belong to the same block. That means the imaginary thread runs to all points of the given grid patch before it enters the next.

The order of the points inside a given grid patch is arbitrary as well as the order of the patches. Both have no influence on the efficiency of the blocking scheme. But every blocking scheme prefers some grid points, namely those that have all their neighbors inside the patch, the inner points of the patch. To estimate the values of the points on the patch boarders only some of their neighbors are available. The situation is worst for the corner points.

One remedy to decrease the number of those "unfortunate" points is to increase the block size, which is extremely expensive numerically, as we will see. Another one is to use different block-diagonal estimates for the operator to give every point the chance to be in the center once. So errors left over by one block smoother can be addressed effectively by another. [3]

In Fig. 3.6 the blocking scheme denoted cb1 represents a patch size of 1×1. That means that only the two degrees of freedom belonging to the two velocity components are combined, resulting in a approximate inverse with 2×2 blocks on the diagonal. This scheme does not prefer any grid points. For the next patch size of 2×2 there exist four versions

1. standard unshifted cb2
2. shifted horizontally cb2sh
3. shifted vertically cb2sv
4. shifted vertically and horizontally cb2svsh

The resulting block in the matrix is always 8×8. All points are boundary points so no point has all its neighbors in the patch. With respect to the boundaries of

---

[3] One may ask if the combined scheme will converge at all. Theoretically the convergence of the combined scheme is ascertained if the convergence of all applied block smoothers can be shown. The worst rate of convergence in the sense $\frac{||r_i||}{||r_i+1||}$ is at least as good as the worst rate of convergence of the worst partaking scheme. In all our examples it is better than the convergence rate of the best performing scheme.

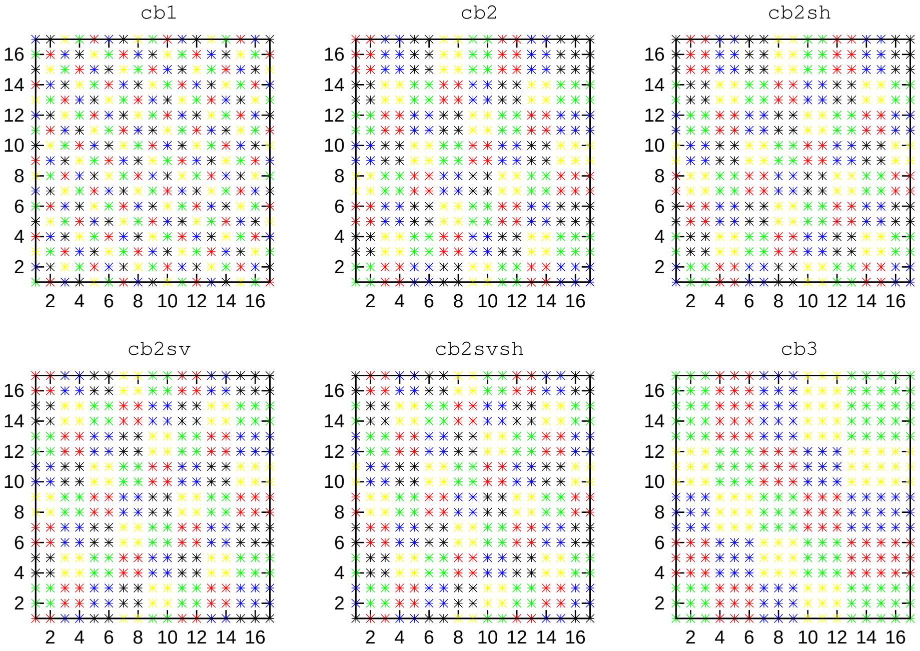


Figure 3.6: Blocking schemes checkerboard 1, all four variants of checkerboard 2, unshifted checkerboard 3

the viscosity patches at least one of the four schemes would be optimal. It is not possible to create a viscosity pattern whose interfaces are located on the patch boundaries for all blocks. This is also true for all the following combinations.

Another common feature of all schemes is that the patch size is adapted to the grid size. Due to the fact that we have a nodal based multigrid the grid size is always 2^{n+2i} in one direction. An equal block size is therefore seldom possible. A small portion of the grid is left over.

One possible solution would be to use smaller patches for this remaining nodes. But the influence of the block size might be crucial if the blocks are to small. In order to avoid this the patches on the boundaries are enlarged since it is assumed that there exists something like a saturation, where larger patches dont have any benefit any longer.

Fig. 3.8 expresses a fact, that is true for all blocking schemes. Patches can be defined across grid boundaries. The reason is that this framework is designed with a spherical domain in mind. This domain is cyclic in two out of three dimensions. In fact it would be artificial not to allow overlapping. However in the setup for the actual results this feature is not important, because I did not run the tests with a cyclic domain. The arising disadvantage for some shifted schemes is that they are not local in the above mentioned sense. The patches contain values that would be useful only if the domain was cyclic. But this disadvantage again is assumed to be smaller than the one arising from the division of the patches.

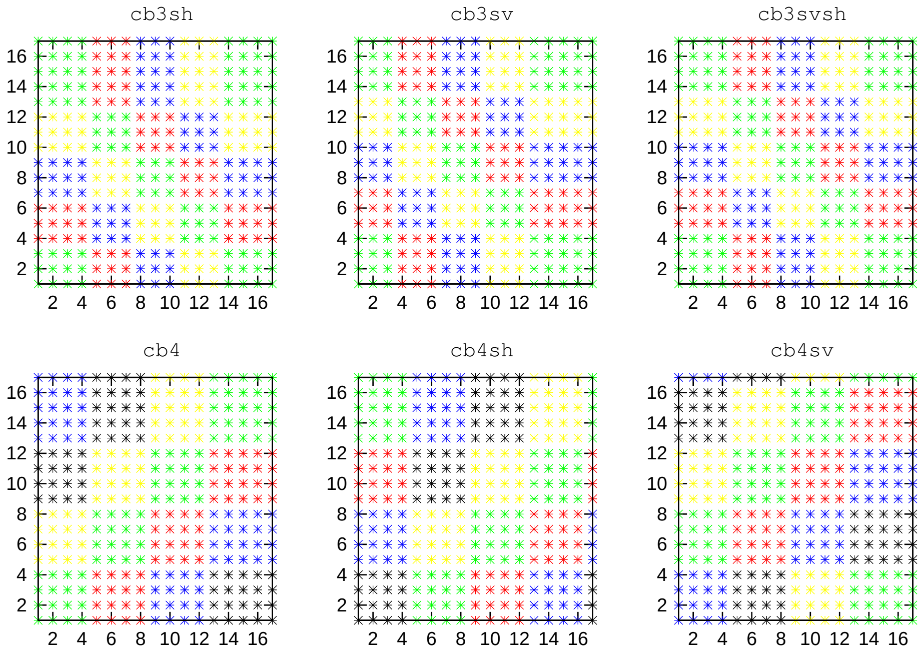


Figure 3.7: The 3×3 patches are very interesting because the center point has all its neighbors in the patch and the shifted variants ensure that every point has this benefit once. It is also the smallest configuration (stencil) that has has this property.

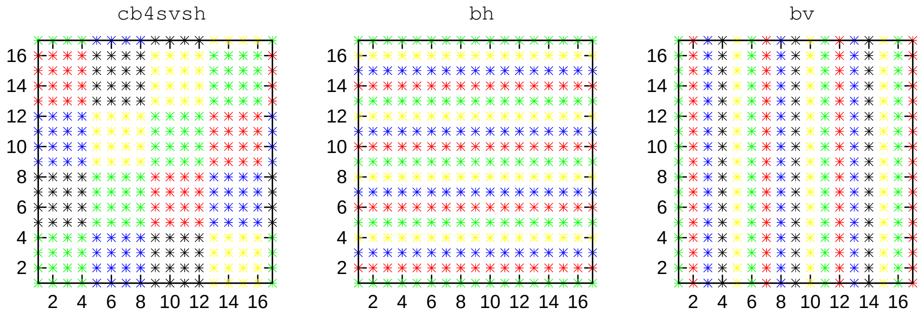


Figure 3.8: The usual line Jacobi smoothers

3.2.4 Multigrid

For strongly varying parameters all literature I know of advises the use of a Galerkin approach. This means that the prolongation and restriction operators as well as the coarse-grid-operators are not independent. The restriction operator is the adjoint prolongator. The coarse grid operator is defined as $A_{coarse} = RA_{fine}P$.
I chose to define the prolongator, because this does not involve a special treatment of the boundaries and is more easy to understand. I implemented two different prolongators. The first one is the canonical prolongator that represents a coarse grid function exactly on the fine grid. Points present in the coarse grid are injected. All other points are interpolated bilinearly, which for the bilinear basis functions is the said canonical inclusion.
The second prolongator is defined according to the suggestion in [49] where injection is also proposed for points inhabiting the nearest neighborhood of a viscosity contrast. This can be accomplished easily by weighting the above mentioned bilinear interpolation with a (coarse grid) viscosity field, as suggested by Baumgardner (personal communication). The value of at a midpoint is computed as follows. $v_m = \frac{\nu_l v_l + \nu_r v_r}{\nu_l + \nu_r}$ If the viscosity ν_l at the left point is several orders of magnitude bigger than the viscosity ν_r at the right point, this is practically injection of v_l. If on the other hand for $\nu_l = \nu_r$ the scheme falls back to canonical inclusion. This way the scheme is adaptive automatically. [4]

3.2.5 Refined expectations

This subsection is intended to make some forecast of the results in an heuristic kind of way. As the results will show some of these heuristics are actually drastically misleading. nevertheless they appear reasonable enough beforehand. I will formulate them in the following assumptions:

1. For a horizontally banded viscosity a vertically banded blocking scheme is most appropriate for the following reasons.
    (a) All viscosity interfaces are caught inside the patches. The patches are even uniform.
    (b) The method becomes very expensive as the grid-size increases, but is used nevertheless in a variety of cases for instance in an important paper [20]. [5] Regarding the rigorous performance competition, there must be some good reason to pay that price.
2. The most simple scheme that just blocks the two dof of the velocity for one grid-point is expected to perform poorly for every non-iso viscosity-field.

---

[4] It is to be mentioned, that a restrictor based definition of this procedure is even more promising since short wave length viscosity contrasts are better represented by a fine grid viscosity field. It is a little bit more difficult in this case to produce a meaningful prolongation on the boundaries. Due to the available time, I have postponed this.

[5] which is by the way worth reading for the wit of its author and the resulting fun of it alone.

This is suggested by its disability to handle the strong coupling between neighboring grid points induced by the viscosity-contrasts.

3. For constant viscosity the said inferiority will be much less prominent but nevertheless present, due to the next two assumptions.
4. Bigger patches perform better, because more coupling can be attended to.
5. Combined blocking will perform better, especially the combination of schemes cb3 and cb4, for the above mentioned reasons [6].
6. Multigrid performance will deteriorate if the isoviscous patches become to small. Weighted prolongation will amend this partly. The worst performance is expected for the smallest isoviscous patches.

3.3 Results

3.3.1 Mode of presentation

I will not restrict myself to convergence-rates. This is justified by the amount of information that can be obtained from the plots. I will give an example first. Fig. 3.9 and 3.10 show the comparison between the history of the residual norm and the error norm. Although the two plots look very similar there *are* some interesting differences.

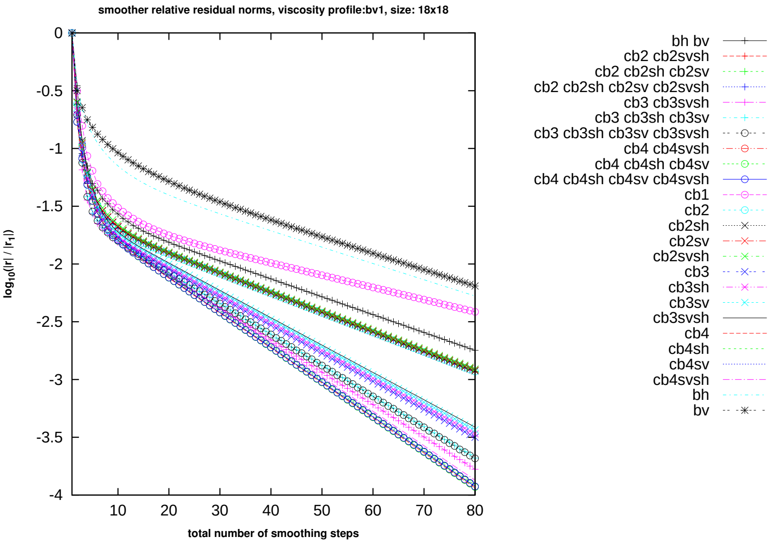


Figure 3.9: Residual history. Please compare with Fig. 3.10! Note that residual norms may be misleading!

---

[6] and because one cannot be expected to be pessimistic about ones own ideas

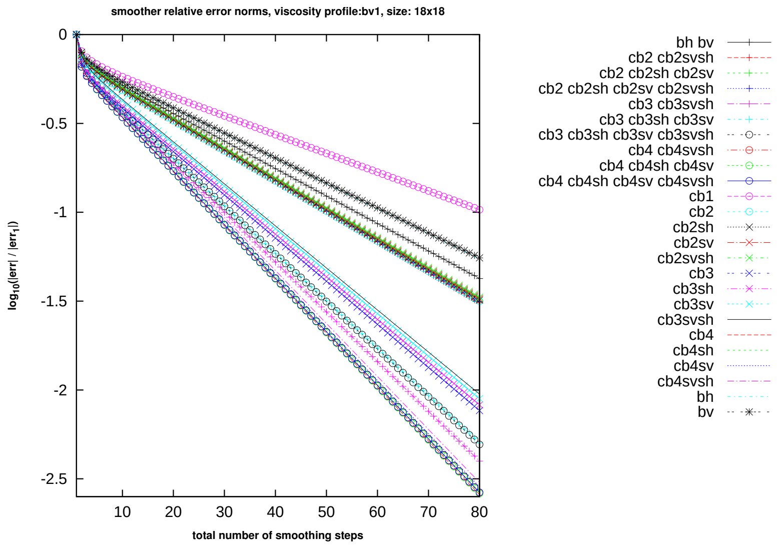


Figure 3.10: Error history. Please compare to Fig. 3.9!

1. The residual plots seem to indicate that the two line Jacobi variants (bv, banded vertically, and bh, banded horizontally, are the worst solvers. But this is not true as the error plot indicates. They certainly handle errors belonging to small eigenvalues better than cb1 (the most simple block smoother).

2. It is a well known property of Jacobi [7] to be a smoother, a solver that handles high frequency errors properly but is slow to remove low frequency errors. This fact is expressed by the residuals very strongly but is much less prominent for the errors. Obviously the errors belonging to huge eigenvalues are removed first. This is not at all surprising, if one remembers, that these iterative solvers compute the correction for the actual step from the residual of the last one. I note it merely to emphasize the fact that we have to deal with matrices with big spectral radii, that means, with ill conditioned systems.

