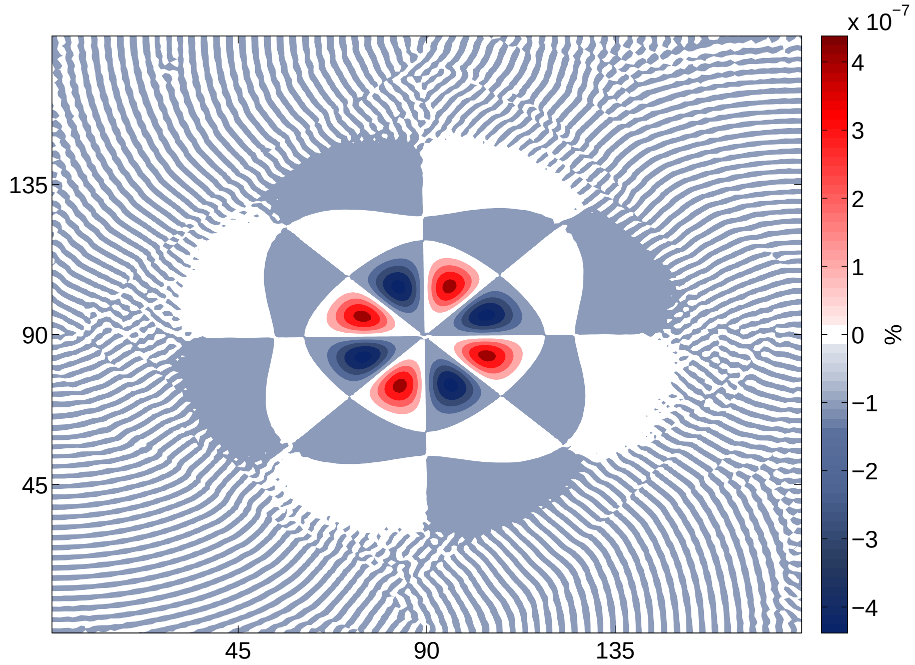
The development and evaluation of an idealized ocean model for the Bergen Dynamic Model



Master thesis in Climate Dynamics

Trond Thorsteinsson

November 2013

UNIVERSITY OF BERGEN
GEOPHYSICAL INSTITUTE

Abstract

Three idealized ocean model versions are developed for the Bergen Dynamic Model (Bedymo). All three model versions fall into the category of slab ocean models. One version is more or less a standard slab ocean model, while the two other versions include two slightly different representations of oceanic heat transport. The model versions are tested through a series of idealized experiments where the model result is compared to analytical solutions. Some simple sensitivity tests are performed, to become familiar with how configurable parameters alter the model output. The model versions are also tested in coupled runs with a one layer atmosphere. The test results are satisfactory, and Bedymo is thus applicable for a whole new range of experiments.

Acknowledgments

I would like to thank my supervisors, Thomas and Clemens, for providing me with an interesting thesis and good advice. Thanks to Thomas for his ever lasting optimism and enthusiasm. Special thanks to Clemens for allowing me to work on his model, and for always being available, also while on the wrong side of the globe. In addition I have to thank my girlfriend Line for continuing to stick around, and in particular for hiding my play station games.

Trond Thorsteinsson
Bergen, 20 November 2013

Contents

Chapter 1

Introduction

One of the worlds greatest challenges relates to the anticipated, and widely debated global warming. As a result the need for reliable predictions regarding the future climate of the Earth is at an all time high. Comprehensive climate models are the essential tool, as the worlds leading model centers compete in producing the most reliable and convincing future forecasts. With the aid of the ever increasing computer power, the complexity of the climate models steadily increases (Held 2005).

As the complexity of the climate models increases, so does the already large gap between simulations and understanding. Held (2005) may be onto the core of the problem when he asks: "What does it mean, after all, to understand a system as complex as the climate, when we can not fully understand idealized nonlinear systems with only a few degrees of freedom?"

To be able to understand our climate models Held (2005) proposes that we build a model hierarchy, from the most idealized models, to the most comprehensive. If we can fully understand the less complex models, we might better understand how key sources of complexity alter the dynamics, bringing us closer to an understanding of the complex climate system. If we are to emphasize our understanding of the climate dynamics, and not just our ability to predict, idealized models clearly have a role to play, apart from just being a tool for the especially interested.

The primary goal of this thesis is to develop an idealized ocean model suited for coupling with the Bergen Dynamic Model (Bedymo). Bedymo is an idealized atmospheric model developed by co-supervisor Clemens Spensberger. Coupling Bedymo to an idealized ocean will make Bedymo applicable for a whole new range of experiments. The first of these experiments shall be presented towards the end of this thesis, be it only simple tests to confirm that the coupling is functional.

Bedymo is originally based on a set of configurable simplifications around Quasi- or Semi-geostrophic theory, but a recent primitive equation version (BedymoPE) is also developed and this is the version that the first coupled experiments are carried out with. The advantage of the primitive equation version in terms of coupling is that the atmospheric wind and temperature are available at the lower vertical boundary.

In the development of the ocean model, I use a slab ocean as a starting point. A slab ocean means a fixed depth motionless ocean mixed layer model where the temperature, which is constant throughout the depth of the slab, is the only prognostic variable. The simplest possible slab ocean model thus has only the ability to interact with the atmosphere through surface heat fluxes and to store heat.

While the coupling of Bedymo to an idealized ocean makes Bedymo applicable for new situations, a redistribution of heat within the slab ocean itself would further increase our options. One could e.g. study the role of oceanic heat transport in the development of atmospheric phenomenon by comparing model runs with and without a representation of oceanic heat transport.

Despite the simplification that a slab ocean model represents with respect to the real ocean it is not uncommon to couple an Atmospheric General Circulation Model (AGCM) to a slab ocean when running climate simulations (Collins et al. 2004). A slab ocean model can be particularily useful in simulations with very long integration time, and when conducting multiple simulations with varying coefficients as is common in for example paleoclimate studies because of many unknown parameters (Donnadieu et al. 2006).

As the temperature of a slab ocean model originally is forced only by local surface heat fluxes, it will differ significantly from observed sea surface temperature (SST) if a slab ocean model is used in a climate simulation without some representation of the heat transport by ocean currents (Codron 2012). A common way to model heat transport in slab oceans when used in climate simulations is to add a corrective flux (Q-flux) based on the results from a control simulation where the atmosphere is forced with observed SST. The Q-flux is determined from monthly means of the surface fluxes, the general idea being that there is an approximate balance between surface fluxes and oceanic heat transport in the mixed layer at such a timescale (Collins et al. 2004). With this method, one can simulate realistic SST with respect to the present climate, but when the assumption of an oceanic heat transport equal to the present becomes less realistic this approach is no longer necessarily the best alternative.

The main objective of implementing a representation of oceanic heat transport in our case is not to model observed SST, and the calculation of a Q-flux is also a more comprehensive method than what we are looking for.

A different way of modelling oceanic heat transport in slab oceans, that is simpler, and thus may be better suited when the slab ocean is to be coupled to an idealized atmopshere, is to include a representation of wind driven Ekman currents (Maze et al. 2011, Codron 2012). The Ekman currents are diagnosed from the atmospheric surface wind by a bulk formula. In the model by Maze et al. (2011) the currents follow the classic Ekman drift formula ($u_{ekman} \sim f v_{surface}$), while Codron (2012) also includes a dissipative parameter ϵ that allows the calculation of wind driven currents also at the equator ($u_{ekman} \sim f v_{surface} + \epsilon u_{surface}$) .

Heat transport by Ekman currents is an optional feature in the slab ocean model I develop, as is also heat transport by user prescribed background currents. Unless the total ocean currents are somehow restrained from being horizontally divergent, their inclusion in the model does immediately raise the question of what is below the slab layer.

The answer to what is below the slab layer is what separates two of a total of three model versions that I present. These versions are named the 0.5-layer model, the 1-layer model and the 1.25-layer model. The 0.5-layer model is a standard slab ocean model that does not include any representation of heat transport by ocean currents. The 1-layer model transports heat by solving the horizontal advection equation in the slab layer, while the 1.25-layer model assumes an infinite reservoir with a user prescribed temperature below the slab layer, and solves the three dimensional flux divergence equation.

As I show in Section 2.1.3, solving the horizontal advection equation in the slab layer (1-layer model) is equivalent to solving the three dimensional flux divergence equation with the assumption of an infinite underlying reservoir where the temperature in the reservoir is equal to the slab temperature. Therefore it is possible to create situations where divergence and the associated upwelling increases the mean temperature in the slab layer. If the domain integrated horizontal divergence is nonzero, there is a source or sink of energy depending on the sign of the vertical flux, and on the distribution of slab temperature. If combined with an atmospheric energy sink, such an oceanic energy source may be infinite. As one would imagine, an infinite energy sink is equally possible.

Upwelling in the ocean is generally associated with a local cooling in the ocean mixed layer (Codron 2012). In the 1.25-layer model cooling is ensured if the prescribed reservoir temperature is colder than the ocean mixed layer temperature in all grid points. Thus in case of divergence and upwelling the 1-layer- and the 1.25-layer model behave differently, while in the case of convergence and downwelling the two versions are exactly the same as the temperature of the water to be downwelled in both cases equals the ocean mixed layer temperature.

While the treatment of heat transport by ocean currents in the 1-layer model is analogous to that in the model by Maze et al. (2011), the 1.25-layer model is one step closer to the models suggested by Codron (2012). Codron (2012) proposes two models, one of which is a 2-layer model that assumes a return flow in the second layer of equal mass flux, but in the opposite direction of the surface flow. Codron (2012) shows that the 2-layer model is able to reproduce the mean meridional heat transport in the ocean in the tropics.

Because Codron's 2-layer model assumes that the flow in the lower layer is in the opposite direction to the surface flow, and the mass flux of equal magnitude, the mass and energy budgets can easily be closed. In a limited domain, as must be the case in Bedymo since the x-y coordinate system is not applicable in global simulations, these budgets are closed with a periodic or a zero flux condition on the boundaries. The advantage of this particular 2-layer model may also be its main drawback, because the assumption that the flow in the lower layer is the reversed copy of the surface flow is in many cases far from applicable.

Any horizontally divergence free flow field could be added to the flow in Codron's second layer, while still conserving mass and energy. If a divergent field were to be added one would need to solve an elliptical equation ($\nabla^2\Psi = \zeta(x, y)$) to balance the wind stress and the continuity simultaneously in the entire model domain, at each time step. Because of the added complexity such an implementation would require solid argumentation.

Bedymo lacks any representation of radiative processes. As a means of to some extent compensating for this lack of radiative processes in Bedymo, an optional temperature relaxation is included in all three versions of the slab ocean model. If e.g. the ocean mixed layer temperature is forced by an annual cycle in atmospheric temperature, the relaxation can be used to lower the amplitude and shift the phase of the resulting cycle in the ocean mixed layer temperature.

The relaxation always acts to restore the temperature to its prescribed equilibrium. Therefore if the distribution of the equilibrium temperature in the model domain is chosen to resemble the present climatology in some part of the world ocean, the effect of the relaxation can be comparable to that of a Q-flux.

In Chapter 2 I present, and evaluate the slab ocean model versions that are developed for Bedymo. I explain the 0.5-layer model, the 1-layer model and the 1.25-layer model in detail in Section 2.1. A brief introduction to the model versions is followed by an overview of the relevant numerics, a subject on which I have benefited greatly from the transport and time integration schemes already implemented in Bedymo. I will not provide a detailed explanation of these schemes.

After the numerics follows the model physics, which is broken down into sensible heat flux, heat transport and relaxation components of the total ocean mixed layer temperature tendency, analogous to how it is done in the model source code. The bulk formula for the calculation of Ekman currents is also provided, and briefly discussed.

The model description (Section 2.1) is followed by a section in which the model is tested in various idealized scenarios (Section 2.2). The model output is compared to corresponding analytical solutions to demonstrate that the model behaves as expected.

Section 2.2.5 is the final section of Chapter 2. Here a set of configurable parameters are varied to become familiar with their effect on the model output, as well as to suggest what are reasonable or less reasonable choices of the parameter values.

Chapter 3 contains the results from a coupled test run that was repeated with all three model versions. After the coupled test follows the summary and conclusion, and in the end three Appendices containing analytical solutions, additional figures for the transport tests and the slab ocean module of the Bedymo source code.

Chapter 2

Slab Ocean Model

2.1 Model Description

2.1.1 The three model versions

The 0.5-layer model (Figure 2.1) is a standard slab ocean model with an optional temperature relaxation towards a prescribed mean state of the ocean mixed layer. Equation (2.1) is the 0.5-layer version of the prognostic equation for ocean mixed layer temperature T. The tendencies on the right hand sides of Equations (2.1) - (2.3) will be explained in the physics section (2.1.3).

$$\frac{\partial T}{\partial t} = \left(\frac{\partial T}{\partial t}\right)_{\mathrm{surface\_heat\_flux}} + \left(\frac{\partial T}{\partial t}\right)_{\mathrm{relaxation}} \tag{2.1}$$

The 1-layer model (Figure 2.2 a) is an extension of the 0.5-layer model by the introduction of diagnostic and/or prescribed ocean currents. Relaxation is an optional feature also for the 1-layer model. Equation (2.2) is the 1-layer version of the prognostic equation for ocean mixed layer temperature.

$$\frac{\partial T}{\partial t} = \left(\frac{\partial T}{\partial t}\right)_{\mathrm{surface\_heat\_flux}} + \left(\frac{\partial T}{\partial t}\right)_{\mathrm{heat\_transport\_1layer}} + \left(\frac{\partial T}{\partial t}\right)_{\mathrm{relaxation}} \tag{2.2}$$

The 1.25-layer model (Figure 2.2 b) is a further extension of the 1-layer model that explicitly assumes an infinite reservoir of a prescribed temperature distribution beneath the ocean mixed layer. Relaxation is still an optional feature. Equation (2.3) is the 1.25-layer version of the prognostic equation for ocean mixed layer temperature.

$$\frac{\partial T}{\partial t} = \left(\frac{\partial T}{\partial t}\right)_{\mathrm{surface\_heat\_flux}} + \left(\frac{\partial T}{\partial t}\right)_{\mathrm{heat\_transport\_1.25layer}} + \left(\frac{\partial T}{\partial t}\right)_{\mathrm{relaxation}} \tag{2.3}$$

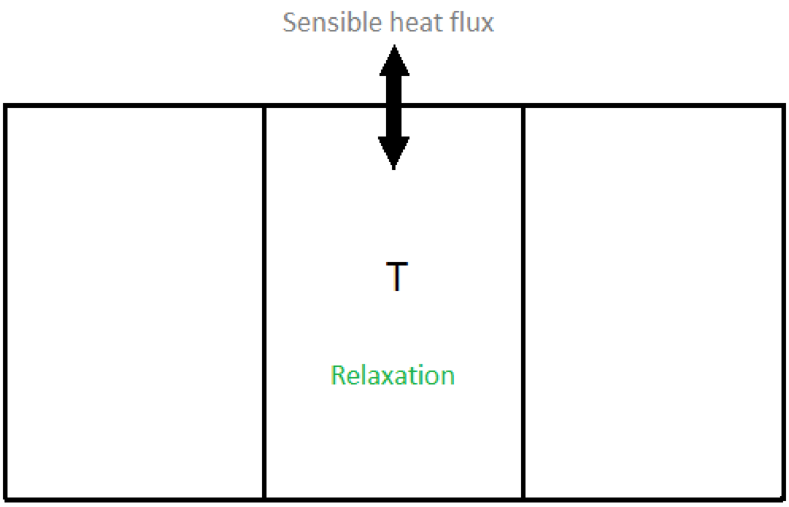
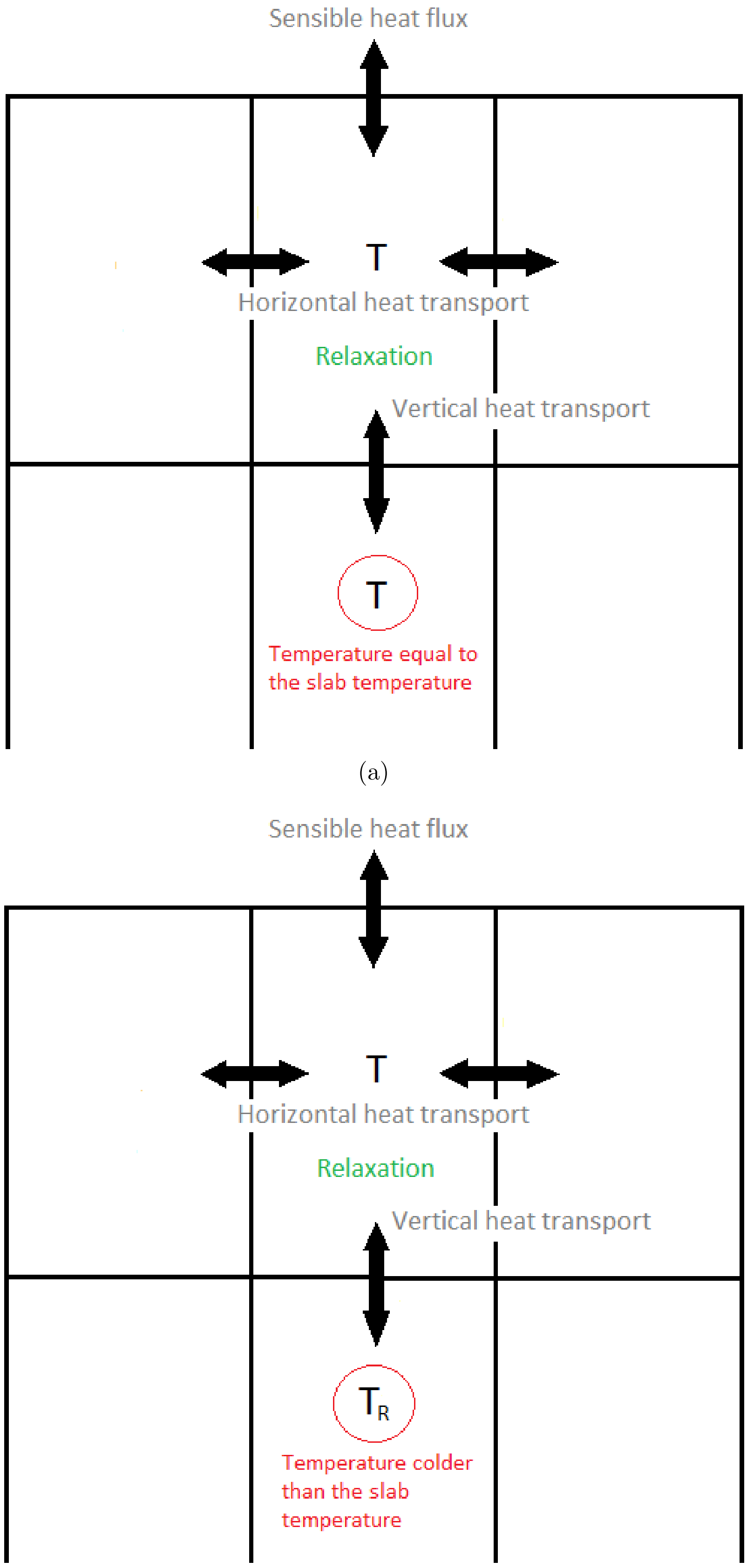


Figure 2.1: A conceptual sketch of the 0.5-layer model.


