A Spectral Element Method for Durran's Pseudo-incompressible Model

by

Geir Kokkvoll Engdahl

THESIS

presented for the degree of

MASTER OF SCIENCE IN COMPUTATIONAL SCIENCE AND ENGINEERING

Faculty of Mathematics and Natural Sciences
University of Oslo

May 2007

Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo

Acknowlegdements

This thesis is submitted for the degree of Master of Science at the University of Oslo. The work was carried out mostly at the Norwegian Defence Research Establishment (FFI) from January 2006 to May 2007. I would like to thank both FFI and the University of Oslo for providing good working conditions during this period.

My supervisors have been Dr. Carl Erik Wasberg and Dr. Øyvind Andreassen. I am extremely grateful for their insights and helpful advice. Big thanks also to Dr. Thor Gjesdal, Dr. Anders Helgeland and Dr. B. A. Petterson Reif at FFI for taking interest in this work and for their useful suggestions.

The challenges turned out to be greater than I had anticipated. I have spent many a day wondering "what went wrong?" and debugging either the code or the equations. During dark moments, I wondered whether or not I would produce any results at all. On such days, it is important to have someone to tell you to "keep moving forward". I thank my family and my girlfriend Lily for encouraging me and believing in me.

Geir K. Engdahl

Slependen, May 21, 2007

Contents

Abstract

A 3-d spectral element method is presented for Durran's pseudo-incompressible model for atmospheric flows. The model is valid in deeply stratified flows and conserves a form of energy close to that which the fully compressible system conserves. Durran's method allows for longer time-steps than the fully compressible model. The spectral element method provides high computational accuracy. The solver supports parallel execution in supercomputers or clusters of computers through domain decomposition.

Solutions are presented for three physical cases: A differentially heated cavity benchmark case, the St. Andrew's Cross formed by internal gravity waves in a continuously stratified fluid, and the breaking of internal gravity waves in two and three dimensions. The behaviour of the computed solutions agree well with cited literature in both linear and non-linear cases.

Chapter 1

Introduction

In this thesis, a partial differential equation solver is developed, mainly to study gravity waves in the atmosphere. Internal gravity waves can arise for instance from winds passing over mountain ranges, called mountain- or lee waves. The study of such waves is important in order to understand the middle atmospheric winds.

To simulate these phenomena, we need to choose an appropriate set of governing equations (a mathematical model), and develop a solution algorithm for this chosen set of equations. The first chapter of this thesis quickly presents the most relevant model equations before it goes on to explain the basics of the chosen numerical solution framework. Chapter two explains the treatment of the mathematical model in more detail. Chapter three shows how the discretisation of the equations leads to systems of linear equations and explicit integration problems and how to solve these. The fourth chapter explains how the numerical model is implemented as a parallel computer program. Chapter five contains the results of a few simulations, while chapter six has some concluding remarks.

1.1 Models for Atmospheric Flows

Various models exist for the purpose of investigating atmospheric phenomena. Their complexity and application vary. A too simple model will fail to accurately represent the physics of interest in many cases. With a too complex model, it might not be feasible to compute the desired information at all, or the programming effort might not be worth the extra accuracy.

The main goal of this thesis has been to be able to produce accurate numerical results on the behaviour of internal gravity waves in deeply stratified, i.e. very tall atmospheres. In one of the cases presented in chapter five, the density decreases by three orders of magnitude from the bottom to the top of the simulation domain. Such very tall atmospheres require specialised mathematical models.

We will try to solve for the pressure p, temperature T, density ρ and the three velocity components $\vec{u} = u\vec{i} + v\vec{j} + w\vec{k}$ of the fluid. The 'Perfect Gas Law' gives an analytical relation

$$p = \rho RT$$

that eliminates one of the state variables. R is the specific gas constant, which is around 287 *J/(kg K)* for air. We only need to find five other equations to solve for the remaining quantities. The pressure and temperature state variables have been replaced by the *Exner function*

$$\pi = \left(\frac{p}{p_0}\right)^{R/c_p}$$

and the *potential temperature*

$$\theta = \frac{T}{\pi}$$

in some of the models mentioned below. Here, p_0 represents a constant reference pressure and c_p is the specific heat of the fluid at constant pressure. The potential temperature is a convenient quantity, since convective instabilities are easy to spot when using it. If the vertical gradient of the potential temperature is negative ($d\theta/dz < 0$), then that region is unstable.

1.1.1 The Fully Compressible Model

Omitting Coriolis and viscous terms, the fully compressible governing equations for a perfect gas are

$$\begin{aligned} \rho\frac{D\vec{u}}{Dt} &= -\nabla p + \rho\vec{g} \\ \rho c_v\frac{DT}{Dt} &= -\nabla\cdot\vec{q} - p(\nabla\cdot u) \\ \frac{1}{\rho}\frac{D\rho}{Dt} &+ \nabla\cdot\vec{u} = 0, \end{aligned} \tag{1.1}$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{u}\cdot\nabla$$

is the substantive derivative. It is common to make some assumptions to further simplify the thermal energy equation (1.1) and still name the model fully compressible. For instance, the heat flux $\vec{q}$ is often assumed to obey Fourier's law ($\vec{q} = -k\nabla T$).

The fully compressible model can describe all the physical phenomena that we are interested in. The challenge, however, is to accurately solve these equations. The model supports sound waves, which are small fluctuations in density that move with a much higher speed than the other flow structures typically do. Because of the high speeds, it is necessary to have a very small time-step if this model is to be solved numerically. Nance and Durran [1], in their comparison between the anelastic and fully compressible models, have quoted time steps of size $\Delta t = 0.25s$ for the fully compressible model and $\Delta t = 10s$ for the anelastic models in some cases. It seems that there is a substantial computational penalty associated with the fully compressible model. Since sound wave propagation is not of any interest, we choose not to use the fully compressible model.

1.1.2 The Boussinesq Approximation

It is acceptable to neglect the density variations in the governing equations under certain circumstances. The resulting model is justified if the Mach number of the flow is lower than 0.3, the propagation of sound waves is of no interest, the vertical stratification is sufficiently small ($\ll 10km$ in the Earth's atmosphere) and the temperature differences in the fluid are small [2].

Let us decompose the density $\rho = \rho_0 + \rho'$, where ρ_0 is the constant density in a hydrostatic equilibrium state with $p = p_0(z)$ and $\nabla p_0 = \rho_0 \vec{g}$. If we neglect the effect of density variations everywhere except in the buoyancy term, we get the Boussinesq momentum equation

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho_0} \nabla p = \frac{\rho'}{\rho_0} \vec{g} + \nu \nabla^2 (\vec{u}).$$

If we were to ignore the density variations also in the buoyancy term, there would be no buoyant motion at all. That would mean that a blob of fluid with a higher density than its surroundings would not sink!

The properties of the fluid such as kinematic viscosity ν and thermal diffusivity κ are considered to be constant. For a perfect gas satisfying Fourier's law of heat conduction, we get the thermal energy equation

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = \kappa \nabla^2 (T).$$

By assuming a constant density in the continuity equation we get the incompressible form

$$\nabla \cdot \vec{u} = 0.$$

We will not use the Boussinesq approximation, other than for comparison purposes, since we are interested in atmospheres much taller than the Boussinesq model can handle.

1.1.3 The Anelastic Models

There are several systems of governing equations in the anelastic category. They all share the form of the anelastic continuity equation

$$\nabla \cdot (\bar{\rho} \vec{u}) = 0, \tag{1.2}$$

which was first discussed by Batchelor [3]. However, the definition of the background density profile $\bar{\rho}$ is different for the various models. The anelastic heat equation

$$\frac{D\theta}{Dt} = \frac{\mathcal{H}}{c_p},$$

where $\mathcal{H}$ is the heating per unit volume, is also similar for the anelastic models discussed here. In the following discussion of the models, viscous effects are omitted. A

viscous friction term may be inserted in the momentum equations, with the appropriate stratified viscosity for each model.

The anelastic models are stated using the potential temperature and the Exner function instead of the basic temperature and pressure quantities. The state variables have also been decomposed into a reference background denoted with a bar, and a perturbation denoted primed, i.e $\rho = \bar{\rho} + \rho'$. The background quantities are either constant or vertically varying, and must sometimes also satisfy additional constraints.

Ogura and Phillips [4] derived their anelastic system through a scale analysis and ended up with the following momentum equation:

$$\frac{D\vec{u}}{Dt} + c_p \Theta \nabla \pi' = \frac{\vec{g}\theta'}{\Theta}.$$

Here, Θ is a constant potential temperature in an isentropic, i.e. adiabatic and frictionless, reference state. An adiabatic process takes place without any heat change.

One of the assumptions that Ogura and Phillips made was that the deviations from the constant background potential temperature Θ are small. Again this is an unacceptable assumption, since the potential temperature will deviate by as much as an order of magnitude between the top and bottom of our simulation domain.

Wilhelmson and Ogura [5] developed another model by assuming a vertically varying background potential temperature $\bar{\theta}(z)$. Their momentum equation is

$$\frac{D\vec{u}}{Dt} + c_p \bar{\theta} \nabla \pi' = \frac{\vec{g}\theta'}{\bar{\theta}}.$$

This model does not preserve the energy of finite-amplitude disturbances.

Lipps and Hemler [6] were able to produce a system that conserves energy in the case of a slowly varying background potential temperature $\bar{\theta}(z)$. The momentum equation in this system is

$$\frac{D\vec{u}}{Dt} + c_p \nabla(\pi' \bar{\theta}) = \frac{\vec{g}\theta'}{\bar{\theta}}.$$

This model is valid if the perturbations in the potential temperature and the Exner function are small compared to the vertically varying background state. This is not the case in the heated cavity benchmark case presented later in this thesis. Also, the background potential temperature is assumed to be slowly varying, which is not the case in the stratosphere [7].

1.1.4 The Pseudo-Incompressible Model

Durran [7] introduced the pseudo-incompressible system

$$\frac{D\vec{u}}{Dt} + c_p \theta \nabla \pi' = \frac{\vec{g}\theta'}{\bar{\theta}}, \tag{1.3}$$

$$\nabla \cdot (\bar{\rho}\bar{\theta}\vec{u}) = \frac{\mathcal{H}}{c_p \bar{\pi}}, \tag{1.4}$$

$$\frac{D\theta}{Dt} = \frac{\mathcal{H}}{c_p \bar{\pi}} \frac{\theta}{\bar{\rho}\bar{\theta}}. \tag{1.5}$$

It is important to note that unlike the original anelastic model by Batchelor, the background quantities in Durran's model need not form an adiabatic state.

Durran derived the pseudo-incompressible equation from the equation of state of dry air

$$\pi = \left(\frac{R}{p_0}\rho\theta\right)^{R/c_v},$$

where c_v is the specific heat of the gas during constant volume processes. The only assumptions he made were that

$$\pi' \ll \bar{\pi},$$

and that the time-scale associated with the perturbations is much longer than that of sound wave propagation. If the perturbation time-scale is similar to that of sound waves, the time-steps for the pseudo-incompressible model would also have to be very small, and there would not be much to save in terms of computational efficiency anyway.

The pseudo-incompressible system conserves a form of energy

$$E = \frac{\bar{\rho}\bar{\theta}}{\theta}\left(\frac{u^2+v^2+w^2}{2} + gz\right) + c_v\bar{\rho}\bar{T}.$$

The only difference compared to the fully compressible energy conservation is that the actual density ρ is replaced by $\frac{\bar{\rho}\bar{\theta}}{\theta}$.

1.2 The Spectral Element Method

1.2.1 The Element Method

Element methods approximate the solution of partial differential equations (PDEs) within a pre-selected function space. The derivation of the numerical model for an element method is usually more involved than for finite difference methods, which approximate the differentiation operators by finite differences. This extra complexity can sometimes be justified by the straightforward handling of geometrically complicated domains and grids, and the flexibility that arises from being able to chose the function space that is used to approximate the solution.

Assume we want to solve the PDE

$$\mathcal{L}(u(\vec{x})) = 0$$

on the domain Ω where $\mathcal{L}$ is some differential operator. We try to approximate the solution $u(\vec{x})$ with a linear combination

$$u(\vec{x}) \approx \hat{u}(\vec{x}) = \sum_{j=1}^{n} u_j h_j(\vec{x})$$

of basis functions $h_j(\vec{x})$. Each basis function is normally only non-zero on a subdomain of Ω, sometimes referred to as an *element*. Solving the PDE is now reduced to finding

the coefficients u_j that somehow give the 'best' approximation to the real solution. There are many good candidates for norms that can be used to measure how well a function approximates another. Intuitively, it seems that to minimise the maximum absolute error is the way to go. Unfortunately, this norm is very hard to deal with. It is much simpler to minimise the sum or integral of the square of the error, called least-squares approximation. Now since we do not generally know the exact solution, we need to work with the residual

$$R(\hat{u}) = \mathcal{L}(\hat{u}),$$

i.e. what is left over when we insert our approximated solution into the PDE. With least squares, it gives the problem

$$\min_{u_j \in \mathbb{R}} \left(\int_\Omega R(\hat{u}, \vec{x})^2 d\Omega \right).$$

This can be differentiated with respect to the unknown coefficients u_j, so that instead we get an equation

$$\int_\Omega R(\hat{u}, \vec{x}) \frac{\partial R(\hat{u}, \vec{x})}{\partial \hat{u}} d\Omega = 0, \tag{1.6}$$

that can be solved to find the unknown coefficients $\hat{u}$. If the basis functions are chosen so that the rest of the integral can be evaluated (either analytically or numerically), then we are left with a linear system of equations to solve. Piecewise linear basis functions (triangle hats) constitute a popular choice, because they yield relatively simple systems. Note that instead of equation (1.6), it is common to use a slightly different version, replacing the differentiated residual with a more convenient 'weight' function, e.g.

$$\int_\Omega R(\hat{u}, \vec{x}) h_j(x) d\Omega = 0, \qquad j = 1...n$$

which is called a Galerkin method, because the weight functions are the same as the basis functions. This is what is used in the code for this project. For a better introduction to element methods see e.g. Langtangen [8].

1.2.2 The Spectral Element Method

A key question is how to chose the basis functions so that we get the most accurate solution, while still arriving at a linear system of equations that can be solved efficiently. Most common choices, including triangle hats, lead to polynomial accuracy, e.g. the error satisfies

$$||u - \hat{u}|| \leq C \Delta x^p$$

where C is some constant, Δx is the grid spacing and p is a constant, usually 1 or 2. The spectral element method uses basis functions that allow for faster spatial convergence. *Spectral accuracy* means that the error is bounded by

$$||u - \hat{u}||_2 \leq C e^{-n}.$$

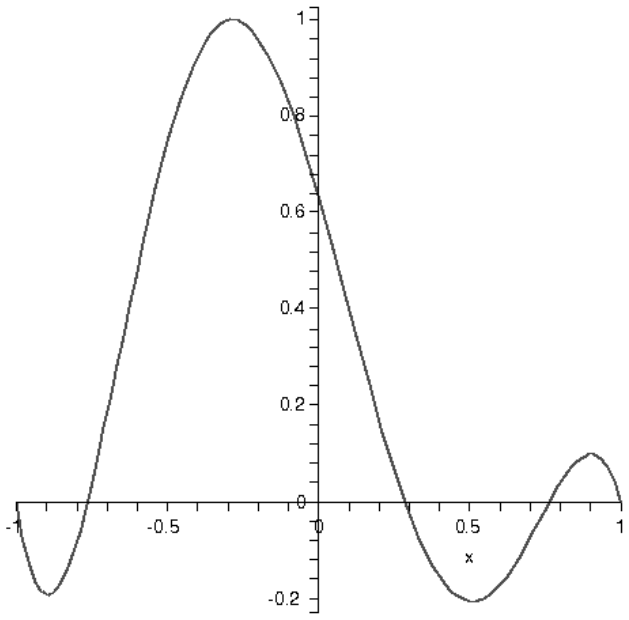

Figure 1.1: The third basis function of a six point Gauss-Lobatto-Legendre grid

i.e. decays exponentially with the number of terms in a series expansion of the solution on each element. The spectral element method that is used here has a family of orthonormal polynomials as a basis for the solution. An orthonormal family of functions $h_i(x)$ with respect to the inner product $\langle \cdot, \cdot \rangle$ satisfies

$$\langle h_i, h_j \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

In order to make numerical integration as accurate and simple as possible, the polynomials are chosen from the Gauss-Lobatto quadrature points. It is possible to integrate a $(2n-1)$th degree polynomial $f(x)$ exactly with the quadrature rule

$$\int_{-1}^{1} f(x)dx = \sum_{j=1}^{n} f(x_j)w_j$$

if the points x_j and the weights w_j are chosen carefully. The Gauss Lobatto points are $-1, 1$, and the extremal points of the $n-1$th degree Legendre polynomial [9]. Assuming that there are n Gauss-Lobatto points, the k'th basis function is the unique polynomial of degree $n-1$ that has value one in the k'th Gauss-Lobatto point and zero in all the others.