3. Due to the same reason the residuals can be reduced faster than the errors. This is also apparent in the plots.

According to these observations I decided to base the whole discussion on the error plots. The error is computed as follows. A random solution is given, the appropriate right hand side is computed. The solver starts with an initial guess of 0. To achieve a fair comparison the x axis counts the total number of smoothing steps. A combi scheme like cb3 cb3sv cb2sh needs three smoothing steps for one cycle. The residual norm is plotted for cb3, cb3sv and cb2sh. The 80 Iterations indicate $\left[\frac{80}{3}\right] = 26$ cycles.
We proceed with an example of the results for the two multigrid variants. See

---

[7] and also block Jacobi

Fig. 3.11 and Fig. 3.12 . The error is again plotted against the number of fine-grid smoothing steps. To include the combi schemes in a fair way, the number of pre and post smoothing steps must at least be four since the largest combinations include four different smoothers. It follows that a complete v-cycle needs 8 (fine grid) smoothing steps. The example result suggests that this is suboptimal for most of the schemes, because the drop between pre and post smoothing indicates that one would rather use the computing time consumed by the smoothing steps for more multigrid-cycles. The only schemes that perform poorly are the line-Jacobi variants bv and bh. The reason for this behavior is obvious in this kind of plot and the motivation for this form of presentation. After a single v-cycle they seem to be unable to benefit from the coarse grid correction.

Note again the misleading residual plots.

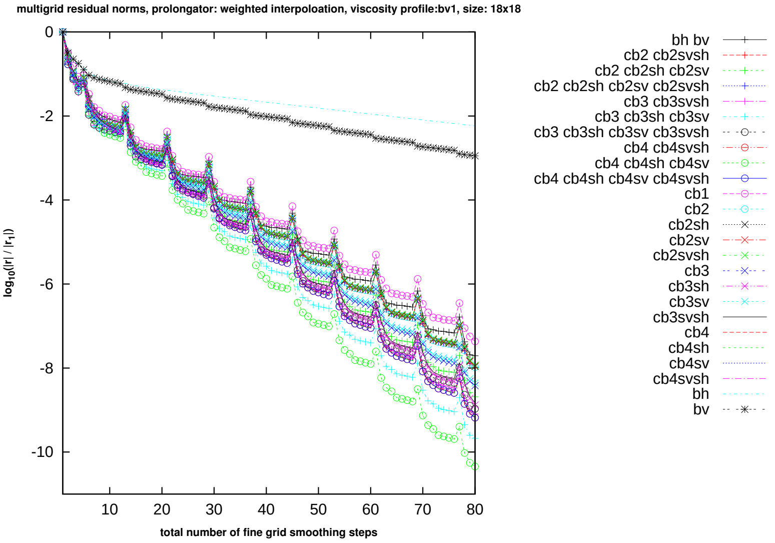


Figure 3.11: Residual history. Please compare with Fig. 3.12!

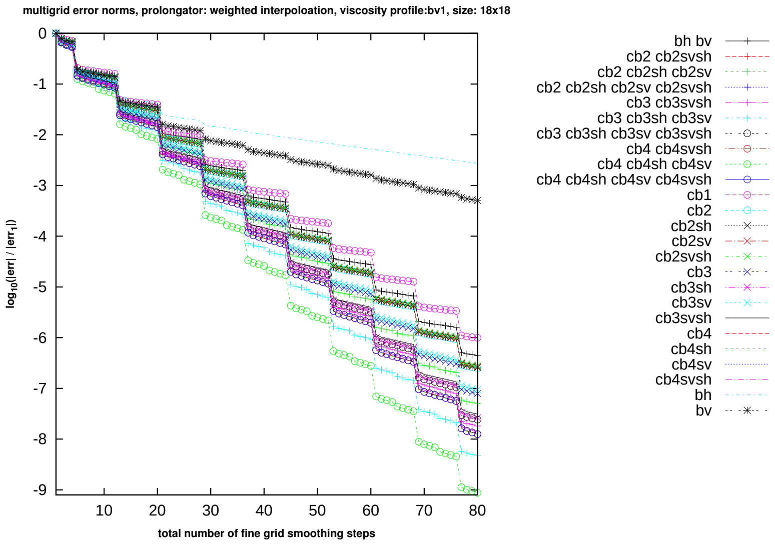


Figure 3.12: Error history. Please compare to Fig. 3.11! The error actually decreases after a multigrid step, although the residual plot seems to indicate the opposit.

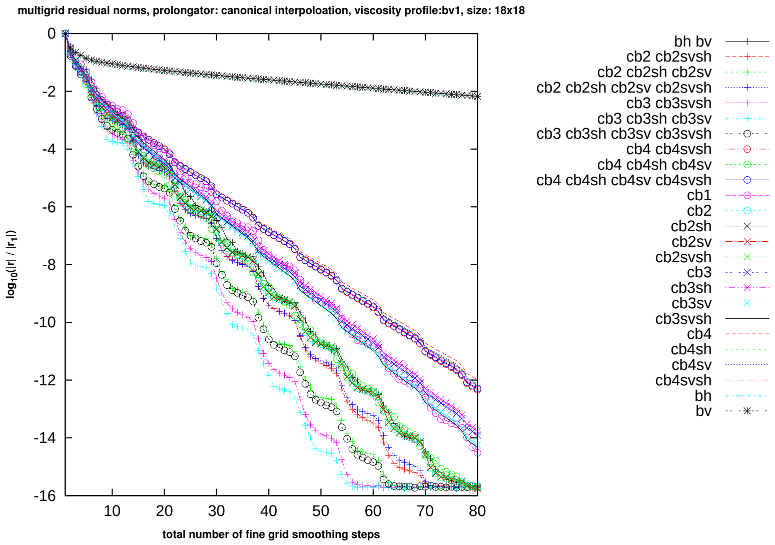


Figure 3.13: Residual history please compare to Fig. 3.14 The almost straight lines indicate that the number of pre and post smoothing steps is nearly optimal. This is, however, not true as the error history makes clear.

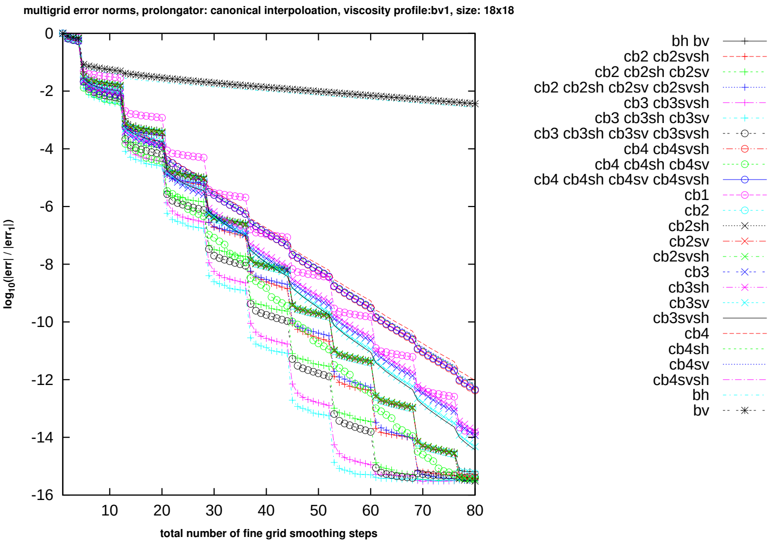


Figure 3.14: This result shows that also for the canonical prolongation 4 pre and post smoothing steps are too much. The optimal curve would be almost straight. Please compare with Fig. 3.13!

3.3.2 Comparison of convergence rates

The plots of the previous subsection contain a lot of useful information. To compare the different schemes we will reduce this information to a single number.

Smoothers as iterative solvers

We start our discussion with an analysis of the smoother as iterative solver. The convergence rate is given by the relative error reduction per smoothing step. For our test cases it has been computed as the geometric mean of three smoothing steps after 15 iterations. [8]

$$r_c = \left(\frac{|err_{15}|}{|err_{15-3}|} \right)^{\frac{1}{3}}$$

From the convergence rate one may estimate the number of iterations needed to achieve a given fixed accuracy (or relative error reduction) tol. Again every smoothing step of a combi scheme counts.

$$n \approx \frac{\ln(tol)}{\ln(r_c)}$$

We will use this estimate to arrange the plots, since it gives the possibility for an overall ranking of both the difficulty of the problems posed by the viscosity fields and the robustness of the smoothers.
Fig. 3.15 and Fig. 3.16 show the estimated number of smoothing steps to achieve a total error reduction of 10^{-10}. We note first the following points concerning the viscosity profiles.

1. To avoid misinterpretation it must be noted that every visc profile represents also a different random solution, so not all differences have a general source.

2. As expected the profiles with large iso visc patches present small difficulties to all schemes.

3. The profile cb1 is apparently not problematic for any scheme. This is less surprising for the schemes with a patch size > 1 since the grid-patches for all those schemes always capture the viscosity boundaries, which was our intention. For the 18x18 grid it is even better handled than the isoviscous profile. The same is true for bh1 and bv1.
   It is surprising that even the cb1 scheme performs not worse for the checkerboard viscosity profile than for the iso visc case.

---

[8] The smoothers are nearly indistinguishable for the first steps, so it is necessary to wait for the random part of the solution to vanish. This is common practice. It is, however, clear that we are not really interested in a Jacobi *solver*, but in its suitability as *smoother*, which probably is not applied 15 times. We will come to this in time, but draw some valuable conclusions from this results first.

4. The banded viscosity profiles bv2 , bh2 ,bv4 and bh4 seem to be the most problematic for all the smoothers. This is true for all of the presented grids although the random solution was different. We will see this again in the multigrid results, so its worth noting that already here a (very small) problem is present.

The more interesting part of Fig. 3.15 and Fig. 3.16 is of course how the solver performance differs. [9] We note some interesting points:

1. For the checkerboard schemes block size is advantageous. The best performing scheme is about 2.5 times faster than the worst (and cheapest) one. One reason for this behavior beside our intended capturing of jumps is probably that the smoothers are (mis)used here to correct low frequency (global) errors and the reach of the big-block smoothers is simply broader.

2. The line Jacobi smoothers perform badly although their block size is greater than that of any other block smoother for grid sizes greater than 16. This will become more pronounced in the sequel.

3. The influence of the combined blocking increases with decreasing block size. For the cb4 variants the best fitting scheme performs always better than any combination. For cb3 some combinations are better. For cb2 combination is always beneficial. This is understandable since the smaller patches have more suboptimal treated boundary points than the bigger ones. They therefore profit more if these points are treated better by another blocking scheme.

4. The capturing of viscosity jumps inside the blocks has no great influence, as a look at the cb2 and cb4 schemes shows. The unshifted cb2, whose patch boundaries coincide with the boundaries of the isoviscous patches of nearly all viscosity schemes, in other words, that falls in all the pits, performs even slightly better than its shifted companions. The cb4 schemes gain something from the shifting but this benefit is much less pronounced than expected.

5. The difference between cb3 and cb4 is very small, which is interesting since the cb3 variants are much cheaper. [10]

The even more interesting question is which of these observations are also true for multigrid.

---

[9] See the description of Fig. 3.17 for the reason of its limited importance for this discussion.

[10] There is also a small unintended advantage for the cb3 scheme. The viscosity schemes due to their binary nature fail to provide a real pitfall for these blocking schemes. All cb3 schemes are bound to catch at least one of the viscosity jumps. At the time I set up the viscosity profiles I had not thought about the cb3 schemes yet, which recommended themselves later. I will correct this in a future version. However, due to point 4 this is not serious.

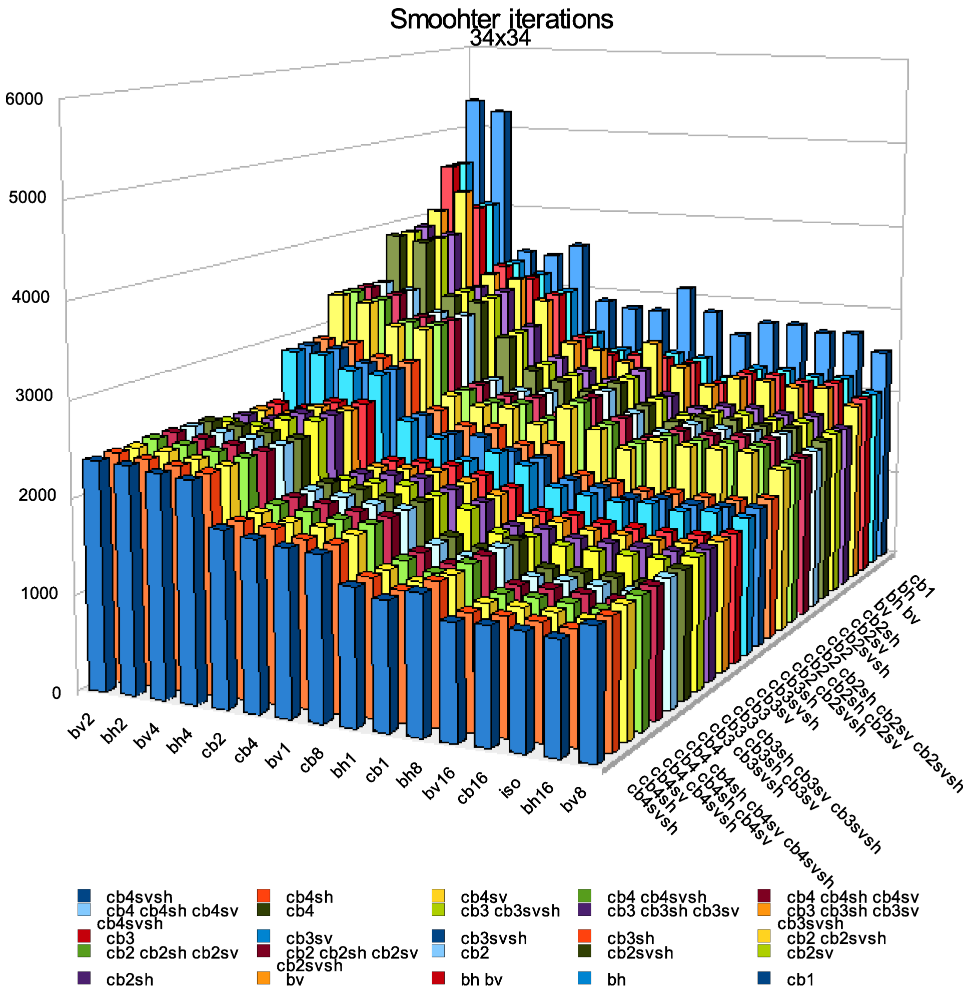


Figure 3.15: Number of Solver iterations for the randomized viscosity profiles on a 34x34 grid. For every viscosity profile the numbers for all blocking schemes are summed up thus yielding an estimate for the total number of smoothing steps to compute solutions for all viscosity profiles. The viscosity profiles are arranged according to this sum from left to right. The one with the smallest sum appears rightmost. Thus the most difficult profile will appear on the left side. The same procedure is performed for the blocking schemes. The blocking scheme that performs best over all viscosity profiles is in the first row, nearest to the observer.