Sensible heat flux
T
Horizontal heat transport
Relaxation
Vertical heat transport
T
Temperature equal to the slab temperature
Sensible heat flux
T
Horizontal heat transport
Relaxation
Vertical heat transport
T_R
Temperature colder than the slab temperature

(a)

(b)

Figure 2.2: Conceptual sketches of the 1-layer model (a) and the 1.25-layer model (b).

2.1.2 Numerics

Bedymo uses an Arakawa C grid. As it is convenient that the atmosphere and ocean temperatures are defined in the same points, as well as it is equally convenient to let the ocean model easily make use of the transport schemes already implemented in Bedymo, it was a natural choice to use an Arakawa C grid also in the ocean model (Figure 2.3). The ocean mixed layer temperature is defined in the middle of the grid cells, while the ocean velocities are defined on grid cell boundaries. The vertical grid (Figure 2.4) is equivalent to the horizontal grid.

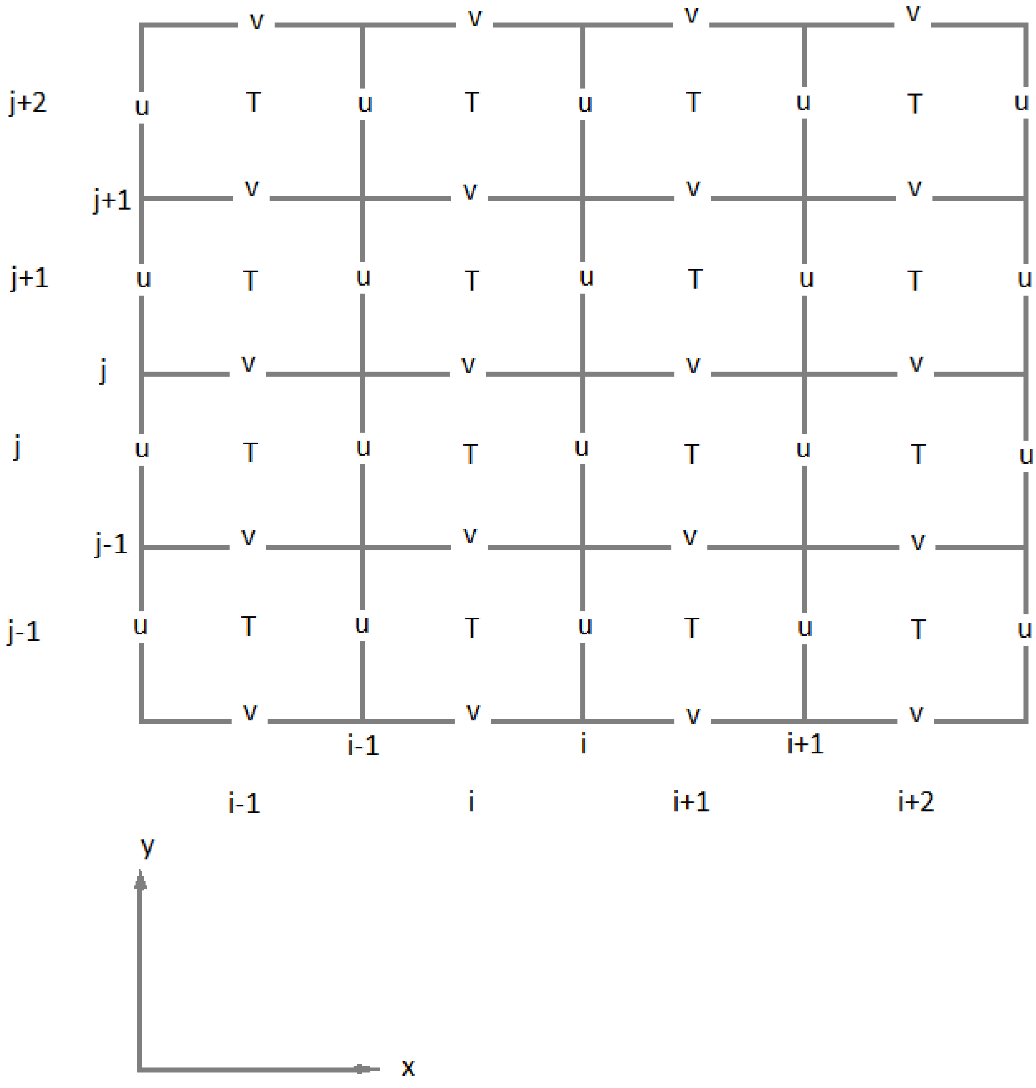


Figure 2.3: Sketch of the Arakawa C grid in the x-y plane. The outer indices represent grid cells and thus unstaggered quantities, while the inner indices represent the convention that is used for the staggered quantities in Fortran.

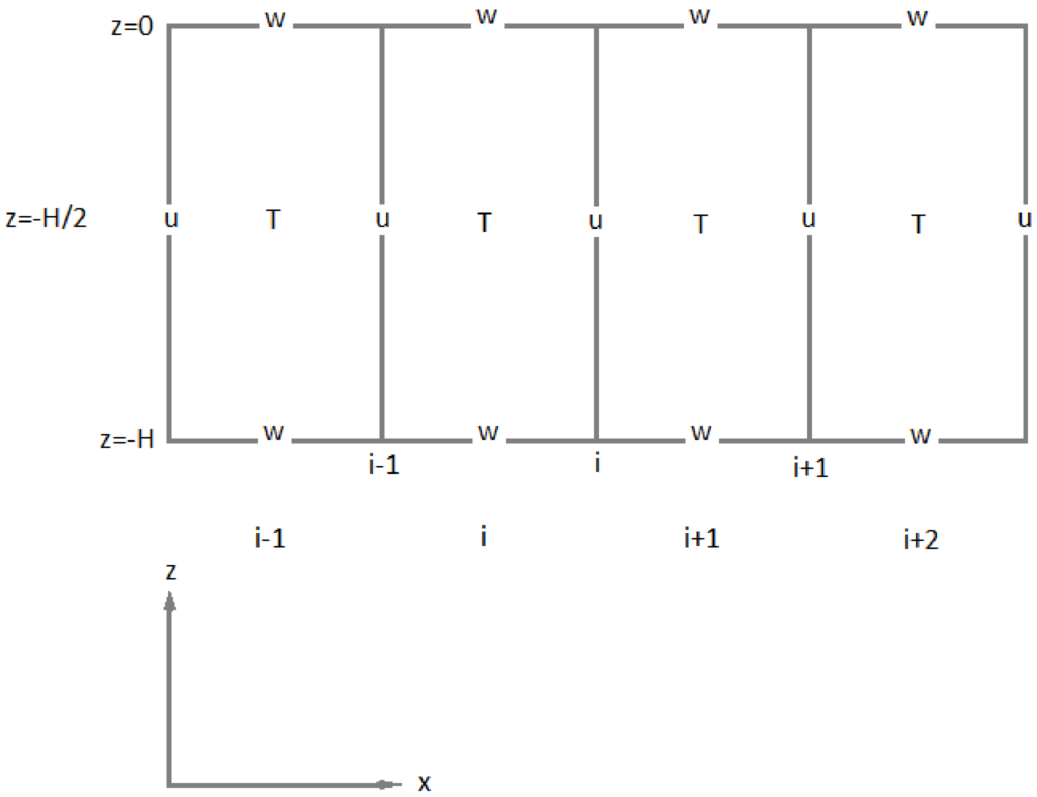


Figure 2.4: Sketch of the grid in the x-z plane. Outer indices represent unstaggered quanitities and inner indices represent staggered quantities.

Upwind schemes of order 1 to 6, were the order is a configurable variable, are already implemented in Bedymo and the ocean model makes use of these in order to calulate the heat transport part of the temperature tendency (Equations (2.2) and (2.3)). The detailed derivation of the schemes can be found in Spensberger (2012) or in Tremback et al. (1987).

Five time integration schemes are implemented in Bedymo. The schemes and their respective ID's are given in Table 2.1.

Time integration scheme	**ID**
Fourth order Runge Kutta	1
Miller Pearce	2
Quasi Implicit	3
Leap frog with RAW filter	4
Euler	5

Table 2.1: The time integration schemes implemented in Bedymo and their respective ID's.

The lateral boundary conditions for Bedymo and the slab ocean are also configurable. Table 2.2 summarizes the options. The periodic condition allows the ocean mixed layer temperature to reenter the model domain on the opposite side. This option is most useful when representing a circumglobal latitude band as model domain (Spensberger 2012).

The impermeable wall condition is represented by a zero gradient condition on ocean mixed layer temperature, a free slip condition for boundary parallel ocean velocities and a zero condition for boundary normal velocities.

Boundary condition	**ID**
Periodic	-1
Impermeable wall	0

Table 2.2: Lateral boundary conditions and their corresponding ID's.

Configurable variables are collected in the model namelist. Table 2.3 presents the namelist variables that are most relevant for the slab ocean model, including a short explanation of each variable, and a set of default values that are referred to as default#1.

Variable	**Explanation**	**Value**
nx	# of gridpoints in x	180
ny	# of gridpoints in y	180
dx	grid distance in x	222 222 m
dy	grid distance in y	222 222 m
dt	time step	0.5 d
latbrdtyp_ew	east-west boundaries	-1
latbdrtyp_sn	south-north boundaries	0
adv_order_max	order of upwind scheme	3
int_scheme_id	time integration scheme	4
tsstop	end time of model run	1095 d
dtout	output timestep	1 d
lcoriolis	coriolis force	false
lbetaeffect	beta effect	false
fcor	coriolis parameter	$1.0 \times 10^{-4}\,\mathrm{s}^{-1}$
betacor	d/dy(fcor)	0
lslab_ocean	slab ocean	true
l050layermodel	0.5-layer model	true
l100layermodel	1-layer model	false
l125layermodel	1.25-layer model	false
lekmantransp	Ekman transport	false
lrelaxation	relaxation	true
relaxation_coeff	relaxation coefficient	$0\,\mathrm{s}^{-1}$
drag_coeff	drag coefficient	1.3×10^{-3}
sh_exchange_coeff	sensible heat exchange coefficient	1.3×10^{-3}
mixed_layer_depth	mixed layer depth	50 m

Table 2.3: The default namelist for test runs with the 0.5-layer model (default#1).

2.1.3 Physics

The physics section explains the calculation of the surface heat flux, heat transport and relaxation tendencies seen in Equations (2.2) - (2.3), and provides the bulk formula for the Ekman currents.

Surface heat flux

For slab ocean models in general, the tendency of ocean mixed layer temperature due to surface heat fluxes is given by the following equation (Collins et al. 2004, Maze et al. 2011).

$$\left(\frac{\partial T}{\partial t}\right)_{\text{surface\_heat\_flux}} = \frac{F_{total}}{\rho_o H C_{po}}$$

Here T is the ocean mixed layer temperature, ρ_o is the ocean density, H the mixed layer depth, C_{po} the ocean heat capacity and F_{total} is the net heat flux through the atmosphere-ocean interface. The net flux F_{total} consists of contributions from shortwave radiation F_{sw}, longwave radiation F_{lw}, sensible heat transfer F_{sh} and latent heat transfer F_{lh}.

$$F_{total} = F_{sw} + F_{lw} + F_{sh} + F_{lh}$$

As Bedymo does not include parametrizations of moisture and radiation, the only contribution to F_{total} in the model that I develop here is the sensible heat flux which can be estimated by a standard bulk formula as given by e.g. Liu et al. (1979).

$$F_{sh} = \rho_a C_{pa} C_{sh} |\vec{u_a}| (T_a - T) \tag{2.4}$$

In Equation (2.4), ρ_a is the density of the atmosphere, C_{pa} the heat capacity of the atmosphere, C_{sh} the exchange coefficient for sensible heat, and $|\vec{u_a}|$ and T_a the surface atmospheric wind speed and temperature respectively ($\vec{u_a} = [u_a, v_a]$). The two latter are typically taken at 10 m height, but in BedymoPE (primitive equation version) they are defined at the lower atmospheric boundary.

Equation (2.4) looks rather simple, but can be vastly complicated by the fact that the exchange coefficient in reality is not at all constant, but depends on several factors in a way that requires iterative methods for solving. The reader is referred to Fairall et al. (2003) for a detailed description of the COARE algorithm, which is one of the most frequently used algorithms for this purpose.

The complexity of the COARE algorithm, and the models that typically use it (e.g. CAM3.0 (Collins et al. 2004)) suggests that it may not be the best fit for Bedymo. Though there may well be a middle way between the COARE algorithm and a constant coefficient, I choose to keep the sensible heat exchange coefficient constant. It is assigned a standard value of 1.3×10^{-3} which is the same value that was used by Maze et al. (2011), and lies somewhere in the middle of the range of values that the COARE algorithm produces for different conditions (Kara et al. 2005).

Heat transport

Equation (2.5) is the basis for the treatment of heat transport both in the 1-layer model and the 1.25-layer model.

$$\left(\frac{\partial T}{\partial t}\right)_{\text{heat\_transport}} = -\nabla \bullet (\vec{u}T) \tag{2.5}$$

Here $\vec{u}$ is the three dimensional velocity vector. Dividing the right hand side of Equation (2.5) into partial derivatives with respect to x, y and z yields the following.

$$\left(\frac{\partial T}{\partial t}\right)_{\text{heat\_transport}} = -\frac{\partial}{\partial x}(uT) - \frac{\partial}{\partial y}(vT) - \frac{\partial}{\partial z}(wT) \tag{2.6}$$

The first two terms on the right hand side are discretized as follows, using the x-direction as example.

$$\frac{\partial}{\partial x}(uT)|_i \approx \frac{(uT)|_{i+\frac{1}{2}} - (uT)|_{i-\frac{1}{2}}}{\Delta x}$$

The horizontal wind is directly available on the staggered grid, while the temperature is interpolated using the implemented upwind schemes (Tremback et al. 1987) of order 1 to 6.

The third term on the right hand side of (2.6) is discretized in an equivalent way.

$$\frac{\partial}{\partial z}(wT)|_{z=-\frac{H}{2}} \approx \frac{(wT)|_{z=0} - (wT)|_{z=-H}}{H}$$

The vertical velocity at the surface $w(x, y, 0)$ is known and equal to zero leading to Equation (2.7).

$$\frac{\partial}{\partial z}(wT)|_{z=-\frac{H}{2}} \approx -\frac{w|_{z=-H}T|_{z=-H}}{H} \tag{2.7}$$

Because the ocean is assumed to be of constant density, the equation of continuity reduces to $\nabla \bullet \vec{u} = 0$ which in turn leads to

$$\int_{-H}^{0} \frac{\partial w}{\partial z}\, dz = -\int_{-H}^{0} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) dz \tag{2.8}$$

and

$$w(x, y, -H) = H\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right). \tag{2.9}$$

The temperature on grid cell vertical boundaries ($z = 0$ and $z = -H$) is determined by interpolation between the adjacent volumes. This is done using a first order upwind scheme to ensure that the temperature at the lower grid cell boundary always equals the temperature in the upwind volume.

The 1-layer model leaves only one option for the temperature at the lower boundary and that is for it to be equal to the slab (ocean mixed layer) temperature both for upward and downward mass fluxes. Combining Equation (2.6) and Equation (2.7) leads to

$$\left(\frac{\partial T}{\partial t}\right)_{\text{heat\_transport\_1layer}} = -\frac{\partial}{\partial x}(uT) - \frac{\partial}{\partial y}(vT) + \frac{wT}{H} \tag{2.10}$$

where $w = w(x, y, -H)$ from now on. Substituting the right hand side of Equation (2.9) for w in Equation (2.10) and applying the chain rule of differentiation to the terms involving the horizontal fluxes shows that Equation (2.10) is just another way of writing the horizontal advection equation (2.11).

$$\left(\frac{\partial T}{\partial t}\right)_{\text{heat\_transport\_1layer}} = -u\frac{\partial T}{\partial x} - v\frac{\partial T}{\partial y} \tag{2.11}$$

The 1.25-layer model assumes a reservoir of infinite depth below the slab layer. The temperature T_R of the water in this reservoir is a prescribed variable. Because the temperature at the lower boundary is determined by the direction of the flux through the lower boundary we get

$$\frac{\partial}{\partial z}(wT)|_{z=-\frac{H}{2}} \approx -\frac{w|_{z=-H}T|_{z=-H}}{H} = -\frac{\max(w,0)T_R}{H} - \frac{\min(w,0)T}{H} \tag{2.12}$$

Equation (2.6) then becomes

$$\left(\frac{\partial T}{\partial t}\right)_{\text{heat\_transport\_1.25layer}} = -\frac{\partial}{\partial x}(uT) - \frac{\partial}{\partial y}(vT) + \frac{\max(w,0)T_R}{H} + \frac{\min(w,0)T}{H} \tag{2.13}$$

Model currents

The model currents consist of an Ekman current that is diagnosed from the atmospheric wind, and an optional background current that must be prescribed. The Ekman currents are

$$u_e = \frac{\rho_a C_D|\vec{u_a}|}{\rho_o H(\epsilon^2 + f^2)}(\epsilon u_a + f v_a) \tag{2.14}$$

$$v_e = \frac{\rho_a C_D|\vec{u_a}|}{\rho_o H(\epsilon^2 + f^2)}(\epsilon v_a - f u_a) \tag{2.15}$$

In (2.14) and (2.15) u_a and v_a are the x and y components of the atmospheric surface wind, C_D the drag coefficient, f is the Coriolis parameter and ϵ is a parameter representing various dissipative effects including horizontal momentum fluxes (Codron 2012). Codron (2012) also refers to ϵ as the inverse damping timescale of oceanic currents. The presence of ϵ allows the computation of wind driven currents also near the equator where $f \approx 0$. When f is much larger than ϵ the classic Ekman drift formula is recovered. In his studies Codron (2012) uses $\epsilon = 10^{-5}\text{s}^{-1}$ which gives a transition latitude ($\epsilon = f$) at 4°. The transition latitude means simply the latitude at which the the resulting Ekman transport drifts 45° to the right or left of the surface atmospheric wind, and it depends on the choice of ϵ.