See e.g. Deville et al[10] for more on spectral element methods.

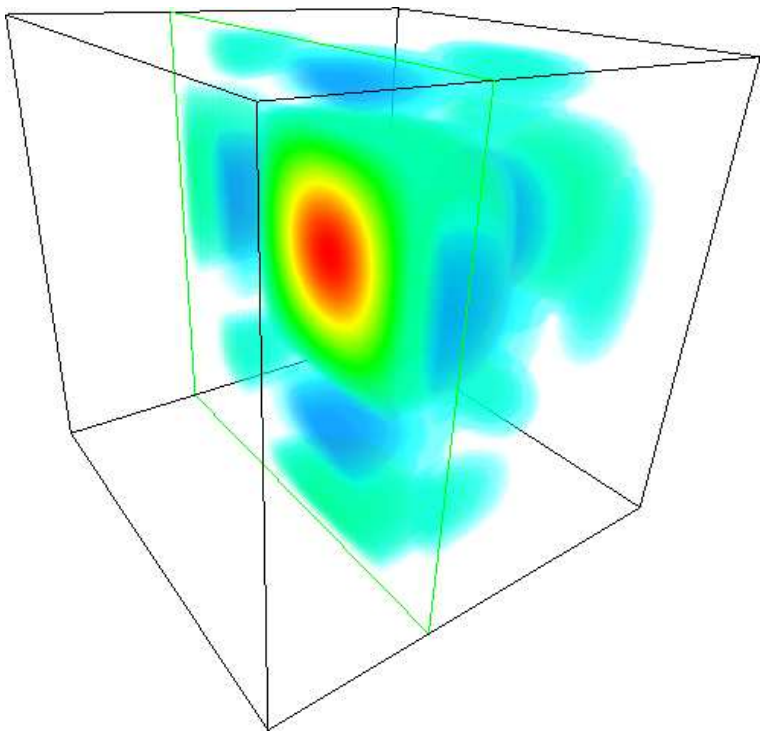

Figure 1.2: The third basis function of a 6 x 6 x 6 point Gauss-Lobatto-Legendre grid (rendered with a clipping plane)

Chapter 2

Mathematical Model

2.1 Durran's Model

We add Coriolis forces, viscous forces and source terms $\vec{F}$, J, to Durran's model (eqs. (1.3) (1.4) (1.5)). From this, we get a closed set of governing equations

$$\frac{Du}{Dt} - 2\Omega v \sin(\alpha) + c_p\theta\frac{\partial\pi'}{\partial x} = F_x(\vec{x},t) + \frac{\mu}{\bar{\rho}(z)}\nabla^2(u), \tag{2.1}$$

$$\frac{Dv}{Dt} + 2\Omega u \sin(\alpha) + c_p\theta\frac{\partial\pi'}{\partial y} = F_y(\vec{x},t) + \frac{\mu}{\bar{\rho}(z)}\nabla^2(v), \tag{2.2}$$

$$\frac{Dw}{Dt} - 2\Omega u \cos(\alpha) + c_p\theta\frac{\partial\pi'}{\partial z} = g\frac{\theta'}{\bar{\theta}} + F_z(\vec{x},t) + \frac{\mu}{\bar{\rho}(z)}\nabla^2(w), \tag{2.3}$$

$$\frac{\partial\theta}{\partial t} + \vec{u}\cdot\nabla\theta = \frac{\mathcal{H}}{c_p\bar{\pi}}\frac{\theta}{\bar{\rho}\bar{\theta}} + \frac{\bar{\theta}}{g}J(\vec{x},t), \tag{2.4}$$

$$\nabla\cdot(\bar{\rho}\bar{\theta}\vec{u}) = \frac{\mathcal{H}}{c_p\bar{\pi}}, \tag{2.5}$$

where the unknowns are the 3-d velocity field $\vec{u} = u\vec{i} + v\vec{j} + w\vec{k}$, the potential temperature θ and the Exner function π. The variables θ and π have been decomposed into temporally constant backgrounds $\bar{\theta}$ and $\bar{\pi}$ and variable perturbations θ' and π'.

$$\pi(\vec{x},t) = \bar{\pi}(z) + \pi'(\vec{x},t)$$

$$\theta(\vec{x},t) = \bar{\theta}(z) + \theta'(\vec{x},t)$$

The potential temperature and the Exner function can be combined to restore the temperature variable through the relation

$$T = \pi\theta.$$

The equation of state for dry air relates the Exner function and the potential temperature to the density:

$$\pi = (\frac{R}{p_0}\rho\theta)^{\frac{R}{c_v}}$$

Symbol	Meaning	Typical value
Ω	Rotation rate of the planet	7.27e-5 s^{-1}
α	Angle north of the equator	0
c_p	Specific heat (of the fluid)	1003.0 $J/(kg\ K)$
$F_{x\|y\|z}(\vec{x},t)$	Momentum forcing term	0
μ	Fluid viscosity	1.8e-5 $Pa\ s$
g	Gravity constant	9.81 m/s^2
$\mathcal{H}$	Heating per unit volume	0
$J(\vec{x},t)$	Heat source term	0
Pr	Prandtl number	0.71

Table 2.1: Overview of symbols.

$$p = \rho RT.$$

It was not necessary to add the Coriolis forces to the momentum equations for the cases presented in this thesis. However, the viscous friction term in the momentum equation is necessary, because it smoothes small-scale structures and makes the algorithm more stable.

2.1.1 Heating

The treatment of the heating term $\mathcal{H}$ is of great importance for the solution algorithms needed to treat the system of equations. If the heat diffusion is to be represented by $\mathcal{H}$, we get

$$\mathcal{H} = \nabla \cdot (k \nabla T) \tag{2.6}$$

where k is the thermal conductivity [11]. It equals

$$k = \frac{c_p \mu}{\mathrm{Pr}}.$$

We will assume that k is approximately constant, and pull it outside of the divergence operator. If we insert $\mathcal{H}$ into the heat and continuity equations, we get

$$\frac{\partial \theta}{\partial t} + u \cdot \nabla \theta = \frac{\mu}{\mathrm{Pr}\bar{\pi}} \frac{\theta}{\bar{\rho}\bar{\theta}} \nabla^2(\pi\theta) + \frac{\bar{\theta}}{g} J(\vec{x},t), \tag{2.7}$$

and

$$\nabla \cdot (\bar{\rho}\bar{\theta}\vec{u}) = \frac{\mu}{\mathrm{Pr}\bar{\pi}} \nabla^2(\pi\theta).$$

Heating is of little importance in some of the physical cases that we investigate later. As can be seen from the thermal convection benchmark case, the treatment of the heating term has not been as successful as could have been hoped for. Therefore, we will consider the model also in the case where $\mathcal{H} = 0$. The derivation of the numerical model will include the full version of $\mathcal{H}$. Once the full method is developed, it is fairly easy to remove the extra heating terms so a simpler method can be tested.

2.2 Linearisation

Notice that the heating introduces a new variable, the Exner function π, into the heat equation. It is also a non-linear term, which will require a different level of computational complexity to solve. As we will see later, the potential temperature θ will be assumed known at the time the continuity equation is solved, so it is not a big problem there.

A closer look at the non-linear terms of the heat equation is needed. Using the perturbation decomposition of π and θ, we get

$$\theta\nabla^2(\pi\theta) = (\bar{\theta} + \theta')\nabla^2(\bar{\pi}\bar{\theta} + \bar{\pi}\theta' + \pi'\bar{\theta} + \pi'\theta').$$

Thus eight terms need to be examined. We explore two alternatives for dealing with these terms. The simplest possible linearisation is to drop the non-linear terms of both equations:

$$\frac{D\theta}{Dt} = \frac{\mu}{\mathrm{Pr}\bar{\pi}\bar{\rho}}\left(\nabla^2(\bar{\pi}\bar{\theta} + \bar{\pi}\theta' + \pi'\bar{\theta}) + \frac{\theta'}{\bar{\theta}}\nabla^2(\bar{\pi}\bar{\theta})\right) + \frac{\bar{\theta}}{g}J(\vec{x}, t), \tag{2.8}$$

$$\nabla \cdot (\bar{\rho}\bar{\theta}\vec{u}) = \frac{\mu}{\mathrm{Pr}\bar{\pi}}\nabla^2(\bar{\pi}\bar{\theta} + \pi'\bar{\theta} + \bar{\pi}\theta').$$

This approach has a few drawbacks, one is that the heating is now inconsistent in the two terms of the heat equation (2.8). There is another problem with dropping the term $\bar{\pi}\theta'$, since Durran does not assume

$$\theta' \ll \bar{\theta}$$

in his derivation of the model. This will impose extra restrictions on the use of the code, which we want to avoid if possible. Lastly, this approach does not remove the problem we have with π' appearing in the heat equation. If we drop π' from the heat equation alltogether, then another inconsistency in the treatment of $\mathcal{H}$ is introduced.

Another approach is to extrapolate some parts of the non-linear terms. We introduce the quantities $\tilde{\pi}'$ and $\tilde{\theta}$. They are extrapolated from the values of π' and θ at previous time-steps, using first or higher-order extrapolation. It remains to investigate the effects of this approach with regards to the stability and accuracy of the scheme and the allowed time-step size.

The chosen approach is to extrapolate terms in the heat equation.

$$\frac{D\theta}{Dt} = \frac{\mu\tilde{\theta}}{\mathrm{Pr}\bar{\pi}\bar{\theta}\bar{\rho}}\nabla^2(\bar{\pi}\bar{\theta} + \bar{\pi}\theta' + \tilde{\pi}'\tilde{\theta}) + \frac{\bar{\theta}}{g}J(\vec{x}, t). \tag{2.9}$$

With this model, we make two additional assumptions that Durran did not make, namely that $\tilde{\theta}$ and $\tilde{\pi}'$ can approximate θ and π' sufficiently well with extrapolation.

Chapter 3

Numerical Scheme

It is essential for the implementation effort that the full set of equations are broken down into manageable parts of similar form. The heat equation has been approximated into equation (2.9) so that it is decoupled from the variable π'. The other equations have been left unmodified.

3.1 Operator Splitting

In order to make the approximated heat equation (2.9) fit into the standard Helmholtz solver, it is necessary to introduce a new quantity

$$\phi = \bar{\pi}\theta = \bar{\phi} + \phi' = \bar{\pi}\bar{\theta} + \bar{\pi}\theta'.$$

Inserting this into equation (2.9), multiplying by $\bar{\pi}$ and reorganising gives

$$\frac{D\phi}{Dt} - \frac{\phi\vec{u}\cdot\nabla\bar{\pi}}{\bar{\pi}} = \frac{\mu\tilde{\theta}}{\mathrm{Pr}\bar{\rho}\bar{\theta}}\left(\nabla^2(\phi) + \nabla^2(\tilde{\pi}'\tilde{\theta})\right) + \frac{\bar{\phi}}{g}J(\vec{x},t),$$

which is split into an explicit part

$$\frac{\partial\phi}{\partial t} + \vec{u}\cdot(\nabla\phi - \frac{\phi\nabla\bar{\pi}}{\bar{\pi}}) = 0, \tag{3.1}$$

and an implicit part

$$\frac{\frac{3}{2}\phi - 2\phi_1^* + \frac{1}{2}\phi_2^*}{\triangle t} - \frac{\mu\tilde{\theta}}{\mathrm{Pr}\bar{\rho}\bar{\theta}}\nabla^2(\phi) = \frac{\mu\tilde{\theta}}{\mathrm{Pr}\bar{\rho}\bar{\theta}}\nabla^2(\tilde{\pi}'\tilde{\theta}) + \frac{\bar{\phi}}{g}J(\vec{x},t). \tag{3.2}$$

Here, ϕ_1^* and ϕ_2^* are the solutions of the explicit part (3.1), and enter the equation through a second-order backwards time differentiation scheme (BDF2). ϕ_i^* models the solution of the explicit problem

$$\begin{aligned}\frac{\partial\phi_i^*(s)}{\partial s} &= A\phi_i^*(s), \qquad 0 < s < t^{n+1} - t^{n+1-i},\\ \phi_i^*(0) &= \phi(t^{n+1-i}),\end{aligned}$$

where

$$A\phi^* = -\vec{u} \cdot \left(\nabla \phi^* - \frac{\phi^* \nabla \bar{\pi}}{\bar{\pi}} \right).$$

The implicit part can be rewritten to fit the variable coefficient Helmholtz equation

$$-\nu \nabla^2(\phi) + \lambda(\vec{x}, t)\phi = f(\vec{x}, t), \tag{3.3}$$

with

$$\nu = \frac{\mu}{\mathrm{Pr}},$$

$$\lambda(\vec{x}, t) = \frac{3\bar{\rho}\bar{\theta}}{2\tilde{\theta}\triangle t},$$

$$f(\vec{x}, t) = \frac{\bar{\rho}\bar{\theta}}{\tilde{\theta}} \left(\frac{4\phi_1^* - 2\phi_2^*}{2\triangle t} + \frac{\bar{\phi}}{g} J(\vec{x}, t) \right) + \frac{\mu}{\mathrm{Pr}} \nabla^2(\tilde{\pi}' \tilde{\theta}).$$

In order to solve the momentum / continuity eqs. (2.1) (2.2) (2.3) (2.5), we use the operator-integration-factor method suggested by Maday et. al. [12], splitting the system into an explicit and an implicit part. The explicit part is

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \mathcal{A}\vec{u} = 0, \tag{3.4}$$

where

$$\mathcal{A} = \begin{bmatrix} 0 & -2\Omega \sin\alpha & 0 \\ 2\Omega \sin\alpha & 0 & 0 \\ -2\Omega \cos\alpha & 0 & 0 \end{bmatrix},$$

and the corresponding implicit problem is

$$\frac{\frac{3}{2}\vec{u} - 2\vec{u_1^*} + \frac{1}{2}\vec{u_2^*}}{\triangle t} - \frac{\mu}{\bar{\rho}} \nabla^2(\vec{u}) + c_p \theta \nabla \pi' = \theta' \frac{g}{\bar{\theta}} \vec{k} + \vec{F}(\vec{x}, t). \tag{3.5}$$

$$\nabla \cdot (\bar{\rho}\bar{\theta}\vec{u}) = \frac{\mu}{\mathrm{Pr}\,\bar{\pi}} \nabla^2(\bar{\pi}\bar{\theta} + \pi'\bar{\theta} + \bar{\pi}\theta' + \pi'\theta').$$

This implicit system will hereafter be referred to as the 'Modified Stokes Problem'. Its discretisation and solution algorithm will be discussed in detail in section 3.2.

The explicit problems in both the heat and momentum equations are solved using a Runge-Kutta (RK) time integration method. The explicit integration scheme can optionally divide each implicit time-step Δt into a number of explicit time steps Δt_*. RK schemes can have a local truncation error of arbitrary order in Δt_*. A second-order RK method will use two function evaluations in each explicit time-step, with higher order methods using more, thus being more expensive. A second and third order RK method is used in the code by Wasberg [13], which is re-used for this project. The stability of the explicit time step length is assessed by looking at the difference between the results of the second and third order methods. The method will decrease the explicit time-step length if the two results disagree too much.

3.2 Discretisation of the Modified Stokes Problem

We want to solve the equations

$$\lambda\vec{u} - \nu(z)\nabla^2(\vec{u}) + \beta(\vec{x},t)\nabla\pi' = \vec{f} \tag{3.6}$$

and

$$\nabla\cdot(\bar{\rho}\bar{\theta}\vec{u}) = \gamma(z)\nabla^2(\theta\bar{\pi}) + \gamma(z)\nabla^2(\pi'\bar{\theta}), \tag{3.7}$$

where

$$\vec{f} = \frac{2}{\Delta t}\vec{u_1^*} - \frac{1}{2\Delta t}\vec{u_2^*} + \frac{g\theta'}{\bar{\theta}}\vec{k} + \vec{F}(\vec{x},t),$$
$$\nu(z) = \frac{\mu}{\bar{\rho}(z)},$$
$$\beta(\vec{x},t) = c_p\theta,$$
$$\lambda = \frac{3}{2\Delta t},$$
$$\gamma(z) = \frac{\mu}{\mathrm{Pr}\bar{\pi}(z)}.$$

3.2.1 Weak Form

The weak formulation of the ith component of eq. (3.6) and eq. (3.7) is: Find $u^i \in X, \pi' \in Y$ so that

$$\lambda\int_\Omega u^i v d\Omega + \int_\Omega \nabla u^i \nabla(\nu v) d\Omega - \int_\Omega \pi' \frac{\partial \beta v}{\partial x_i} d\Omega = \int_\Omega f^i v d\Omega, \qquad \forall v \in X,$$

and

$$\int_\Omega \nabla\cdot\bar{\rho}\bar{\theta}uq d\Omega = \int_\Omega \gamma(z)\nabla^2(\bar{\pi}\theta)q d\Omega + \int_\Omega \gamma(z)\nabla^2(\pi'\theta)q d\Omega, \qquad \forall q \in Y,$$

where X and Y are the function spaces we restrict u^i and π' to, respectively.