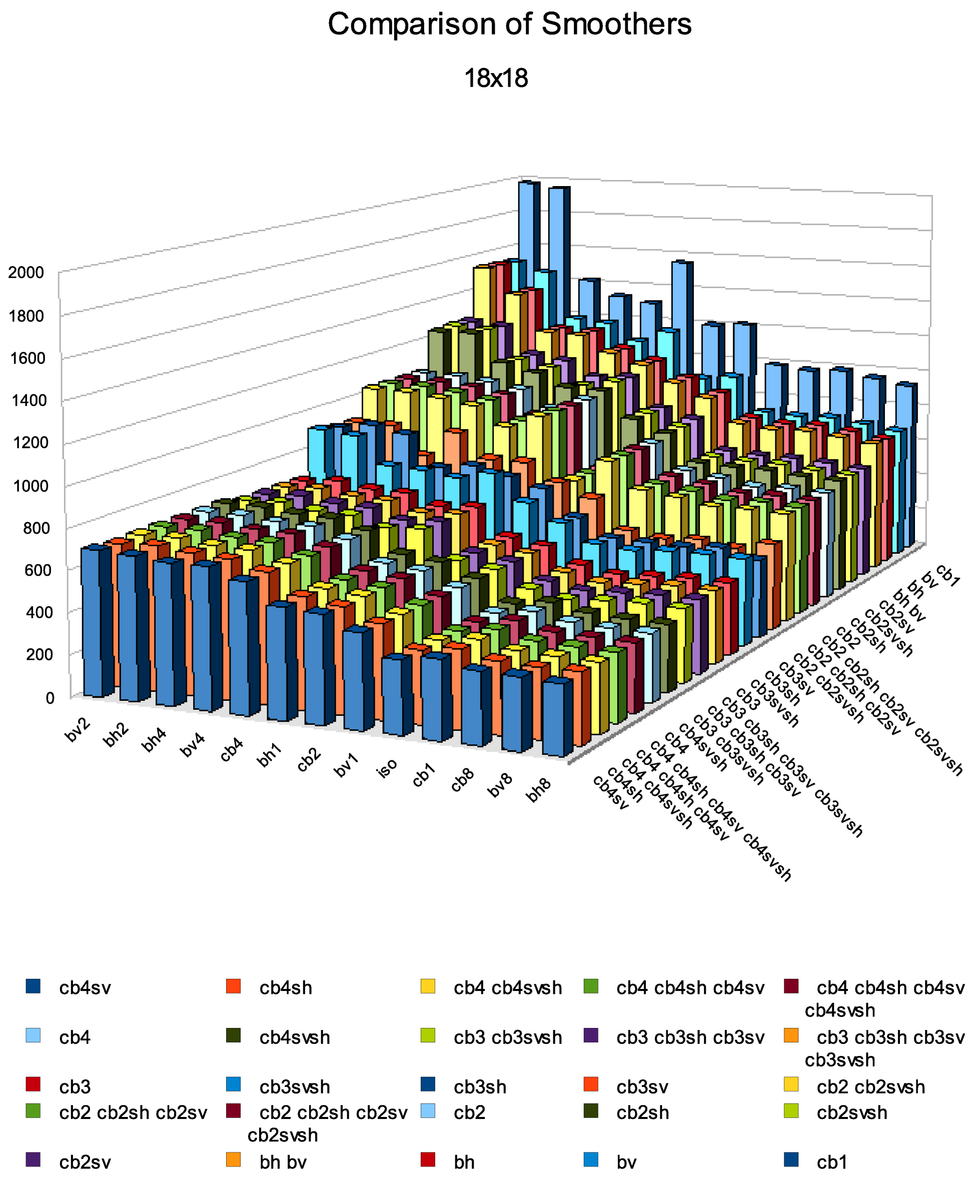


Figure 3.16: Number of solver iterations for the randomized viscosity profiles on a 18x18 grid

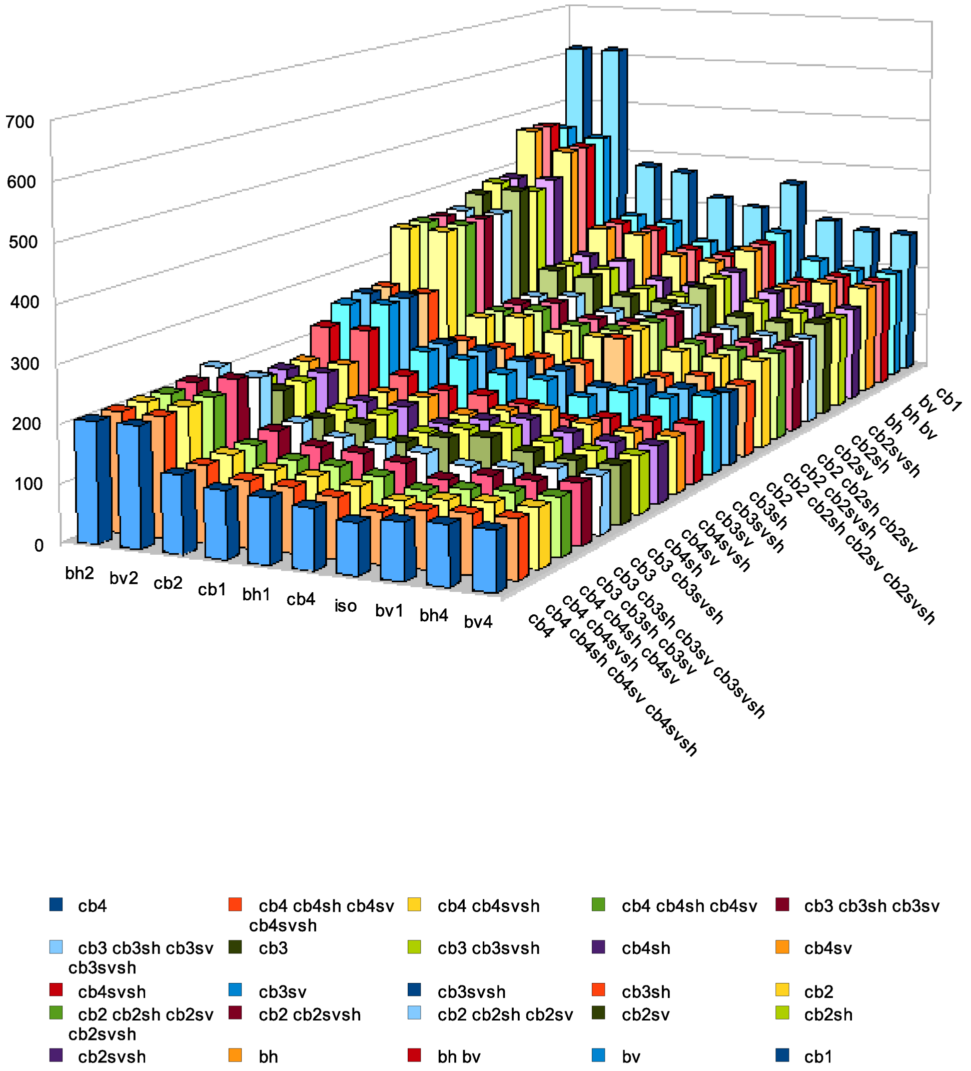


Figure 3.17: Solver iteration needed for $tol < 10^{-10}$ for the randomized viscosity profiles on a 10x10 grid. Note that this result is more influenced by the block overlap on the boundaries, simply because the ratio of boundary to non boundary blocks is greater for smaller grids. It is somewhat less suited for general conclusions than its bigger companions, but good enough as an additional check.

Smoother performance in the multigrid context

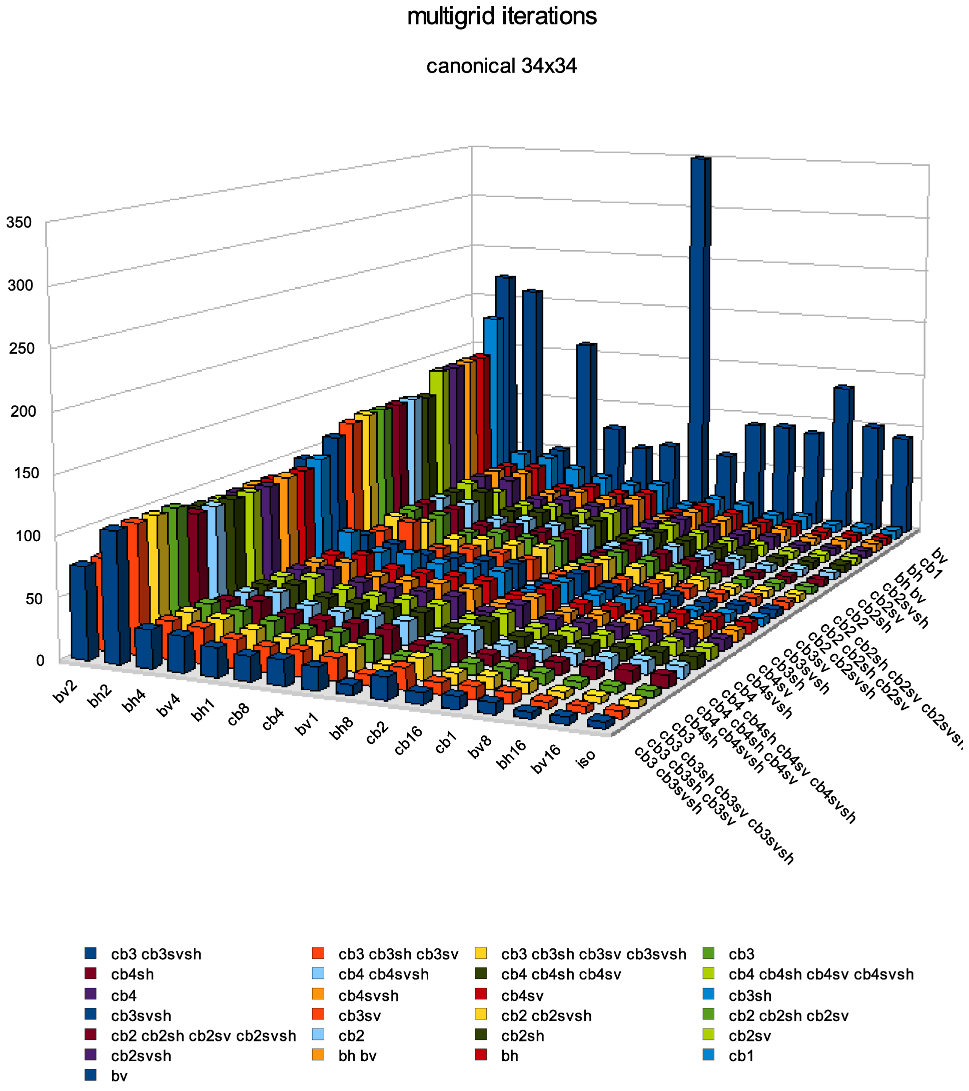
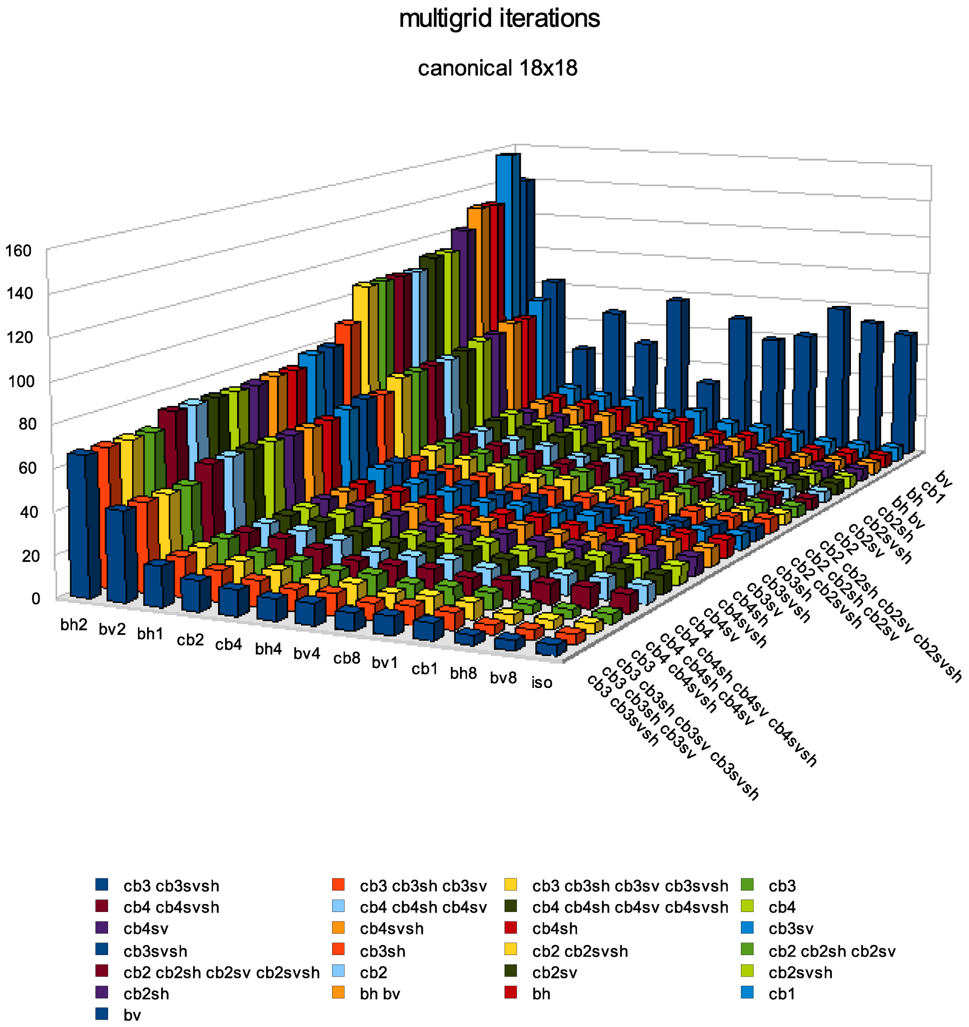


Figure 3.18: Note the nearly divergent line-Jacobi smoother (bv) and the challenging viscosity schemes bv2 and bh2.

Fig. 3.18 shows the number of iterations needed to obtain a relative error reduction of 10^{-10}.

Note that for all schemes the number of pre and post smoothing steps is 4. We make the following observations:

1. The performance of the best performing scheme is about a factor of two better than for the most simple scheme. This is also true if the most challenging profiles (bh2 and bv2) and the nearly divergent bv blocking scheme are re-



moved. The result can be seen in Fig. 3.19. This is much less than our sanguine hopes.

2. Neither bigger patches nor combinations with more schemes have any significant advantage.

3. The performance of the line-Jacobi smoothers is again disappointing. One of them is even divergent in some cases.

We will deal with point 3 first, this will also make clear point 2, and thus explain point 1.

Often both of the line Jacoby schemes and sometimes even the combination of the two converge very slowly. This happens even for the isoviscous case and depends on the random solution. However it does not happen if the method is applied as solver. I have also checked the spectral radii of the block matrices. They are quite bad for many viscosity profiles but not for the isoviscous case, in agreement with the normal performance as solver. To see what happens we look at an example error-history plot on Fig. 3.20 that shows the norm of the error after every fine-grid smoothing-step. The smoother seems to correct the wrong kind of error. This corresponds with the observation 2.

Greater block-size is *not* always beneficial in the context of multigrid. Patch-size 3×3 seems to be the optimal choice. For a patch-size of 4×4 without combination the effect of multigrid actually decreases for some viscosity profiles, namely for those with few viscosity jumps. In comparison to all other schemes the drops in the error due to the coarse grid correction are much smaller, the curve in Fig. 3.21 is nearly straight. This means that the overall performance for small grid sizes is merely achieved by smoothing.

This indicates that the division of work between smoothing and coarse-grid correction is invalidated if smoothing is attempted in a widespread way. The *local* error reduction which is essential for multigrid actually becomes damped sometimes by bigger patches. In this cases the smother simply reduces the slightly wrong kind of error, namely the part that would be handled more efficiently on the coarser grid. This is a very valuable information, since it is clear that after a certain amount of coupling is achieved further performance improvements cannot be gained by bigger patches, even if the cost of the block-matrix-inversion is not taken into account.