Far away from the equator ϵ does not matter, but near the equator there will be a region where the Coriolis force is weak, the width of this region depending on the value of ϵ. Depending on the setup it might even be less wide than the model resolution, causing an intense and narrow upwelling centered at the equator, in contrast to a wider and weaker upwelling for a larger value of ϵ. In the studies by Codron (2012) the structure of the upwelling had consequences for tropical SST and precipitation.

The background currents are left for the user to prescribe. The source code includes the fields that I used for the ocean tests in Section 2.2, but nothing more at the moment. The variables 'ub' and 'vb' in the slab ocean module of the Fortran source code (bedymo/src_ocean/slab.f95) are the background currents that together with the Ekman currents form the total ocean currents.

Relaxation

All three model versions (0.5-layer-, 1-layer- and 1.25-layer model) include an optional temperature relaxation towards a prescribed equilibrium state of the ocean mixed layer. The relaxation part of the ocean mixed layer temperature tendency is given in Equation (2.16) where α is the relaxation coefficient and T_E is the equilibrium temperature.

$$\left(\frac{\partial T}{\partial t}\right)_{\text{relaxation}} = -\alpha(T - T_E) \tag{2.16}$$

The relaxation always acts to restore the ocean mixed layer temperature T to it's prescribed equilibrium temperature T_E. If $T \neq T_E$ and nothing else forces the ocean mixed layer temperature, the equilibrium will be reached after a period of time depending on the relaxation coefficient.

The choice of relaxation coefficient is not necessarily straight forward. A relaxation coefficient that is too small will make the relaxation component of the total temperature tendency negligible compared to the surface flux and transport components, while a relaxation coefficient that is too large will do the exact opposite, and make it almost impossible to change the ocean mixed layer temperature. Based on the tests presented in Section 2.2.5 reasonable values for the relaxation coefficient appear to range from 1.0×10^{-8} to 1.0×10^{-6}, where both ends of the range are best considered extreme cases.

The best suitable value of α depends on the application. For an experiment involving a certain part of the world ocean an analysis of the time lag between maximum atmospheric and maximum oceanic temperatures from the region in question could suggest a reasonable value. This method is most reasonable if the equilibrium temperature is also prescribed based on analyses of ocean mixed layer temperature from the region in question.

In a different scenario if e.g. relaxation is used to maintain an anomaly in the ocean mixed layer temperature to force the atmosphere, the best value might be larger than what seems physically reasonable based on an analysis of atmosphere and ocean temperatures. In such a case the solution may just as well be to find a value by trial and error.

2.2 Model Evaluation

Here I present the results from a series of idealized test runs with the three model versions. First the 0.5-layer model is forced with an annual cycle in atmospheric temperature in four slightly different setups. Then the 1-layer model and the 1.25-layer model are both run with four idealized background flow fields. A warm temperature anomaly that is to be advected by the different flow fields is prescribed as the initial condition for ocean mixed layer temperature. The same anomaly is the initial condition for the next set of runs, where the 1-layer- and 1.25-layer models are forced by a rigid body atmospheric vortex to test the calculation of the Ekman currents, and to highlight the difference between these two model versions. The rigid body vortex is reused to perform a rotational test in the ocean. Then follows a couple of sensitivity tests, and the main discussion, to mark the end of Section 2.2.

The aim of Section 2.2 is to demonstrate that the slab ocean behaves as expected. To achieve this I compare the model results to analytical solutions. The analytical solutions are presented in Appendix A. For the atmospheric rigid body vortex case, no analytical solution is provided. The focus is instead on the difference between the 1-layer- and 1.25-layer models. For the rotational tests in the ocean, the analytical solution is equal to the initial anomaly.

2.2.1 The 0.5-layer model forced with an annual cycle in atmospheric temperature: Numerical versus Analytical solutions

The atmospheric temperature (T_a) cycle is given by (2.17). A constant atmospheric surface wind of $10\,\mathrm{m\,s^{-1}}$ is also prescribed as the sensible heat flux would be zero in case of zero surface wind (2.4).

$$T_a(t) = T_0 + A\sin(\frac{2\pi t}{P} - \frac{\pi}{2}) \tag{2.17}$$

The amplitude A is equal to 10 K, the period P equal to one year and T_0 is equal to 283.15 K. As indicated by (2.17) no spatial dependency is prescribed for the atmospheric temperature. The same is true for the initial ocean mixed layer temperature. As there is also no transport of heat in the 0.5-layer model the consequence is that there is no spatial dependency in the solution.

The initial ocean mixed layer temperature is equal to T_0 in all four runs. For the numerical solution to be comparable to the analytical the ocean first needs to adapt to the forcing. This is why I have removed the first year from all time series.

The default namelist for the test runs in Section 2.2.1 is provided in Table 2.3 (Section 2.1.2). The first run is performed with the default setup. Another run is performed with the same time step, but a nonzero relaxation coefficient, and then these two runs are repeated, the time step set to ten times the default.

Figure 2.5 shows the numerical solution for ocean mixed layer temperature normalized between zero and one (solid curves), and the atmospheric temperature scaled accordingly (dashed curves). The subfigure titles indicate the values of the time step dt and the relaxation coefficient α.

Figure 2.6 shows the error in the numerical solution measured in percentage of two times the amplitude of the oceanic temperature cycle. A larger initial error, and a slight decay in the error with time can be seen in the two subfigures on the left hand side, where the relaxation coefficient is equal to zero. This is just a result of the ocean mixed layer being initialized at a temperature that in most cases, though depending on a set of parameters, does not satisfy the analytical solution at time $t = 0$. Thus the reason that the same decay in the error is not seen when the relaxation coefficent is nonzero, is likely a result of the initial temperature being a better fit for that particular combination of parameter values, or that the relaxation helps the ocean adjust quicker to the atmospheric forcing.

The two subfigures on the right hand side of Figure 2.6 are, out of these few examples, considered the best representation of the error actually related to the numerical integration. As is expected the larger time step, produces a larger relative error. Although both time steps used here are way larger than what the atmosphere would be expected to handle, they might both be used if the ocean and atmosphere were to be run with separate time steps.

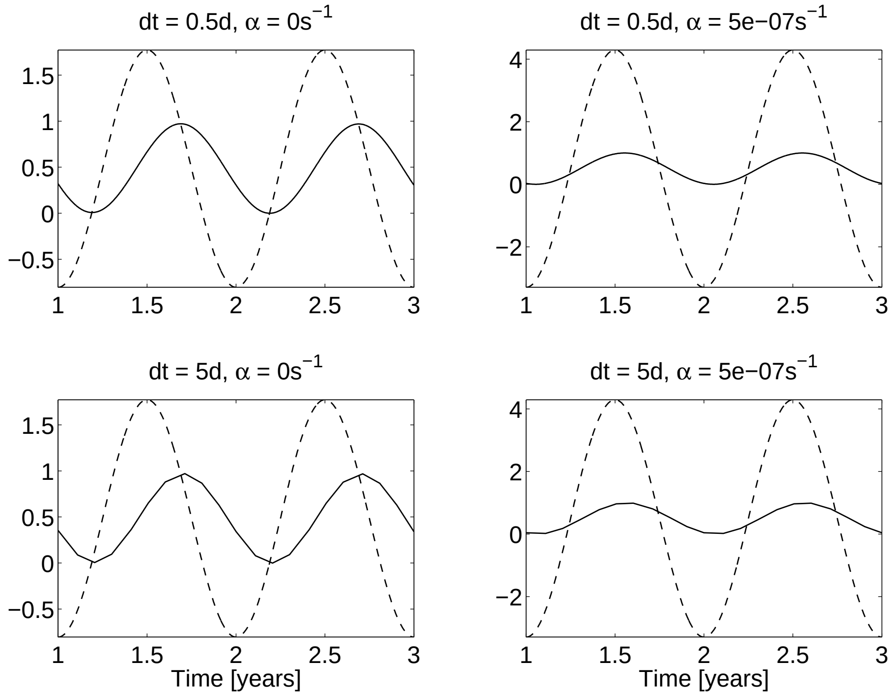


Figure 2.5: Time series of atmospheric temperature (dashed curves), and the resulting ocean mixed layer temperature (solid curves).

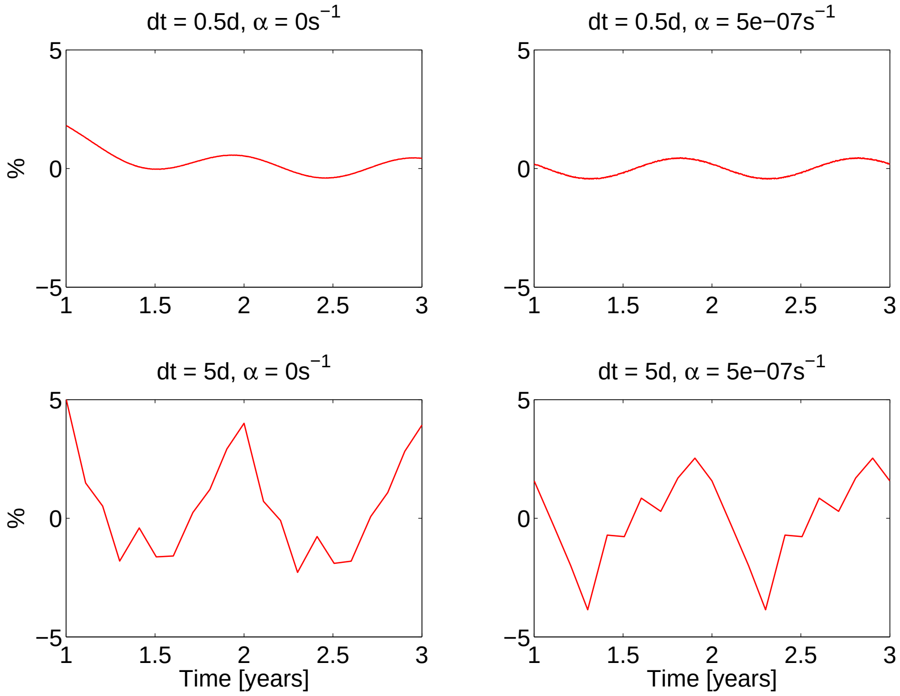


Figure 2.6: The difference between the numerical solutions (Figure 2.5), and the corresponding analytical solutions.

2.2.2 Heat transport by prescribed idealized flow fields in the 1-layer- and 1.25-layer model: Numerical versus Analytical solutions

Variable	**Explanation**	**Value**
nx	# of gridpoints in x	180
ny	# of gridpoints in y	180
dx	grid distance in x	222 222 m
dy	grid distance in y	222 222 m
dt	time step	2.5 d
latbrdtyp_ew	east-west boundaries	-1
latbdrtyp_sn	south-north boundaries	0
adv_order_max	order of upwind scheme	3
int_scheme_id	time integration scheme	4
tsstop	end time of model run	3650 d
dtout	output time step	20 d
lcoriolis	coriolis force	true
lbetaeffect	beta effect	false
fcor	coriolis parameter	$1.0 \times 10^{-4}\,\mathrm{s}^{-1}$
betacor	d/dy(fcor)	0
lslab_ocean	slab ocean	true
l050layermodel	0.5-layer model	false
l100layermodel	1-layer model	true
l125layermodel	1.25-layer model	false
lekmantransp	Ekman transport	false
lrelaxation	relaxation	false
relaxation_coeff	relaxation coefficient	$1.0 \times 10^{-7}\,\mathrm{s}^{-1}$
drag_coeff	drag coefficient	1.3×10^{-3}
sh_exchange_coeff	sensible heat exchange coefficient	1.3×10^{-3}
mixed_layer_depth	mixed layer depth	50 m

Table 2.4: The default namelist for test runs with the 1-layer model and the 1.25-layer model (default#2).

Table 2.4 provides the default namelist for the transport tests. The only deviations from the default occurs when the 1.25-layer model is used instead of the 1-layer model, or when the prescribed flow field is given by (2.20) or (2.21) (latbdrtyp_ew = 0).

The idealized flow fields are given in Equations (2.18) - (2.21) where $j = 1, \ldots, n_y$, $i = 0, \ldots, n_x$ and $u_0 = 0.1\,\mathrm{m\,s^{-1}}$. There is no flow in the north-south direction in any of these fields, i.e. $v(j,i) = 0$.

$$\text{Homogeneous flow:} \quad u(j,i) = u_0 \tag{2.18}$$

$$\text{Shear flow:} \quad u(j,i) = u_0 \frac{j}{n_y} \tag{2.19}$$

$$\text{Convergent flow:} \quad u(j,i) = u_0 \left(1 - \frac{i}{n_x}\right) \tag{2.20}$$

$$\text{Divergent flow:} \quad u(j,i) = u_0 \frac{i}{n_x} \tag{2.21}$$

As the velocities are prescribed also on the western and eastern boundaries of the model domain the only quantity to be set by an additional condition is the temperature. For the homogeneous and shear flow fields a periodic condition makes sense, but not so much for the convergent and divergent fields. When the convergent or divergent field is used the temperature is set by a zero gradient condition. As there are no waves, but only advection in the slab ocean model the boundaries may thus, in these two cases, be viewed as one open boundary and one impermeable wall.

The first test is done with the homogeneous field (2.18). Figure 2.7 shows the numerical solution, normalized with respect to the amplitude of the initial anomaly. As one would hope the shape of the anomaly looks to be well conserved. Figure 2.8 shows the error in the numerical solution, measured as the difference between the numerical and analytical solution, in percentage of the amplitude of the initial anomaly. The maximum error is very small ($< 0.35\%$) already with an advection scheme of third order. The anomaly itself is subject to a minor damping, and surrounded by an almost continuous region, covering more than half the model domain where the relative error is around -1%.

Now what happens if a linear y-dependence is prescribed in the flow field? Figure 2.9 shows the result from the 1-layer model, when the shear flow is used. In this case it would be more of a surprise if the shape of the anomaly did not change. As seen in Figure 2.9 the shape is deformed, but the total area covered by the anomaly looks like it may not have changed. Figure 2.10 shows the error, which is one order of magnitude larger than for the homogeneous field, and distributed in a slightly different pattern.

The analytical solutions become more complex once an x-dependence is prescribed in the x-component of the velocity fields (Section A.2.3). Does the error in the numerical solution increase accordingly? Figures 2.11 and 2.13 show the numerical solutions from the 1-layer model for the convergent and divergent flow fields. The initial shape is now not only deformed, but also stretched or squeezed to cover a larger or smaller area. Figure 2.12 shows the error for the convergent field, while Figure 2.14 shows the error for the divergent field. The maximum error encountered so far is seen in Figure 2.12 for the convergent field.

The transport tests so far, are all performed with the 1-layer model. For the homogeneous, shear, and convergent flow fields no differences are expected with the 1.25-layer model, and no differences do occur (Appendix B, Figure B.1 and Figure B.2). In the case of divergence a noteable difference is expected if the prescribed reservoir temperature is colder than the ocean mixed layer temperature. To evaluate also the upwelling component of the 1.25-layer model, the result (Figure 2.15) is again compared to the corresponding analytical solution (Figure 2.16). The error is comparable to the error produced by the 1-layer model for the same flow field.

It should be mentioned that in the figures that follow (2.7 - 2.16) the quantitity on the x and y axes is number of grid points.