The variables will be discretised on two different grids, because the pressure-like quantity π can give rise to spurious modes if solved for on the same grid as the other quantities. This would destroy the uniqueness of the solution [14]. The grids are an N-point Gauss-Lobatto grid (as described in chapter 1) which includes the element boundaries $-1, 1$, and one $N-2$ point Gauss grid with different points that do not include the element boundaries. The Gauss points are based on the zeroes of the $N-3$ degree Legendre polynomial.

Let $\{\sigma_n\}_{n=1}^{N-1}$ and $\{\eta_n\}_{n=1}^{N-1}$ be the Gauss quadrature weights and points respectively, and let $\{\rho_j\}_{j=0}^{N}$ and $\{\xi_j\}_{j=0}^{N}$ be the Gauss-Lobatto quadrature weights and points respectively.

We define two discrete inner products to approximate the integrals:

$$\langle f(x), g(x)\rangle := \sum_{j=1}^{N-1} f(\eta_j)g(\eta_j)\sigma_j$$

$$(f(x), g(x)) := \sum_{j=0}^{N} f(\xi_j) g(\xi_j) \rho_j$$

With these inner products, we can write the approximated weak form of the momentum and continuity equations

$$\lambda(u^i, v) + (\nabla u^i, \nabla(\nu v)) - \langle \pi', \frac{\partial \beta v}{\partial x_i} \rangle = (f, v), \qquad \forall v \in X,$$

and

$$\langle \nabla \cdot (\bar{\rho}\bar{\theta}u), q \rangle = \langle \gamma(z)\nabla^2(\bar{\pi}\theta), q \rangle + \langle \gamma(z)\nabla^2(\pi'\theta), q \rangle, \qquad \forall q \in Y,$$

respectively. Note that partial integration of the viscous friction and Exner gradient terms (Green's Lemma) imposes the boundary conditions

$$\int_{\partial\Omega} \nu v \frac{\partial u}{\partial n} d\Gamma = 0, \qquad \forall v \in X,$$

and

$$\int_{\partial\Omega} \pi' \beta v d\Gamma = 0, \qquad \forall v \in X.$$

It is tempting to avoid the partial integration in the Exner gradient term of the momentum equation. This leads to a simpler form with one term instead of the two we get from the product rule of differentiation of β and v. This was tried and in practice that form was less stable than the one we get from partial integration.

3.2.2 Discretisation in 1-d

The product rule of differentiation and the assumption that we work with the field of real numbers, give

$$\lambda(u, v) + (\nabla u, \nu \nabla v) + (\nabla u, v \nabla \nu) - \langle \pi', \beta \frac{\partial v}{\partial x_i} \rangle - \langle \pi', v \frac{\partial \beta}{\partial x_i} \rangle = (f, v), \quad \forall v \in X,$$

and

$$\begin{aligned} &\langle \bar{\rho}\bar{\theta}\nabla \cdot u, q \rangle + \langle \bar{\rho} u \cdot \nabla \bar{\theta}, q \rangle + \langle \bar{\theta} u \cdot \nabla \bar{\rho}, q \rangle \\ &= \langle \gamma \bar{\pi} \nabla^2(\theta), q \rangle + \langle \gamma \theta \nabla^2(\bar{\pi}), q \rangle + 2\langle \gamma \nabla \bar{\pi} \cdot \nabla \theta, q \rangle \\ &+ \langle \gamma \pi' \nabla^2(\theta), q \rangle + \langle \gamma \theta \nabla^2(\pi'), q \rangle + 2\langle \gamma \nabla \pi' \cdot \nabla \theta, q \rangle, \quad \forall q \in Y. \end{aligned}$$

Let the test functions for the Gauss-Lobatto and the Gauss grid be $v^{(i,k)}(x) = h_i^k(x)$ and $q^{(j,k)} = g_j^k(x)$. The function $h_i^k(x)$ is the Lagrange interpolant that has value zero in all except the ith Gauss-Lobatto grid points of the kth element, where it has value one. Similarly, $g_j^k(x)$ is the Lagrange interpolant of the Gauss grid. Here, and in the following, $i = 0, 1, ..., N$ and $j = 1, 2, ..., N-1$. Further, let $h_{i,r} = h_i(\eta_r)$ when $r = 1, ..., N-1$ and $h_{i,r} = h_i(\xi_r)$ when $r = 0, ..., N$. Similarly, we define $g_{i,r} = g_i(\eta_r)$.

We approximate the variables and background quantities using these basis functions. Note that θ is calculated on a Gauss-Lobatto grid in the heat equation solver, but it is interpolated onto a Gauss grid in the Stokes problem.

$$u^k(x) \approx \sum_{n=0}^{N} u_n^k h_n^k(x), \quad \theta^k(x) \approx \sum_{n=1}^{N-1} \theta_n^k g_n^k(x), \quad \pi^k(x) \approx \sum_{n=1}^{N-1} \pi_n^k g_n^k(x)$$
$$\nu^k(x) \approx \sum_{n=0}^{N} \nu_n^k h_n^k(x), \quad \beta^k(x) \approx \sum_{n=1}^{N-1} \beta_n^k g_n^k(x), \quad \gamma^k(x) \approx \sum_{n=1}^{N-1} \gamma_n^k g_n^k(x)$$

Since X is spanned by the basis functions $h_i(x)$, we can replace the $\forall v \in X$ restriction by requiring that the equation holds for all the basis functions. The same applies to the $\forall q \in Y$ requirement. We note that sums over the K elements appear in all terms, and drop these summations and the superscripts k, to avoid cluttering. From this, we get the equations:

$$\begin{aligned}
\lambda \sum_{r=0}^{N}\sum_{m=0}^{N} u_m h_{m,r} h_{i,r} \rho_r + \sum_{r=0}^{N}\sum_{m=0}^{N}\sum_{n=0}^{N} u_m \nu_n h'_{m,r} h_{n,r} h'_{i,r} \rho_r + \sum_{r=0}^{N}\sum_{m=0}^{N}\sum_{n=0}^{N} u_m \nu_n h'_{m,r} h'_{n,r} h_{i,r} \rho_r \\
- \sum_{r=1}^{N-1}\sum_{m=1}^{N-1}\sum_{n=1}^{N-1} \pi'_m \beta_n g_{m,r} g_{n,r} h'_{i,r} \sigma_r - \sum_{r=1}^{N-1}\sum_{m=1}^{N-1}\sum_{n=1}^{N-1} \pi'_m \beta_n g_{m,r} g'_{n,r} h_{i,r} \sigma_r \\
= \sum_{r=0}^{N}\sum_{m=0}^{N} f_m h_{m,r} h_{i,r} \rho_r, \qquad i = 0...N,
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{r=1}^{N-1}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N} u_{m_1} \bar{\rho}_{m_2} \bar{\theta}_{m_3} h'_{m_1,r} h_{m_2,r} h_{m_3,r} g_{j,r} \sigma_r \\
&+ \sum_{r=1}^{N-1}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N} u_{m_1} \bar{\rho}_{m_2} \bar{\theta}_{m_3} h_{m_1,r} h'_{m_2,r} h_{m_3,r} g_{j,r} \sigma_r \\
&+ \sum_{r=1}^{N-1}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N} u_{m_1} \bar{\rho}_{m_2} \bar{\theta}_{m_3} h_{m_1,r} h_{m_2,r} h'_{m_3,r} g_{j,r} \sigma_r \\
&= \sum_{r=1}^{N-1}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\sum_{m_3=1}^{N-1} \theta_{m_1} \bar{\pi}_{m_2} \gamma_{m_3} g''_{m_1,r} g_{m_2,r} g_{m_3,r} g_{j,r} \sigma_r \\
&+ \sum_{r=1}^{N-1}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\sum_{m_3=1}^{N-1} \theta_{m_1} \bar{\pi}_{m_2} \gamma_{m_3} g_{m_1,r} g''_{m_2,r} g_{m_3,r} g_{j,r} \sigma_r \\
&+ 2 \sum_{r=1}^{N-1}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\sum_{m_3=1}^{N-1} \theta_{m_1} \bar{\pi}_{m_2} \gamma_{m_3} g'_{m_1,r} g'_{m_2,r} g_{m_3,r} g_{j,r} \sigma_r \\
&+ \sum_{r=1}^{N-1}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\sum_{m_3=1}^{N-1} \theta_{m_1} \pi'_{m_2} \gamma_{m_3} g''_{m_1,r} g_{m_2,r} g_{m_3,r} g_{j,r} \sigma_r \\
&+ \sum_{r=1}^{N-1}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\sum_{m_3=1}^{N-1} \theta_{m_1} \pi'_{m_2} \gamma_{m_3} g_{m_1,r} g''_{m_2,r} g_{m_3,r} g_{j,r} \sigma_r \\
&+ 2 \sum_{r=1}^{N-1}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\sum_{m_3=1}^{N-1} \theta_{m_1} \pi'_{m_2} \gamma_{m_3} g'_{m_1,r} g'_{m_2,r} g_{m_3,r} g_{j,r} \sigma_r, \quad j = 1...N-1.
\end{aligned}$$

Now, introduce the operators

$$\frac{2}{l_k}D_{ij} = \frac{\partial h_j(\xi_i)}{\partial x}, \quad \frac{2}{l_k}\tilde{D}_{ij} = \frac{\partial h_j(\eta_i)}{\partial x}, \quad \frac{2}{l_k}\hat{D}_{ij} = \frac{\partial g_j(\eta_i)}{\partial x},$$
$$\frac{4}{l_k^2}\hat{D}^2_{ij} = \frac{\partial^2 g_j(\eta_i)}{\partial x^2}, \qquad \tilde{T}_{ij} = h_j(\eta_i).$$

Using these operators, and orthonormality of the basis functions $h_{i,r} = \delta_{ir}$ (Kronecker delta) for $r = 0, ..., N$ and $g_{j,r} = \delta_{jr}$ for $r = 1, ..., N-1$, we get

$$\lambda u_i \rho_i + \frac{4}{l_k^2}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N} u_{m_1}\nu_{m_2}D_{m_2 i}D_{m_2 m_1}\rho_{m_2} + \frac{4}{l_k^2}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N} u_{m_1}\nu_{m_2}D_{i m_1}D_{i m_2}\rho_i$$
$$-\frac{2}{l_k}\sum_{r=1}^{N-1}\pi'_r\beta_r\tilde{D}_{ri}\sigma_r - \frac{2}{l_k}\sum_{r=1}^{N-1}\sum_{n=1}^{N-1}\pi'_r\beta_n\hat{D}_{rn}\tilde{T}_{ri}\sigma_r = f_i\rho_i, \quad i = 0...N,$$

and

$$\begin{aligned}
&\frac{2}{l_k}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N} u_{m_1}\bar{\rho}_{m_2}\bar{\theta}_{2,m_3}\tilde{D}_{jm_1}\tilde{T}_{jm_2}\tilde{T}_{jm_3}\sigma_j \\
&+\frac{2}{l_k}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N} u_{m_1}\bar{\rho}_{m_2}\bar{\theta}_{2,m_3}\tilde{T}_{jm_1}\tilde{D}_{jm_2}\tilde{T}_{jm_3}\sigma_j \\
&+\frac{2}{l_k}\sum_{m_1=0}^{N}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N} u_{m_1}\bar{\rho}_{m_2}\bar{\theta}_{2,m_3}\tilde{T}_{jm_1}\tilde{T}_{jm_2}\tilde{D}_{jm_3}\sigma_j \\
&= \frac{4}{l_k^2}\sum_{m_1=1}^{N-1}\theta_{m_1}\bar{\pi}_j\gamma_j\hat{D}^2_{jm_1}\sigma_j + \frac{4}{l_k^2}\sum_{m_2=1}^{N-1}\theta_j\bar{\pi}_{m_2}\gamma_j\hat{D}^2_{jm_2}\sigma_j \\
&\qquad +\frac{8}{l_k^2}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\theta_{m_1}\bar{\pi}_{m_2}\gamma_j\hat{D}_{jm_1}\hat{D}_{jm_2}\sigma_j \\
&+\frac{4}{l_k^2}\sum_{m_1=1}^{N-1}\theta_{m_1}\pi_j\gamma_j\hat{D}^2_{jm_1}\sigma_j + \frac{4}{l_k^2}\sum_{m_2=1}^{N-1}\theta_j\pi_{m_2}\gamma_j\hat{D}^2_{jm_2}\sigma_j \\
&+\frac{8}{l_k^2}\sum_{m_1=1}^{N-1}\sum_{m_2=1}^{N-1}\theta_{m_1}\pi_{m_2}\gamma_j\hat{D}_{jm_1}\hat{D}_{jm_2}\sigma_j, \quad j = 1...N-1.
\end{aligned}$$

Putting it all together on matrix form, we get

$$\begin{bmatrix} M & -G \\ N & -S \end{bmatrix}\begin{bmatrix} u \\ \pi' \end{bmatrix} = \begin{bmatrix} Bf \\ S\bar{\pi} \end{bmatrix}$$

Here, the matrices are given by

$$M_{ij} = \lambda\rho_i\delta_{ij} + \frac{4}{l_k^2}\sum_{m_2=0}^{N}\nu_{m_2}D_{m_2i}D_{m_2j}\rho_{m_2} + \frac{4}{l_k^2}\sum_{m_2=0}^{N}\nu_{m_2}D_{ij}D_{im_2}\rho_i$$

$$G_{ij} = \frac{2}{l_k}(\beta_j\tilde{D}_{ji}\sigma_j + \sum_{n=1}^{N-1}\beta_n\hat{D}_{jn}\tilde{T}_{ji}\sigma_j)$$

$$B_{ij} = \rho_i\delta_{ij}$$

$$N_{ij} = \frac{2}{l_k}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N}\bar{\rho}_{m_2}\bar{\theta}_{m_3}\tilde{D}_{ij}\tilde{T}_{im_2}\tilde{T}_{im_3}\sigma_i + \frac{2}{l_k}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N}\bar{\rho}_{m_2}\bar{\theta}_{m_3}\tilde{T}_{ij}\tilde{D}_{im_2}\tilde{T}_{im_3}\sigma_i$$
$$+\frac{2}{l_k}\sum_{m_2=0}^{N}\sum_{m_3=0}^{N}\bar{\rho}_{m_2}\bar{\theta}_{m_3}\tilde{T}_{ij}\tilde{T}_{im_2}\tilde{D}_{im_3}\sigma_i$$

$$S_{ij} = \frac{4}{l_k^2}\sum_{m=1}^{N-1}\theta_m\gamma_i\hat{D}_{im}^2\sigma_i\delta_{ij} + \frac{4}{l_k^2}\theta_i\gamma_i\hat{D}_{ij}^2\sigma_i + \frac{8}{l_k^2}\sum_{m=1}^{N-1}\theta_m\gamma_i\hat{D}_{im}\hat{D}_{ij}\sigma_i.$$

By using the block Gauss elimination procedure, we get

$$\begin{bmatrix} H & -G \\ 0 & S + MH^{-1}G \end{bmatrix}\begin{bmatrix} u \\ \pi' \end{bmatrix} = \begin{bmatrix} Bf \\ -s - MH^{-1}Bf \end{bmatrix}.$$

How to solve this system is discussed in the last part of the 3-d section.

3.2.3 Discretisation in 3-d

Before we dive into the gory details of the 3-d discretisation, a small comment about the notation is helpful. When we write for instance $u_{i_xi_yi_z}$, you may still think of u as being a vector, with $u_{i_xi_yi_z} = u_{i_x+N_x(i_y+N_yi_z)}$. The same thing goes with matrices: $M_{i_xi_yi_zj_xj_yj_z}$ can be thought of as a regular 2-d matrix M_{ij} with $i = i_x + N_x(i_y + N_yi_z)$ and $j = j_x + N_x(j_y + N_yj_z)$.