This corresponds to the poor performance of the line-Jacobi schemes. The effect becomes worse for larger problems (32×32) because the convergence rate of the smoother (as a solver) deteriorates and the defect of accuracy due to the ineffective local smoothing cannot be compensated. See Fig. 3.20. The combination of different blocking schemes can also be seen as a coupling strategy that attempts a task that is better performed by multigrid. *The proposed benefit of capturing the viscosity jumps turned out to be not important for our discretization.* From this angle point 1 is not so surprising.

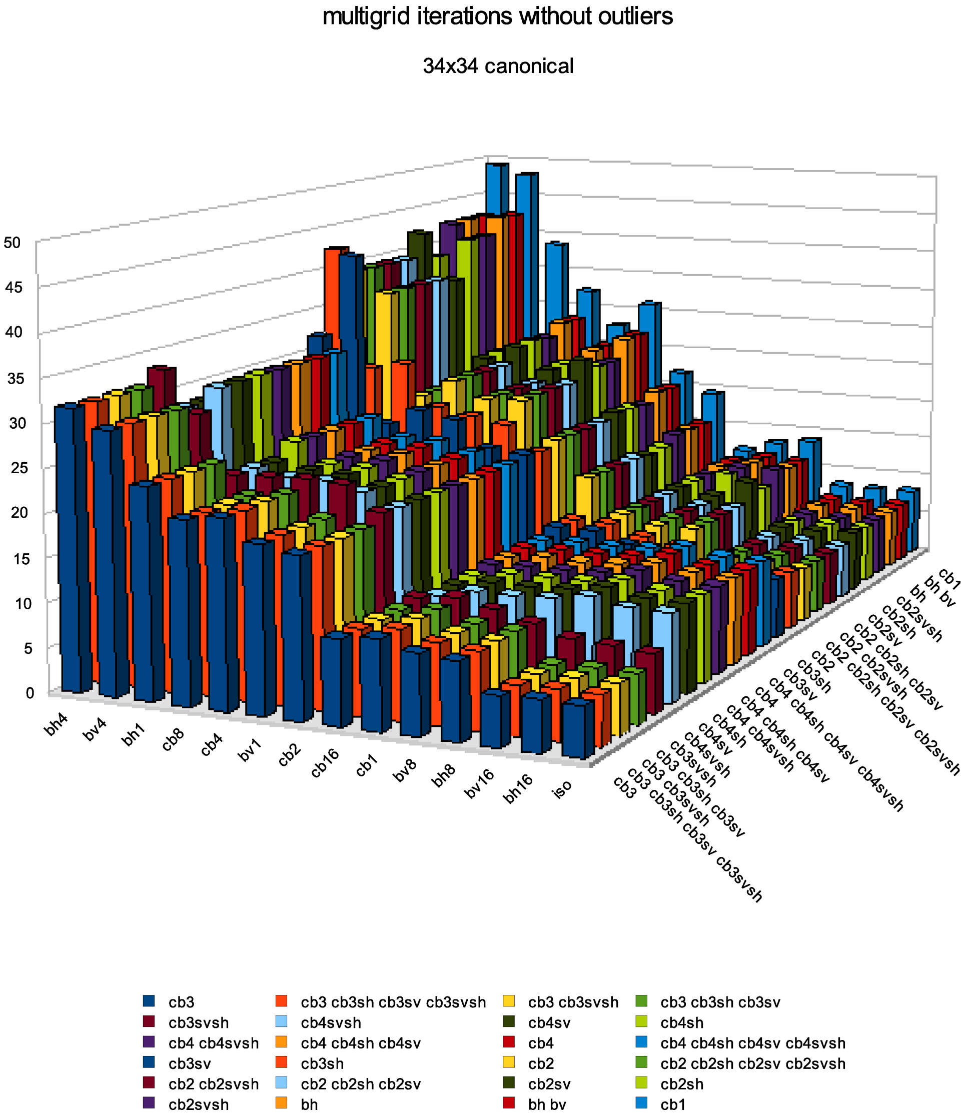


Figure 3.19: This is a version of the data of Fig. 3.18 where the most challenging profiles bh2 and bv2 have been removed and also the bad performing bv scheme.

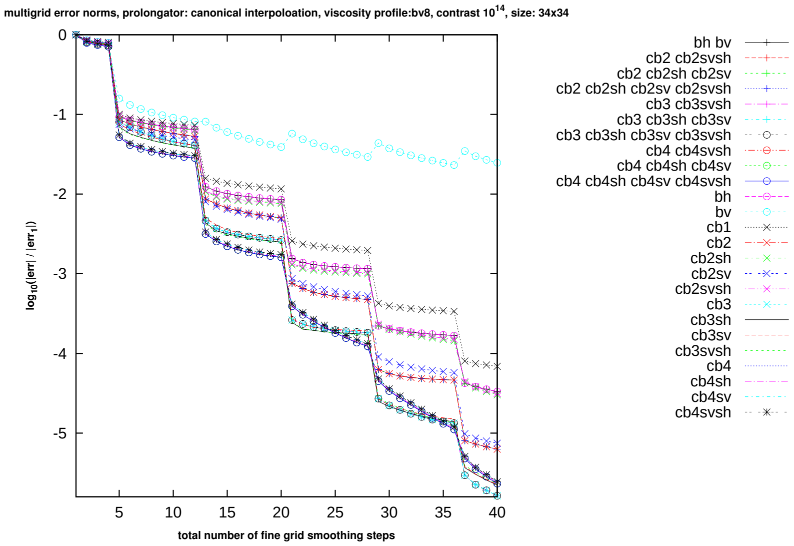


Figure 3.20: fine-grid error-history. The bv line-Jacoby smoother (cyan circles) has no benefit from the coarse-grid “corrections”. Surprisingly this does not happen for the first v-cycle and (only in this plot) never for the bh scheme. Note also the steep error drops after the coarse grid correction and the almost level steps, where only the smoothers work.

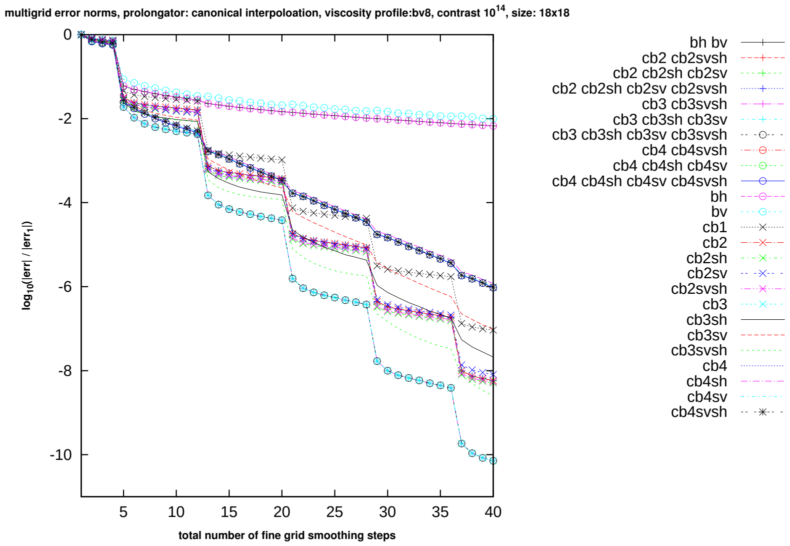


Figure 3.21: Both line-Jacobi smoothers and their combination suffer from useless coarse-grid corrections. Note also: The bigger the patch size the smaller the v-cycle error drops.

Conclusions for the smoothing strategy

The lesson to be learned in the last paragraph was not to meddle with multigrid specific tasks. Remember that Fig. 3.20 and Fig. 3.21 imply that most of the schemes would perform better if they would spend their time in more multigrid cycles instead of smoothing already smooth enough solutions. Therefore the next test will allow a minimal number of smoothing steps.

Thus a single block scheme will use only one pre smoothing and one post smoothing step, a combi scheme will use as many smoothing steps as it has members. Of course one cannot compare the error reduction per multigrid cycle in this case, but rather the error reduction per smoothing step. [11]

The data underlying Fig. 3.22 and Fig. 3.3.2 indicate a factor of $1,5$ between the best performing scheme and cb1. The line-Jacobi smoothers should not be taken as a reference. If one prescribes at least two pre and two post smoothing steps for all methods this factor increases two 2.

The cb3 variants seem to be optimal, with very little difference between them. The combination of blocking schemes does not improve the solution significantly.

---

[11] The result would be overwhelmingly in favor of the combined schemes, and suggest a performance increase of factor 10 for the four-member schemes. But this is misleading because the number of smoothing steps (and therefore the approximate numerical cost) is four times bigger for such an iteration.

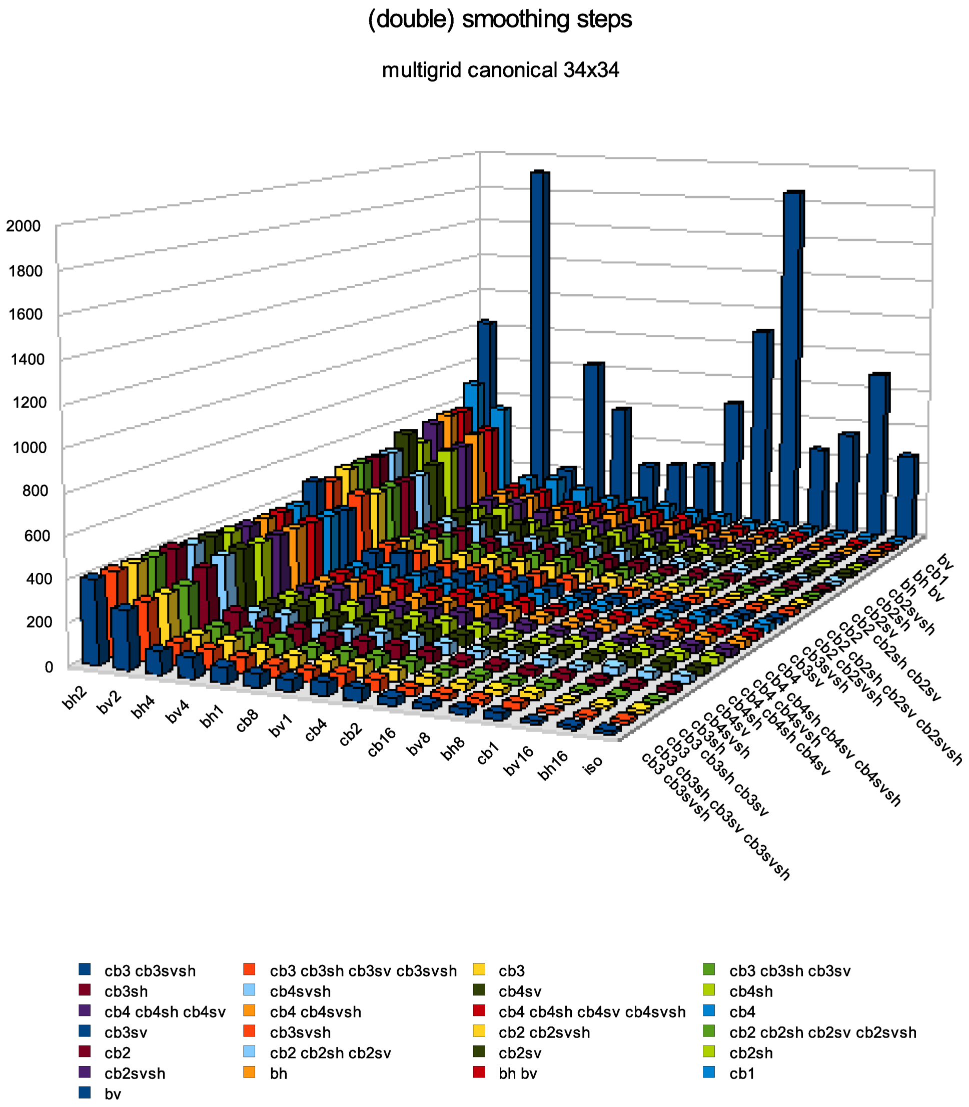
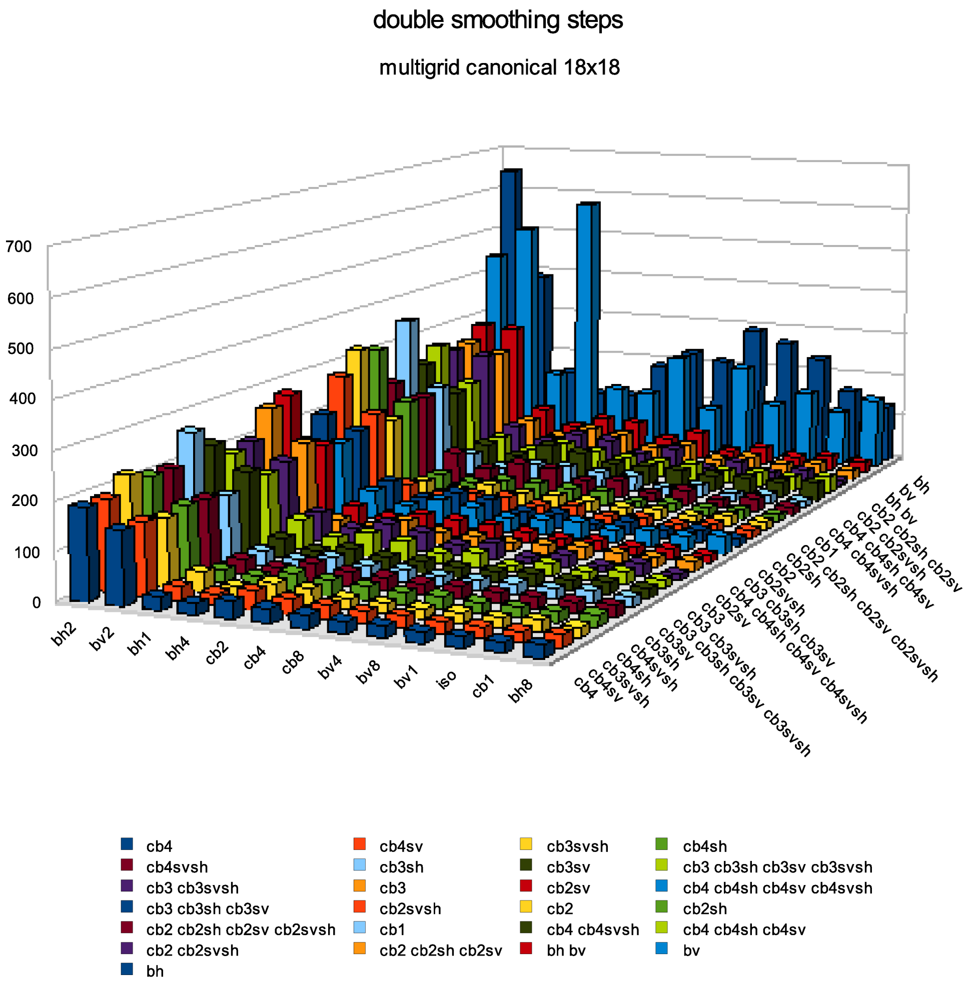