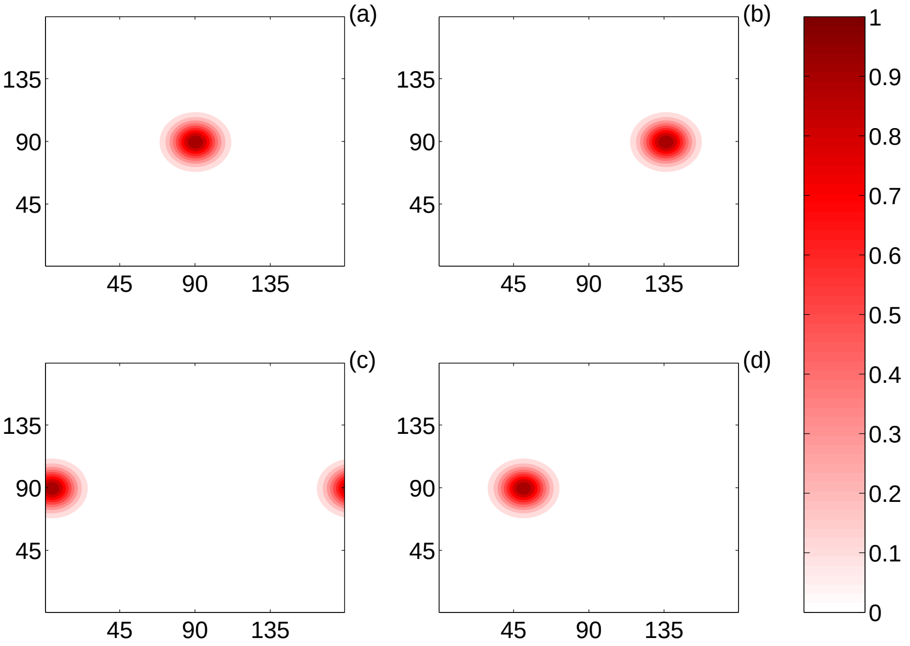


Figure 2.7: Normalized ocean mixed layer temperature from the 1-layer model when the prescribed flow field is given by Equation (2.18) at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

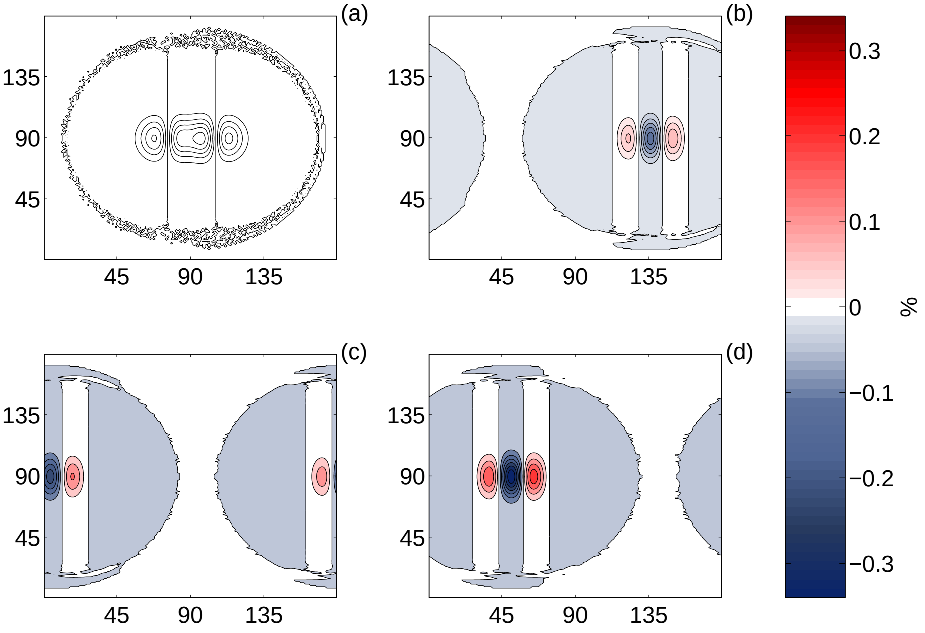


Figure 2.8: The error in the numerical solution with respect to the analytical, measured in % of the magnitude of the initial anomaly, after (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

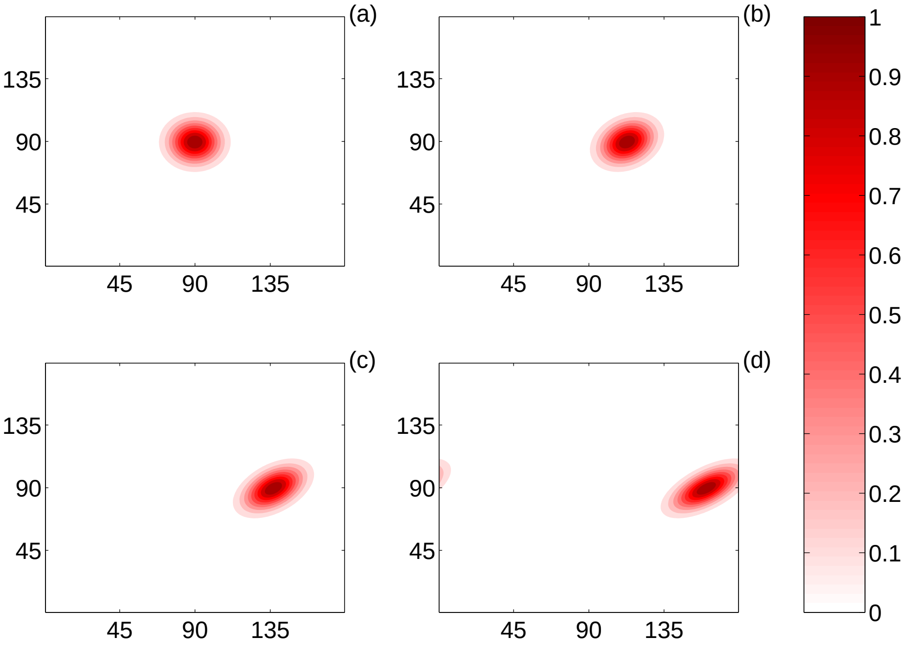


Figure 2.9: Normalized ocean mixed layer temperature from the 1-layer model when the prescribed flow field is given by Equation (2.19) at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

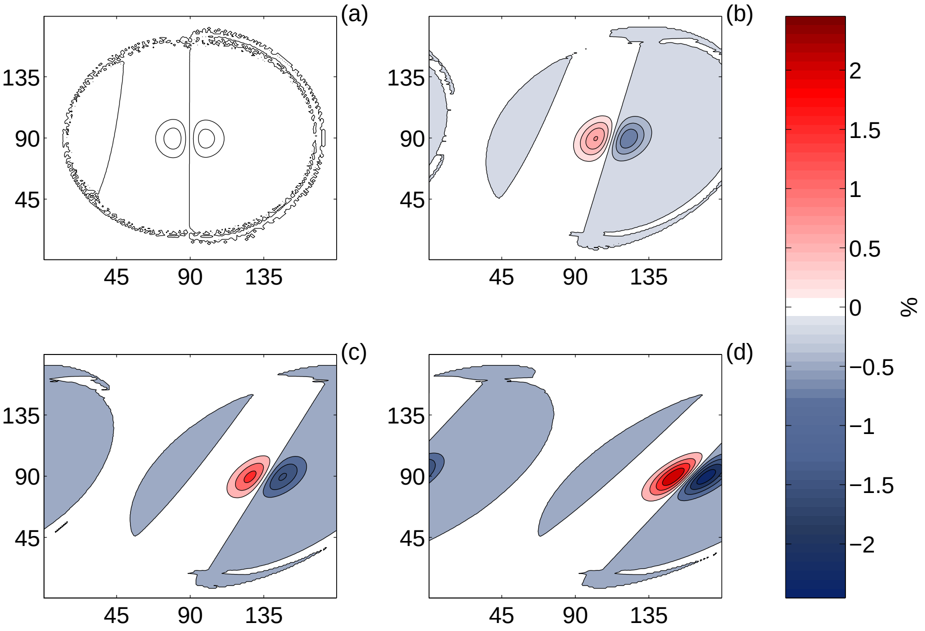


Figure 2.10: The error in the numerical solution with respect to the analytical, measured in % of the magnitude of the initial anomaly, after (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

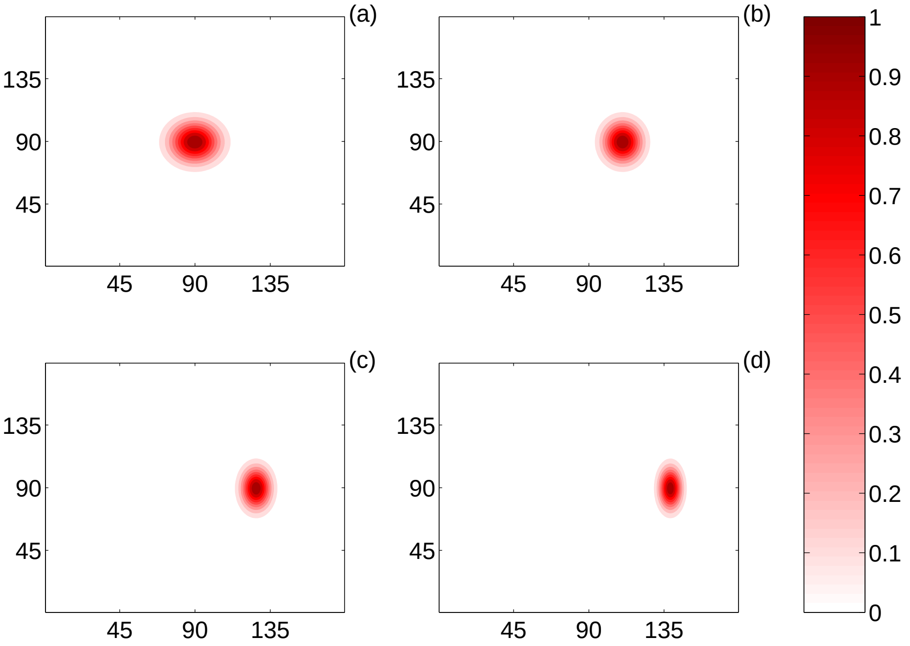


Figure 2.11: Normalized ocean mixed layer temperature from the 1-layer model when the prescribed flow field is given by Equation (2.20) at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

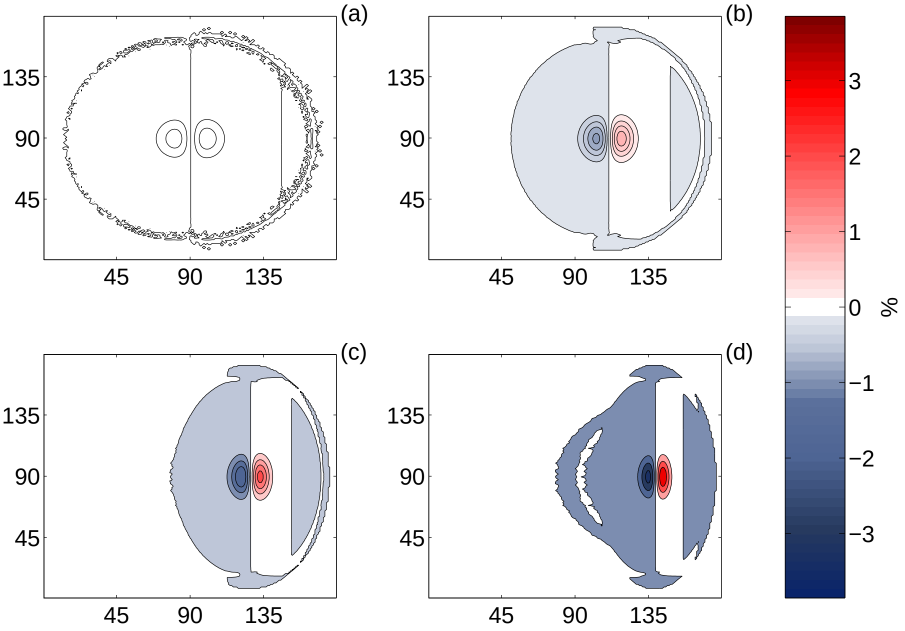


Figure 2.12: The error in the numerical solution with respect to the analytical, measured in % of the magnitude of the initial anomaly, after (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

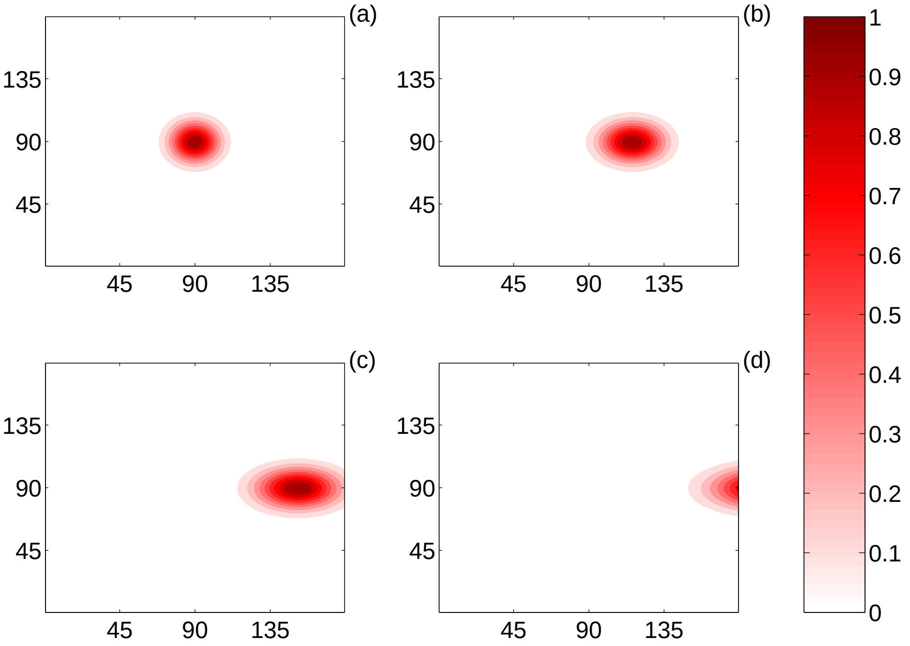


Figure 2.13: Normalized ocean mixed layer temperature from the 1-layer model when the prescribed flow field is given by Equation (2.21) at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

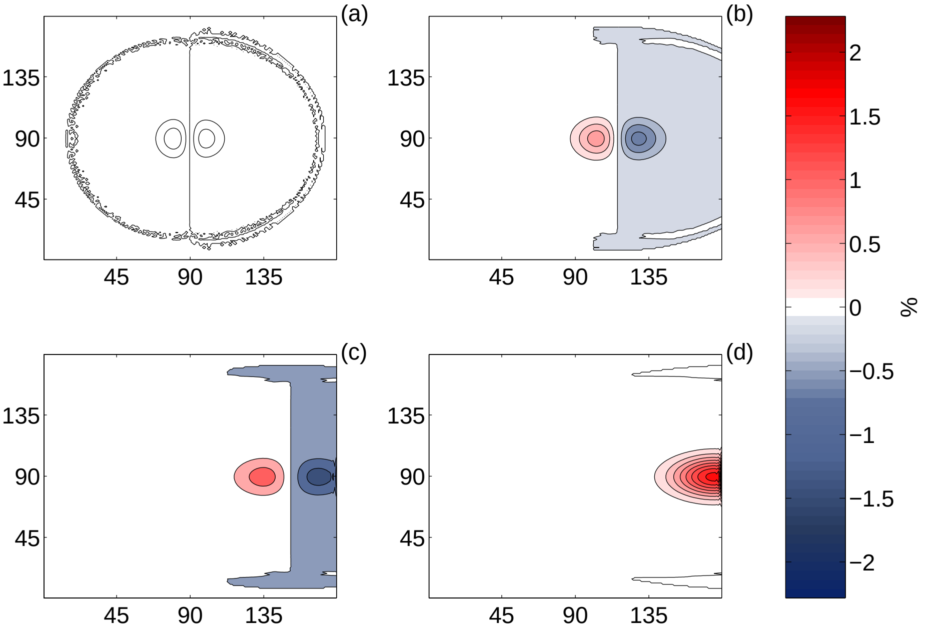


Figure 2.14: The error in the numerical solution with respect to the analytical, measured in % of the magnitude of the initial anomaly, after (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

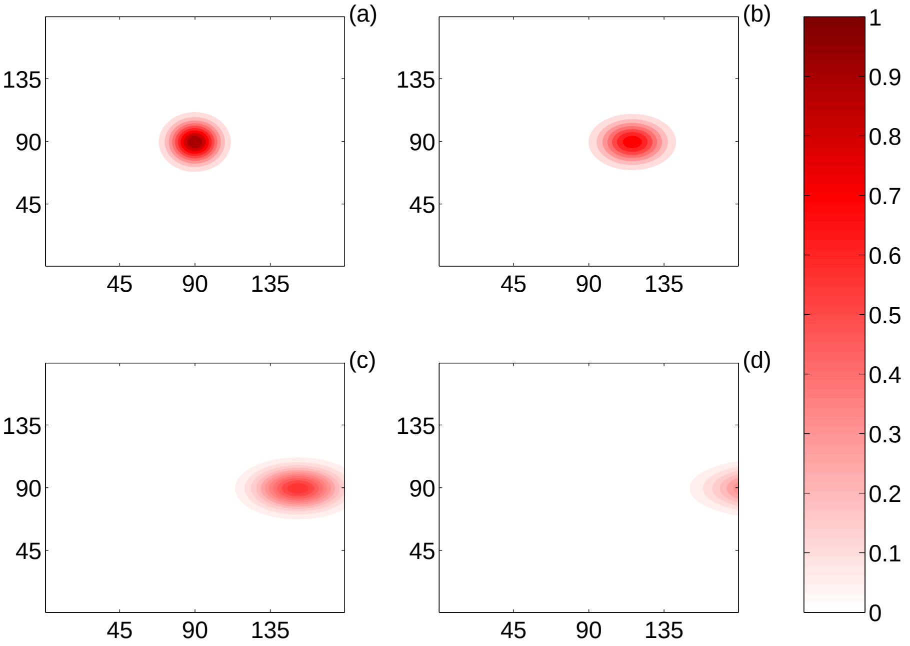


Figure 2.15: Normalized ocean mixed layer temperature from the 1.25-layer model when the prescribed flow field is given by Equation (2.21) at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

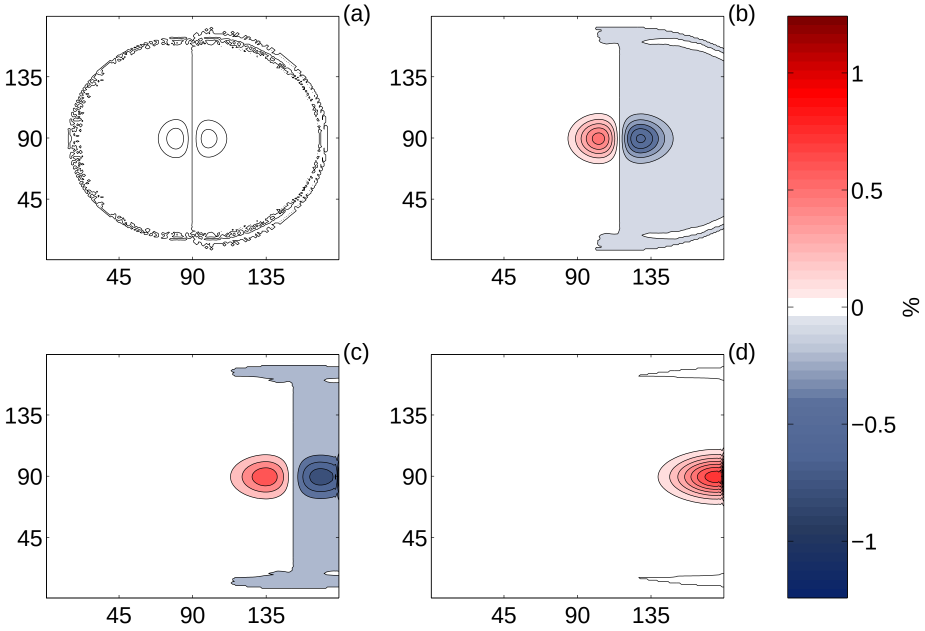


Figure 2.16: The error in the numerical solution with respect to the analytical, measured in % of the magnitude of the initial anomaly, after (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

2.2.3 Ekman transport in the 1-layer model versus the 1.25-layer model when forced with a rigid body atmospheric vortex

$$u(j,i) = -\frac{u_0}{0.5n_y}(j - 0.5n_y) \quad j = 1, \ldots, n_y \quad i = 0, \ldots, n_x \tag{2.22}$$

$$v(j,i) = \frac{u_0}{0.5n_x}(i - 0.5n_x) \quad j = 0, \ldots, n_y \quad i = 1, \ldots, n_x \tag{2.23}$$

The main objective of this test is to confirm that the calculation of the Ekman currents turns out as expected. The background currents are set to zero, and the Ekman currents are enabled by setting the logical 'lekmantransp' to true. The ocean is forced by a rigid body cyclonic atmospheric vortex, (2.22) and (2.23), that is easily prescribed, and ensures divergence in the entire model domain, which helps highlighting the difference between the 1-layer model and the 1.25-layer model. The constant u_0 is equal to $15\,\mathrm{m\,s^{-1}}$. The boundaries for ocean mixed layer temperature are set by a zero gradient condition, and the initial condtion is the same warm anomaly as before.