Note that here, as in the 1-d section, the element summations and superscripts k are dropped to avoid cluttering. We write

$$\vec{u} = u^1\vec{i} + u^2\vec{j} + u^3\vec{k}$$

and split the momentum equation into its three components. Each of the components of $\vec{u}$ will be approximated by

$$u^l(x,y,z) \approx \sum_{m_x=0}^{N_x}\sum_{m_y=0}^{N_y}\sum_{m_z=0}^{N_z}u^l_{m_xm_ym_z}h_{m_x}(x)h_{m_y}(y)h_{m_z}(z).$$

The test functions v remain scalar, and are given by

$$v^{(i_x,i_y,i_z)}(x,y,z) = h_{i_x}(x)h_{i_y}(y)h_{i_z}(z).$$

The weak form of each component, $l = 1, 2, 3$, of the momentum equation is

$$\lambda(u^l, v) + \sum_{i=1}^{3}(\frac{\partial u^l}{\partial x_i}, \nu\frac{\partial v}{\partial x_i}) + \sum_{i=1}^{3}(\frac{\partial u^l}{\partial x_i}, v\frac{\partial \nu}{\partial x_i}) - \langle \pi', \beta\frac{\partial v}{\partial x_l}\rangle - \langle \pi', v\frac{\partial \beta}{\partial x_l}\rangle = (f^l, v),$$

$\forall v \in X$. The weak form of the continuity equation is

$$\begin{aligned}
&\sum_{l=1}^{3}\langle \bar{\rho}\bar{\theta}\frac{\partial u^l}{\partial x_l}, q\rangle + \sum_{l=1}^{3}\langle \bar{\rho}u^l\frac{\partial \bar{\theta}}{\partial x_l}, q\rangle + \sum_{l=1}^{3}\langle \bar{\theta}u^l\frac{\partial \bar{\rho}}{\partial x_l}, q\rangle \\
&= \sum_{l=1}^{3}\langle \gamma\bar{\pi}\frac{\partial^2\theta}{\partial x_l^2}, q\rangle + \sum_{l=1}^{3}\langle \gamma\theta\frac{\partial^2\bar{\pi}}{\partial x_l^2}, q\rangle + 2\sum_{l=1}^{3}\langle \gamma\frac{\partial\bar{\pi}}{\partial x_l}\frac{\partial\theta}{\partial x_l}, q\rangle \\
&+ \sum_{l=1}^{3}\langle \gamma\pi'\frac{\partial^2\theta}{\partial x_l^2}, q\rangle + \sum_{l=1}^{3}\langle \gamma\theta\frac{\partial^2\pi'}{\partial x_l^2}, q\rangle + 2\sum_{l=1}^{3}\langle \gamma\frac{\partial\pi'}{\partial x_l}\frac{\partial\theta}{\partial x_l}, q\rangle, \quad \forall q \in Y.
\end{aligned}$$

Notice that this discretisation now contains 27 inner product terms! Writing it out with the approximations and search functions and using the orthogonality of the basis functions, we get the lth momentum equation

$$\begin{aligned}
&\lambda u^l_{i_x i_y i_z}\rho_{i_x}\rho_{i_y}\rho_{i_z} \\
&+ \sum_{r_x=0}^{N_x}\sum_{m_x=0}^{N_x} u^l_{m_x i_y i_z}\nu_{i_x i_y i_z}h'_{m_x,r_x}h'_{i_x,r_x}\rho_{r_x}\rho_{i_y}\rho_{i_z} + \sum_{r_y=0}^{N_y}\sum_{m_y=0}^{N_y} u^l_{i_x m_y i_z}\nu_{i_x i_y i_z}h'_{m_y,r_y}h'_{i_y,r_y}\rho_{i_x}\rho_{r_y}\rho_{i_z} \\
&+ \sum_{r_z=0}^{N_z}\sum_{m_z=0}^{N_z} u^l_{i_x i_y m_z}\nu_{i_x i_y i_z}h'_{m_z,r_z}h'_{i_z,r_z}\rho_{i_x}\rho_{i_y}\rho_{r_z} + \sum_{m_x=0}^{N_x}\sum_{n_x=0}^{N_x} u^l_{m_x i_y i_z}\nu_{n_x i_y i_z}h'_{m_x,i_x}h'_{n_x,i_x}\rho_{i_x}\rho_{i_y}\rho_{i_z} \\
&+ \sum_{m_y=0}^{N_y}\sum_{n_y=0}^{N_y} u^l_{i_x m_y i_z}\nu_{i_x n_y i_z}h'_{m_y,i_y}h'_{n_y,i_y}\rho_{i_x}\rho_{i_y}\rho_{i_z} + \sum_{m_z=0}^{N_z}\sum_{n_z=0}^{N_z} u^l_{i_x i_y m_z}\nu_{i_x i_y n_z}h'_{m_z,i_z}h'_{n_z,i_z}\rho_{i_x}\rho_{i_y}\rho_{i_z} \\
&-R_l = f^l_{i_x i_y i_z}\rho_{i_x}\rho_{i_y}\rho_{i_z}, \qquad i_x = 0...N_x, \quad i_y = 0...N_y, \quad i_z = 0...N_z,
\end{aligned}$$

where

$$\begin{aligned}
R_1 &= \sum_{r_x=1}^{N_x-1}\sum_{r_y=1}^{N_y-1}\sum_{r_z=1}^{N_z-1}\pi'_{r_x r_y r_z}\beta_{r_x r_y r_z}h'_{i_x,r_x}h_{i_y,r_y}h_{i_z,r_z}\sigma_{r_x}\sigma_{r_y}\sigma_{r_z} \\
&+ \sum_{r_x=1}^{N_x-1}\sum_{r_y=1}^{N_y-1}\sum_{r_z=1}^{N_z-1}\sum_{n_x=1}^{N_x-1}\pi'_{r_x r_y r_z}\beta_{n_x r_y r_z}g'_{n_x,r_x}h_{i_x,r_x}h_{i_y,r_y}h_{i_z,r_z}\sigma_{r_x}\sigma_{r_y}\sigma_{r_z} \\
R_2 &= \sum_{r_x=1}^{N_x-1}\sum_{r_y=1}^{N_y-1}\sum_{r_z=1}^{N_z-1}\pi'_{r_x r_y r_z}\beta_{r_x r_y r_z}h_{i_x,r_x}h'_{i_y,r_y}h_{i_z,r_z}\sigma_{r_x}\sigma_{r_y}\sigma_{r_z} \\
&+ \sum_{r_x=1}^{N_x-1}\sum_{r_y=1}^{N_y-1}\sum_{r_z=1}^{N_z-1}\sum_{n_y=1}^{N_y-1}\pi'_{r_x r_y r_z}\beta_{r_x n_y r_z}g'_{n_y,r_y}h_{i_x,r_x}h_{i_y,r_y}h_{i_z,r_z}\sigma_{r_x}\sigma_{r_y}\sigma_{r_z} \\
R_3 &= \sum_{r_x=1}^{N_x-1}\sum_{r_y=1}^{N_y-1}\sum_{r_z=1}^{N_z-1}\pi'_{r_x r_y r_z}\beta_{r_x r_y r_z}h_{i_x,r_x}h_{i_y,r_y}h'_{i_z,r_z}\sigma_{r_x}\sigma_{r_y}\sigma_{r_z} \\
&+ \sum_{r_x=1}^{N_x-1}\sum_{r_y=1}^{N_y-1}\sum_{r_z=1}^{N_z-1}\sum_{n_z=1}^{N_z-1}\pi'_{r_x r_y r_z}\beta_{r_x r_y n_z}g'_{n_z,r_z}h_{i_x,r_x}h_{i_y,r_y}h_{i_z,r_z}\sigma_{r_x}\sigma_{r_y}\sigma_{r_z}.
\end{aligned}$$

The continuity equation becomes

$$\sum_{m_x=0}^{N_x}\sum_{m_y=0}^{N_y}\sum_{m_z=0}^{N_z}\sum_{n_x=0}^{N_x}\sum_{n_y=0}^{N_y}\sum_{n_z=0}^{N_z}\sum_{s_x=0}^{N_x}\sum_{s_y=0}^{N_y}\sum_{s_z=0}^{N_z}$$

$$\begin{aligned}
&(\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^1_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h'_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^2_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h'_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^3_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h'_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^1_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h'_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^2_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h'_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^3_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h'_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^1_{s_x s_y s_z}h'_{m_x,i_x}h_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^2_{s_x s_y s_z}h_{m_x,i_x}h'_{m_y,i_y}h_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}u^3_{s_x s_y s_z}h_{m_x,i_x}h_{m_y,i_y}h'_{m_z,i_z}h_{n_x,i_x}h_{n_y,i_y}h_{n_z,i_z}h_{s_x,i_x}h_{s_y,i_y}h_{s_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z})\\
=&\sum_{m_x=1}^{N_x-1}\theta_{m_x i_y i_z}\bar{\pi}_{i_x i_y i_z}\gamma_{i_x i_y i_z}g''_{m_x,i_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}+\sum_{m_y=1}^{N_y-1}\theta_{i_x m_y i_z}\bar{\pi}_{i_x i_y i_z}\gamma_{i_x i_y i_z}g''_{m_y,i_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{m_z=1}^{N_z-1}\theta_{i_x i_y m_z}\bar{\pi}_{i_x i_y i_z}\gamma_{i_x i_y i_z}g''_{m_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}+\sum_{n_x=1}^{N_x-1}\theta_{i_x i_y i_z}\bar{\pi}_{n_x i_y i_z}\gamma_{i_x i_y i_z}g''_{n_x,i_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{n_y=1}^{N_y-1}\theta_{i_x i_y i_z}\bar{\pi}_{i_x n_y i_z}\gamma_{i_x i_y i_z}g''_{n_y,i_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}+\sum_{n_z=1}^{N_z-1}\theta_{i_x i_y i_z}\bar{\pi}_{i_x i_y n_z}\gamma_{i_x i_y i_z}g''_{n_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{m_x=1}^{N_x-1}\sum_{n_x=1}^{N_x-1}\left(\theta_{m_x i_y i_z}\bar{\pi}_{n_x i_y i_z}+\theta_{m_x i_y i_z}\pi'_{n_x i_y i_z}\right)\gamma_{i_x i_y i_z}g'_{m_x,i_x}g'_{n_x,i_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{m_y=1}^{N_y-1}\sum_{n_y=1}^{N_y-1}\left(\theta_{i_x m_y i_z}\bar{\pi}_{i_x n_y i_z}+\theta_{i_x m_y i_z}\pi'_{i_x n_y i_z}\right)\gamma_{i_x i_y i_z}g'_{m_y,i_y}g'_{n_y,i_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{m_z=1}^{N_z-1}\sum_{n_z=1}^{N_z-1}\left(\theta_{i_x i_y m_z}\bar{\pi}_{i_x i_y n_z}+\theta_{i_x i_y m_z}\pi'_{i_x i_y n_z}\right)\gamma_{i_x i_y i_z}g'_{m_z,i_z}g'_{n_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{m_x=1}^{N_x-1}\theta_{m_x i_y i_z}\pi'_{i_x i_y i_z}\gamma_{i_x i_y i_z}g''_{m_x,i_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}+\sum_{m_y=1}^{N_y-1}\theta_{i_x m_y i_z}\pi'_{i_x i_y i_z}\gamma_{i_x i_y i_z}g''_{m_y,i_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{m_z=1}^{N_z-1}\theta_{i_x i_y m_z}\pi'_{i_x i_y i_z}\gamma_{i_x i_y i_z}g''_{m_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}+\sum_{n_x=1}^{N_x-1}\theta_{i_x i_y i_z}\pi'_{n_x i_y i_z}\gamma_{i_x i_y i_z}g''_{n_x,i_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\\
&+\sum_{n_y=1}^{N_y-1}\theta_{i_x i_y i_z}\pi'_{i_x n_y i_z}\gamma_{i_x i_y i_z}g''_{n_y,i_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}+\sum_{n_z=1}^{N_z-1}\theta_{i_x i_y i_z}\pi'_{i_x i_y n_z}\gamma_{i_x i_y i_z}g''_{n_z,i_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}.
\end{aligned}$$

$$j_x = 1...N_x-1, \quad j_y = 1...N_y-1, \quad j_z = 1...N_z-1.$$

The whole discretised system can be written on matrix form as

$$\begin{bmatrix} M & 0 & 0 & -G^x \\ 0 & M & 0 & -G^y \\ 0 & 0 & M & -G^z \\ N^x & N^y & N^z & -S \end{bmatrix} \begin{bmatrix} u^1 \\ u^2 \\ u^3 \\ \pi' \end{bmatrix} = \begin{bmatrix} Bf^1 \\ Bf^2 \\ Bf^3 \\ S\bar{\pi} \end{bmatrix}, \tag{3.8}$$

where

$$\begin{aligned}
M_{i_x i_y i_z j_x j_y j_z} &= \lambda \rho_{i_x} \rho_{i_y} \rho_{i_z} \delta_{i_x j_x} \delta_{i_y j_y} \delta_{i_z j_z} \\
&+ \frac{4}{l_k^2} \sum_{r_x=0}^{N_x} \nu_{i_x i_y i_z} D_{r_x j_x} D_{r_x i_x} \rho_{r_x} \rho_{i_y} \rho_{i_z} \delta_{i_y j_y} \delta_{i_z j_z} \\
&+ \frac{4}{l_k^2} \sum_{r_y=0}^{N_y} \nu_{i_x i_y i_z} D_{r_y j_y} D_{r_y i_y} \rho_{i_x} \rho_{r_y} \rho_{i_z} \delta_{i_x j_x} \delta_{i_z j_z} \\
&+ \frac{4}{l_k^2} \sum_{r_z=0}^{N_z} \nu_{i_x i_y i_z} D_{r_z j_z} D_{r_z i_z} \rho_{i_x} \rho_{i_y} \rho_{r_z} \delta_{i_x j_x} \delta_{i_y j_y} \\
&+ \frac{4}{l_k^2} \sum_{n_x=0}^{N_x} \nu_{n_x i_y i_z} D_{i_x j_x} D_{i_x n_x} \rho_{i_x} \rho_{i_y} \rho_{i_z} \delta_{i_y j_y} \delta_{i_z j_z} \\
&+ \frac{4}{l_k^2} \sum_{n_y=0}^{N_y} \nu_{i_x n_y i_z} D_{i_y j_y} D_{i_y n_y} \rho_{i_x} \rho_{i_y} \rho_{i_z} \delta_{i_x j_x} \delta_{i_z j_z} \\
&+ \frac{4}{l_k^2} \sum_{n_z=0}^{N_z} \nu_{i_x i_y n_z} D_{i_z j_z} D_{i_z n_z} \rho_{i_x} \rho_{i_y} \rho_{i_z} \delta_{i_x j_x} \delta_{i_y j_y}, \\
G^x_{i_x i_y i_z j_x j_y j_z} &= \frac{2}{l_k} (\beta_{j_x j_y j_z} \tilde{D}_{j_x i_x} \tilde{T}_{j_y i_y} \tilde{T}_{j_z i_z} \sigma_{j_x} \sigma_{j_y} \sigma_{j_z} \\
&+ \sum_{n_x=1}^{N_x-1} \beta_{n_x j_y j_z} \hat{D}_{j_x n_x} \tilde{T}_{j_x i_x} \tilde{T}_{j_y i_y} \tilde{T}_{j_z i_z} \sigma_{j_x} \sigma_{j_y} \sigma_{j_z}), \\
G^y_{i_x i_y i_z j_x j_y j_z} &= \frac{2}{l_k} (\beta_{j_x j_y j_z} \tilde{D}_{j_y i_y} \tilde{T}_{j_x i_x} \tilde{T}_{j_z i_z} \sigma_{j_x} \sigma_{j_y} \sigma_{j_z} \\
&+ \sum_{n_y=1}^{N_y-1} \beta_{j_x n_y j_z} \hat{D}_{j_y n_y} \tilde{T}_{j_x i_x} \tilde{T}_{j_y i_y} \tilde{T}_{j_z i_z} \sigma_{j_x} \sigma_{j_y} \sigma_{j_z}), \\
G^z_{i_x i_y i_z j_x j_y j_z} &= \frac{2}{l_k} (\beta_{j_x j_y j_z} \tilde{D}_{j_z i_z} \tilde{T}_{j_x i_x} \tilde{T}_{j_y i_y} \sigma_{j_x} \sigma_{j_y} \sigma_{j_z} \\
&+ \sum_{n_z=1}^{N_z-1} \beta_{j_x j_y n_z} \hat{D}_{j_z n_z} \tilde{T}_{j_x i_x} \tilde{T}_{j_y i_y} \tilde{T}_{j_z i_z} \sigma_{j_x} \sigma_{j_y} \sigma_{j_z}),
\end{aligned}$$

$$B_{i_x i_y i_z j_x j_y j_z} = \rho_{i_x}\rho_{i_y}\rho_{i_z}\delta_{i_x j_x}\delta_{i_y j_y}\delta_{i_z j_z},$$

$$N^x_{i_x i_y i_z j_x j_y j_z} = \frac{2}{l_k}\sum_{m_x=0}^{N_x}\sum_{m_y=0}^{N_y}\sum_{m_z=0}^{N_z}\sum_{n_x=0}^{N_x}\sum_{n_y=0}^{N_y}\sum_{n_z=0}^{N_z}$$

$$(\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}\tilde{T}_{i_x m_x}\tilde{T}_{i_y m_y}\tilde{T}_{i_z m_z}\tilde{T}_{i_x n_x}\tilde{T}_{i_y n_y}\tilde{T}_{i_z n_z}\tilde{D}_{i_x j_x}\tilde{T}_{i_y j_y}\tilde{T}_{i_z j_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}$$

$$+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}\tilde{T}_{i_x m_x}\tilde{T}_{i_y m_y}\tilde{T}_{i_z m_z}\tilde{D}_{i_x n_x}\tilde{T}_{i_y n_y}\tilde{T}_{i_z n_z}\tilde{T}_{i_x j_x}\tilde{T}_{i_y j_y}\tilde{T}_{i_z j_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}$$