Figure 3.22: The number of fine-grid (double) smoothing steps needed for an error reduction of 10^{-10}. For the single-element combinations this is equivalent with the number of multigrid v-cycles, for a dual-element combi it is the number of v-cycles times two and so on. It is a (rough) estimate of the numerical cost if the inversion of the block matrices is not taken into account.


double smoothing steps
multigrid canonical 18x18
700
600
500
400
300
200
100
0
bh2
bv2
bh1
bh4
cb2
cb4
cb8
bv4
bv8
bv1
iso
cb1
bh8
cb4
cb4sv
cb3svsh
cb4sh
cb4svsh
cb3sh
cb3sv
cb3 cb3sh cb3sv cb3svsh
cb3 cb3svsh
cb3
cb2sv
cb4 cb4sh cb4sv cb4svsh
cb3 cb3sh cb3sv
cb2svsh
cb2
cb2sh
cb2 cb2sh cb2sv cb2svsh
cb1
cb4 cb4svsh
cb4 cb4sh cb4sv
cb2 cb2svsh
cb2 cb2sh cb2sv
bh bv
bv
bh
cb4
cb4sv
cb3svsh
cb4sh
cb4svsh
cb3sh
cb3sv
cb3 cb3sh cb3sv cb3svsh
cb3 cb3svsh
cb3
cb2sv
cb4 cb4sh cb4sv cb4svsh
cb3 cb3sh cb3sv
cb2svsh
cb2
cb2sh
cb2 cb2sh cb2sv cb2svsh
cb1
cb4 cb4svsh
cb4 cb4sh cb4sv
cb2 cb2svsh
cb2 cb2sh cb2sv
bh bv
bv
bh

Comparison of the prolongation schemes

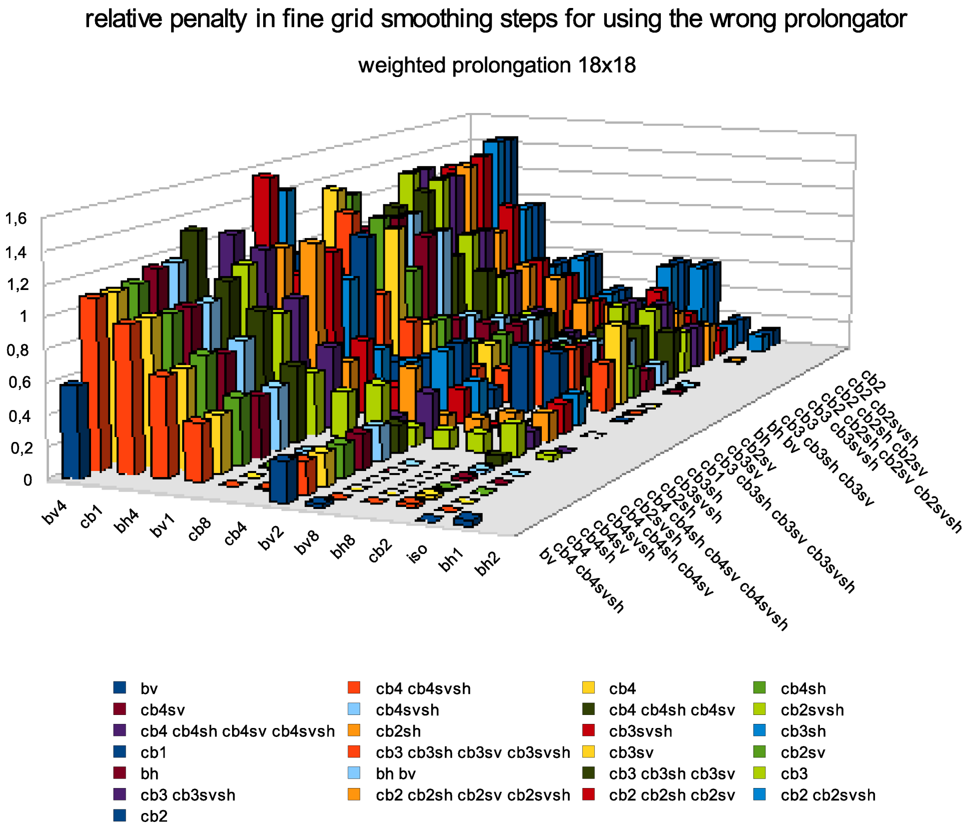
We now discuss the viscosity weighted prolongation, which turns to injection for strong viscosity contrasts. We compare it to the canonical prolongation by counting the number of fine grid smoothing steps needed to obtain a relative error reduction of 10^{-10}.

If we sum up this numbers for all smoothers and all viscosity schemes we find that the simple canonical inclusion is about 10% better. But this is mostly due to the fact that the challenging viscosity profiles have a much bigger influence in this rating.

It is interesting to distinguish between the cases for which the canonical inclusion is more appropriate and those for which the viscosity weighted prolongation is better.

To make the effect visible the difference in the number of smoothing steps between the two multigrid versions is computed. The result is divided by the number of smoothing steps of the faster converging scheme. This is equivalent to the performance ratio between the two schemes minus 1.

$$z = \frac{n_{slow}}{n_{fast}} - 1$$


relative penalty in fine grid smoothing steps for using the wrong prolongator
weighted prolongation 18x18
1,6
1,4
1,2
1
0,8
0,6
0,4
0,2
0
bv4
cb1
bh4
bv1
cb8
cb4
bv2
bv8
bh8
cb2
iso
bh1
bh2
bv
cb4 cb4svsh
cb4
cb4sh
cb4sv
cb4svsh
cb4 cb4sh cb4sv
cb2svsh
cb4 cb4sh cb4sv cb4svsh
cb2sh
cb3svsh
cb3sh
cb1
cb3 cb3sh cb3sv cb3svsh
cb3sv
cb2sv
bh
bh bv
cb3 cb3sh cb3sv
cb3
cb3 cb3svsh
cb2 cb2sh cb2sv cb2svsh
cb2 cb2sh cb2sv
cb2 cb2svsh
cb2
bv
cb4sv
cb4 cb4sh cb4sv cb4svsh
cb1
bh
cb3 cb3svsh
cb2
cb4 cb4svsh
cb4svsh
cb2sh
cb3 cb3sh cb3sv cb3svsh
bh bv
cb2 cb2sh cb2sv cb2svsh
cb4
cb4 cb4sh cb4sv
cb3svsh
cb3sv
cb3 cb3sh cb3sv
cb2 cb2sh cb2sv
cb4sh
cb2svsh
cb3sh
cb2sv
cb3
cb2 cb2svsh

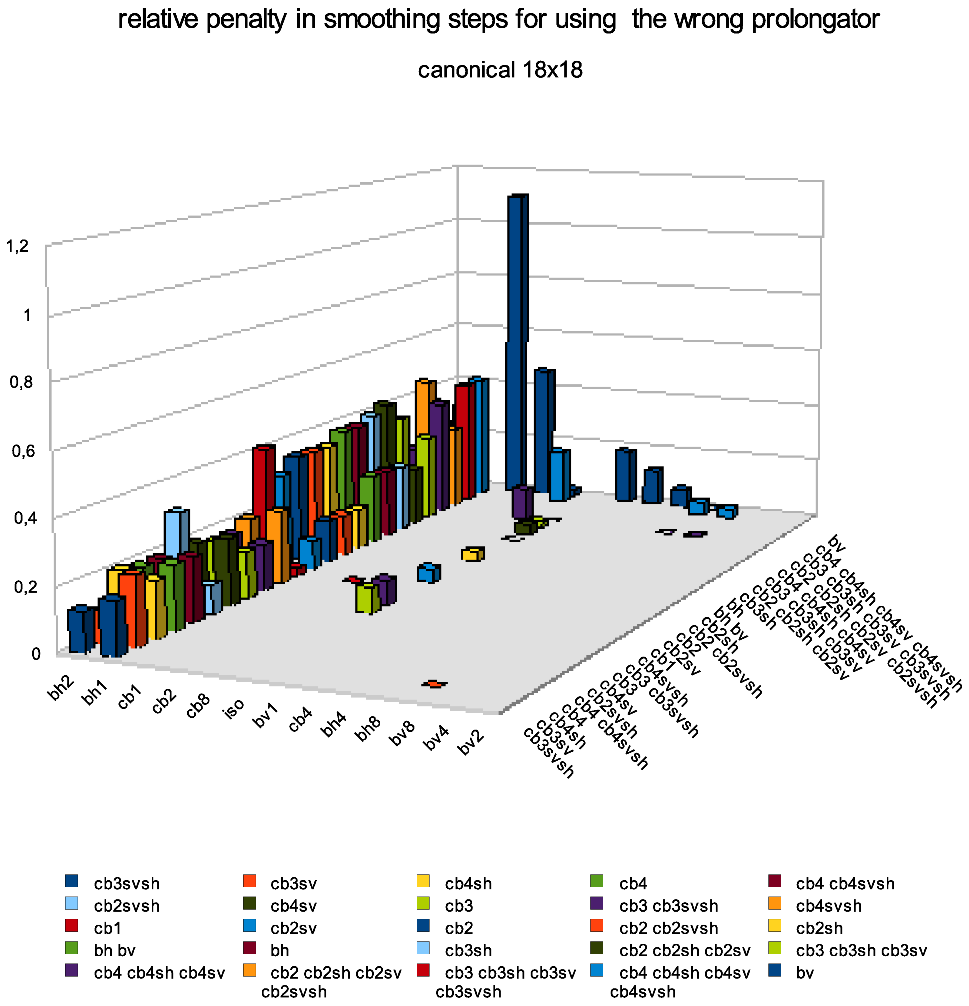

relative penalty in smoothing steps for using the wrong prolongator
canonical 18x18
1,2
1
0,8
0,6
0,4
0,2
0
bh2
bh1
cb1
cb2
cb8
iso
bv1
cb4
bh4
bh8
bv8
bv4
bv2
cb3svsh
cb3sv
cb4sh
cb4
cb4 cb4svsh
cb2svsh
cb4sv
cb3
cb3 cb3svsh
cb4svsh
cb1
cb2sv
cb2
cb2 cb2svsh
cb2sh
bh bv
bh
cb3sh
cb2 cb2sh cb2sv
cb3 cb3sh cb3sv
cb4 cb4sh cb4sv
cb2 cb2sh cb2sv cb2svsh
cb3 cb3sh cb3sv cb3svsh
cb4 cb4sh cb4sv cb4svsh
bv
cb3svsh
cb3sv
cb4sh
cb4
cb4 cb4svsh
cb2svsh
cb4sv
cb3
cb3 cb3svsh
cb4svsh
cb1
cb2sv
cb2
cb2 cb2svsh
cb2sh
bh bv
bh
cb3sh
cb2 cb2sh cb2sv
cb3 cb3sh cb3sv
cb4 cb4sh cb4sv
cb2 cb2sh cb2sv cb2svsh
cb3 cb3sh cb3sv cb3svsh
cb4 cb4sh cb4sv cb4svsh
bv

Fig. 3.3.2 shows that the canonical inclusion is almost always the method of choice. Only for two profiles bh2 and bh1 the weighted prolongation performed better. This is very interesting since bh2 is definitely the most challenging scheme. [12]

3.3.3 Semi coarsening

The semi coarsening approach should have been integrated in all the tests presented in this chapter. There was, however, no time to do so, since the idea and possibility to implement it came as a surprise at the end of this work. I am aware of this, but nevertheless could not resist to present the exiting results.

Semi coarsening can be seamlessly incorporated into the previously introduced framework. As we have seen, the block smoothers need the description of the space-filling curve to assemble the matrix. This description is contained in the data type that represents a blocking scheme.

If a block smoother is applied, which combines two blocking schemes, a permutation of the dof takes place. The permutation matrix can be computed from the combined blocking schemes, because the space-filling curves induce bijections from the grid point to the positions of the appropriate dof in the solution vectors.

To determine the position of a dof in an other ordering of unknowns one just has to compute the grid point with the first blocking scheme and then the position in the other ordering with the second blocking scheme.

If the target scheme describes a smaller grid, then the permutation becomes a restriction. If the target grid is larger it becomes a prolongation. The ordering of the dof is arbitrary. It is for instance even possible to restrict from a fine cb3 to a coarse bh scheme. To derive the prolongator matrix, beside the blocking schemes, only a geometrical interpolation scheme is needed.

1. For standard multigrid this is either the canonical inclusion or the viscosity weighted interpolation from $(2^n+1)^2$ to $(2^{n+1}+1)^2$ grid points. That means we have (for 2D) approximately four fine-grid points per coarse-grid point.
2. For semi-coarsening similar interpolations are applied, but from $(2^n + 1)^2$ to only $(2^{n+1} + 1)(2^n + 1)$ grid points. Thus we only have to interpolate about two fine-grid points from one coarse-grid point. This indicates that the prolongation is easier, which is in fact the case.
3. For the combined block smoothers without coarsening it is the identical "interpolation".

Technically the size of the grid is stored in two instance members of the blocking-scheme data type. To alter the dimension of the target grid just these two variables

---

[12] If we had based the plots on the symmetric profiles the results would have shown the same difficulty for bv2. Although one would assume that bh2 and bv2 should give symmetrical results this is not true for the randomized versions which have been used up to now. A closer look at Fig. 3.5 and Fig. 3.4 reveals that for bh2 two jumps with maximal amplitude are situated directly side by side whereas for bv2 the situation is slightly less serious.

have to be changed. For the standard multigrid procedure this blocking-scheme restriction or prolongation affects both grid dimensions, x and y. For a semi coarsening scheme just one number less is to be changed. The prolongation is also easier than for standard multigrid.
Accordingly it took only two hours to implement the first test for the viscosity profile bh2. To compare it to the previously presented results, I ran it with four pre and post smoothing steps. If semi coarsening perpendicular to the isoviscous patches is applied, it only needs two v-cycles to achieve the error reduction of 10^{-12} for the most challenging viscosity profile bh2. This is about *35 times faster* than the fastest scheme up to now and *89 times faster than cb1* in terms of multigrid iterations. So semi coarsening is by far the most advantageous option for this profile.
However, this huge increase in performance, does not occur for the schemes where the standard multigrid procedure works well. This indicates that semi coarsening is especially interesting when prolongation is an issue. Prolongation is always easier for this approach since the ratio of fine-grid to coarse-grid points is always about 2 for a dyadic refinement, even for the n-dimensional case, whereas this ratio increases with the space dimension to 2^{dim} for the standard multigrid approach.

3.3.4 Numerical cost

Block smoothers

The actual implementation in octave shows that the usage of combined different blocking schemes is not an expensive thing to do as long as the inversion of the block matrices is not too costly. In fact the need to invert several block diagonals is the only runtime penalty.
The application of different blocking schemes is equivalent to a permutation of unknowns in the operator. The permutation matrices can be computed at compile time. In any real code the operator would be applied in a grid based fashion anyway, so that the ordering of unknowns is hardwired. If one still wants to have different schemes available, automatic code generation could be implemented in some scripting language, so that the Fortran code produced in this way can be optimized by the compiler without the necessity to be able to handle different ordering of unknowns. [13].
If the blocks of the combined scheme are smaller than the blocks of a similar performing single blocking approach, the cost will actually decrease. This becomes obvious when the cost of a single block inversion is considered.
Since the block matrices are symmetric and positive definite, which can be deduced from the properties of the whole operator, Cholesky factorization can be applied.