Figure 2.17 shows the result from the 1-layer model. The anomaly is stretched in every direction, and as a result the domain integrated mixed layer temperature increases. What actually happens is that the net horizontal transport of water out of a grid cell is compensated by an upwelling of equally warm water. Thus there is an energy source that, if the model run was extended for long enough, would heat the mixed layer until every grid cell had the same temperature.

Figure 2.18 shows the result from the 1.25-layer model. The prescribed reservoir temperature is constant at 283.15 K. In this case the anomaly is weakened, and the domain integrated mixed layer temperature decreases. If this run were to be extended long enough the mixed layer temperature would eventually become equal to the reservoir temperature in all grid points.

Figure 2.19 shows the difference between the results from the 1-layer model and the 1.25-layer model. The maximum difference between the two model versions is larger than 30% of the amplitude of the initial temperature anomaly.

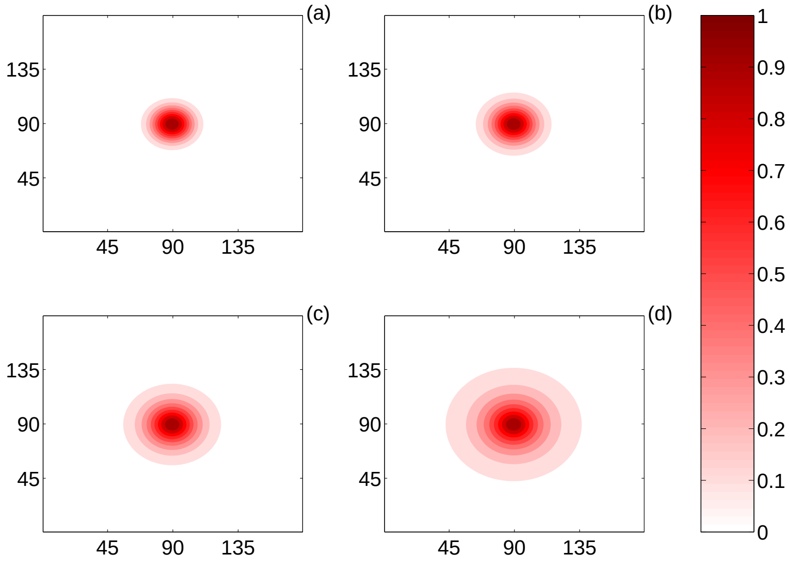


Figure 2.17: The result from the 1-layer model when forced with a rigid body cyclonic vortex at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d. Normalized with respect to the amplitude of the initial anomaly.

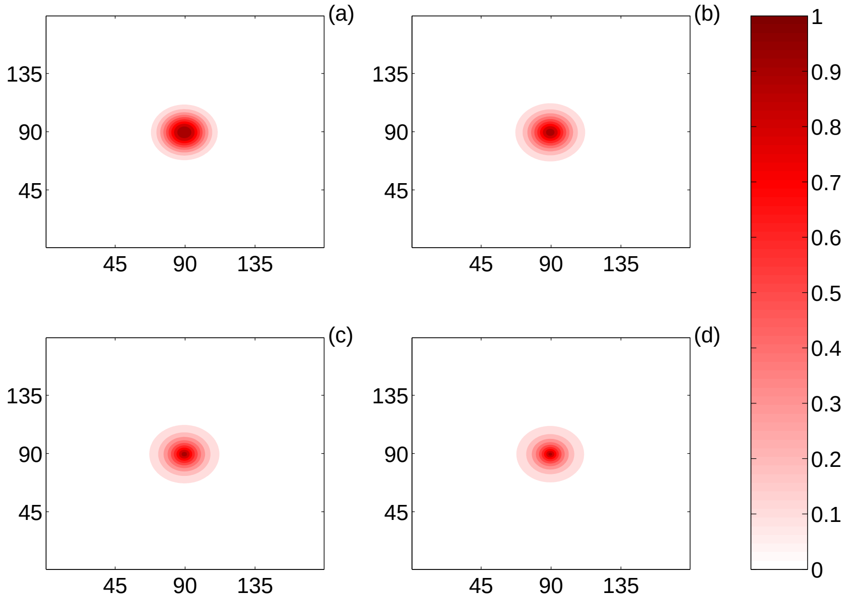


Figure 2.18: The result from the 1.25-layer model when forced with a rigid body cyclonic vortex at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d. Normalized with respect to the amplitude of the initial anomaly.

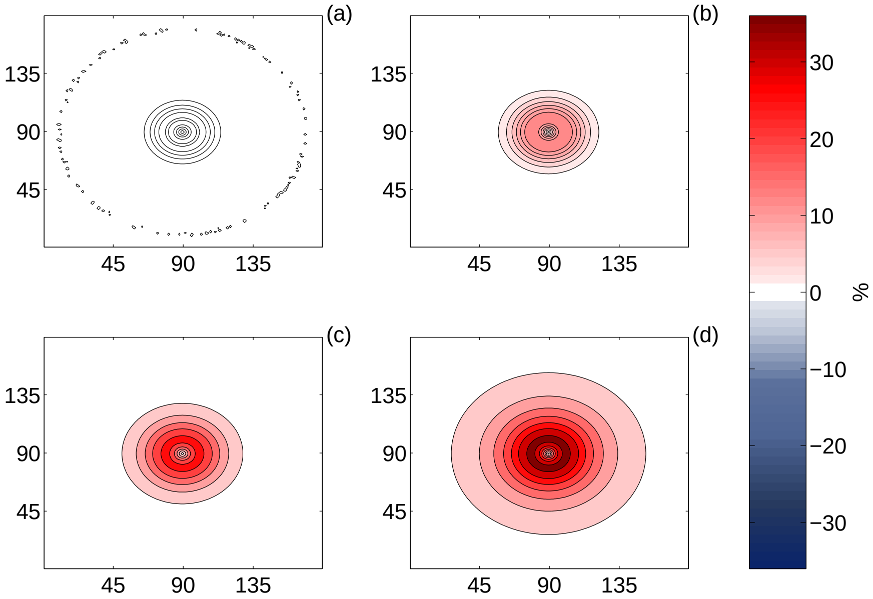


Figure 2.19: The difference between the 1-layer model and the 1.25-layer model measured in percentage of the amplitude of the initial anomaly.

2.2.4 Tests with a rigid body vortex in the ocean

The rigid body cyclonic vortex, (2.22) and (2.23), is reused here, with $u_0 = 0.4\,\mathrm{m\,s^{-1}}$. The vortex is now prescribed as the background flow in the ocean to perform a test resembling that by Tremback et al. (1987). The test is repeated for the second, third, fifth and sixth order of the upwind schemes. The model runs end after one full rotation, and the final result is compared to the initial temperature profile, which remains the same as in the previous tests (e.g Figure 2.18 (a)).

Figure 2.20 shows the resulting error with the second order upwind scheme, Figure 2.21 the error with the third order scheme, Figure 2.22 the error produced by the fifth order scheme and Figure 2.23 shows the error that results when the sixth order scheme is used.

The odd and even ordered schemes display significant differences in the pattern of the errors. While the odd ordered schemes mainly damp the initial anomaly, the errors from the even ordered schemes have a more dispersive nature. This is believed to be consistent with the results of Tremback et al. (1987). Eventual differences may come from the length of the runs, a different method of time integration, different grids and resolution etc.

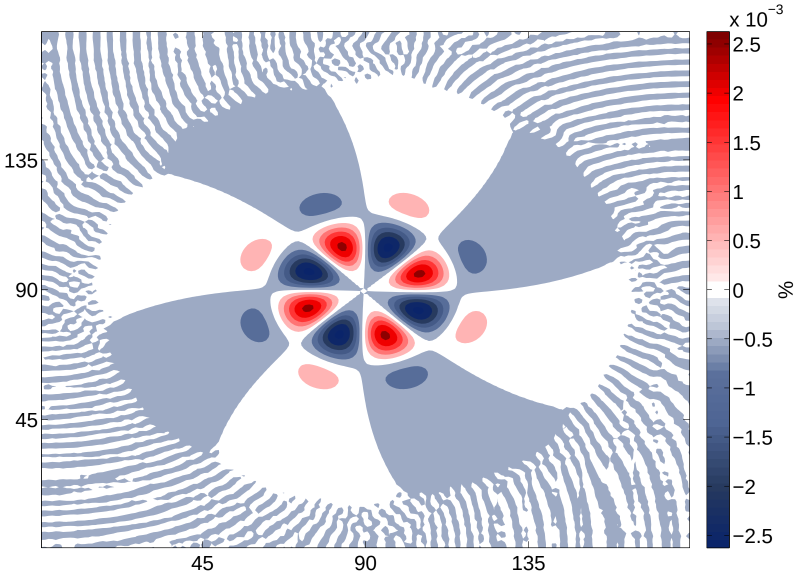


Figure 2.20: The error with the second order upwind scheme.

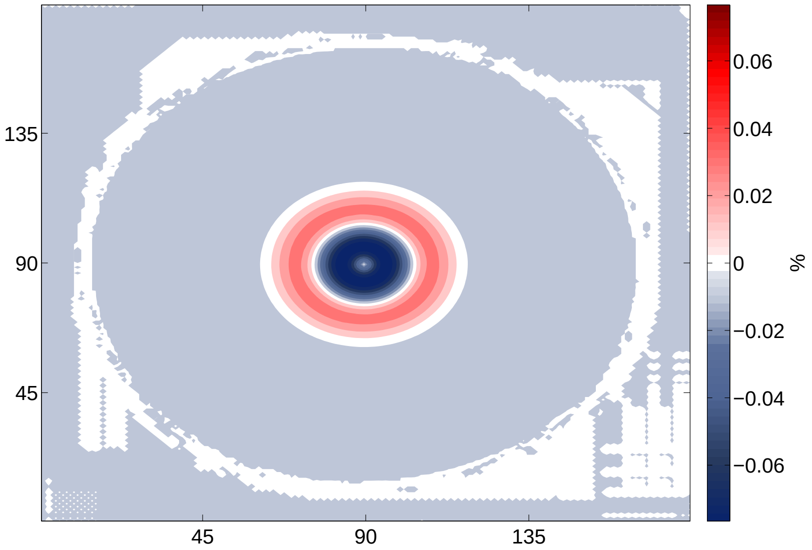


Figure 2.21: The error with the third order upwind scheme.

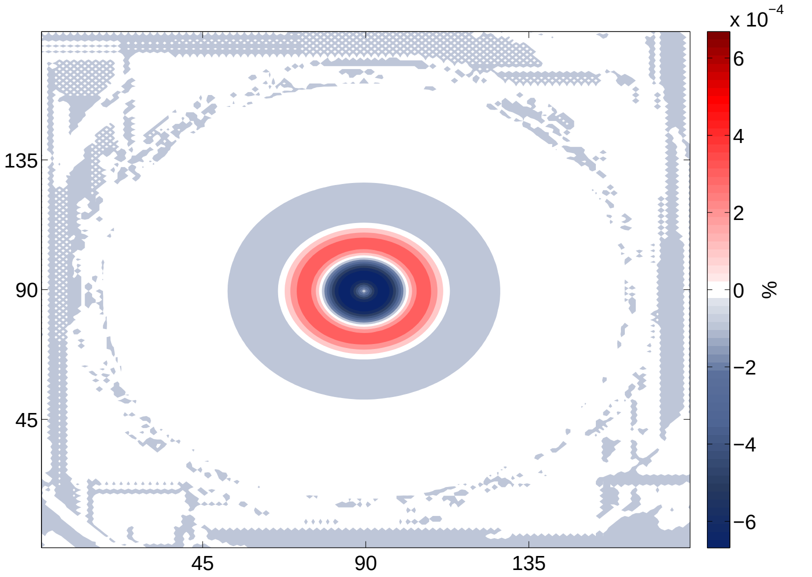


Figure 2.22: The error with the fifth order upwind scheme.

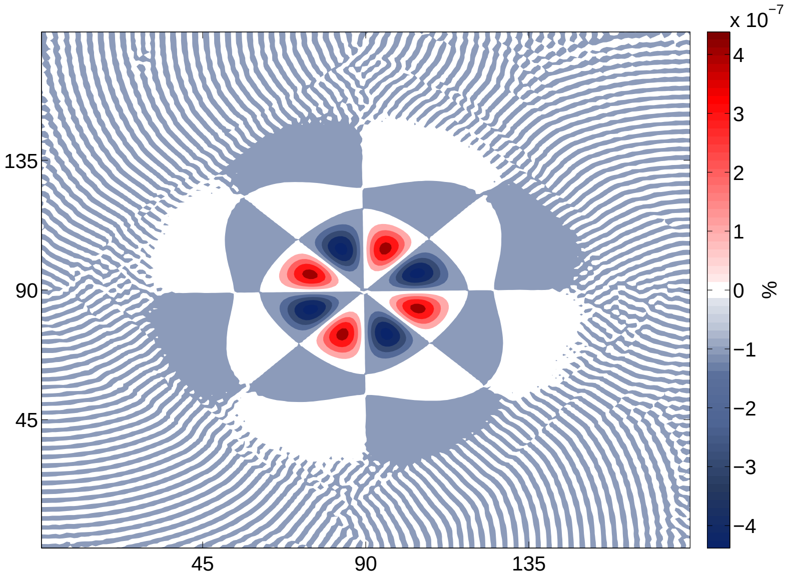


Figure 2.23: The error with the sixth order upwind scheme.

2.2.5 Sensitivity tests

The model contains a set of parameters that if changed will have an impact on the model output. Most of these parameters are configurable, an exception is the ϵ that is used in the bulk formula for the Ekman currents.

The perhaps most interesting of these configurable parameters is the relaxation coefficient. To illustrate some of its effect, and to help develop an idea of which values might actually be reasonable, model runs with an annual atmospheric temperature forcing are repeated for a range of coefficients.

Equivalent runs are also carried out for a range of mixed layer depths and sensible heat exchange coefficients. As for the relaxation coefficient, these tests help build a relationship to these parameters.

In this particular setup the variation of parameters is easily visible in a change of phase and amplitude of the resulting cycle in ocean temperature. The analytical solutions in Appendix A are used to construct plots of the amplitude and time lag as functions of the model parameters.

If nothing else is specified the parameter values are set by default#1 (Table 2.3).

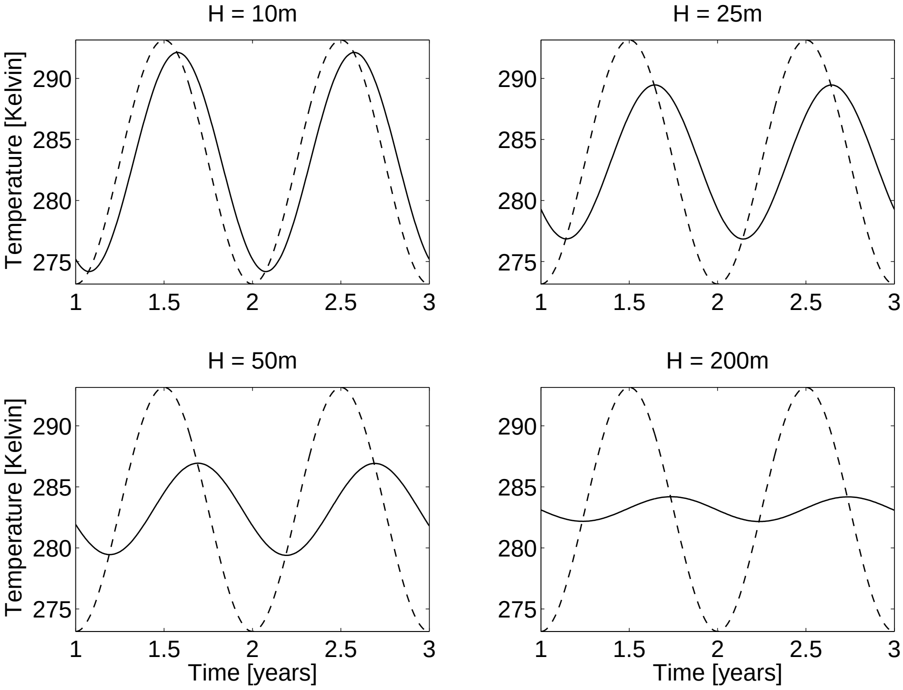


Figure 2.24: The atmospheric temperature forcing (dashed curves), and the resulting ocean mixed layer temperature (solid curves) from the 0.5-layer model for four different mixed layer depths (subplots).