$$+\bar{\rho}_{m_x m_y m_z}\bar{\theta}_{n_x n_y n_z}\tilde{D}_{i_x m_x}\tilde{T}_{i_y m_y}\tilde{T}_{i_z m_z}\tilde{T}_{i_x n_x}\tilde{T}_{i_y n_y}\tilde{T}_{i_z n_z}\tilde{T}_{i_x j_x}\tilde{T}_{i_y j_y}\tilde{T}_{i_z j_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}),$$

with similar forms for N^y and N^z (derivative in y- and z- components instead of x-component), and

$$\begin{aligned}
S_{i_x i_y i_z j_x j_y j_z} = {} & \frac{4}{l_k^2}\sum_{r_x=1}^{N_x-1}\theta_{r_x i_y i_z}\gamma_{i_x i_y i_z}\hat{D}^2_{i_x r_x}\sigma_{r_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_x j_x}\delta_{i_y j_y}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\sum_{r_y=1}^{N_y-1}\theta_{i_x r_y i_z}\gamma_{i_x i_y i_z}\hat{D}^2_{i_y r_y}\sigma_{i_x}\sigma_{r_y}\sigma_{i_z}\delta_{i_x j_x}\delta_{i_y j_y}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\sum_{r_z=1}^{N_z-1}\theta_{i_x i_y r_z}\gamma_{i_x i_y i_z}\hat{D}^2_{i_z r_z}\sigma_{i_x}\sigma_{i_y}\sigma_{r_z}\delta_{i_x j_x}\delta_{i_y j_y}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\theta_{i_x i_y i_z}\gamma_{i_x i_y i_z}\hat{D}^2_{i_x j_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_y j_y}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\theta_{i_x i_y i_z}\gamma_{i_x i_y i_z}\hat{D}^2_{i_y j_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_x j_x}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\theta_{i_x i_y i_z}\gamma_{i_x i_y i_z}\hat{D}^2_{i_z j_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_x j_x}\delta_{i_y j_y} \\
& +\frac{4}{l_k^2}\sum_{r_x=1}^{N_x-1}\theta_{r_x i_y i_z}\gamma_{i_x i_y i_z}\hat{D}_{i_x r_x}\hat{D}_{i_x j_x}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_y j_y}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\sum_{r_y=1}^{N_y-1}\theta_{i_x r_y i_z}\gamma_{i_x i_y i_z}\hat{D}_{i_y r_y}\hat{D}_{i_y j_y}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_x j_x}\delta_{i_z j_z} \\
& +\frac{4}{l_k^2}\sum_{r_x=1}^{N_z-1}\theta_{i_x i_y r_z}\gamma_{i_x i_y i_z}\hat{D}_{i_z r_z}\hat{D}_{i_z j_z}\sigma_{i_x}\sigma_{i_y}\sigma_{i_z}\delta_{i_x j_x}\delta_{i_y j_y}.
\end{aligned}$$

The first component of equation (3.8) can be transformed in the following manner:

$$\begin{aligned}
Mu_1^{n+1} - G^x\pi'^{n+1} &= Bf^1 \\
Mu_1^{n+1} - G^x\pi'^{n+1} - MQG^x\pi'^{n+1} &= Bf^1 - MQG^x\pi'^{n+1} \\
Mu_1^{n+1} - MQG^x(\pi'^{n+1} - \pi'^n) &= Bf^1 - MQG^x(\pi'^{n+1} - \pi'^n) + G^x\pi'^{n+1} \\
Mu_1^{n+1} - MQG^x(\pi'^{n+1} - \pi'^n) &= Bf^1 + G^x\pi'^n + r^x
\end{aligned}$$

Here, the term

$$r^x = -(MQ - I)G^x(\pi'^{n+1} - \pi'^n)$$

can be dropped if we choose Q carefully. To obtain an $O(\Delta t^3)$ asymptotic bound on the error term r^x, let

$$Q = \lambda^{-1} B^{-1}.$$

Insert this into the error term and remember that $M = \lambda B + A$, where A is some matrix independent from Δt, to get

$$r^x = -\lambda^{-1} A B^{-1} G^x (\pi'^{n+1} - \pi'^n).$$

We have that $\pi'^{n+1} - \pi'^n$ is of order $O(\Delta t)$. As $\Delta t \to 0$, M becomes dominated by the diagonal λB term. Since u_1 is multiplied by M, the truncation error in u_1 from dropping r^x is $O(\Delta t^3)$ when $\Delta t \to 0$. This makes the overall scheme have accuracy $O(\Delta^2)$.

Similar transformations applied to the second and third component give rise to the system

$$\begin{bmatrix} M & 0 & 0 & -MQG^x \\ 0 & M & 0 & -MQG^y \\ 0 & 0 & M & -MQG^z \\ N^x & N^y & N^z & -S \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \pi'^{n+1} - \pi'^n \end{bmatrix} = \begin{bmatrix} Bf^1 + G^x \pi'^n \\ Bf^2 + G^y \pi'^n \\ Bf^3 + G^z \pi'^n \\ S\bar{\pi} + S\pi'^n \end{bmatrix},$$

which can be rewritten on block triangular form as

$$\begin{bmatrix} M & 0 & 0 & -MQG^x \\ 0 & M & 0 & -MQG^y \\ 0 & 0 & M & -MQG^z \\ 0 & 0 & 0 & -S + \mathcal{N} \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \pi'^{n+1} - \pi'^n \end{bmatrix} = \begin{bmatrix} Bf^1 + G^x \pi'^n \\ Bf^2 + G^y \pi'^n \\ Bf^3 + G^z \pi'^n \\ S\bar{\pi} + S\pi'^n - \mathcal{M}\vec{f} - \mathcal{P}\pi'^n \end{bmatrix}.$$

The matrices

$$\begin{aligned} \mathcal{N} &= N^x Q G^x + N^y Q G^y + N^z Q G^z, \\ \mathcal{M}\vec{f} &= N^x M^{-1} B f^1 + N^y M^{-1} B f^2 + N^z M^{-1} B f^3, \\ \mathcal{P}\pi &= N^x M^{-1} G^x \pi + N^y M^{-1} G^y \pi + N^z M^{-1} G^z \pi, \end{aligned}$$

are introduced for convenience of notation. We get the following solution procedure

for the Exner function and the velocities:

$$\begin{aligned}
Mu_1^* &= Bf^1 + G^x\pi'^n \\
Mu_2^* &= Bf^2 + G^y\pi'^n \\
Mu_3^* &= Bf^3 + G^z\pi'^n \\
(-S+\mathcal{N})(\pi'^{n+1} - \pi'^n) &= S\bar{\pi} + S\pi'^n - N^x u_1^* - N^y u_2^* - N^z u_3^* \\
u_1^{n+1} &= u_1^* + QG^x(\pi'^{n+1} - \pi'^n) \\
u_2^{n+1} &= u_2^* + QG^y(\pi'^{n+1} - \pi'^n) \\
u_3^{n+1} &= u_3^* + QG^z(\pi'^{n+1} - \pi'^n)
\end{aligned} \tag{3.9}$$

3.3 Solving Systems of Linear Equations

The discretisation in the previous section boils down to several systems of linear equations. These systems are large in all but trivial cases. 9th order polynomials as basis functions give element matrices with one million entries in 3-d. Non-trivial cases can sport several hundreds or even thousands of elements, making the fully assembled system unwieldingly large. The solver methods that require the full matrix of coefficients to be stored in memory are not useful in such a setting. Instead, iterative methods, which evaluate one or more matrix-vector products several times, must be used. There are plenty of different iterative algorithms. The ones that are used in the code are the preconditioned conjugate gradient method and the preconditioned bi-conjugate gradient method with stabilisation (BiCGstab). These are advanced algorithms which were chosen because of their fast rate of convergence. See table 3.2 to see that one of the linear equation solvers (BiCGstab) dominates the CPU-time for the overall PDE solver.

3.3.1 Classical Iterative Methods

Assume we want to solve the equation

$$Ax = b.$$

Introduce the splitting $A = M - N$ to get the equivalent system of equations

$$Mx = Nx + b.$$

Now, we can hope that the scheme

$$Mx^{n+1} = Nx^n + b$$

for which x is an equilibrium, will eventually produce an x^n that is a sufficiently good approximation for x. We start out with an initial guess x_0. A better guess will use less time to approximate the real solution to a certain accuracy, but x_0 can be any value. Several classical schemes arise from different choices of M and N. For instance, Jacobi iteration is retrieved when selecting M as the diagonal D of A, and $N = A - D$:

$$x^{n+1} = D^{-1}(A-D)x^n + D^{-1}b = D^{-1}(Ax^n + b) - x^n$$

Another common classical scheme is Gauss-Seidel iteration, which arise when $M = D + L$, i.e. the diagonal and the lower triangular part of A:

$$(D + L)x^{n+1} = Ux^n + b$$

This system is trivial to solve, because $D + L$ is lower triangular. Gauss-Seidel iteration can be shown to converge whenever the spectral radius (largest absolute value of an eigenvalue) of $M^{-1}N < 1$. The rate of convergence for classical methods is usually much slower than for modern projection-based methods.

3.3.2 The Conjugate Gradient Method

The conjugate gradient method is a member of the family known as 'projection' or 'Krylov-subspace' solvers. The idea is to find the closest possible approximation to x in a given low-dimensional subspace $V^* \subset \mathbb{R}^N$, where N is the number of unknowns. Such a solution x^* is characterised by

$$||x - x^*|| \leq ||x - v||, \qquad \forall v \in V^*.$$

From a geometrical point of view, this means that the difference $x - x^*$ is perpendicular to all vectors in V^*. The solution is constructed by iteratively expanding the subspace. The procedure adds one linearly independent vector to the search subspace in each iteration, which means that an exact solution will be reached in at most N iterations. It is normal to use far less than N iterations. The number of iterations that are required to reach a desired accuracy depends on the condition number

$$\kappa(A) = \frac{\lambda_{max}}{\lambda_{min}}$$

where λ_{min} and λ_{max} are the smallest and largest eigenvalues of A, respectively. The number of iterations is proportional to the square-root of $\kappa(A)$. One way to construct the search subspace is based on the Gram-Schmidt orthogonalisation procedure. An A-orthogonal basis $v_1, ..., v_k$ for V^* can be iteratively constructed, for instance by using the residual of the previous iteration as the next expansion vector. The solution x^* in the kth iteration is expressed in terms of the basis vectors as

$$x^* = \sum_{j=1}^{k} \alpha_j v_j.$$

We choose the α_j's so that $x - x^* \perp v_j, \quad \forall v_j \in V^*$, which together with the A-orthogonality of the basis vectors leads to an explicit formula for the α_j's:

$$\alpha_j = \frac{v_j^T b}{v_j^T A v_j}$$

This formula breaks down if the denominator is zero, so the conjugate gradient method cannot be used if A is not positive definite, i.e. $y^T Ay > 0, \quad \forall y \neq 0$.

Preconditioner	Iterations u_2	Iterations Exner
Identity	172	N/A
Diagonal	35	N/A
Schwarz fine grid only	N/A	230
Schwarz fine + coarse grid	N/A	196

Table 3.1: Performance of different preconditioners. Average number of iterations required to reach a relative (compared to the norm of the right-hand side) tolerance of 2.2e-14 for u_2 and 1e-5 for the Exner solver. Thermal convection case with dt=0.001, t = 1 to 1.01.

One challenge with spectral element methods is that the condition number of the system that is solved in the standard Helmholtz equation grows as $O(N^4)$ (N is the polynomial order of the basis functions). This result is due to Vandeven [15]. This means that $O(N^2)$ iterations is required to reach an acceptable accuracy. Luckily, *preconditioning* can improve this. The preconditioned conjugate gradient method (PCG) involves a new matrix P^{-1} which is applied to the residual to create a second residual vector which is used to construct the new search vector. The number of PCG iterations that are required to solve a system of linear equations to a certain accuracy is governed by the condition number of $P^{-1}A$. Indeed, if $P^{-1} = A^{-1}$, this number becomes 1.

Unfortunately, finding A^{-1} would require us to solve the full system that we seek to solve. We should aim to find a preconditioner P^{-1} that is as close as possible to A^{-1}, but still relatively easy to compute. A popular choice is to extract the diagonal of A, which is then trivial to invert. This is what is done in the implementation of the implicit part of the heat equation (3.3) and the momentum equations (the first three parts of equation (3.9)) in the current code. This preconditioner works fairly well, and is ideal for the conjugate gradient method, since it preserves symmetry and positive definiteness. Table 3.1 clearly demonstrates that the diagonal preconditioner works well for the velocity solver. It also shows that even though the preconditioner for the pressure equation saves a great number of iterations, much is left to desire from the performance of the pressure solver. As table 3.2 shows, most of the CPU-time is spent solving the pressure (or Exner) equation (part four of equation (3.9)).

3.3.3 The Stabilised Bi-Conjugate Gradient Method

As was noted in the previous section, the standard conjugate gradient method does not handle cases where A is not positive definite. The discretised momentum equations and the derived pressure equation (equations 1-4 in equation (3.9)) are neither symmetric nor positive definite, so a method which handles this case is needed. The BiCGstab method does this. A detailed description of the BiCGstab method is available in e.g. [16]. The basic algorithm is given in figure 3.1.

1. Compute $r_0 := b - Ax_0$. Choose r_0^* s.t. $(r_0, r_0^*) \neq 0$.
2. Set $p_0 := r_0$, $p_0^* := r_0^*$
3. For $j = 0, 1, ...,$ until convergence DO:
4. $\alpha_j := (r_j, r_j^*)/(Ap_j, p_j^*)$
5. $x_{j+1} := x_j + \alpha_j p_j$
6. $r_{j+1} := r_j - \alpha_j A p_j$
7. $r_{j+1}^* := r_j^* - \alpha_j A^T p_j^*$
8. $\beta_j := (r_{j+1}, r_{j+1}^*)/(r_j, r_j^*)$
9. $p_{j+1} := r_{j+1} + \beta_j p_j$
10. $p_{j+1}^* := r_{j+1}^* + \beta_j p_j^*$
11. END DO

Figure 3.1: The basic BiCGstab algorithm.

As was the case with the conjugate gradient method, a preconditioner can be applied to the BiCGstab method as well. For the momentum equations, diagonal preconditioning was used. With the pressure equation, a two-level preconditioner was chosen based on Wasberg's implementation [13] of the overlapping Schwarz method described in [17]. The preconditioner P^{-1} is constructed from finite element operators on the spectral element grid:

$$P^{-1} = R_0^T A_0^{-1} R_0 + \sum_{k=1}^{K} R_k^T A_k^{-1} R_k$$

where the R_ks are restriction operators that converts data from global- to subproblem space and R^T converts back. The A_k^{-1}s are approximate finite element Laplacians of each subproblem. The 0th index k represents a global coarse grid component, with grid nodes on each local spectral element corner, while the rest are triangular elements created from each point on the spectral element grid. The A_ks (except for A_0) have a special structure which allows them to be inverted fast. The coarse subproblem must be solved numerically, and is redundantly solved globally on each processor node. The performance of this preconditioner is still not quite satisfactory, and also gets worse in cases of very deep stratification, probably because it makes the operator NQG behave less like the Laplace operator.

Function	**Share CPU time**
`IterativeSolvers::solve_bcgs`	84.3%
`RungeKutta23::rk23`	9.8%
`IterativeSolvers::solve_cg`	3.0%
`PseudoIncompressible::solve`	0.8%
`HeatedSquareCavity::average_nusselt_number`	0.6%
`VP_split_pseudoinc::u_update`	0.6%
`VP_split_pseudoinc::u_solve`	0.4%
`VP_split_pseudoinc::p_rhs`	0.3%
`main`	0.2%
`PseudoIncompressible::buoyancy`	0.1%
`VP_split_pseudoinc::solve`	0.1%

Table 3.2: Profiling chart for a run of the thermal convection case. The time spent in most low-level functions has been charged to their higher-level callers.

Chapter 4

Software Implementation

The implementation is based on a C++ spectral element solver for the Boussinesq approximation, written by Carl Erik Wasberg and Thor Gjesdal.

4.1 Solution Algorithm

A brief overview of the overall solution algorithm is given in figure 4.1. Most details are omitted, so that it is easy to see the order of computation.

4.2 Program Structure

The object-orientation of the original spectral element solver made it possible to re-use the code for this project. The whole program is broken down into data structure classes, equation solver classes and higher-level classes that describe the physical properties of the simulation.

The standard grid and field paradigm was applied to the data structure abstraction, but instead of one field class, there are two types. One is for the element level and is called Field. It contains the data values for one variable on one element. Because the elements are 3-d and typically use around ten basis functions in each direction, each Field instance will typically contain around one thousand coefficient values. Each Field instance is also associated with one Grid instance. It contains the data that is needed to interpret and work with the field values, for instance integration weights and scale factors needed for global assembly. Operators for interpolation to other grids are also included in the grid class. The other type of field class is called GlobalField and contains all the pointers to the various Field classes, as well as convenience functions that will call functions in all the Field classes to compute various quantities. See figure 4.2 for an overview of the data structure classes.