---

[13] I have already done automatic code generation for other parts of the actual code (namely the test framework which is written in ruby)

Let n denote the number of dof combined in a block. The cost of a Cholesky factorization is then given by

$$\frac{2}{3}n^3$$

For a given mesh with N dof the number of blocks is given by

$$\frac{N}{n}$$

Hence the numerical cost to compute the approximate inverse of the whole operator is given by

$$\frac{2}{3}Nn^2$$

The cost of the application of this inverse is given by

$$\frac{N}{n}n^2 = Nn$$

So we obtain the following result for the numerical cost of computation and application of the approximate inverse:

$$c_{inv} = N(\frac{2}{3}n^2 + n)$$

This shows that the line Jacobi methods are absolutely undesirable for larger grids. They even destroy the optimality of multigrid, since the block size increases with the grid size. It also shows that the combination of smaller patches is less expensive than the usage of a larger patch.

Semi coarsening

The bad news is, that semi-coarsening schemes are not optimal. This is obvious for this small example, since the same argument applies as for the line-Jacobi smoothers.

However, whereas the line-Jacobi solvers do not pay off as smoothers in the standard multigrid context, they are rehabilitated for a semi-coarsening multigrid, where they are unevitably used to compute the exact solution on the coarsest grid.

Note that they do *not* have to be applied necessarily as *smoothers* in the semi-coarsening approach. Their smoothing performance remains disappointing. The astonishing result, mentioned before, used a cb3 smoother.

This means, that for semi coarsening we have only one expensive application of a line-Jacobi *solver* that is highly effective.

Instead, we have many, reiterated, fruitless, and thus even more expensive applications of the line-Jacobi *smoothers* an all grid levels, including the finest, in standard multigrid.

Concluding we find that we will try to avoid semi coarsening if we can achieve similar performance without the loss of optimality.

But if we are nevertheless forced to abandon it, we can do it much more comfortably with semi coarsening than with line-Jacobi *smoothers*.

3.4 Conclusions and outlook

3.4.1 Discussion of the proposed numerical improvements

Although this discussion is based on the actual data it does not attempt to decide exactly whether or not a particular scheme is the best choice. We want to form a reasonable strategy. Since this experiment aims at the improvement of a 3D code with a similar but different grid the answer is somewhat preliminary anyway.

Expectations

We start with a résumé of our expectations formulated in 3.2.5.

1. The assumption that the line-Jacoby smoothers are superior cannot be confirmed by any of the run tests. They are nearly always the worst performing schemes.
   The reason is that the coarse-grid correction is nearly useless after the first v-cycle as the missing drops in the error-history curves indicate.
   A combination of the two line-Jacobi schemes performs, however, sometimes better than the single point scheme and some uncombined patching schemes.
   But the numerical cost for this slight improvement is prohibitive.

2. Neither excessive combinations nor large block sizes are silver bullets. At most a factor of 2.5 in terms of multigrid iterations can be expected.
   As already mentioned capturing of jumps inside the blocks has nearly no effect. A patch size greater than 3 is not advisable.
   The combination $cb3, cb3shsv$ is the only one that seems to be worth the effort. Whether or not this is an option at all depends on some implementational details. Multigrid has a parallelization (or communication) penalty on the coarse grids. On the other hand bigger or more block inverses may cause cache issues.

3. The expected disaster for the high frequency viscosity profile cb1 did not happen. In fact it was solved very efficiently. The real challenge consisted in the schemes with a band width of two.

4. A real surprise is, that even the single point scheme performs often well.

5. The weighted prolongation in the actual form has no great benefit, except for the most challenging viscosity profile.

6. At least semi coarsening surpasses our sanguine hopes.

Apparent contradictions

Nearly all this seems to be quite contradictory to the experiences described in [49] and the commonly held notion to use line Jacobi smoothers for problems with discontinuities, which is for 3D extended to smoothing along 2D faces where the coefficient jumps. I will explain why this is not really the case.

In [49] the proposed blocking schemes are the remedy for divergent or nearly divergent solvers.

This problem does not exist in our tests, expect for line-Jacobi methods which, ironically, have the biggest block size. Even for the most simple scheme in the most challenging viscosity configuration our convergence rate is about 0.95, which is not very well but still convergent.

For the viscosity profiles that are comparable to those, used in [49], our most simple solver achieves 0.44, although the jump in the viscosity is 10^8 times larger in our case. So we actually do not have the same kind of numerical problem that was successfully solved in [49].

From another point of view the problems described in this chapter are harder than those in [49] or [39, 70]. In our viscosity profiles the number of jumps increases with the grid size. As we have seen the crucial point is the bandwidth. The destructive effect of the latter is worse for larger grids.

This is understandable if the erroneous values are situated at grid points near the jumps. If the bandwidth is measured in points between the bands, then for a constant bandwidth the number of bands increases with the resolution, and so does the number of points with erroneous values.

This, by the way, explains the increased errors for seemingly the same viscosity profiles for higher resolutions, which, at first glance, contradicts the expected multigrid behavior.

But geometrically the bandwidth *does* differ between the profiles 16x16 bh2 and 32x32 bh2. That even a comparison between 16x16 bh2 and 32x32 bh4 shows differences is due to the bandwidth induced inapplicability of the central idea of multigrid. So the results really confirm the theory and do not contradict it.

Another feature, in which [49] differs from this work, is the position of the jumps relative to coarse grid boundaries.

In [49] the jumps were situated at coarse grid boundaries, which was generally not the case in our tests.

According to these observations and our experience with Terra's difficulties to handle strongly varying viscosity, I suggest that

some common problems in dealing with parameter jumps do not occur for this kind of viscosity discretization.

I have not yet implemented another discretization and run similar tests to confirm this suggestion, but there is another point in favor of this idea. The discretization is optimal for the canonical interpolation in the sense that stress is continuous at cell faces.

This is often taken as a guideline for the construction of matrix dependent prolongation operators. See e.g. [1, 2]. As a short computation shows, this continuity is automatically fulfilled if viscosity is continuous at the cell faces and the velocity is interpolated linearly.

Nevertheless the proposed weighted prolongation cannot be abandoned yet. It has an advantage for the most challenging schemes that cannot be smoothed directly on the finest grid. The method could probably be improved if the viscosity would not be restricted before it is used as weighting factor. This would require the definition of an efficient restrictor, which has not been implemented yet.

However, the presented framework allows this kind of change to be made very rapidly. The canonical version was generalized to the weighted prolongation within two days . so it would be very interesting to test some more sophisticated versions.

Summary

The most interesting, quite unexpected result is that the implemented viscosity discretization seems to simplify he numerics considerably. Of course this accidental discovery must be backed up by further experiments with an piecewise constant viscosity approximation for similar profiles.

If we can confirm the results for the 3D case, we are able to abandon the expensive and inefficient line Jacoby smoothers.

For the resulting linear system even the simple cb1 blocking and canonical prolongation which avoids the time consuming set up of the matrix dependent transfer operators, yield very good results. Of course one would prefer this most simple scheme if it works, since it is also the cheapest one, regarding computational cost.

If we encounter serious problems for larger models in 3D spherical geometry that can not be cured with the new discretization, then semi coarsening is definitely interesting.

3.4.2 Necessary enhancement of the framework

The test cases should include viscosity profiles with round patches and oblique edges. It would be profitable, to include the test problems of [39, 70] to be able to compare the method to other approaches found in the literature.

A performance comparison with algebraic multigrid could be interesting for a fully optimized version.

The framework must be ported to 3D. This includes a refined semi coarsening strategy. There are 3D variants that use semi coarsening multigrid recursively to solve the 2D sub problems arising from semi coarsening the original 3D problem. Probably something like this will be necessary.

3.4.3 Implementational aspects

Programming effort

The effort needed to answer the questions asked at the beginning was about four months. First symbolic test cases were implemented in mupad to check some operator properties for cyclic domains exactly. Then the combined smoothers were implemented, at first in mupad.
Since mupad's numerical capabilities turned out to be insufficient, the code had to be ported to octave which was done in a test driven way. [14] This enforced some recoding since octave is not really a high level language, compared to object oriented and functional languages like mupad. [15]
After this the code was generalized to arbitrary domain sizes to allow multigrid.
Although this has been quite tedious, it is not comparable to the effort needed to do all this in the actual huge MPI parallelized code.
From this point of view it has not only been sensible to write this little framework, it would just not have been possible to test the whole bunch of ideas.

Efficiency of the code

The actual solver is, thanks to octaves sparse-matrix library, reasonably fast. I have also already parallelized it, which was easy since no communication between the different test cases is needed. But one part of the framework is much to slow for a reasonably sized 3D problem.
This is the matrix assembly step, which, up to now, uses loops which are prohibitively slow in an interpreted language.
The remedy would be either to write this function in C and call it from octave, or to port the whole framework.
Since a possible future change of the solver in terra against something new would probably amount to a C++ Interface either way [16], I tend to prefer the latter.

---

[14] That means that every new feature was preceded by an automated test for this feature.

[15] OO support in mupad is rather rudimentary but can be retrofitted by means of its functional abilities

[16] Dune

Chapter 4

Implementation in Terra

This chapter is intended to shed some light on the infrastructure necessary to maintain a code as huge as Terra. The benefit of these efforts cannot be measured as easily as e.g. a new multigrid strategy. Nevertheless it is crucial for the success of the whole project. Accordingly the amount of code, that has been written for this purpose is comparable to the amount of code developed for all other parts of this work.

4.1 Motivation

I will present an example for a change in the code and how it convinced me to write a test framework.

4.1.1 An example

We first describe the new algorithm to give an overview about the amount of work expected to implement it. By the way it is of course a reasonable change in the code that was suggested to me by Prof. Zumbusch and Irad Yavneh. It cannot only be justified by experience but also by theory, which, among other things, surprisingly connects it to a very much simplified version of algebraic multigrid. [22].

Multigrid preconditioned CG

We want to replace the multigrid algorithm by a Krylov-subspace method e.g. the conjugate gradients method, that uses multigrid as a preconditioner. [1] We expect greater robustness, since the Krylov-subspace method can handle some error components better than the pure multigrid. To see how this change can be implemented in the code let us consider the linear system:

$$Ax = b$$

---

[1] The optimality of the multigrid method is not endangered, since only a fixed number of CG iterations will be performed.

If we multiply from the left a preconditioning matrix P we get the system:

$$PAx = Pb$$

with the same solution x Now let us look on a standard Conjugate Gradient algorithm

```
choose startestimate $x$
compute first residual $r = b - Ax$
compute first $\alpha = (r, r)$
compute first $p = r$

while $\alpha > \epsilon$ do
$v = Ap$
$\lambda = \frac{\alpha}{(v,p)}$
update $x = x + \lambda p$
update $r = r - \lambda v$
update $\alpha_{new} = (r, r)$
update $p = r + \frac{\alpha_{new}}{\alpha} p$
$\alpha = \alpha_{new}$
end while

                                                  (4.1)
```

Now we substitute A by PA and b by Pb to get the preconditioned Algorithm.

```
choose startestimate $x$
compute first residual $r = Pb - PAx = P(b - Ax)$
compute first $\alpha = (r, r)$
compute first $p = r$

while $\alpha > \epsilon$ do
$v = PAp$
$\lambda = \frac{\alpha}{(v,p)}$
update $x = x + \lambda p$
update $r = r - \lambda v$
update $\alpha_{new} = (r, r)$
update $p = r + \frac{\alpha_{new}}{\alpha} p$
$\alpha = \alpha_{new}$
end while

                                                  (4.2)
```

We now proceed to choose a concrete preconditioning matrix. Our choice is the approximate inverse obtained by multigrid. Of course we want to avoid computing this $\tilde{A}^{-1}$ but only replace the occurrences of multiplications with $P = \tilde{A}^{-1}$ by a call to multigrid.

```
choose startestimate $x$
compute first residual
$r = multigrid(A, (b - Ax))$
compute first $\alpha = (r, r)$
compute first $p = r$

while $\alpha > \epsilon$ do
$v = multigrid(A, Ap)$
$\lambda = \frac{\alpha}{(v,p)}$
update $x = x + \lambda p$
update $r = r - \lambda v$
update $\alpha_{new} = (r, r)$
update $p = r + \frac{\alpha_{new}}{\alpha} p$
$\alpha = \alpha_{new}$
end while

                                        (4.3)
```

Expected implementational effort

The algorithm only uses subroutines that already provide a fairly high level of abstraction. Thus the implementation has not to deal with parallelization and grid specific issues. [2] It took me half a day to implement the changes and only some hours more to use the limited object oriented features of Fortran 90 to hide the routines behind overloaded operators, so that the program code looked nearly exactly like the pseudo code presented above. The code worked well in the old version.

4.1.2 Difficulties in the implementation

However, the old version of Terra does not provide the ability to treat variable viscosity. Since this is an main focus of this thesis, the actual aim was of course to implement the algorithm in the new version, that had been developed for several

---

[2] In fact, when I asked John Baumgarder, the author of Terra, what he expected the change would cost in terms of programming hours, he answered: "a day". He was exactly right for the older version of Terra, I was working with at the time.

years by different people all over the world. Although the interfaces of the necessary routines had not been changed, the code did not work.
We soon found out, that it did not receive its data via the parameter list, but via common blocks, which linked it to other parts of the code in a very undesirable way. The originally clean code had been cluttered up over the years.
In view of the changes, proposed in this work, I decided to change this, before I started to implement them. It soon turned out, that even the limited task, to provide clean interfaces at least for the solver, would entail a major refactoring of the code. To do this my colleague Christoph Köstler and I have worked for several months. This cannot be done without tests that ensure that the results remain the same under the refactoring.

4.2 Test framework

4.2.1 Features

Although the changes in the code would be huge, they firstly would only change the structure, not the functionality. So at first the sole task was to check that, the results did not *change*. However, since there are some different branches in the code and different machines the code runs on, this is not a single test, but a fairly large number, which cannot be tested without automation. The developed framework provides the following features.

1. Automatic generation of of input files setting up the physical input for different test cases
2. Automatic generation of header files for the compilation
    (a) with different MPI libraries,
    (b) on different machines (the altix at munich and several pc's)
    (c) for different parallelization schemes
3. Automatic job setup for the queue on the altix supercomputer and collection of the results of parallel run tests
4. Automatic check of up-to-dateness of results with respect to the code [3]
5. Automatic check of computing times and timeout control for the jobs
6. Automatic cross architecture check of results
7. Scheduled nightly builds and test runs

---

[3] It is of course necessary to check that the correct result was computed, *after* the code had been changed. This naturally includes tests for failing compilation and runtime errors in every test run. Otherwise the correct result would only remain correct because it was out of date and had never been overwritten.

Some of these features are of course obvious, I only present them to make clear, what a lot of work, not directly related to the code, has been necessary.

4.2.2 Development

The framework has been written in ruby, since Fortran, the language of Terra, does not provide the necessary flexibility. The code for the framework would have been much longer.
The preferred way to develop a test framework is of course test driven. See e.g. [8]. That means that the implementation of the self tests for the framework preceded the implementation of the according functionality. [4]

4.3 Outlook

4.3.1 Further reasons for a test framework

The numerical tests of the previous chapter show, that it is difficult to use recipes. To make an informed decision, one has to test. As mentioned before this is not possible in a reasonable time frame, without reuse of existing software for parts of the solution process. To be able to integrate external solutions we need clean interfaces.