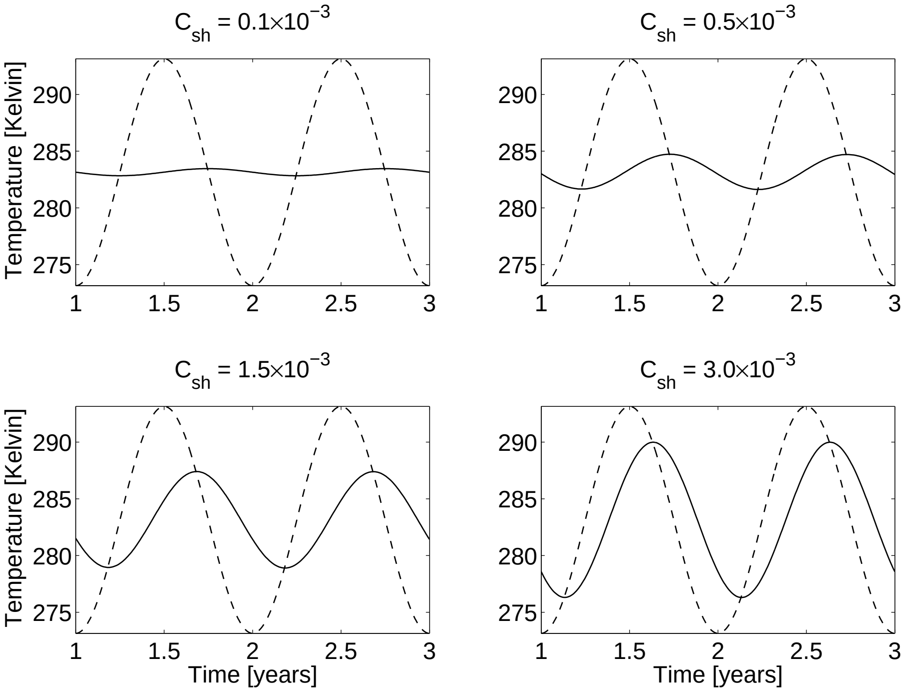


Figure 2.25: The atmospheric temperature forcing (dashed curves), and the resulting ocean mixed layer temperature (solid curves) from the 0.5-layer model for four values of the sensible heat exchange coefficient (subplots).

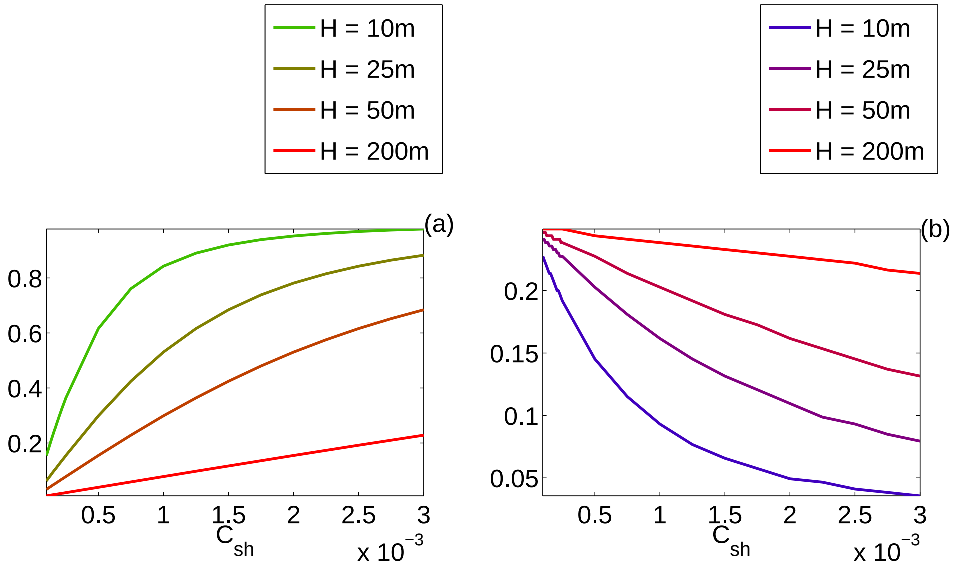


Figure 2.26: (a) The amplitude of the oceanic temperature cycle (normalized with respect to the amplitude of atmospheric temperature), as a function of the sensible heat exchange coefficient for four different mixed layer depths. (b) The time lag between atmosphere and ocean temperature maximums (normalized with respect to the period), as a function of the sensible heat exchange coefficient for four different mixed layer depths.

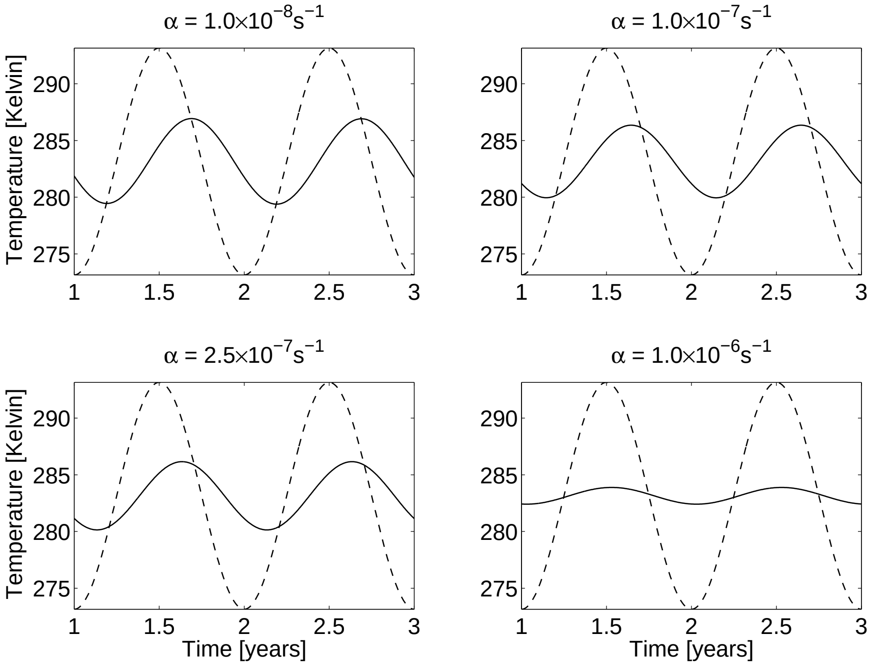


Figure 2.27: The atmospheric temperature forcing (dashed curves), and the resulting ocean mixed layer temperature (solid curves) from the 0.5-layer model for four different values of the relaxation coefficient (subplots).

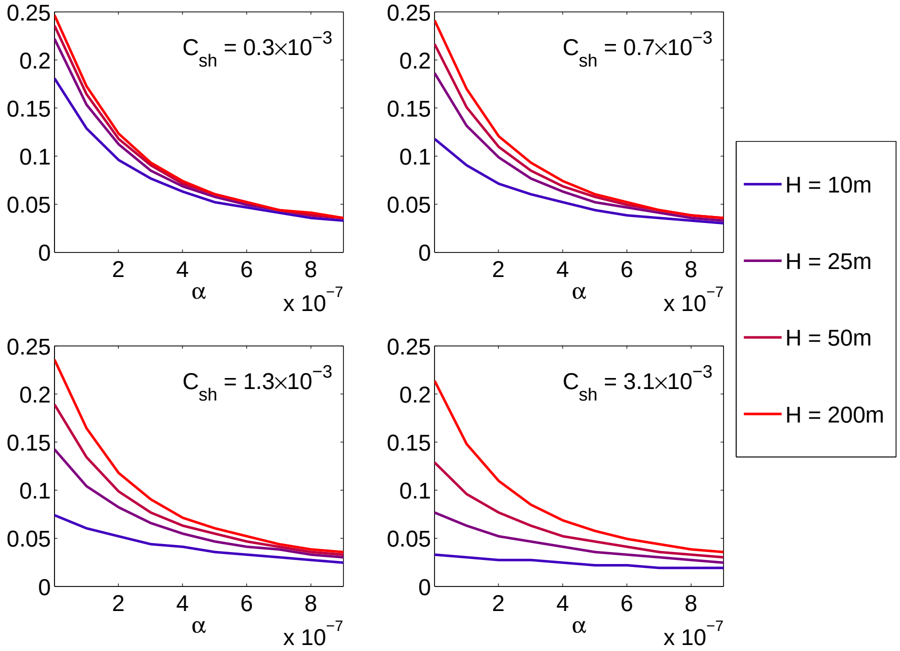


Figure 2.28: (a) The time lag (normalized with respect to the period) between atmosphere and ocean temperature maximums, as a function of the relaxation coefficient for four mixed layer depths (legend), and four exchange coefficients (subplots).

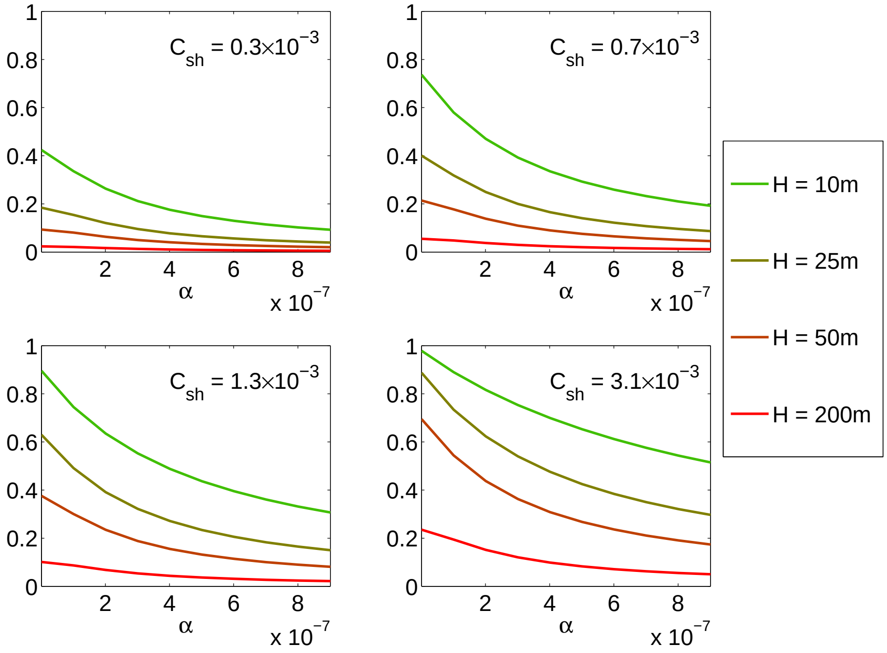


Figure 2.29: (a) The amplitude (normalized with respect to the amplitude of atmospheric temperature) of the oceanic temperature cycle, as a function of the relaxation coefficient for four mixed layer depths (legend), and four exchange coefficients (subplots).

2.3 Discussion

The simple tests in Section 2.2.1 demonstrate that the 0.5-layer model responds as expected to an annual cycle in atmospheric temperature, both with and without relaxation. The magnitude of the error depends on the choice of time step, but even with a time step of five days the accuracy is acceptable. The surface flux alone imposes no particular stability criterion on the model, while the relaxation might do so, depending on the time integration. With an Euler integration at least, the relaxation is conditionally unstable. However, the largest constraint on the time step is related to the transport schemes unless a very large relaxation coefficient is chosen.

The tests in Section 2.2.2 are all performed with the third order upwind scheme (Tremback et al. 1987), the Leap Frog time integration with a Robert-Asselin-Williams (RAW) time filter (Williams 2011), and a Courant number of approximately 0.1. Thus eventual differences in accuracy between these tests are related to the nature of the flow fields.

Appendix B contains a set of additional figures. These are all from tests with a homogeneous flow (2.18) where either the Courant number or the order of the advection scheme, are changed with respect to the default (default#2).

All the implemented upwind schemes have the Courant-Friedrich-Levy stability criterion (Tremback et al. 1987). Because of the staggered Arakawa C-grid this criterion is further reduced to $C_\alpha \leq 0.5$. However, as can be seen in Figure B.10, the solution for a constant advecting velocity blows up badly for $C_\alpha \approx 0.36$ when the model is run using the Leap Frog time integration, and the third order upwind scheme. For this particular combination of transport scheme and time integration method, Figures B.7 - B.10 show a general decrease in the relative error as the Courant number is reduced.

As the order of the upwind scheme increases, so does the overall accuracy (Spensberger 2012), at the cost of increased computational expenses. In the opinion of Tremback et al. (1987) the best balance between efficiency and accuracy might be the sixth order scheme. The homogeneous flow field test was repeated also with a sixth order scheme (Figure B.6), the Courant number being as set by default#2 ($C_\alpha \approx 0.1$). The maximum percentage error in this run, relative to the initial magnitude of the anomaly, was reduced to just above 0.01%, one order of magnitude smaller than for the third order scheme.

The 0.5-layer-, 1-layer- and 1.25-layer models have been tested and are declared functional. The numerical errors in the tests that are presented are in general less than 5% of the magnitude of the original temperature anomaly, or the amplitude of the annual temperature cycle. Other and better tests than those performed here may well exist, and more tests might also have been performed. Nonlinear flow fields could have been prescribed, the method of time integration could have been changed etc. The heat transport is without question the most complex component of the slab ocean model, and therefore also the most extensively tested.

The results from the sensitivity tests are as expected. Figures 2.24, 2.25 and 2.26 show that the sensible heat exchange coefficient, and the mixed layer depth have similar effects on the oceanic temperature cycle, though in opposite proportions. A change in sensible heat exchange coefficient or mixed layer depth either increases the amplitude of the oceanic cycle and decreases the time lag between atmosphere and ocean temperature maximums, or decreases the amplitude and increases the lag.

Mixed layer depths of 10 m and 200 m stand out as relatively extreme. The first value in particular gives an amplitude of the oceanic cycle that is at least 80 % of the atmospheric amplitude for exchange coefficients above 1.0×10^{-3} (Figure 2.26). For a mixed layer depth of 200 m the oceanic amplitude remains below 20 % of the atmospheric for the chosen range of exchange coefficients. While both these mixed layer depths might be applicable in some situations one should at least consider carefully before combining very deep mixed layers with a very weak exchange coefficient, or very shallow mixed layers with a strong exchange coefficient.

The relaxation decreases both the amplitude and the time lag of the resulting cycle in ocean mixed layer temperature. Figure 2.28 shows the lag as a function of relaxation coefficient for four different mixed layer depths and exchange coefficients, while Figure 2.29 shows the amplitude. These plots are constructed with the analytical solutions in Appendix A, but with a reference to Section 2.2.1 they should be representative enough for the model output.

Default values are set for the mixed layer depth, the sensible heat exchange coefficient and the relaxation coefficient, with the results in Section 2.2.5 providing some guidaince in the process.

A somewhat hidden, but still important part of the testing process is the debugging of the source code. Not every day did the runs go as smoothly as one would want, but the coding errors were for the most part I would hope, discovered and elimininated one by one.

Chapter 3

Coupled Test

Gill (1980) presents elegant analytical solutions to a set of problems involving heat induced tropical circulation. It was our intension to perform some experiments along those lines with Bedymo using a two layered atmosphere. However, it became evident along the road, that some more work is required before this is possible. For the moment we are thus limited to a one layer atmosphere.

A test with some similarity to the Gill setup (Gill 1980) is performed with a one layer atmosphere. The model domain is a circumglobal band centered at the equator, and stretching from 63° south to 63° north. The ocean is initialized with a warm anomaly centered at equator, halfway into the eastern part of the domain. The same anomaly is prescribed as the equilibrium temperature, thus the relaxation attempts to maintain this anomaly.

The atmosphere is initialized at a constant temperature, which except from the anomaly is equal to the ocean temperature. A westward cosine shaped basic state atmospheric wind centered at the equator and dying out at 45° north and south is prescribed, mainly to force an Ekman transport in the ocean. The experiment is repeated with all three versions of the slab ccean model.

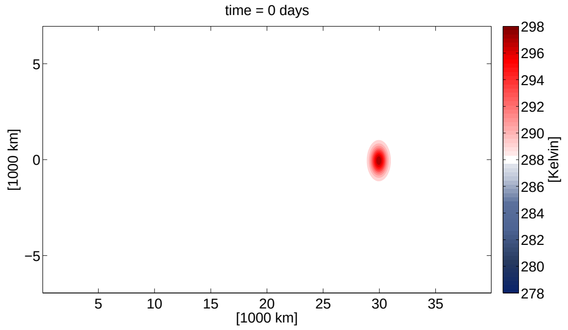


Figure 3.1: The initial ocean mixed layer temperature, and the prescribed equilibrium of the ocean.

Variable	**Explanation**	**Value**
nx	# of gridpoints in x	400
ny	# of gridpoints in y	140
dx	grid distance in x	100 000 m
dy	grid distance in y	100 000 m
dt	time step	300 s
latbrdtyp_ew	east-west boundaries	-1
latbdrtyp_sn	south-north boundaries	0
adv_order_max	order of upwind scheme	4
int_scheme_id	time integration scheme	1
tsstop	end time of model run	730 d
dtout	output time step	1 d
lcoriolis	coriolis force	true
lbetaeffect	beta effect	true
fcor	coriolis parameter	$0\,\mathrm{s}^{-1}$
betacor	d/dy(fcor)	$2.28 \times 10^{-11}\,\mathrm{m}^{-1}\,\mathrm{s}^{-1}$
lslab_ocean	slab ocean	true
l050layermodel	0.5-layer model	false
l100layermodel	1-layer model	true
l125layermodel	1.25-layer model	false
lekmantransp	Ekman transport	true
lrelaxation	relaxation	true
relaxation_coeff	relaxation coefficient	$1.0 \times 10^{-7}\,\mathrm{s}^{-1}$
drag_coeff	drag coefficient	3.0×10^{-3}
sh_exchange_coeff	sensible heat exchange coefficient	1.3×10^{-3}
mixed_layer_depth	mixed layer depth	50 m

Table 3.1: The default namelist for the coupled tests (default#3).

Figure 3.1 shows the warm anomaly used for the initial ocean mixed layer temperature, and the prescribed equilibrium temperature. It has an amplitude of 10 K which causes an initial sensible heat flux $\sim 10\,\mathrm{W\,m^{-2}}$.

Table 3.1 presents the default setup for the coupled tests. Note that a strong drag coefficient is used to enhance the Ekman transport in these tests, while the relaxation coefficient is kept at a moderat value. The maximum basic state wind speed is $5\,\mathrm{m\,s^{-1}}$.

Figure 3.2 shows the ocean mixed layer temperature after 99 days from the different versions of the slab ocean model. The result from the 1-layer model is, at least down to visual accuracy, equal to the result from the 0.5-layer model. In the 0.5-layer and 1-layer models the anomaly is slightly weakened, but apart from that, there is no apparent change in the ocean mixed layer temperature with respect to Figure 3.1.