The solution algorithm is divided into solver classes to reflect how the system of equations has been split into a heat equation part and a velocity/pressure part, both with explicit advection parts and implicit parts. The object-oriented approach also promotes re-use of the most generic classes such as RungeKutta23 and IterativeSolvers

Fill $u_0, v_0, w_0, \pi'_0, \theta_0$ with initial values.
$n \leftarrow 0$, $t \leftarrow 0$.
WHILE $t < t_{stop}$ DO:
    $\tilde{\theta} \leftarrow$ extrapolate$(\theta_n, \theta_{n-1}, ...)$.
    $\tilde{\pi}' \leftarrow$ extrapolate$(\pi'_n, \pi'_{n-1}, ...)$.
    $\phi_n \leftarrow \theta_n \bar{\pi}$.
    $\phi_1^*, \phi_2^* \leftarrow$ solution of equation (3.1) with $\vec{u} = \vec{u}_n$.
    $\phi_{n+1} \leftarrow$ solution of equation (3.2).
    $\theta_{n+1} \leftarrow \phi_{n+1}/\bar{\pi}$.
    $u_1^*, u_2^* \leftarrow$ solution of x-component of equation (3.4).
    $v_1^*, v_2^* \leftarrow$ solution of y-component of equation (3.4).
    $w_1^*, w_2^* \leftarrow$ solution of z-component of equation (3.4).
    $\pi'_{n+1}, u_{n+1}, v_{n+1}, w_{n+1}$ are updated according to (3.9).
    $t \leftarrow t + \triangle t$.
    $n \leftarrow n + 1$.
END DO

Figure 4.1: The simplified overall solution algorithm. Notice that the variable θ is solved for first, then π' and then the velocities. In the case of no heating, $\mathcal{H} = 0$, there is no need for extrapolation of θ and π'.

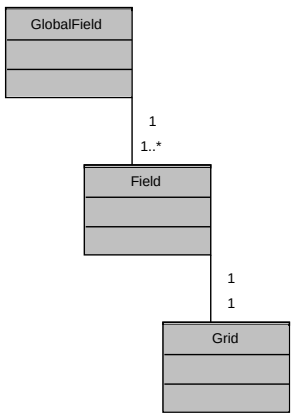

Figure 4.2: The basic data structure classes.

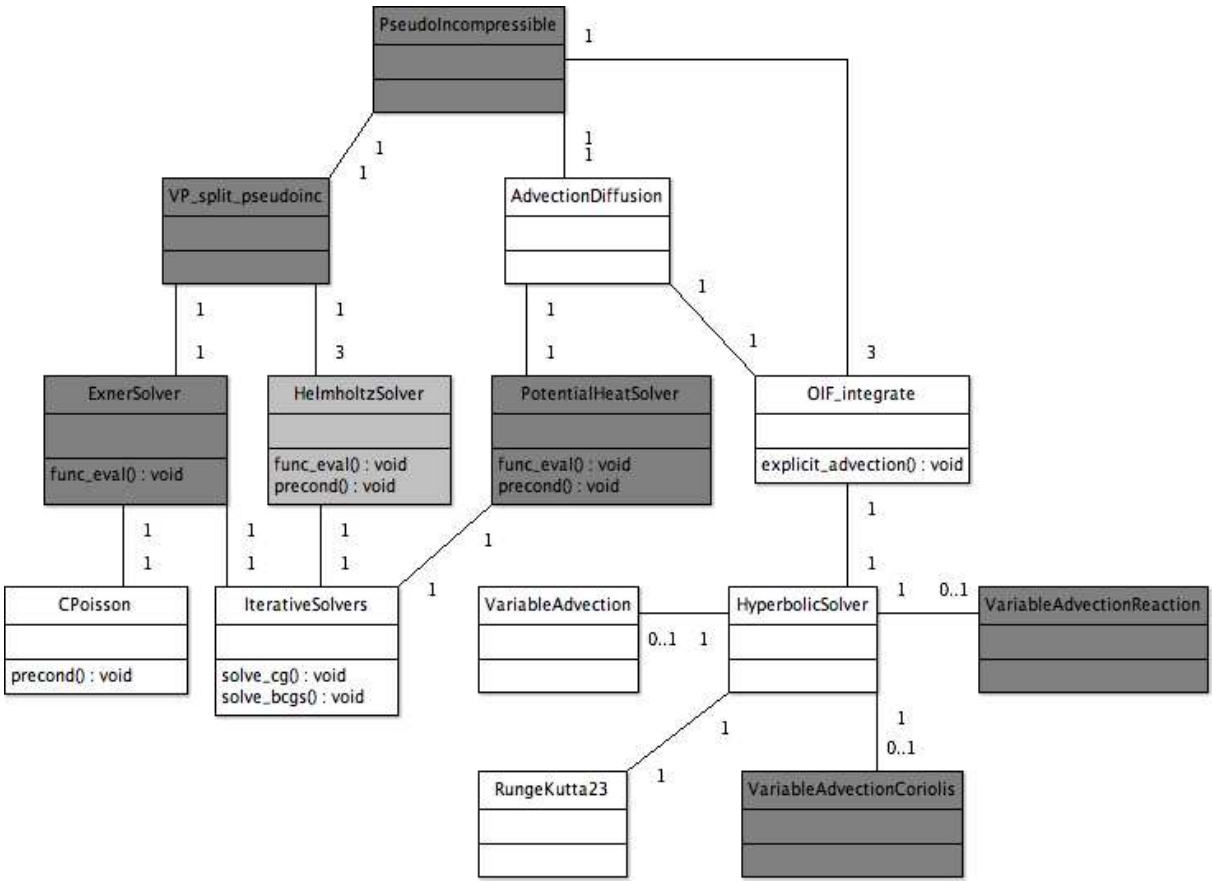

Figure 4.3: The solver classes, from high- to low-level. White, grey and dark grey boxes indicate old, modified and new classes, respectively.

(see figure 4.3). The distinction between data and algorithms is not complete, since the operators derived from the discretisation are implemented in the field classes.

Phyiscal case classes were made in order to have a consistent container for all the physical parameters associated with each simulation case. A common interface (abstract class) is used so that the algorithms can function without modification when new physical cases are introduced. See figure 4.4.

4.3 Parallel Execution

The full solver algorithm can be parallellised by breaking the spatial domain down into several subdomains. Several processors or machines can then cooperate by solving their assigned subproblem. Some amount of communication between each process is required, since there are element boundary grid points shared between the different subdomains. The serial version of the code also features communication between neighbouring elements in the assembly of the linear system of equations. The parallell implementation comes naturally if the assembly routine is modified with cross-process communication using MPI (Message Passing Interface).

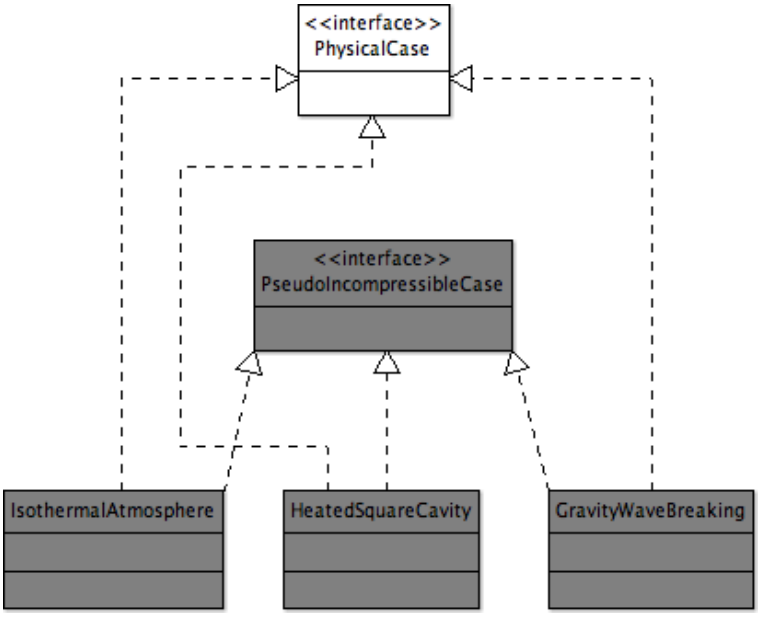

Figure 4.4: The physical case classes. White and dark grey boxes indicate old and new classes, respectively.

The communication involves a certain amount of overhead. There is also the coarse part of the overlapping Schwarz preconditioner mentioned in chapter 3, which is not parallellised. These factors tend to make the speed-up less than linear with the number of processors. Each processor has a certain amount of cache (fast memory), and dividing the overall problem between many different processors tends to put a greater portion of the overall problem in this fast memory. This effect can actually make the speed-up better than linear in some cases. Some results were obtained from running the code on a 32-CPU SGI Altix machine and can be seen in figure 4.5. A dual-core laptop achieves a near 2x speed-up.

4.4 Visualisation

The output from the solver can be extremely large, involving velocity, temperature and pressure in millions of data points. In the cases where the time-dependency is important, the total dataset size can be many gigabytes. The results can sometimes be parameterised into one or a few numbers, but in most cases it is useful to investigate the full output fields.

Visualisation was done using VoluViz [18], a volume renderer developed at FFI by Trond Gaarder, Anders Helgeland and myself. VoluViz uses slice-based volume rendering to show large sets of three-dimensional data. A number of slices perpendicular to the view direction are placed in the data domain. Data values are interpolated onto each slice, and assigned a colour and transparency from a colour table. The slices are then blended together in back-to-front order to produce the final image.

Vector fields pose a challenge, and volume rendering has traditionally not been

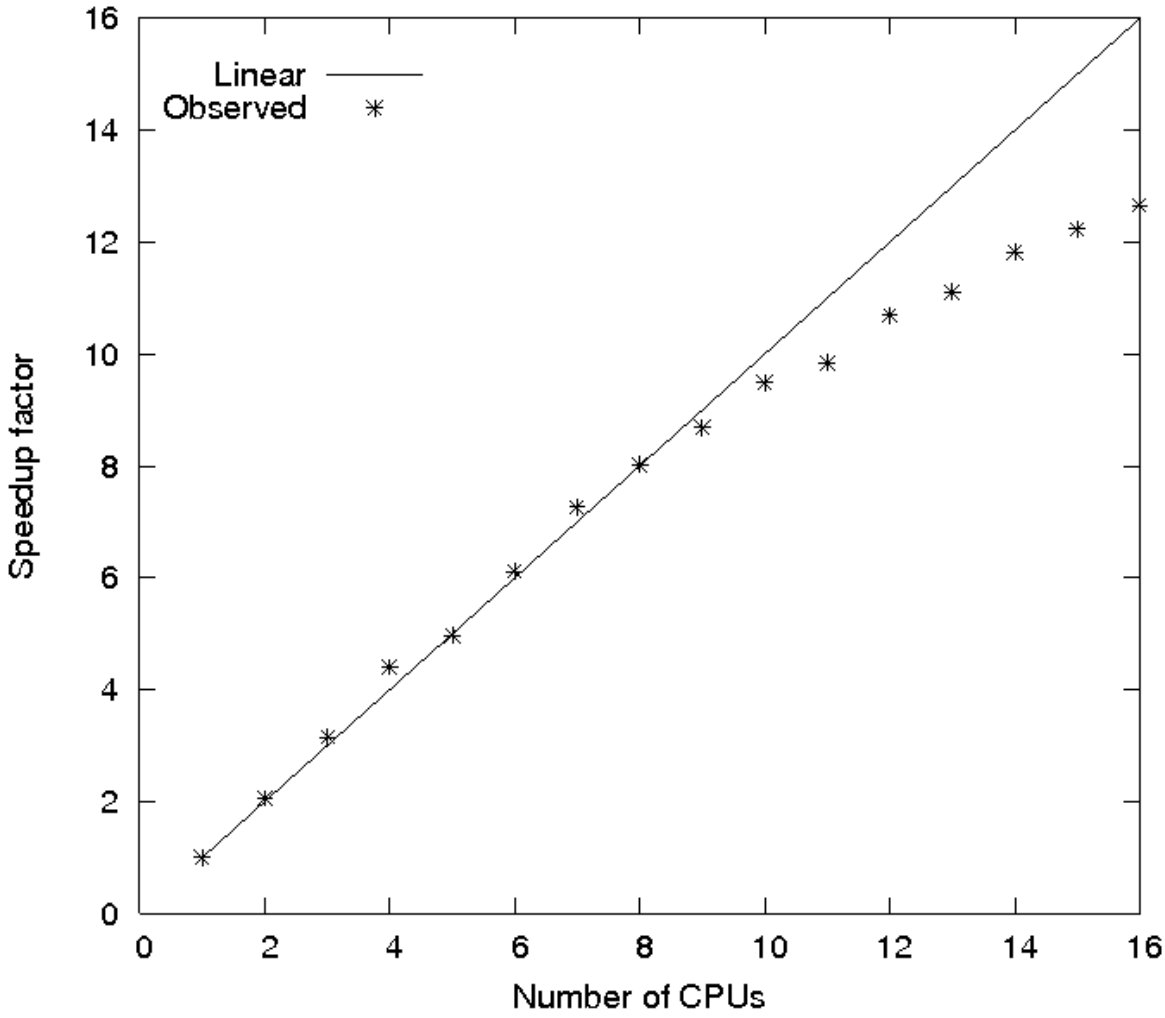

Figure 4.5: Parallel speedup for simulation of thermal convection case from t=1 to t=1.01 with dt=0.001.

used as much for the purpose of investigating such fields. A technique called seed LIC [19] was used to create volume data from the velocity fields.

Chapter 5

Results

5.1 2-d Thermal Convection

We study flow in a square cavity with one hot and one cold vertical wall. The other walls are insulated. Hot air will rise along the hot wall, and cold air will sink near the cold wall. Since the fluid cannot accumulate anywhere, we expect to see a circulation. Because we do not assume an incompressible fluid, the solution is not skew-symmetric.

5.1.1 Setup

The simulation domain is a prism as shown in figure 5.2, even though the physical problem is 2-d. This is done to stress test the 3-d code. The physical parameters are shown in tables 5.1 and 5.2. The boundary conditions are listed in table 5.3, and the initial conditions are listed in table 5.4. The setup is intended to match the benchmark case by Le Quere and Paillere [20], for which a reference solution exists.

The stratification is defined using the background functions $\bar{\rho}$, $\bar{\pi}$ and $\bar{\theta}$. The background quantities are set so that they are at a hydrostatic equilibrium

$$\frac{\partial \bar{p}}{\partial z} = -\bar{\rho} g,$$

where $\bar{p} = \bar{\rho} R \bar{T}$. Solved for $\bar{\rho}$, this ordinary differential equation gives

$$\bar{\rho} = \rho_0 \exp(-\frac{gz}{R\bar{T}}).$$

Direction	Length
X	0.067074 *m*
Y	0.067074 *m*
Z	0.067074 *m*

Table 5.1: Volume dimensions

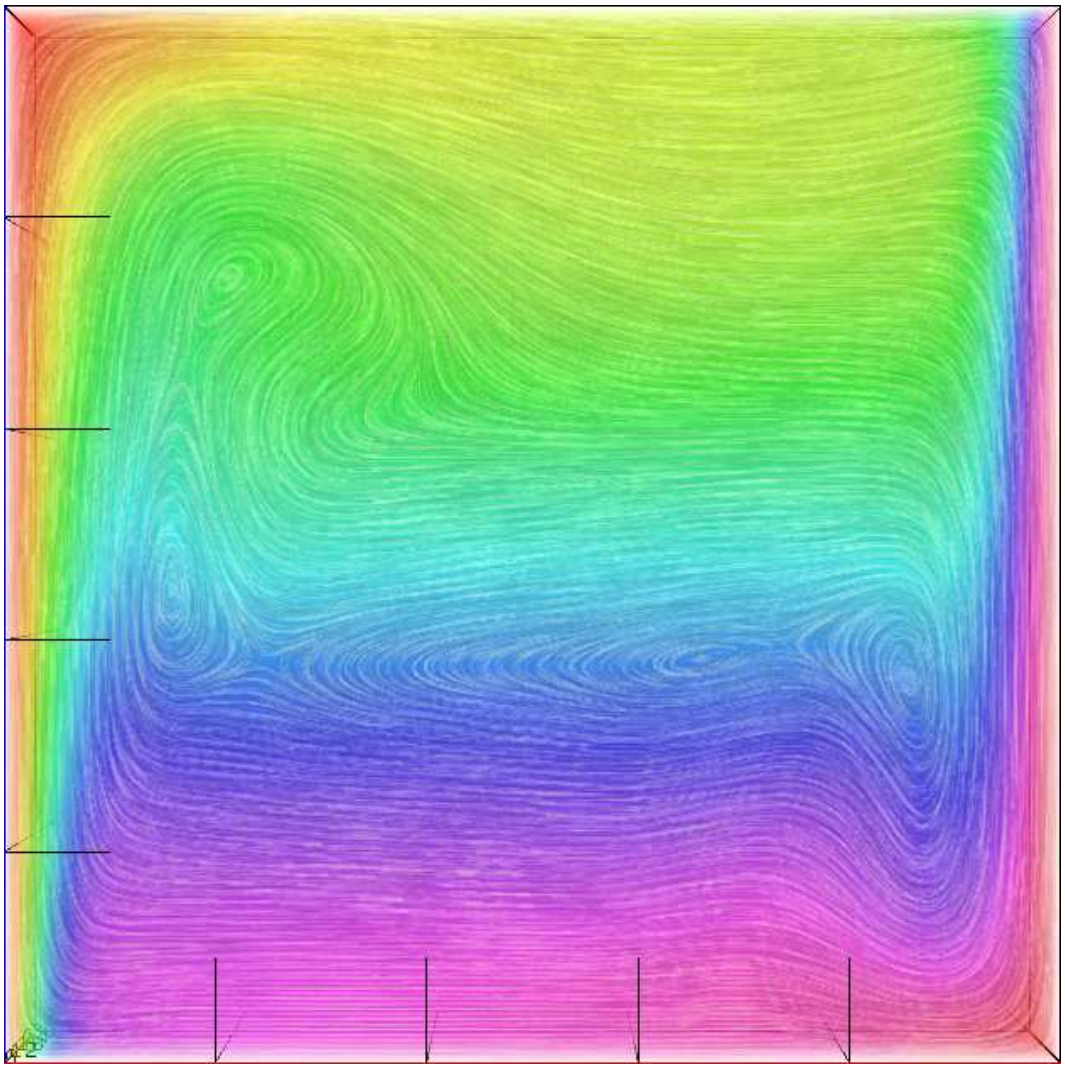

Figure 5.1: Seed LIC visualisation of the time-converged flow field.