4.3.2 Necessary extensions

Up to now automatical, and therefore constantly running, tests are only available to compare Terra against its own history. This might be called regression testing. To facilitate a fast code development, we also need:

1. A unit testing framework that checks parts of the code independently.
2. Physical benchmark cases for the complete solver. See e.g. [15]

---

[4] How important it is to test the tests, became obvious when by mistake the self tests were switched of for some weeks. A major bug, introduced by the refactoring remained unrecognized for several revisions of the code, and had to be tracked down afterwards via the version control system.

Chapter 5

Summary and outlook

5.1 Summary

In this work mantle convection simulation with Terra has been investigated from a numerical point of view, theoretical analysis as well as practical tests have been performed. The following results have been achieved.

5.1.1 Connection of the physical model to numerical stability criteria

Stability of the incompressible Dirichlet problem

For the incompressible case and the Terra specific treatment of the anelastic approximation, two inf-sup stable grid modifications can be applied, that are both compatible with hanging nodes. It has been shown that

1. For the $Q_{1_h} Q_{1_{2h}}$ element pair a simple numeric test can be used to prove the stability for any given grid.

2. For the $Q_{1_h} P^{disc}_{1_{2h}}$ element pair an existing general proof can be adopted, for 1-regular refinements with hangig nodes.

Extension to the anelastic approximation

The necessary conditions for the expansion of the stability result to the anelastic approximation have been shown.

Slip boundary condition

The influence of the slip boundary condition is destabilizing. For the incompressible case a cure can be adopted from the literature.

5.1.2 Multigrid-test framework

A numerical framework has been developed for different numerical handling of strongly varying viscosity. By application of this framework to 2D testcases the following results were found.

1. The continous viscosity discretization peformes well even for large viscosity gradients simple block smoothers and simple prolongation.
2. The most sophisticated combinations of block smoothers performs about 2.5 times better in terms of multigrid iterations.
3. Matrix dependend prolongation sometimes improves performance to a certain extendt, but cannot resolve the most challenginging profiles. In most cases canonical inclusion is superior.
4. For the most challenging viscosity profile semi coarsening performs 89 times better in terms of multigrid iterations. It clearly provides a remedy, where standard multigrid techniques fail due to fundamental inabilities of the multigrid algorithm.
5. All new schemes perform better than the already implemented line-Jacobi smoothers.

5.1.3 Regression-test framework

An automatic regression-test framework runs on several machines including the supercomputer for production and helps to refactor the code.

5.2 Outlook

The outlook sections of the previous chapters already mentioned necessary or promising further steps. I will now summerize them and emphasize their connections.

5.2.1 Generalization of stability results

Stability and well posedness of the numerical formulation are even more important than efficiency of the solver for a limited number of tests. Clearly it does not matter how fast a solution can be obtained if it cannot be trusted.

Therefore the first aim should be a complete proof of the stability conditions for the anelastic approximation and the slip boundary condition for a suitable finite element pair.

Up to now, the stability of a numerical implementation can be proved *either* for space dependent density *or* for an incompressible fluid with free slip boundary.

The inf-sup stability of the proposed grid modifications, indicate that this task will be not more difficult for Terra's adapted grid than for any other inf-sup stable pair of finite elements, but up to now I do not know about a proof for any pair.
It would be very interesting, and perhaps not too difficult, to change this.

5.2.2 Unit tests

After the unique solvability is established, tests are much more usefull for the code development, since only then a failing test indicates an error in the *code*. I would like to enhance the regression-test framework with unit tests. Under the control of an extensive test framework it is possible to clean up the huge code and remove duplications without destruction of functionality.
Instead a lot of versions that differ only in small detailes and thus contain a lot of duplication, I want to have a single source base that is constantly, automatically tested for all the specific tasks, it can be used for.
As previously pointed out, this is an important precondition to be able to integrate external libraries.

5.2.3 Physical benchmarks

Supposing we are able to integrate external libraries seamlessly, it still remains difficult to predict the impact of a special strategy *quantitatively*, as the experience with the multigrid framework shows.
Accordingly we need as many physical different bench-mark scenarios as possible. Recently the possibillity to produce such testcases for variable viscosity analytically has been the subject of a diploma thesis at our group [71], but this must be an aim for the whole community.

5.2.4 Numerical improvements

If the benchmarks and tests are in place, we can proceed with an in-depth comparison of alternative numerical approaches. I would at least be very interested in an algebraic multigrid solver, an adaptive grid refinement, and accordingly space dependent time step sizes, various versions of 3D semi coarsening, and a different discretization with finite volumes (provided the stability can be proved). It is clear that this can only be achieved by the integration of external software.

5.2.5 Enhanced physical models

An extensive testframework also provides the flexibility, to adapt the code to varying scientific requirements. Our group is e.g. interested in the interaction of mantle convection and the orogenesis of the Andes.
If something like this is to be modeled, than we clearly need a really free surface. This requires a Lagrangian or Eulerian Lagrangian description of the problem like

in [30], and is therefore quite different from our, up to now, purely Eulerian approach.

However the difficulties in the treatment of strongly variable viscosity will not vanish for this more complex problem.

It would be fascinating if the strengths of Terra could be generalized to be able to treat problems like this. This will definitly not be possible without the use of adaptive grids, and thus external software. This emphasizes the importance of the preceding steps mentioned above.

Appendix A

A.1 Fréchet derivative of A

To express the fact that τ is an argument of the operator $\nabla\cdot$ we write $\mathbf{Div}(\tau)$ instead of $\nabla \cdot \tau(\mathbf{v})$. This is easier to read.

$$
\begin{aligned}
A\mathbf{v} &= \mathbf{Div}\left[(\tau(\mathbf{v})\right] \\
&= \nabla \cdot \tau(\mathbf{v}) \\
&= \nabla \cdot \left[\mu(\mathbf{v})\dot{\hat{\varepsilon}}(\mathbf{v})\right] \\
&= \nabla \cdot \left[k\{\dot{\hat{\varepsilon}}_0(\mathbf{v})\}^r\dot{\hat{\varepsilon}}(\mathbf{v})\right] \\
&= \nabla \cdot \left[k\left\{\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}\right\}^r \dot{\hat{\varepsilon}}(\mathbf{v})\right] \\
&= \nabla \cdot \left[k\left\{\sqrt{\tfrac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^t - \tfrac{1}{3}\nabla\cdot\mathbf{v}I\right) : \tfrac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^t - \tfrac{1}{3}\nabla\cdot\mathbf{v}I\right)}\right\}^r\right. \\
&\qquad \left.\tfrac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^t - \tfrac{1}{3}\nabla\cdot\mathbf{v}I\right)\right]
\end{aligned}
$$

For the Fréchet derivative F_A the chain rule holds. Because it is more suggestive, I use the Leibniz notation $F_A = \frac{\delta A}{\delta \mathbf{v}}$.

$$
\frac{\delta A}{\delta \mathbf{v}} = \frac{\delta \mathbf{Div}}{\delta \tau}\frac{\delta \tau}{\delta \dot{\hat{\varepsilon}}}\frac{\delta \dot{\hat{\varepsilon}}}{\delta \mathbf{v}} \tag{A.1}
$$

Since $\mathbf{Div}$ and $\dot{\hat{\varepsilon}}$ are linear they are reproduced. Since we want to compute a linear approximation in operator form for $\delta\mathbf{v}$, I write all derivatives with the argument they are applied to.

$$
\begin{aligned}
\tfrac{\delta \mathbf{Div}}{\delta \tau}\delta\tau &= \mathbf{Div}(\delta\tau) \\
\tfrac{\delta \tau}{\delta \dot{\hat{\varepsilon}}}\delta\dot{\hat{\varepsilon}} &= \tfrac{\delta \mu}{\delta \dot{\hat{\varepsilon}}}\dot{\hat{\varepsilon}} + \mu(\dot{\hat{\varepsilon}})\delta\dot{\hat{\varepsilon}} \\
\tfrac{\delta \mu}{\delta \dot{\hat{\varepsilon}}}\delta\dot{\hat{\varepsilon}} &= \tfrac{\delta \mu}{\delta \dot{\hat{\varepsilon}}_0}\tfrac{\delta \dot{\hat{\varepsilon}}_0}{\delta \dot{\hat{\varepsilon}}} \\
\tfrac{\delta \mu}{\delta \dot{\hat{\varepsilon}}_0}\delta\dot{\hat{\varepsilon}}_0 &= kr\dot{\hat{\varepsilon}}_0^{r-1}\delta\dot{\hat{\varepsilon}}_0 \\
\tfrac{\delta \dot{\hat{\varepsilon}}_0}{\delta \dot{\hat{\varepsilon}}}\delta\dot{\hat{\varepsilon}} &= \tfrac{1}{2\sqrt{\dot{\hat{\varepsilon}}:\dot{\hat{\varepsilon}}}}\dot{\hat{\varepsilon}} : \delta\dot{\hat{\varepsilon}} \\
\tfrac{\delta \dot{\hat{\varepsilon}}}{\delta \mathbf{v}}\delta\mathbf{v} &= \dot{\hat{\varepsilon}}(\delta\mathbf{v})
\end{aligned}
$$

We now can simply derive the operator form by successive back substitution in A.1.

$$\begin{aligned}
\tfrac{\delta \mathbf{Div}}{\delta \tau}\delta\tau &= \mathbf{Div}\,(\delta\tau)\\
&= \mathbf{Div}\left(\tfrac{\delta\mu}{\delta\dot{\hat{\varepsilon}}}\dot{\hat{\varepsilon}}(\mathbf{v}) + \mu(\mathbf{v})\delta\dot{\hat{\varepsilon}}\right)\\
&= \mathbf{Div}\left(\tfrac{\delta\mu}{\delta\dot{\hat{\varepsilon}}_0}\tfrac{\delta\dot{\hat{\varepsilon}}_0}{\delta\dot{\hat{\varepsilon}}}\dot{\hat{\varepsilon}}(\mathbf{v}) + \mu(\mathbf{v})\delta\dot{\hat{\varepsilon}}\right)\\
&= \mathbf{Div}\left(\left[kr\dot{\hat{\varepsilon}}_0(\mathbf{v})^{r-1}\tfrac{1}{2\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}):\dot{\hat{\varepsilon}}(\mathbf{v})}}\dot{\hat{\varepsilon}}(\mathbf{v}) : \delta\dot{\hat{\varepsilon}}\right]\,\dot{\hat{\varepsilon}}(\mathbf{v}) + \mu(\mathbf{v})\delta\dot{\hat{\varepsilon}}\right)\\
&= \mathbf{Div}\left(\left[kr\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}^{\,r-1}\tfrac{1}{2\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}):\dot{\hat{\varepsilon}}(\mathbf{v})}}\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\delta\mathbf{v})\right]\,\dot{\hat{\varepsilon}}(\mathbf{v}) + \mu(\mathbf{v})\dot{\hat{\varepsilon}}(\delta\mathbf{v})\right)\\
&= \mathbf{Div}\left(\left[\tfrac{1}{2}kr\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}^{\,r-2}\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\delta\mathbf{v})\right]\,\dot{\hat{\varepsilon}}(\mathbf{v}) + \mu(\mathbf{v})\dot{\hat{\varepsilon}}(\delta\mathbf{v})\right)
\end{aligned}$$

with $\dot{\hat{\varepsilon}}(\delta\mathbf{v}) = \frac{1}{2}\left(\nabla\delta\mathbf{v} + \nabla\delta\mathbf{v}^t - \frac{1}{3}\nabla\cdot\delta\mathbf{v}I\right)$ and $\dot{\hat{\varepsilon}}(\mathbf{v}) = \frac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^t - \frac{1}{3}\nabla\cdot\mathbf{v}I\right)$

A.2 Symmetry of $a'(.,.)$

The bilinear form defined by

$$a'_{\mathbf{v}}(\mathbf{u},\mathbf{w}) = \int\limits_{\Omega} \mathbf{u}F_A(\mathbf{v})\mathbf{w}\; d\Omega$$

is symmetric for $\mathbf{u},\mathbf{w} \in H_0^1(\Omega)$. The second part $\mathbf{Div}\Big(\mu\frac{1}{2}\left(\nabla\mathbf{w} + \nabla\mathbf{w}^t - \frac{1}{3}\nabla\cdot\mathbf{w}I\right)\Big)$ is symmetric because its A itself. We have to proof the symmetry only for the first part.

$$\begin{aligned}
a'_{\mathbf{v}}(\mathbf{u},\mathbf{w}) - a(\mathbf{u},\mathbf{w}) &= \int\limits_{\Omega}\mathbf{u}F_A(\mathbf{v})\mathbf{w}\;d\Omega - \int\limits_{\Omega}\mathbf{u}A\mathbf{w}\;d\Omega\\
&= \int\limits_{\Omega}\mathbf{u}\cdot\mathbf{Div}\left(\left[\frac{1}{2}kr\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}^{\,r-2}\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{w})\right]\dot{\hat{\varepsilon}}(\mathbf{v})\right)\;d\Omega\\
&= -\int\limits_{\Omega}\nabla\mathbf{u} : \dot{\hat{\varepsilon}}(\mathbf{w})\left[\frac{1}{2}kr\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}^{\,r-2}\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{w})\right]\;d\Omega\\
&\quad \underbrace{+\int\limits_{\partial\Omega}\left[\frac{1}{2}kr\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}^{\,r-2}\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{w})\right]\;\dot{\hat{\varepsilon}}\mathbf{nu}\;dS}_{=0\text{ because }\mathbf{u}\in H_0^1(\Omega)\rightarrow\mathbf{u}=0\text{ on }\partial\Omega}\\
&= -\int\limits_{\Omega}\nabla(\mathbf{u}) : \dot{\hat{\varepsilon}}(\mathbf{w})\left[\frac{1}{2}kr\sqrt{\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{v})}^{\,r-2}\dot{\hat{\varepsilon}}(\mathbf{v}) : \dot{\hat{\varepsilon}}(\mathbf{w})\right]\;d\Omega
\end{aligned}$$

The second factor is obviously symmetric and assumes the the role of the viscosity in the initial definition of A. The first factor is also symmetric, which has already been used to prove the symmetry of A.

A.3 Fréchet derivative of D

The dissipation is defined by

$$D(\mathbf{u}) = \tau(\dot{\hat{\varepsilon}}(\mathbf{u})) : \varepsilon(\mathbf{u})$$

To derive F_D we first apply the product rule and then the chain rule.

$$\begin{aligned} \frac{\delta D}{\delta \mathbf{u}} &= \frac{\delta \tau}{\delta \dot{\hat{\varepsilon}}} \frac{\delta \dot{\hat{\varepsilon}}}{\delta \mathbf{u}} : \varepsilon(\mathbf{u}) + \tau(\dot{\hat{\varepsilon}}(\mathbf{u})) \frac{\delta \varepsilon}{\delta \mathbf{u}} \\ &= \frac{\delta \tau}{\delta \dot{\hat{\varepsilon}}} \dot{\hat{\varepsilon}}(\delta \mathbf{u}) : \varepsilon(\mathbf{u}) + \tau(\dot{\hat{\varepsilon}}(\mathbf{u})) \varepsilon(\delta \mathbf{u}) \end{aligned}$$

The remaining substitutions are identical to those applied to derive F_A.

Appendix B

B.1 Selbständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und unter Verwendung der angegebenen Hilfsmittel, persönlichen Mitteilungen und Quellen angefertigt habe.