In the 1.25-layer model a region of upwelling of reservoir water, and intense cooling, forms along the equator. In the region of the prescribed warm anomaly the strength of the upwelling is enough to destroy the anomaly completely in its center, but as the basic state wind speed decreases towards the north and south, the relaxation regains some of its intended dominance to form two separate, and weaker anomalies. The fact that the anomaly intended to be a heat source is to a large extent destroyed by upwelling provides some food for thought.

The prescribed reservoir temperature in the 1.25-layer model in these tests is a constant, and equal to 283.15 K. Thus the reservoir is initially colder than the ocean mixed layer in all grid points, and any eventual upwelling is expected to cool the mixed layer. The temperature difference between the slab layer and the reservoir in this case is too large to be considered especially realistic.

The cooling along the northern and southern boundaries, is not something that was expected, as the basic state wind is set to zero. The reason for this cooling, stands out as something in need of an explanation.

Figure 3.3 shows the perturbation x-wind after 1, 3, and 5 days, while Figure 3.4 shows the perturbation y-wind at the same times. In Figure 3.3 it seems a Kelvin wave might be propagating eastward from the region of the warm anomaly. This particular result seems consistent with the solution by Gill (1980) for a similar heating.

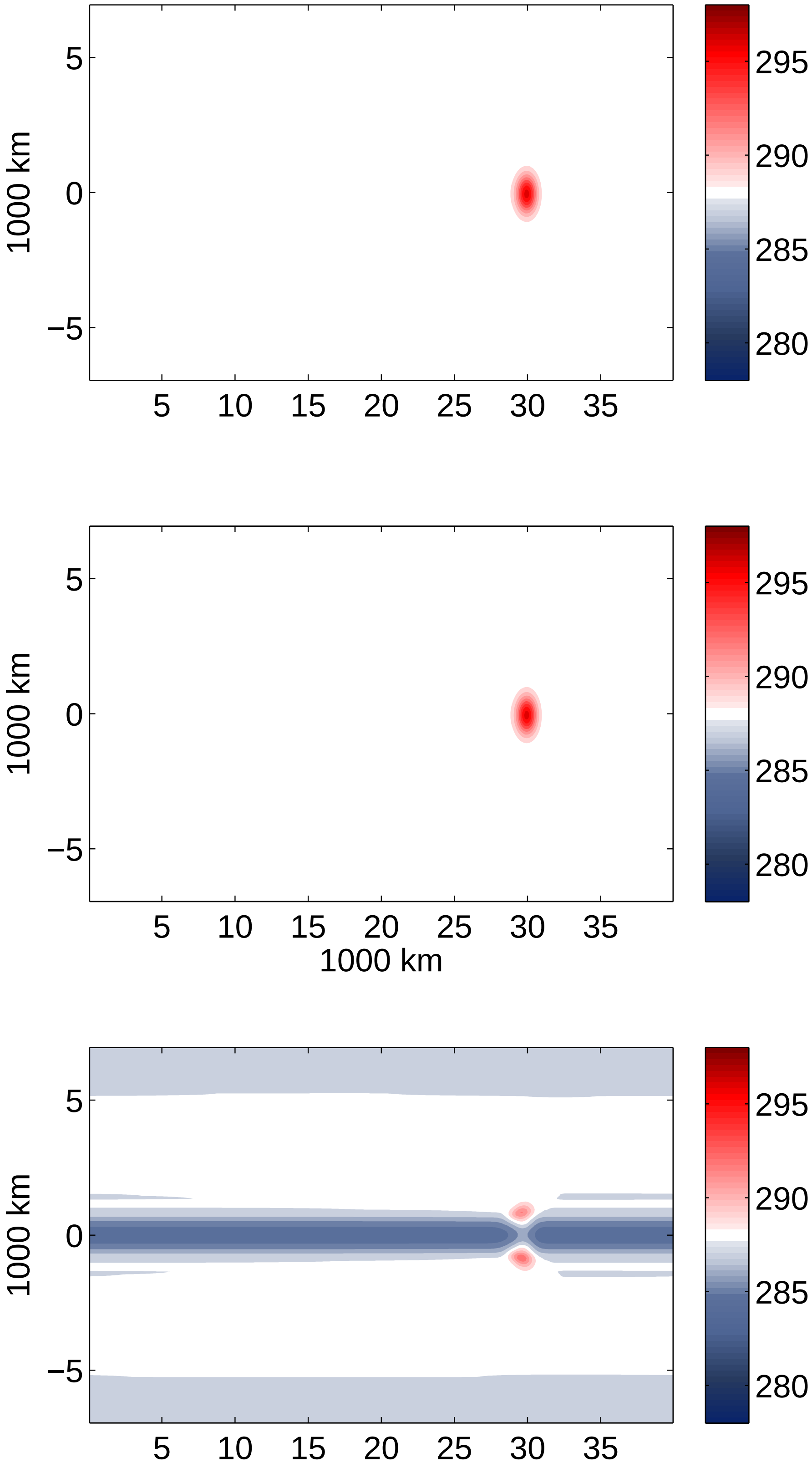


Figure 3.2: The resulting ocean mixed layer temperature [Kelvin] after 99 d from the 0.5-layer model (top), the 1-layer model (middle) and 1.25-layer model (bottom).

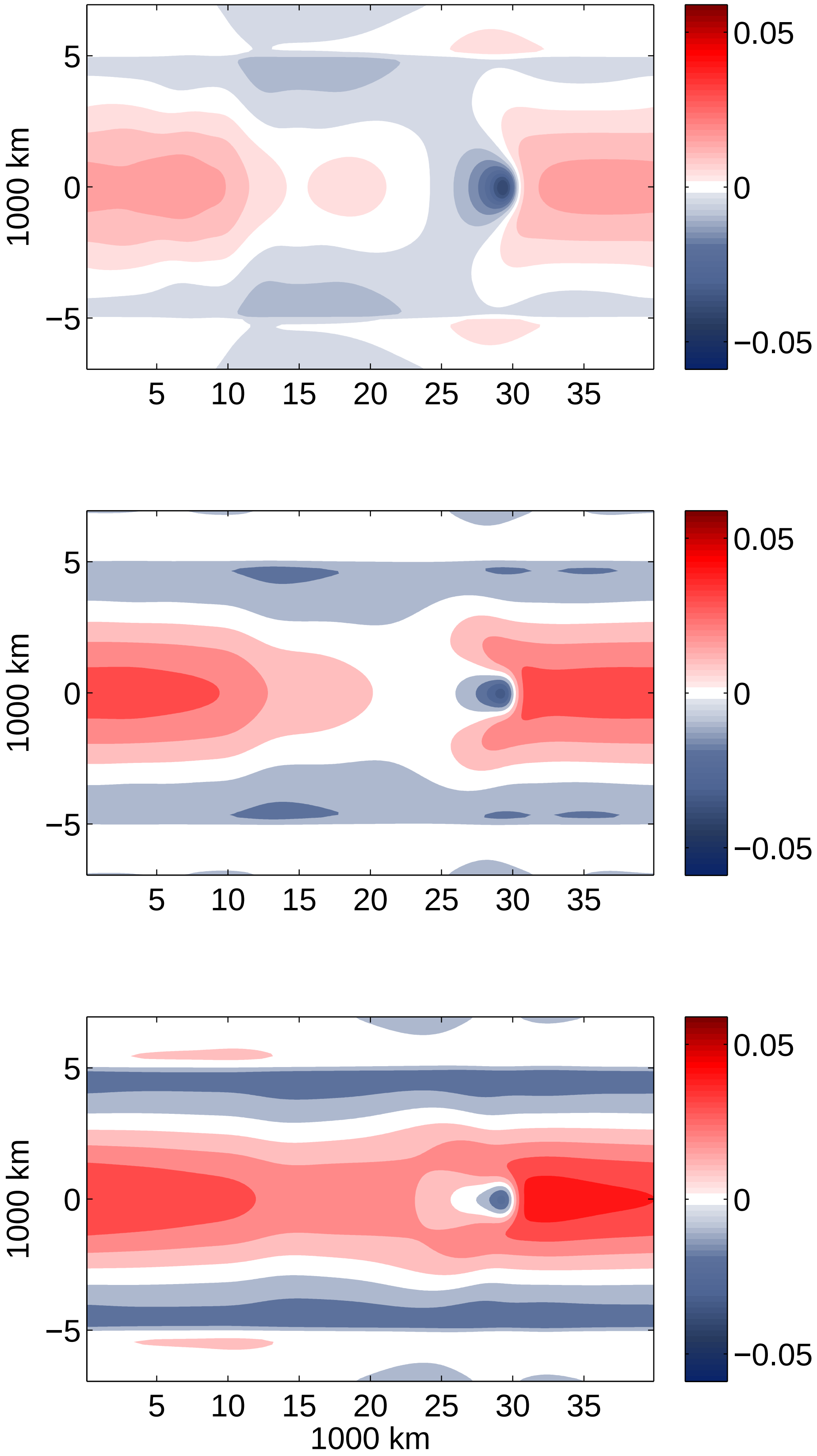


Figure 3.3: The perturbation x-wind [m s^{-1}] after 1 d (top), 3 d (middle) and 5 d (bottom).

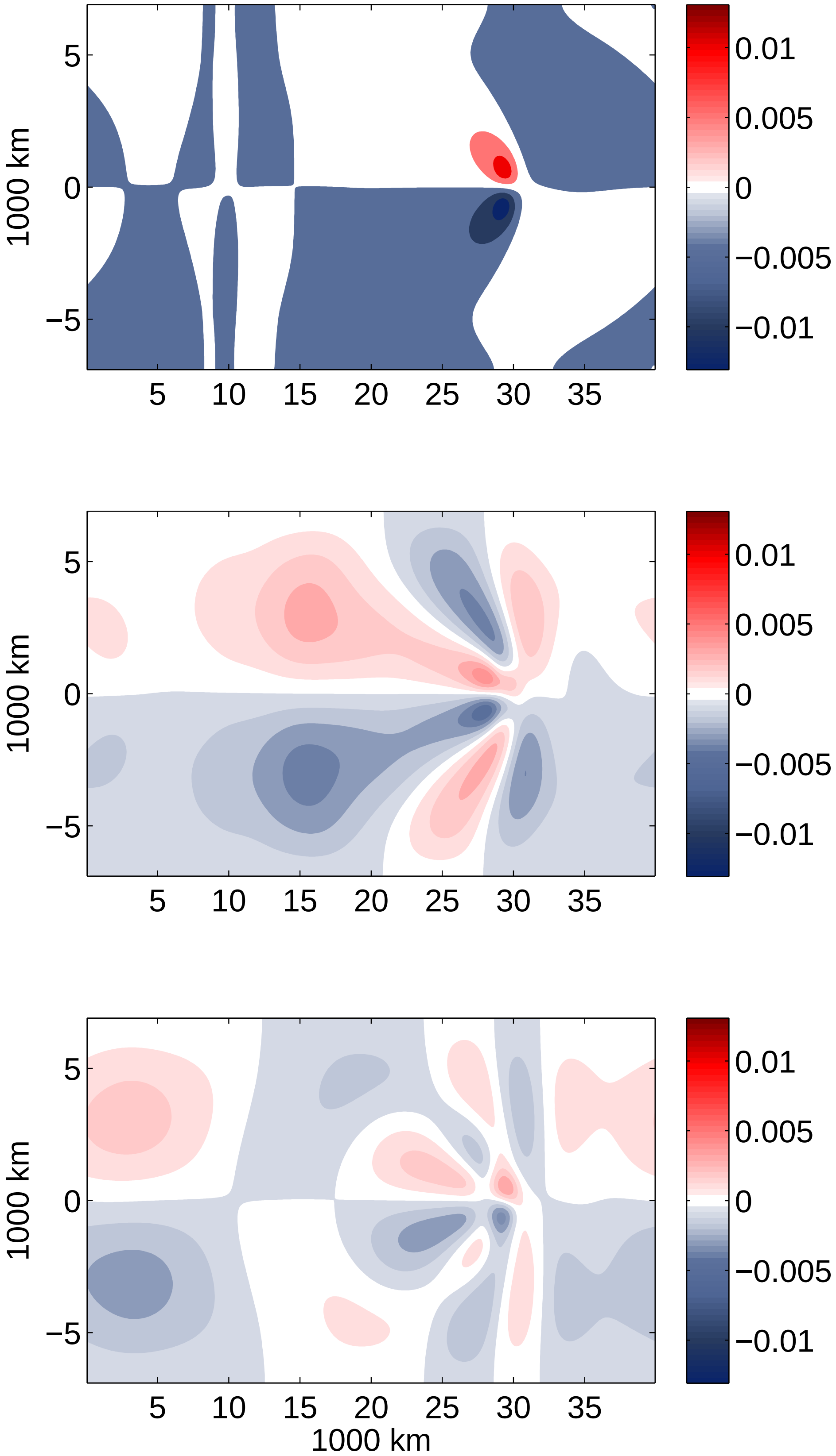


Figure 3.4: The perturbation y-wind [$\mathrm{m\,s^{-1}}$] after 1 d (top), 3 d (middle) and 5 d (bottom).

3.1 Discussion

There is a significant difference in the resulting ocean mixed layer temperature from the run with the 1-layer model, and the run with the 1.25-layer model. The intense divergence and upwelling that is clearly seen for the 1.25-layer model, is invisible, although still present in the 1-layer model, only the upwelled water has the same temperature as the water above. Given that the drag coefficient was assigned a value on the large side, I had expected to observe some trace of the Ekman transport also in the 1-layer model. Apparently the combination of the relaxation, and the cooling by the sensible heat flux that occurs when the anomaly water is transported away from the equator balances the heating tendency of the Ekman currents.

Figure 3.2 (bottom subfigure) shows the result from the run with the 1.25-layer model. This run successfully creates a region of cold upwelling along the equator that bears some resemblance to the real ocean. On the less positive side, no explanation is found at present, for the cooling along the northern and southern boundaries. One way or the other there has to be either upwelling of cold water, or some process that cools the atmosphere which again cools the ocean. There is nothing in the basic state wind suggesting divergence and upwelling in these regions, and it also seems far fetched that the perturbation winds would cause this. There might be reason to have a careful look at the setting of boundary conditions, to check if there is something there that could cause this cooling.

Another issue with 1.25-layer run, is the fact that the relaxation that was meant as the heat source, is dominated by the intense upwelling. A weaker drag coefficient, and a stronger relaxation coefficient, might be enough to solve this problem though. The coupled test with the 1.25-layer model may have benfitted particularily from a larger relaxation coefficient in the region of the warm anomaly, although not necessarily elsewhere, suggesting that the possibility to prescribe a spatially variyng relaxation coefficient may be useful. The relaxation coefficient in the source code is a 2-dimensional array, where every location by default is assigned the constant value from the configuration. Thus to prescribe a spatially varying coefficient, only minor modification is needed in the source code.

It seems that for experiments of this particular kind the 0.5-layer model or the 1.25-layer are the better options, as the 1-layer model apparantly adds no change in the ocean mixed layer temperature relative to the 0.5-layer model, even though the drag coefficient is three times its default value. Without the relaxation though, and the situation might be different.

Chapter 4

Summary and Conclusion

Three model versions are developed and tested. These are the 0.5-layer model, a standard slab ocean model with an optional temperature relaxation. The 1-layer model, an extension of the 0.5-layer model that introduces wind driven or prescribed ocean currents and solves the horizontal advection equation for the slab temperature. And last, but not least, the 1.25-layer model, a further extension of the 1-layer model that assumes an infinite reservoir of prescribed temperature below the slab layer, and solves the three dimensional flux divergence equation for the slab temperature.

The model versions are tested, by comparing the results from idealized simulations with corresponding analytical solutions. The tests result in errors that are considered reasonably small.

A simple sensitivity test where the 0.5-layer model is forced with an annual cycle in atmospheric temperature is repeated for various values of the mixed layer depth, the sensible heat exchange coefficient and the relaxation coefficient, to become familiar with these parameter's effect on model output, and to assign them a set of default values. The chosen defaults are considered moderate values in most cases, though a relaxation coefficient of $1.0 \times 10^{-7}\,\mathrm{s}^{-1}$ may be in the lower part of the range for certain experiments.

The coupling is tested, and does work, for the three models versions and a one layer atmosphere. The ocean responds more or less as expected to the overlying atmosphere, the most interesting part being the Ekman transport, and the significant difference between the 1-layer model, and the 1.25-layer model. The coupled tests suggest that some care must be taken when heat transport and relaxation are used together, to ensure that one component does not supress the expected or intended effect of the other. The prescribed reservoir temperature that was used in the coupled tests is too cold, at least compared to the above mixed layer temperature to be considered realistic. At the same time a high value was used for the drag coefficient, further enhancing the cooling due to upwelling. I want to emphasize that the main objective of these tests is to demonstrate that the ocean works, also when coupled to an atmospheric model. For the coupling of the Slab Ocean Model versions to a multiple layer atmosphere to function at a satsifactory level, it is evident that some more work and testing is required.

Bedymo is provided with an idealized ocean model as was the aim of this thesis to begin with. This opens the possibility for e.g. studies of heat induced tropical circulation, following the work of Gill (1980). How would the upwelling of cold water in the 1.25-layer model alter the results with respect to the 0.5-layer model? Or one can study e.g. the effect of different temperature gradients in the ocean on the storm tracks, and the role or the importance of oceanic heating in the developement and intensification of synoptic systems. The possbilities are many, and the best we can hope, is for some of these experiments to help us take a few of infinitely many small steps toward a better understanding of our complex climate models, and thus the climate system.

Held (2005) writes that no small commitee can decide what are the appropriate models and model hierarchies. Rather each and every model must prove itself over time. I believe that Bedymo is up for the challenge.

Appendix A

Analytical solutions

A.1 The 0.5-layer model forced with an annual cycle in atmospheric temperature

$$T_a(t) = T_0 + A \sin\left(\frac{2\pi t}{P} + \phi_0\right) \tag{A.1}$$

The 0.5-layer model is forced with an atmospheric temperature on the form of (A.1), where A is the amplitude, P the period, ϕ_0 the phase and T_0 some constant temperature. When the relaxation coefficient is set to zero, the prognostic equation for ocean mixed layer temperature T can be written

$$\frac{\partial T}{\partial t} + CT = C\left(T_0 + A \sin\left(\frac{2\pi t}{P} + \phi_0\right)\right)$$

where

$$C = \frac{\rho_a C_{pa} C_{sh} |\vec{u_a}|}{\rho_o H C_{po}}$$

As seen C contains several parameters and variables, all of which are kept constant in the relevant model runs. An explanation of these variables is given in Section 2.1.3.