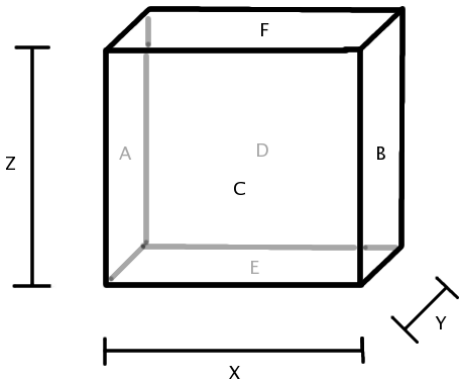

Figure 5.2: Simulation domain.

Symbol	Meaning	Value
c_p	Specific heat	1004.5 $J/(kg\ K)$
μ	Fluid viscosity	2.955e-5 $Pa\ s$
ρ_0	Density at bottom	0.5884 kg/m^3
g	Gravity constant	9.81 m/s^2
$\bar{T}$	Background temperature	600 K
Pr	Prandtl number	0.71
γ	Constant density/volume heat capacity ratio	1.4
Ra	Rayleigh number	1e6
R	Specific gas constant for dry air	287 $J/(kg\ K)$
p_0	Background ground pressure	101325 Pa

Table 5.2: Overview of the physical parameters.

Volume face	Temperature	Velocity
A	Dirichlet (960 K)	Dirichlet
B	Dirichlet (240 K)	Dirichlet
C	Neumann	Periodic
D	Neumann	Periodic
E	Neumann	Dirichlet
F	Neumann	Dirichlet

Table 5.3: Overview of the boundary conditions.

Variable	Initial value
u	0
v	0
w	0
π'	0
θ	$\bar{\theta}(z)$

Table 5.4: Overview of the initial conditions.

The mean-state Exner function is defined as

$$\bar{\pi} = \left(\frac{\bar{p}}{p_0}\right)^{\frac{R}{c_p}}$$

which combined with $\bar{T} = \bar{\pi}\bar{\theta}$ gives

$$\bar{\pi} = \exp\left(-\frac{gz + \ln(\frac{p_0}{R\rho_0\bar{T}})R\bar{T}}{(c_p + R)\bar{T}}\right).$$

5.1.2 Results

The solution of this problem is the value of each variable in a large number of different points. The Nusselt number is chosen as a convenient diagnostic variable, so that the solution accuracy can be evaluated with one number only. Also, the Nusselt number is the only number that both the cited reference solutions contain. The Nusselt number Nu is the ratio of convective to conductive heat transport

$$\mathrm{Nu} = \frac{Q}{Q_o},$$

where

$$Q = \int_\Omega \left(uT - \kappa\frac{\partial T}{\partial x}\right) d\Omega$$

and Q_0 is defined similarly, but with $u = 0$ and T as in the case with pure heat conduction. The thermal diffusivity κ is

$$\kappa = \frac{c_p\mu}{\mathrm{Pr}}.$$

The numbers presented in table 5.5 indicate that the implementation of the pseudo-incompressible model does not work well in this benchmark case. The value of the Nusselt number is closer to the reference value if the heating in the continuity equation is ignored. This may be explained by observing that this change makes the mathematical model closer to the Boussinesq model.

Run	Hot wall Nu	Cold wall Nu
Compressible reference [20]	8.85978	8.85978
Boussinesq reference [21]	8.825	8.825
Pseudo-incompressible	9.24	9.24
Pseudo-incompressible $\mathcal{H} = 0$ *	9.08	9.08

Table 5.5: Average Nusselt numbers in different locations for the different runs. * $\mathcal{H} = 0$ only in the continuity equation. There would be no motion if it was set to zero in the heat equation.

Direction	Length
X	8000 m
Y	1000 m
Z	8000 m

Table 5.6: Simulation domain dimensions for the St. Andrew's Cross case.

5.2 2-d St. Andrew's Cross

We want to study internal gravity waves in a stratified fluid. We do this by introducing a disturbance in a hydrostatically stratified fluid. The disturbance is oscillating and this oscillation causes internal waves. Depending on the frequency of oscillation compared to the Brunt-Väisälä (buoyancy) frequency, wave patterns may arise. For the right frequencies, we expext to observe the density disturbance form beams. This X-shaped structure is referred to as St. Andrew's Cross. St. Andrew was one of the Apostles and was crucified on an X-shaped cross, thus the name.

5.2.1 Setup

The simulation domain is a prism as shown in figure 5.2, even though the physical problem is 2-d. All the variables are uniform in the y-direction. The physical parameters are shown in tables 5.6 and 5.7. The boundary conditions are listed in table 5.8.

The stratification is defined using the background functions $\bar{\rho}$, $\bar{\pi}$ and $\bar{\theta}$. The density is set to decrease exponentially with height, and the Exner function is set so that it satisfies the third momentum equation when the velocities are zero. The potential

Symbol	Meaning	Value
c_p	Specific heat	1003.0 $J/(kg\ K)$
μ	Fluid viscosity	1.8e-5 $Pa\ s$
ρ_0	Density at bottom	1.168 kg/m^3
g	Gravity constant	9.81 m/s^2
$\bar{T}$	Background temperature	300 K
N	Buoyancy frequency	0.0179 s^{-1}
γ	Constant density/volume heat capacity ratio	1.4

Table 5.7: Overview of physical parameters for the St. Andrew's Cross case.

Volume face	Boundary Condition Type
A	Dirichlet
B	Dirichlet
C	Neumann
D	Neumann
E	Dirichlet
F	Neumann

Table 5.8: Overview of boundary conditions for the St. Andrew's Cross case.

temperature is chosen so that the background temperature is constant. This gives

$$\bar{\rho}(z) = \rho_0 \exp(-\frac{(z - z_0)\lambda N^2}{g(\lambda - 1)}),$$

$$\bar{\pi}(z) = \exp(-\frac{gz}{c_p \bar{T}}),$$

and

$$\bar{\theta} = \frac{\bar{T}}{\bar{\pi}}.$$

The disturbance is modelled with a periodic forcing term $F(\vec{x}, t)$ on the right hand side of the vertical momentum equation.

$$F(\vec{x}, t) = 0.01 \cos(\omega t) e^{-r/\sigma^2}.$$

Here, σ is 33 m and r is the distance to the xz-center of the volume, i.e. ignoring the distance along the y-axis. ω is the angular speed of the disturbance. We can vary ω to achieve different angles between the vertical axis and the wave beams.

5.2.2 Results

See figure 5.3 to see the temporal evolution of the St. Andrew's Cross. It was stated earlier that the buoyancy frequency is 0.179 in this setup. This is a simplification, since the buoyancy frequency is really vertically varying according to

$$N(z) = \sqrt{\frac{g}{\theta} \frac{d\theta}{dz}}$$

in the current setup. With a spatially constant N, we should expect to see beams form only at one specific angle to the vertical. In our case, however, linear theory for internal waves predicts multiple beams, with ever more beams as time increases. In fact, the angle between neighbouring wave crests that are around an angle α to the vertical should be

$$\frac{2\pi}{Nt \sin(\alpha)}.$$

This means that the wave crests closest to the vertical will be the widest, and that the width of each beam is decreasing with time. The beams move toward the horizontal line, i.e beams above the source move downwards and wave beams below the source move upwards. The energy propagates upwards above the disturbance source and downwards below the source. This means very little energy should be present on the horizontal line that passes through the source. See e.g. Lighthill [22] for a derivation of the above theory on internal waves and some photographs of laboratory experiments that exhibit the same patterns as figure 5.3.

In short, we see that the behaviour of the experiment agrees with the theory, both with regards to the temporal and spatial evolution of the beams. The computed solution gets somewhat polluted after t=1000s, because of reflections from the boundaries.

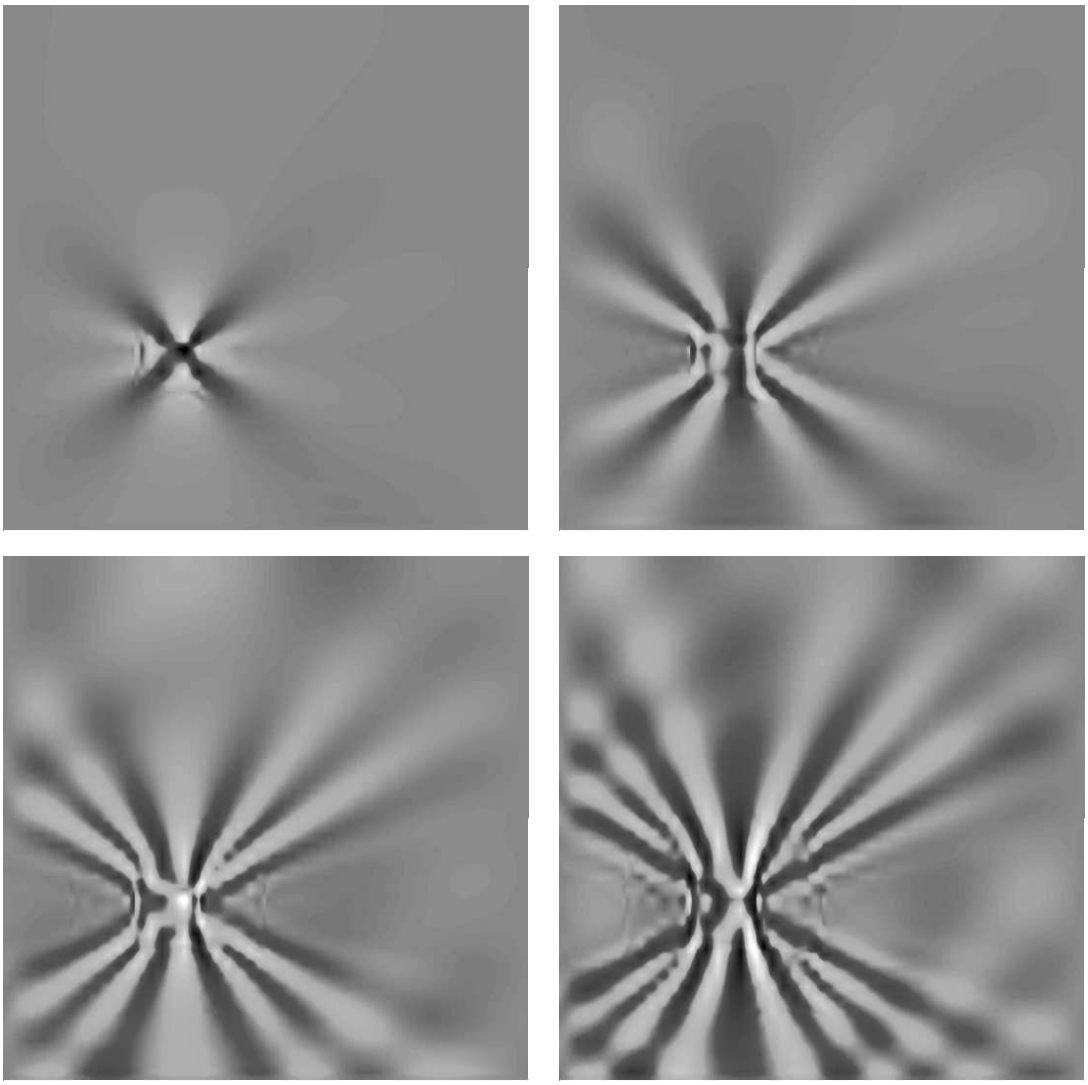

Figure 5.3: Density perturbation plotted at t = 500, 1000, 1500 and 2000.

Quantity	Scale	Value
Length	Density scale height H	8777 m
Speed	c_s	347 m/s
Time	H/c_s	25.3 s

Table 5.9: The scales that were used in the case of the breaking gravity wave.

Direction	Length
X	4H
Y	2H
Z	6H

Table 5.10: Volume dimensions in the case of the breaking gravity wave.

5.3 The Breaking of a Gravity Wave in 2-d/3-d

Interesting dynamics arise when internal gravity waves break. This can happen for instance when the z-component of the potential temperature gradient gets negative. An oscillating disturbance is introduced in a tall hydrostatically stratified atmosphere. This oscillation excites internal waves. Waves propagating downwards are damped and are of little interest. The waves that propagate upwards increase in amplitude. A critical layer with horizontal speed that equals the phase-speed of the disturbance is placed near the top of the simulation domain. The wave energy cannot sufficiently pass through the critical layer. This causes the gravity waves to break slightly below the critical layer.

5.3.1 Setup

The simulation domain is a prism as shown in figure 5.2. The physical parameters are shown in tables 5.10 and 5.11. They are scaled so that they match the conditions in the simulation by Andreassen et. al. [23] closely. The scales that are used are given in table 5.9. The boundary conditions are listed in table 5.12.

The stratification is defined using the background functions $\bar{\rho}$, $\bar{\pi}$ and $\bar{\theta}$. The density is set to decrease exponentially with height, and the Exner function is set so that it satisfies the third momentum equation when the velocities are zero. The potential temperature is chosen so that the background temperature is constant. This gives

$$\bar{\rho}(z) = \rho_0 e^{-z}.$$

Symbol	Meaning	Value
c_p	Specific heat	1003.0 $J/(kg\ K)$
μ	Fluid viscosity	1.0e-6 $Pa\ s$
ρ_0	Density at bottom	1.168 kg/m^3
g	Gravity constant	9.81 m/s^2
$\bar{T}$	Background temperature	300 K
γ	Constant density/volume heat capacity ratio	1.4

Table 5.11: Overview of physical parameters in the case of the breaking gravity wave.

Volume face	Boundary Condition Type
A	Periodic
B	Periodic
C	Periodic
D	Periodic
E	Dirichlet
F	Neumann

Table 5.12: Overview of boundary conditions in the case of the breaking gravity wave.

$$\bar{\pi} = \exp(-\frac{gzH}{c_p \bar{T}}),$$

and

$$\bar{\theta} = \frac{\bar{T}}{\bar{\pi}}.$$

The disturbance is modelled with a periodic forcing term $F(\vec{x}, t)$ on the right hand side of the vertical momentum equation:

$$F(\vec{x}, t) = 0.125\xi(t) \exp(-(z - z_{force})^2/\sigma^2) \sin(\omega t - k_0 x)$$

Here, σ is 0.2, z_{force} is the height of maximum forcing and is set to 3, the angular speed ω is 0.314 s^{-1}, and the wave number k_0 is 1.57. $\xi(t)$ constrains the disturbance to a specific time interval:

$$\xi(t) = \begin{cases} \sqrt{t/10}, & 0 \leq t \leq 10, \\ 1, & 10 < t \leq 50, \\ \sqrt{(60-t)/10}, & 50 < t \leq 60, \\ 0, & 60 < t. \end{cases}$$

The wave number k_0 is set so that the phase speed is equal to the horizontal speed at $z = 5H$. The initial horizontal speed is set to

$$u_0 = \begin{cases} 0, & 0 \leq z \leq 4, \\ 0.2(1 + \cos((3 - x/2.0)\pi)), & 4 < z \leq 6. \end{cases}$$

The upper boundary causes some headaches, since it is an open boundary with no periodic properties. The Neumann boundary condition gave some trouble, because some flow structures inevitably made it past the critical layer and interacted with the boundary. Non-physical reflections would then interfere with the region of interest after some time. To combat this, the simulation box was made taller than necessary and an extra damping layer was introduced. This damping was accomplished by increasing the viscosity, thus smoothing small structures. Some amount of trial and error was required in order to find a sufficiently small, i.e. does not affect the overall flow, viscosity that provided enough damping:

$$\mu = \begin{cases} 2 * 10^{-4}(0.5 - z) + 2 * 10^{-6} z, & 0 \leq z \leq 0.5, \\ 10^{-6}, & 0.5 < z \leq 5.5, \\ 2 * 10^{-4}(z - 5.5) + 2 * 10^{-6}(6 - z), & 5.5 < z \leq 6. \end{cases}$$

In order to induce 3-d flow, weak random noise was added to the potential temperature field at t = 40.