Jena, 04.08.2008

B.2 Curriculum vitae

Angaben zur Person

Name:	Markus Müller
Eltern:	Detmar Müller ∗20.8.1940, †16.01.1985 Veronika Müller, geb. Jack ∗14.1.1942
Geburtsdatum:	16.10.1975
Geburtsort:	Sondershausen
Staatsangehörigkeit	Deutsch
Familienstand	verheiratet
Adresse:	Friedrich-Engels-Str. 83 07749 Jena

Schulbildung

08/1981 bis 08/1990	Polytechnische Oberschule Anton Saefkow, Sondershausen
09/1990 bis 08/1993	Staatliches Gymnasium "Prof. Dr. Irmisch", Sondershausen
Schulabschluss:	allgemeine Hochschulreife (Durchschnittsnote 1,3)

Studium

Zeitraum:	10/1995 bis 09/2001
Universität	Friedrich-Schiller-Universität Jena
Studienfach:	Gymnasiallehramt für Mathematik und Physik
Hochschulabschluss:	Erstes Staatsexamen
Examensarbeit:	“Quantenstatistische Eigenschaften des Strahlungsfeldes am Strahlteiler”

Arbeitsverhältnisse

10/1993 bis 11/1993	Grundwehrdienst in Saarlouis
11/1993 bis 12/1993	Praktikum an einer Schule für Kinder mit geistiger Behinderung
02/1994 bis 12/1994	Freiwilliges Soziales (Halb)Jahr in der ambulanten Altenpflege
12/1994 bis 12/1995	Zivildienst: Alten- und Pflegeheim des Deutschen Roten Kreuzes in Sondershausen
09/2001 bis 08/2002	Referendariat am Staatlichen Studienseminar für Lehrerausbildung Erfurt
09/2002 bis 06/2003	diakonisches Jahr beim CVJM Thüringen in Erfurt nebenberuflich Nachhilfelehrer bei der Schülerhilfe in Erfurt
seit 07/2003	Wissenschaftlicher Mitarbeiter am Institut für Geowissenschaften der Friedrich-Schiller-Universität Jena

Jena, 04.08.2008
Markus Müller

B.3 Poster und Vorträge

Poster

August 2005
Integration of variable viscosity in the numerical simulation of global mantle convection of Mars
Köstler, Sändig, Müller, Walzer, Hendel
Kolloquium, DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets

Deutsche Gesellschaft für Luft- und Raumfahrt, Berlin Adlershof

August 2006
A stable grid refinement for modelling martian mantle convection
Müller, Sändig, Köstler
Untergruppenworkshop, DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets
Mainz

August 2006
An improved multigrid algorithm for modelling thermal convection in the martian mantle
Köstler, Müller, Sändig
Untergruppenworkshop, DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets
Mainz

Vorträge

August 2006
Martian mantle convection, numerical improvements of the Terra code
Müller, Köstler, Sändig
Untergruppenworkshop, DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets
Mainz

September 2006
An inf-sup stable local grid refinement
Müller
FEM-Symposium
Chemnitz

Februar 2007
Mantle convection in terrestrial planets. New possibilities of numerical modelling
Köstler, Müller, Sändig, Walzer, Hendel, Baumgardner
DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets
Deutsche Gesellschaft für Luft- und Raumfahrt, Berlin Adlershof

November 2007
Improving numerics in modelling mantle convection, iterative solver
Köstler, Müller, Baumgardner
Untergruppenworkshop, DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets
Köln

Februar 2008
Local grid refinement and iterative solvers for modeling mantle convection
Köstler, Müller, Sändig, Baumgardner
Untergruppenworkshop, DFG Schwerpunktprogramm SPP 15: Mars and the terrestrial planets
Münster

Bibliography

[1] Albers Michael. A local mesh refinement multigrid method for 3-d convection problems with strongly variable viscosity. *Journal of Computational Physics*, 160:126–150, 2000.

[2] R. E. Alcouffe, Achi Brandt, Jr. J. E. Dendy, and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM Journal on Scientific and Statistical Computing*, 2(4):430–454, 1981.

[3] Arnold Krechel and Klaus Stüben. Parallel algebraic multigrid based on subdomain blocking. *Parallel Computing*, 27:1009–1031, 2001.

[4] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[5] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuss, H. RentzReichert, and C. Wieners. Ug – a flexible software toolbox for solving partial differential equations, 1997.

[6] John Baumgardner. *A Three-Dimensional Finite Element Model for Mantle Convection*. PhD thesis, University of California, Los Angeles, 1983.

[7] John R. Baumgardner and Paul O. Frederickson. Icosahedral discretization of the two-sphere. *SIAM J. Numer. Anal.*, 22:1107–1115, 1985.

[8] Kent Beck. *Test-driven development*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[9] Christine Bernardi, Claudio Canuto, and Yvon Maday. Generalized inf-sup conditions for chebyshev spectral approximation of the stokes problem. *SIAM Journal on Numerical Analysis*, 25(6):1237–1271, 1988.

[10] Christine Bernardi, Frédéric Laval, Brigitte Métivet, and Bernadette Pernaud-Thomas. Finite element approximation of viscous flows with varying density. *SIAM Journal on Numerical Analysis*, 29(5):1203–1243, 1992.

[11] J. M. Boland and R. A. Nicolaides. Stability of finite elements under divergence constraints. *SIAM Journal on Numerical Analysis*, 20(4):722–731, 1983.

[12] D. Braess and R. Sarazin. An efficient smoother for the stokes problem. *Appl. Numer. Math.*, 23(1):3–19, 1997.

[13] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A multigrid tutorial: second edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[14] H.-P. Bunge and J. R. Baumgardner. Mantle convection modeling on parallel virtual machines. *Computers in Physics*, 9:207–215, March 1995.

[15] F. H. Busse. Patterns of convection in spherical shells. *Journal of Fluid Mechanics*, 72:67–+, March 2006.

[16] Geneviève Raugel Christine Bernardi. Analysis of some finite elements for the stokes problem. *Mathematics of Computation*.

[17] Andrew J. Cleary, Robert D. Falgout, Van Enden Henson, and Jim E. Jones. Coarse-grid selection for parallel algebraic multigrid. In *Workshop on Parallel Algorithms for Irregularly Structured Problems*, pages 104–115, 1998.

[18] D. Schötzau, C. Schwab. On the inf-sup condition for mixed hp-fem on meshes with hanging nodes. *Mathematical Models and Methods in Applied Sciences*, 8:787–820, 1998.

[19] D. Schötzau, C. Schwab, R. Stenberg. Mixed hp-fem on anisotropic meshes ii: Hanging nodes and tensor products of boundary layer meshes. *Numerische Mathematik*, 83:667–697, 1999.

[20] J. E. Dendy, Jr. Black Box Multigrid. *Journal of Computational Physics*, 48:366–+, December 1982.

[21] Dune. Dune Web page, 2001. www.dune-project.org/.

[22] R.D. Falgout. A note on the relationship between adaptive amg and pcg. *Lawrence Livermore National Laboratory (LLNL), Livermore, CA Report UCRL-TR-205838*, August, 2004.

[23] Michel Fortin. Old and new finite elements for incompressible flows. *International Journal for Numerical Methods in Fluids*.

[24] Franco Brezzi, Michel Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag New York, Berlin, Heidelberg, 1991.

[25] Abteilung für angewandte Mathematik Albert-Ludwigs-Universität Freiburg. ALUGrid Web page, 2001. www.mathematik.uni-freiburg.de/IAM/Research/alugrid/.

[26] K.-D. Gottschaldt, U. Walzer, R. Hendel, D. R. Stegman, J. R. Baumgardner, and H.-B. Mühlhaus. Stirring in 3-d spherical models of convection in the Earth's mantle. *Philosophical Magazine*, 86:3175–3204, 2006.

[27] G. Haase. A parallel AMG for overlapping and non-overlapping domain decomposition. *Elect. Trans. Numer. Anal.*, 10:41–55, 2000.

[28] Gundolf Haase, Michael Kuhn, and Stefan Reitzinger. Parallel algebraic multigrid methods on distributed memory computers. *SIAM J. Sci. Comput.*, 24(2):410–427, 2002.

[29] Wolfgang Hackbusch. *Multi-Grid Methods and Applications*. Springer, first edition, 1985.

[30] Arbitrary Lagrangian Eulerian finite element analysis of free surface flow Henning Braess, Peter Wriggers. *Computer Methods in Applied Mechanics and Engineering*, pages 95–109, October 2000.

[31] Van Emden Henson and Ulrike Meier Yang. Boomeramg: a parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41(1):155–177, 2002.

[32] Heuveline V. ,Schieweck F. . On the inf-sup condition for mixed hp-fem on meshes with hanging nodes. *ESAIM Mathematical Modelling and Numerical Analysis*, 41:1–20, 2007.

[33] Howard C. Elman, David J. Silvester and Andrew J. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, 2005.

[34] A. T. Hsui, W.-S. Yang, and J. R. Baumgardner. A preliminary study of the effects of some flow parameters in the generation of poloidal and toroidal energies within a 3-D spherical thermal-convective system with variable viscosity. *Pure and Applied Geophysics*, 145:487–503, September 1995.

[35] Andrew Hunt and David Thomas. *The pragmatic programmer: from journeyman to master*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[36] Jr. J. E. Dendy. Two multigrid methods for three-dimensional problems with discontinuous and anisotropic coefficients. *SIAM Journal on Scientific and Statistical Computing*, 8(5):673–685, 1987.

[37] S.-I. Karato and P. Li. Diffusion Creep in Perovskite: Implications for the Rheology of the Lower Mantle. *Science*, 255:1238–1240, March 1992.

[38] S.-I. Karato and P. Wu. Rheology of the upper mantle: A synthesis. *Science*, 260:771–778, May 1993.

[39] M. Khalil and P. Wesseling. Vertex-centered and cell-centered multigrid for interface problems. *J. Comput. Phys.*, 98(1):1–10, 1992.

[40] Barry Koren. Multigrid and defect correction for the steady navier-stokes equations. *J. Comput. Phys.*, 87(1):25–46, 1990.

[41] Larin, M. Reusken, A. A comparative study of efficient iterative solvers for generalized Stokes equations. *NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS*, 15(1):13, 2008.

[42] G. Matthies and L. Tobiska. The inf-sup condition for the mapped qk - pk-1disc element in arbitrary space dimensions. *Computing*, 69(2):119–139, 2002.

[43] O.A. Ladyzhenskaya. *The Mathematical Theory of Viscous Incompressible Flow*. Gordon and Breach New York, 1969.

[44] S. V. Patankar. *Numerical heat transfer and fluid flow*. Washington, DC, Hemisphere Publishing Corp., 1980. 210 p., 1980.

[45] K. Friedrichs R. Courant and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[46] K. Friedrichs R. Courant and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal*, pages 215–234, 1967.

[47] G. Ranalli. The microphysical approach to mantle rheology. *Glacial Isostacy, Sea Level and Mantle Rheology, ed. R. Sabadini Kluwer Academic Publishers, Dordrecht*, pages 343–78, 1991.

[48] Rolf Rannacher. Numerische Mathematik 2 (Numerik partieller Differentialgleichungen) Vorlesungsscriptum, 2004.

[49] Rainer Schmachtel. *Robuste lineare und nichtlineare Lösungsverfahren für die inkompressiblen Navier-Stokes-Gleichungen*. PhD thesis, Universität Dortmund, 2003.

[50] Yair Shapira. *Matrix-based multigrid : theory and applications*. Boston: Kluwer Academic Publishers, 2003.

[51] Yair Shapira, Moshe Israeli, and Avram Sidi. Towards automatic multigrid algorithms for spd, nonsymmetric and indefinite problems. *SIAM J. Sci. Comput.*, 17(2):439–453, 1996.

[52] K. Stemmer, H. Harder, and U. Hansen. A new method to simulate convection with strongly temperature- and pressure-dependent viscosity in a spherical shell: Applications to the Earth's mantle. *Physics of the Earth and Planetary Interiors*, 157:223–249, August 2006.

[53] Masahisa Tabata. Slip boundary conditions and rigid body movements infinite element analysis. *Advances in numerical mathematics*, 12(6):117–124, 1999.

[54] Paul J. Tackley. Modelling Compressible Mantle Convection with Large Viscosity Contrasts in Three Dimensional Spherical Shell Using the Yin-Yang Grid. *Physics of the Earth and Planetary Interior, PEPI*, submitted on 30 October 2007.

[55] R. Trompert and U. Hansen. Mantle convection simulations with rheologies that generate plate-like behaviour. *Nature*, 395:686–689, October 1998.

[56] R. A. Trompert and U. Hansen. The application of a finite volume multigrid method to three-dimensional flow problems in a highly viscous fluid with a variable viscosity. *Geophysical and Astrophysical Fluid Dynamics*, 83:261–291, December 1996.

[57] Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schüller. *Multigrid*. Academic Press (San Diego), 2001.

[58] S. P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65:138–158, July 1986.

[59] R. Verfürth. Finite element approximation of incompressible navier-stokes equations with slip boundary condition. *Numer. Math.*, 50(6):697–721, 1987.

[60] R. Verfürth. Finite element approximation of incompressible navier-stokes equations with slip boundary condition ii. *Numer. Math.*, 59(6):615–636, 1987.

[61] Vivette Girault, Pierre-Arnaud Raviart. *Finite Element Methods for Navier-Stokes Equations*. Springer-Verlag New York, Berlin, Heidelberg, 1986.

[62] U. Walzer and R. Hendel. Tectonic episodicity and convective feed-back mechanisms. *Phys. Earth Planet. Int.*, 100:167–188, 1997.

[63] U. Walzer and R. Hendel. Time-dependent thermal convection, mantle differentiation, and continental crust growth. *Geophys. J. Int.*, 130:303–325, 1997.

[64] U. Walzer and R. Hendel. A new convection-fractionation model for the evolution of the principal geochemical reservoirs of the Earth's mantle. *Phys. Earth Planet. Int.*, 112:211–256, 1999.

[65] U. Walzer and R. Hendel. A new convection-fractionation model for the evolution of the principal geochemical reservoirs of the Earth's mantle. *EOS Transactions*, 80(46):F1171, 2000.

[66] U. Walzer and R. Hendel. Mantle convection and evolution with growing continents. *J. Geophys. Res.*, accepted, May 6 2008.

[67] U. Walzer, R. Hendel, and J. Baumgardner. The effects of a variation of the radial viscosity profile on mantle evolution. *Tectonophysics*, 384:55–90, 2004.

[68] U. Walzer, R. Hendel, and J. Baumgardner. Whole-mantle convection, continent generation, and preservation of geochemical heterogeneity. In W.E. Nagel, W. Jäger, and M. Resch, editors, *High Performance Computing in Science and Engineering '07*, pages 603–645. Springer, Berlin, 2008.

[69] J. Weertman. The Creep Strength of the Earth's Mantle. *Reviews of Geophysics and Space Physics*, 8:145–+, February 1970.

[70] P. Wesseling. Cell-Centered Multigrid for Interface Problems. *Journal of Computational Physics*, 79:85–+, November 1988.

[71] Roland Wolters. Entwicklung einer Testsuite zur numerischen Simulation der Konvektion im Erdmantel, Diplomarbeit, Friedrich-Schiller-Universität Jena, 2008.

[72] Woo-Sun Yang. *Variable Viscosity Thermal Convection at Infinite Prandtl Number in a Thick Spherical Shell.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[73] S. Zhong and M. Gurnis. Interaction of weak faults and non-newtonian rheology produces plate tectonics in a 3D model of mantle flow. *Nature*, 383:245–247, September 1996.