5.3.2 Results

In order to visualise the vortices present in the solution, it was necessary to compute a quantity known as the vorticity or the curl of the velocity field. The vorticity is a vector

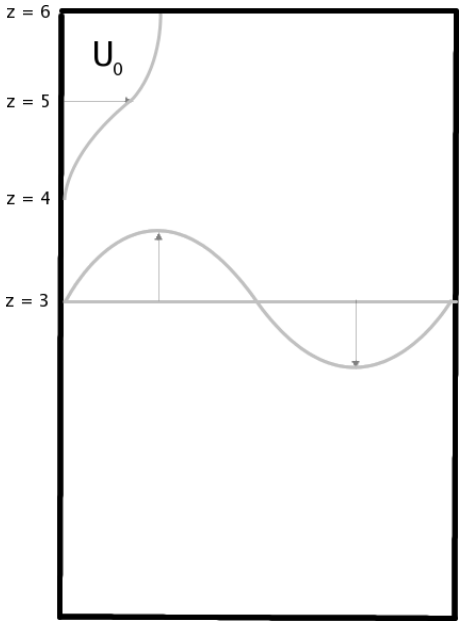

Figure 5.4: This figure shows the vertical forcing in the middle of the volume, and the horizontal velocity initial condition in the upper model atmosphere.

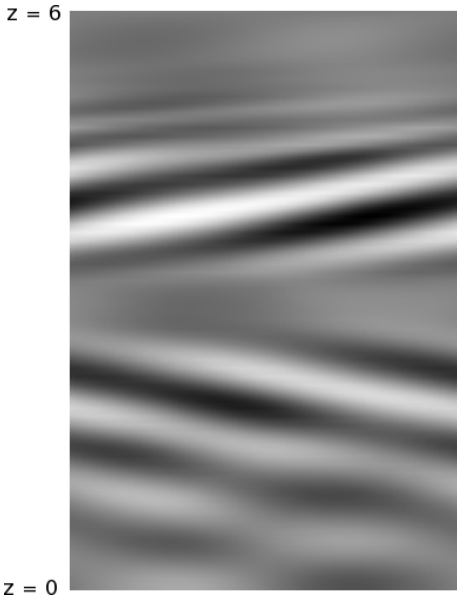

Figure 5.5: The Exner function perturbation at t = 40. Waves propagate up and down as in the St. Andrews case. The area between z = 4.5 and z = 5 is where the waves will eventually break.

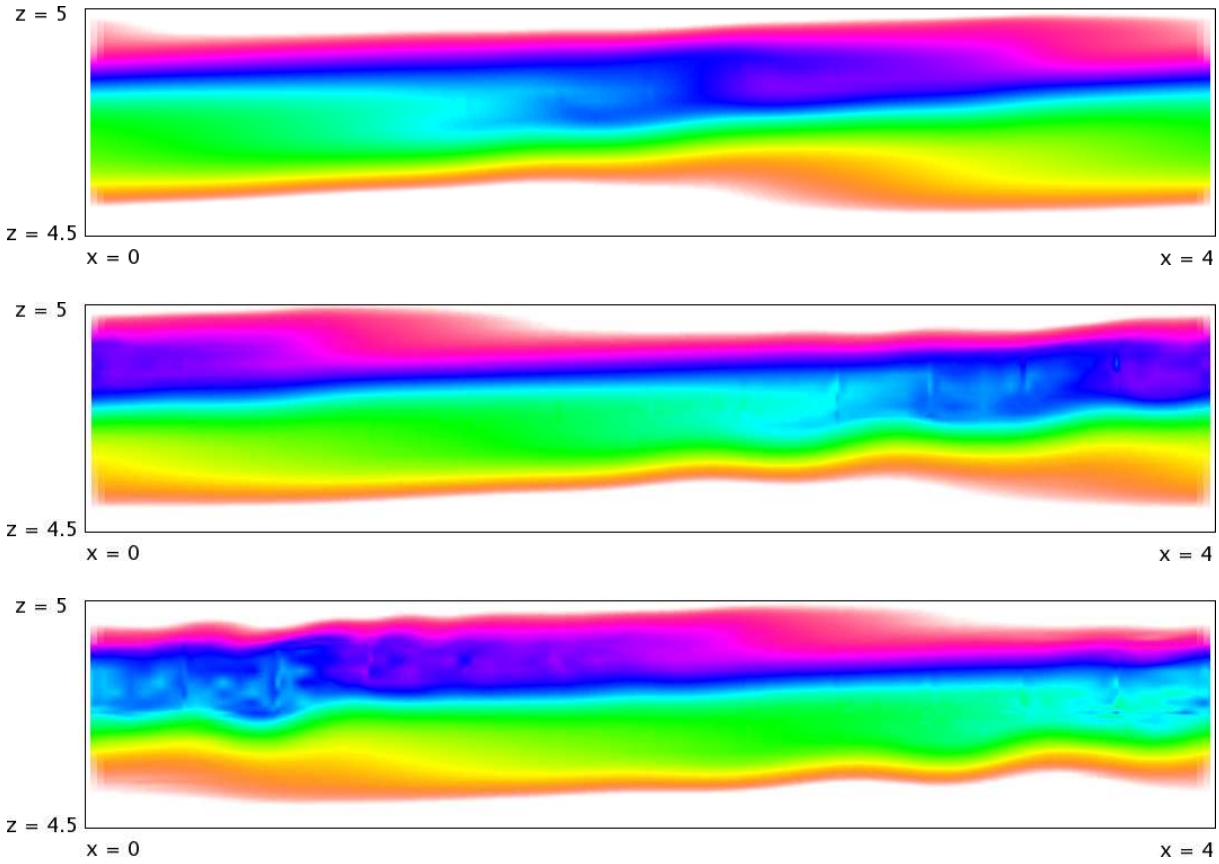

Figure 5.6: Potential temperature in the 2-d case plotted at t = 80, 85, 90. In this case, no noise was added, so the value of all variables are uniform in the y-direction, which is perpendicular to the paper plane.

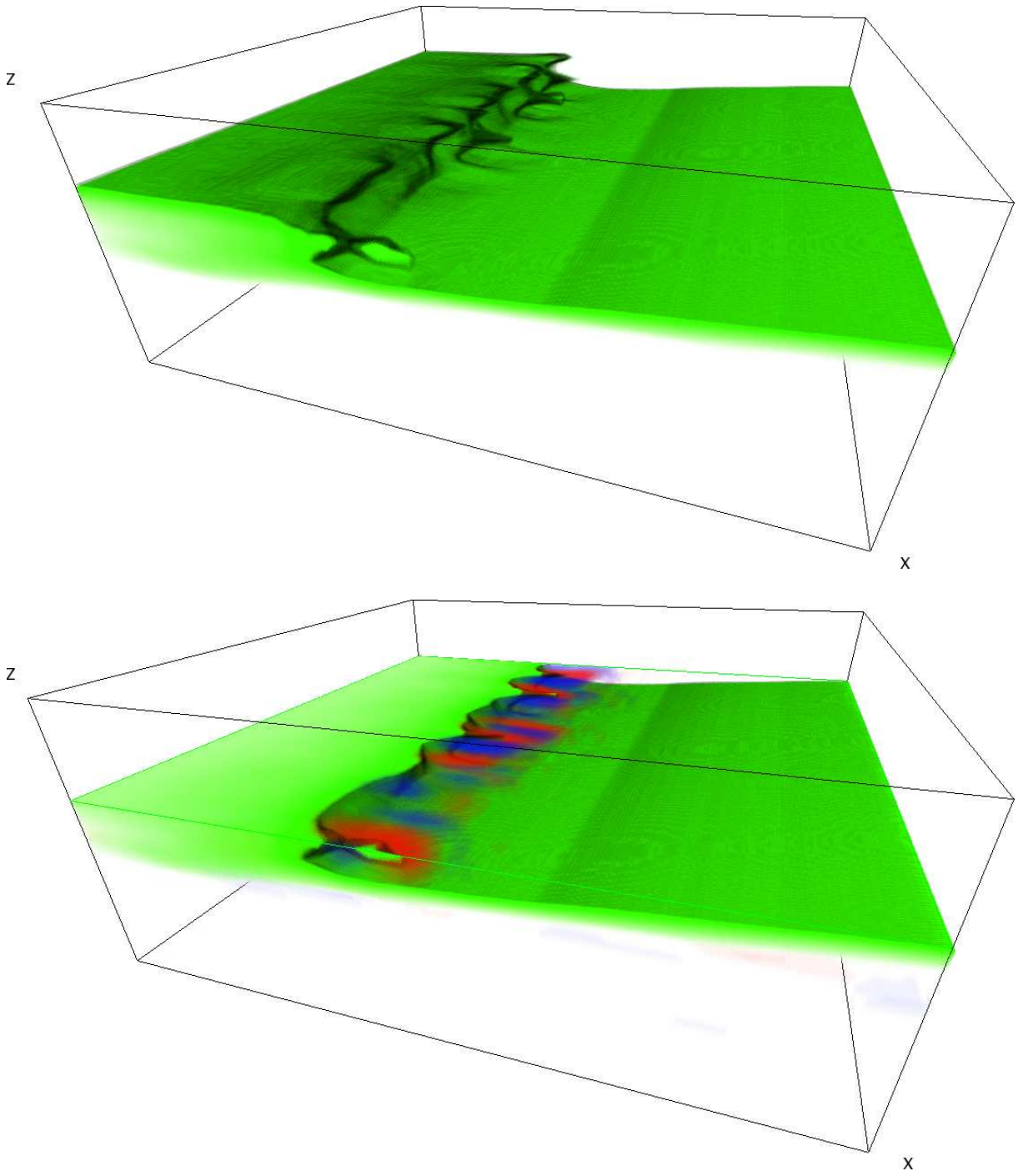

Figure 5.7: Potential temperature isosurface (green) and x-vorticity (red and blue tubes) plotted in the case of random noise added. The lower scene is rendered with a clipping plane, so it is easier to see the vortex tubes. The screenshot is from t=77.5

quantity that indicates rotational movement:

$$\nabla \times \vec{u} = \left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right)\vec{i} + \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right)\vec{j} + \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)\vec{k}$$

Figure 5.6 shows that a convective instability arises from the gravity wave near the critical layer. The instability causes the flow structures to rapidly cascade into smaller structures. It is interesting to note that the 3-d case is qualitatively different from the 2-d case. In 2-d, there are vortices oriented along the y-axis only. In 3-d, a number of counter-rotating vortex pairs oriented along the x-axis appear as the wave breaks. Figure 5.7 shows these vortices as blue and red tubes. The perturbations associated with these vortices are several orders of magnitude larger than the noise that was added. This indicates that the vortices oriented along the X-direction are indeed the favoured way of breaking for such waves. This result is in agreement with the results of Andreassen et. al. [23], even though they obtained their results with a different model (the Euler equations), and a different numerical framework (the spectral viscosity method).

Chapter 6

Conclusions

A spectral element method for the pseudo-incompressible model by Durran has been implemented and run against several test cases.

The results from the thermal convection benchmark case in section 5.1 are not as good as we wished. The Nusselt number is off by 4 percent, which is worse than the Boussinesq model. There is a small concern that this may indicate a programming error, but it can just as well indicate that the pseudo-incompressible model is unsuitable for this problem. Remember that we are applying a mathematical model that is taylored for atmosphere calculations onto a tiny box with huge temperature differences. The reference numbers are for a fully compressible solver, and we cannot expect to match it exactly with a pseudo-incompressible model. It is disappointing nevertheless that it was off by more than the Boussinesq model.

The St. Andrew's Cross case is more relevant as a sanity check for the method, both because it is a stratified case and it concerns internal gravity waves. This case shows exactly the behaviour expected. Thus, we have re-established some faith in the properties of the solver, even though the St. Andrew's Cross is a linear case.

The third case that the method has been applied to, is the breaking of a gravity wave. This is a non-linear case which concerns the transition from laminar to turbulent flow. In this case, the results are in good agreement with cited results obtained through different methods.

In all three cases, the results have been in qualitative agreement with either cited reference material or analytical results. Ideally, the accuracy of the code should also be assessed with a non-linear tall atmosphere reference case. Such reference data was not found. The lack of hard evidence to 'prove' that we achieve the high accuracy expected from a spectral element method is perhaps the greatest weakness of this work.

There is a lot of room for improvement in the current implementation. Perhaps the most pressing, are the pressure preconditioner and the parallell performance. The code scales well up to around 10 CPUs, but after that, the overhead starts to be a problem. This is unacceptable in a 1000 CPU environment. The Exner problem preconditioner does not perform good enough. Convergence is slow, so most of the overall time spent to solve a problem is spent solving for the Exner function.

Bibliography

[1] L. B. Nance and D. R. Durran. A Comparison of the Accuracy of Three Anelastic Systems and the Pseudo-Incompressible System. *J. Atmos. Sci.*, 51(24):3549–3565, December 1994.

[2] P. K. Kundu. *Fluid Mechanics*. Academic Press, 1990.

[3] G. K. Batchelor. The condition for dynamical similarity of motions of a frictionless perfect-gas atmosphere. *Quart. J. Roy. Meteor. Soc.*, 79:224–235, 1953.

[4] Y. Ogura and N. Phillips. Scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.*, 19:173–179, 1962.

[5] R. Wilhelmson and Y. Ogura. The pressure perturbation and the numerical modeling of a cloud. *J. Atmos. Sci.*, 29:1295–1307, 1972.

[6] F. B. Lipps and R. S. Hemler. A scale analysis of deep moist convection and some related numerical calculations. *J. Atmos. Sci.*, 39:2192–2210, 1982.

[7] D. R. Durran. Improving the Anelastic Approximation. *J. Atmos. Sci.*, 46(11):1453–1461, June 1989.

[8] H. P. Langtangen. *Computational Partial Differential Equations*. Springer-Verlag, 2003.

[9] E. W. Weisstein. Lobatto quadrature. MathWorld, http://mathworld.wolfram.com/LobattoQuadrature.html.

[10] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-Order Methods for Incompressible Fluid Flow*. Cambridge University Press, 2002.

[11] S. Vadas, C. E. Wasberg, and T. Gjesdal. On the implementation of Durrans pseudo-incompressible model. Correspondence.

[12] Y. Maday, A. T. Patera, and E. M. Rønquist. An operator-integration-factor splitting method for time-dependent problems: Application to incompressible fluid flow. *J. Sci. Comput.*, pages 263–292, 1990.

[13] C. E. Wasberg. Solution of the Navier-Stokes equations in 2d by a spectral element method. Technical report, FFI, 2002.

[14] F. Brezzi. On the existence, uniqueness and approximation of saddle point problems arising from Lagrange multipliers. *RAIRO Anal. Numer.*, pages 129–151, 1974.

[15] H. Vandeven. On the eigenvalues of second-order spectral differentiation operators. *Comput. Methods Appl. Mech. Engrg.*, 80:313–318, 1990.

[16] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[17] P. F. Fischer, N. I. Miller, and H. M. Tufo. An overlapping Schwarz method for spectral element simulation of three-dimensional incompressible flows. *Parallel Solution of Partial Differential Equations*, pages 159–181, 2000.

[18] T. Gaarder and A. Helgeland. VoluViz 1.0 Report. Technical report, FFI, 2002.

[19] A. Helgeland and Ø. Andreassen. Visualization of vector fields using seed lic and volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):673–682, November-December 2004.

[20] P. Le Quere, C. Weisman, H. Paillere, J. Vierendeels, E. Dick, R. Becker, M. Braack, and J. Locke. Modelling of natural convection flows with large temperature differences: A benchmark problem for low mach number solvers. part 1. reference solutions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39:609–616, 2005.

[21] T. Gjesdal, C. E. Wasberg, and B. A. Pettersson Reif. Spectral element benchmark simulations of natural convection in two-dimensional cavities. *Int. J. Numer. Meth. Fluids*, 2005.

[22] J. Lighthill. *Waves in Fluids*. Cambridge University Press, 1978.

[23] Ø. Andreassen, C. E. Wasberg, D. C. Fritts, and J. R. Isler. Gravity wave breaking in two and three dimensions, 1. Model description and comparison of two-dimensional evolutions. *Journal of Geophysical Research*, 99:8095–8108, 1994.