The constants T_0 and ϕ_0 are ignored for now as they can simply be added to the solution that is found when they are both zero.

$$\frac{\partial T}{\partial t} + CT = C\left(A \sin\left(\frac{2\pi t}{P} + \phi\right)\right)$$

Assume a solution on the form of an annual cycle with a change in phase and amplitude relative to the atmospheric temperature.

$$T(t) = A_* \sin\left(\frac{2\pi t}{P} + \phi\right) \tag{A.2}$$

Taking the partial derivative with respect to t of the above solution yields

$$\frac{\partial T}{\partial t} = \frac{2\pi}{P} A_* \cos\left(\frac{2\pi t}{P} + \phi\right)$$

The prognostic equation for ocean mixed layer temperature can thus be written

$$\frac{2\pi}{P} A_* \cos\left(\frac{2\pi t}{P} + \phi\right) + C A_* \sin\left(\frac{2\pi t}{P} + \phi\right) = C A \sin\left(\frac{2\pi t}{P}\right) \tag{A.3}$$

Apply this equation at $t = 0$ to obtain an expression for ϕ.

$$\frac{2\pi}{P} A_* \cos(\phi) + C A_* \sin(\phi) = 0$$

$$\frac{2\pi}{P} \cos(\phi) = -C \sin\phi$$

$$\tan(\phi) = -\frac{2\pi}{CP}$$

$$\phi = \tan^{-1}\left(-\frac{2\pi}{CP}\right)$$

Then apply (A.3) at a time t^* where $\frac{\partial T}{\partial t} = 0$ to find an expression for A_*.

$$\frac{\partial T}{\partial t} = 0 \Rightarrow \cos\left(\frac{2\pi t^*}{P} + \phi\right) = 0 \Rightarrow \frac{2\pi t^*}{P} + \phi = \frac{\pi}{2}$$

$$t^* = (\frac{\pi}{2} - \phi)\frac{P}{2\pi}$$

$$A_* = A|\sin\left(\frac{\pi}{2} - \phi\right)| = A|\cos\phi|$$

Add T_0 and ϕ_0 to Equation (A.2) and the solution becomes

$$T(t) = T_0 + A_* \sin(\frac{2\pi t}{P} + \phi_0 + \phi)$$

Because of the manner in which this solution is obtained I will give a proof that it fullfils Equation (A.1) for all T_0 and ϕ_0. The proof is carried out by substituting for T and T_a in (A.1) and then using the following trigonometric identities.

$$\sin(a \pm b) = \sin a \cos b \pm \cos a \sin b \tag{A.4a}$$

$$\cos(a \pm b) = \cos a \cos b \mp \sin a \sin b \tag{A.4b}$$

$$\cos a = \pm\frac{1}{\sqrt{1 + \tan^2 a}} \tag{A.4c}$$

$$\sin a = \pm\frac{\tan a}{\sqrt{1 + \tan^2 a}} \tag{A.4d}$$

$$\tan(\tan^{-1} a) = a \tag{A.4e}$$

$$
\begin{aligned}
&\frac{2\pi A_*}{P}\cos\left(\frac{2\pi t}{P}+\phi_0+\phi\right)+C\left(\overline{T_a}+A_*\sin\left(\frac{2\pi t}{P}+\phi_0+\phi\right)\right)\\
&-C\left(\overline{T_a}+A\sin\left(\frac{2\pi t}{P}+\phi\right)\right)=0
\end{aligned}
$$

$$\underbrace{\text{(A.4a) and (A.4b)}}_{\Downarrow}$$

$$
\begin{aligned}
&\frac{2\pi A_*}{P}\left(\cos\left(\frac{2\pi t}{P}+\phi_0\right)\cos\phi-\sin\left(\frac{2\pi t}{P}+\phi_0\right)\sin\phi\right)\\
&+CA_*\left(\sin\left(\frac{2\pi t}{P}+\phi_0\right)\cos\phi+\cos\left(\frac{2\pi t}{P}+\phi_0\right)\sin\phi\right)\\
&-CA\sin\left(\frac{2\pi t}{P}+\phi_0\right)=0
\end{aligned}
$$

$$
\begin{aligned}
&A_*\cos\phi\left(\frac{2\pi}{P}\cos\left(\frac{2\pi t}{P}+\phi_0\right)+C\sin\left(\frac{2\pi t}{P}+\phi_0\right)\right)\\
&+A_*\sin\phi\left(C\cos\left(\frac{2\pi t}{P}+\phi_0\right)-\frac{2\pi}{P}\sin\left(\frac{2\pi t}{P}+\phi_0\right)\right)\\
&-CA\sin\left(\frac{2\pi t}{P}+\phi_0\right)=0
\end{aligned}
$$

$$\underbrace{\text{(A.1)}}_{\Downarrow}$$

$$
\begin{aligned}
&\cos^2\phi\left(\frac{2\pi}{P}\cos\left(\frac{2\pi t}{P}+\phi_0\right)+C\sin\left(\frac{2\pi t}{P}+\phi_0\right)\right)\\
&+\cos\phi\sin\phi\left(C\cos\left(\frac{2\pi t}{P}+\phi_0\right)-\frac{2\pi}{P}\sin\left(\frac{2\pi t}{P}+\phi_0\right)\right)\\
&-C\sin\left(\frac{2\pi t}{P}+\phi_0\right)=0
\end{aligned}
$$

$$\underbrace{\text{(A.4c), (A.4d), (A.4e) and (A.1)}}_{\Downarrow}$$

$$
\begin{aligned}
&\frac{1}{1+(\frac{2\pi}{CP})^2}\left(\frac{2\pi}{P}\cos\left(\frac{2\pi t}{P}+\phi_0\right)+C\sin\left(\frac{2\pi t}{P}+\phi_0\right)\right)\\
&-\frac{\frac{2\pi}{CP}}{1+(\frac{2\pi}{CP})^2}\left(C\cos\left(\frac{2\pi t}{P}+\phi_0\right)-\frac{2\pi}{P}\sin\left(\frac{2\pi t}{P}+\phi_0\right)\right)\\
&-C\sin\left(\frac{2\pi t}{P}+\phi_0\right)=0
\end{aligned}
$$

$$\frac{1}{1+(\frac{2\pi}{CP})^2}\left(C(1+(\frac{2\pi}{CP})^2)\sin\left(\frac{2\pi t}{P}+\phi_0\right)+(\frac{2\pi}{P}-\frac{2\pi}{P})\cos\left(\frac{2\pi t}{P}+\phi_0\right)\right)$$
$$-C\sin\left(\frac{2\pi t}{P}+\phi_0\right)=0$$

$$C\sin\left(\frac{2\pi t}{P}+\phi_0\right)-C\sin\left(\frac{2\pi t}{P}+\phi_0\right)=0$$

With a nonzero relaxation coefficient the prognostic temperature equation becomes

$$\frac{\partial T}{\partial t}+(C+\alpha)T=CT_a+\alpha T_E$$

where T_E must be equal to T_0 to be consistent with the model runs in question. The solution is similar, but with slighty different expressions for ϕ and A_*.

$$\phi=\tan^{-1}(-\frac{2\pi}{(C+\alpha)P})$$

$$A_*=\frac{CA}{C+\alpha}|\cos(\phi)|$$

$$T(t)=T_0+A_*\sin(\frac{2\pi t}{P}+\phi_0+\phi)$$

A.2 Heat transport by prescribed idealized flow fields in the 1-layer- and 1.25-layer model

Analytical solutions to the transport tests that were performed with the 1-layer- and 1.25-layer model (Section 2.2.2) are presented here. They are all solutions to an equation on the form of (A.5), with an initial condition on the form of (A.6). For simplicity all the test runs were performed with $v(x,y,t)=0$, therefore $v(x,y,t)=0$ throughout this section.

$$\frac{\partial T}{\partial t}+\frac{\partial}{\partial x}(uT)+\frac{\partial}{\partial z}(wT)=0 \tag{A.5}$$

$$T(x,y,0)=f(x,y) \tag{A.6}$$

For the 1-layer model (and the 1.25-layer model when the flow field is not divergent) (A.5) reduces to a two dimensional advection equation (Equation A.7).

$$\frac{\partial T}{\partial t}+u(x,y,t)\frac{\partial T}{\partial x}=0 \tag{A.7}$$

As long as the velocity u does not depend on x and/or t analytical solutions are easily found, but when u does depend on these variables the challenge is increased. In this thesis I have relied on the method of characteristics as desribed in e.g. Farlow (1993). The method of characteristics will be explained in Section A.2.3.

A.2.1 Homogeneous flow

When the flow velocity is independent of both time and space the governing equation is an advection equation with constant advecting velocity (Equation A.8). The initial condition is a two dimensional Gaussian distribution (Equation A.9).

$$\frac{\partial T}{\partial t} + u_0 \frac{\partial T}{\partial x} = 0 \tag{A.8}$$

$$T(x, y, 0) = f(x, y) = T_0 + A \exp{[-(\frac{(x - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2})]} \tag{A.9}$$

The general solution to such a problem for any initial distribution $f(x, y)$ is

$$T(x, y, t) = f(x - u_0 t, y) \tag{A.10}$$

This is easily verified by taking the partial derivatives with respect to t and x of Equation (A.10) and then substituting for $\partial T/\partial t$ and $\partial T/\partial x$ in (A.8). Combining (A.9) and (A.10) gives us the solution to this particular problem.

$$T(x, y, t) = T_0 + A \exp{[-(\frac{(x - u_0 t - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2})]} \tag{A.11}$$

A.2.2 Shear flow

Here the flow velocity is a function of y, the direction normal to the flow direction. The initial condition remains the same (Equation A.9).

$$\frac{\partial T}{\partial t} + u(y) \frac{\partial T}{\partial x} = 0 \tag{A.12}$$

$$T(x, y, 0) = f(x, y) = T_0 + A \exp{[-(\frac{(x - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2})]} \tag{A.13}$$

The general solution (Equation A.14) is equivalent to that for a homogeneous flow field and can be verified in the same manner. The particular solution (Equation A.15) is again obtained by combining the general solution with the initial condition.

$$T(x, y, t) = f(x - u(y)t, y) \tag{A.14}$$

$$T(x, y, t) = T_0 + A \exp{[-(\frac{(x - u(y)t - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2})]} \tag{A.15}$$

A.2.3 The method of characteristics

When the flow field is divergent or time dependent obtaining analytical solutions is not as straight forward. Consider the problem posed by the following first order partial differential equation (PDE) and initial condition

$$a(x,t)\frac{\partial T}{\partial x} + b(x,t)\frac{\partial T}{\partial t} + c(x,t)T = 0 \tag{A.16}$$

$$T(x,0) = f(x) \tag{A.17}$$

where $-\infty < x < \infty$ and $0 \leq t < \infty$.

If $a = 1$, b is a constant and $c = 0$ (A.16) reduces to a simple advection equation with the solution $T(x,t) = f(x - bt)$. In this case the initial profile propagates along the x-axis while conserving its shape.

Solutions to (A.16) are in general based on the fact that the initial disturbance at a point x propagates along a line in the tx-plane, called a characteristic or a characteristic curve (Farlow 1993). The method of characteristics introduces two new coordinates s and τ that have the following properties.

- s changes along the characteristic curves.
- τ changes along the initial curve ($t = 0$).

In this way one can think of s as the new x and τ as the new t. The characteristic curves are chosen so that

$$\frac{dx}{ds} = a(x,t) \tag{A.18}$$

$$\frac{dt}{ds} = b(x,t) \tag{A.19}$$

Equations (A.18) and (A.19) are called the characteristic equations. As a result Equation (A.16) can be rewritten as

$$\frac{dT}{ds} + c(x(s,\tau), t(s,\tau))T = 0 \tag{A.20}$$

making the PDE an ordinary differential equation (ODE) along the curves $[x(s), t(s)] : 0 < s < \infty$. We now need to solve these three ODE's with the following initial conditions

- $t(s = 0) = 0$
- $x(s = 0) = \tau$
- $T(s = 0) = f(\tau)$

The solutions to the characteristic equations are solved for s and τ in terms of x and t, and then the last step is to substitute the resulting expressions into the solution of (A.20).

A.2.4 Convergent flow

In this case the flow velocity is a linear function of x, the velocity decreasing as x increases $(0 \leq x \leq L)$. From here on the method of characteristics (Section A.2.3) is applied.

For the homogeneous and shear flow we found general solutions for $u = u_0$ and $u = u(y)$. As the nature of the characteristic equations depends very much on the flow field itself, this is not attempted for the remaining cases.

The initial condition is as before, but now the governing equation is

$$\frac{\partial T}{\partial t} + u_0(1 - \frac{x}{L})\frac{\partial T}{\partial x} = 0 \tag{A.21}$$

Equation (A.21) gives the characteristic equations

$$\frac{dt}{ds} = 1 \tag{A.22}$$

$$\frac{dx}{ds} = u_0(1 - \frac{x}{L}) \tag{A.23}$$

The solutions to these equations when the appropriate initial conditions ($t(0) = 0$ and $x(0) = \tau$) are applied are as follows.

$$t(s) = s + c_1 = s \tag{A.24}$$

$$x(s) = L + c_2 \exp{(-\frac{u_0 s}{L})} = L + (\tau - L) \exp{(-\frac{u_0 s}{L})} \tag{A.25}$$

There is one ODE left to solve.

$$\frac{dT}{ds} = 0 \tag{A.26}$$

Apply the appropriate initial condition ($T(s = 0) = f(\tau)$) and obtain an expression for T as a function of s and τ.

$$T(s, \tau) = f(\tau) \tag{A.27}$$

Solve (A.24) and (A.25) for s and τ in terms of x and t to obtain the coordinate transformations. Equation (A.27) may then be rewritten as follows.

$$T(x, y, t) = f((x - L) \exp{(\frac{u_0 t}{L})} + L, y) \tag{A.28}$$

At last combining (A.9) and (A.28) yields the complete solution to this specific problem.

$$T(x, y, t) = T_0 + A \exp -[(\frac{((x - L) \exp{(\frac{u_0 t}{L})} + L - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2})] \tag{A.29}$$

A.2.5 Divergent flow, 1-layer model

The velocity is still a linear function of x, but now increasing as x increases ($0 \leq x \leq L$). The initial condition is not changed.

$$\frac{\partial T}{\partial t} + u_0 \frac{x}{L} \frac{\partial T}{\partial x} = 0 \tag{A.30}$$

$$T(x, y, 0) = f(x, y) = T_0 + A \exp\left[-\left(\frac{(x - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2}\right)\right] \tag{A.31}$$

¿From Equation (A.30) we find the characteristic equations (A.32 and A.33).

$$\frac{dt}{ds} = 1 \tag{A.32}$$

$$\frac{dx}{ds} = u_0 \frac{x}{L} \tag{A.33}$$

The solutions to the characteric equations are

$$t(s) = s + c_1 = s \tag{A.34}$$

$$x(s) = C_2 \exp\left(\frac{u_0 s}{L}\right) = \tau \exp\left(\frac{u_0 s}{L}\right) \tag{A.35}$$

There is one more ODE to solve, that is

$$\frac{dT}{ds} = 0 \tag{A.36}$$

Applying the initial condition, $T(s = 0) = f(\tau)$, yields the following expression for T as a function of s and τ.

$$T(s, \tau) = f(\tau) \tag{A.37}$$

It is time to solve the characteristic equations for s and τ in terms of x and t. Then we can write

$$T(x, y, t) = f\left(x \exp\left(-\frac{u_0 t}{L}\right)\right) \tag{A.38}$$

Again we need to combine (A.38) with the initial temperature profile (A.9) to obtain the specific solution to this particular problem.

$$T(x, y, t) = T_0 + A \exp\left[-\left(\frac{(x \exp(-\frac{u_0 t}{L}) - x_0)^2}{2{s_x}^2} + \frac{(y - y_0)^2}{2{s_y}^2}\right)\right] \tag{A.39}$$

A.2.6 Divergent flow, 1.25-layer model

In this case the flow is the same field as in Section A.2.5, the initial condition is still given by (A.9). The governing equation is

$$\frac{\partial T}{\partial t} + u_0 \frac{x}{L}\frac{\partial T}{\partial x} + \frac{u_0}{L}T = \frac{u_0}{L}T_R \tag{A.40}$$

Although the governing equation has changed the characteristic equations are the exact same as in Section A.2.5, and hence also their solutions.

$$t(s) = s \tag{A.41}$$

$$x(s) = \tau \exp \frac{u_0 s}{L} \tag{A.42}$$

The difference lies in the ODE (Equation A.43) resulting from the coordinate transformation.

$$\frac{dT}{ds} + \frac{u_0}{L}T_1 = \frac{u_0}{L}T_R \tag{A.43}$$

This equation can be solved using the method of integrating factors (Boyce & DiPrima 2005) where the integrating factor in this case is $\mu(s) = \exp \frac{u_0 s}{L}$.

$$\int \frac{d}{ds}(\exp \frac{u_0 s}{L} T) ds = \frac{u_0 s}{L} T_R \int \exp \frac{u_0 s}{L} ds$$

$$\exp \frac{u_0 s}{L} T = T_R \exp \frac{u_0 s}{L} + C_1$$

$$T(s) = T_R + C_1 \exp(-\frac{u_0 s}{L})$$

The integration constant C_1 is determined by applying the initial condition ($T(0) = f(\tau)$).

$$T(s,\tau) = T_R + (f(\tau) - T_R)\exp(-\frac{u_0 s}{L}) \tag{A.44}$$

Combining (A.44) and (A.9) gives the solution

$$T(x,y,t) = T_R + \left(T_0 + A\exp[-(\frac{(x\exp(-\frac{u_0 t}{L}) - x_0)^2}{2{s_x}^2} + \frac{(y-y_0)^2}{2{s_y}^2})] - T_R\right)\exp(-\frac{u_0 s}{L})$$

Appendix B

Additional figures for Section 2.2.2

All figures presented here, with the exception of Figures B.1 and B.2, are from repetitions of the transport test with a homogeneous flow field. The order of the advection scheme, or the Courant number is varied with respect to the default setup (default#2).

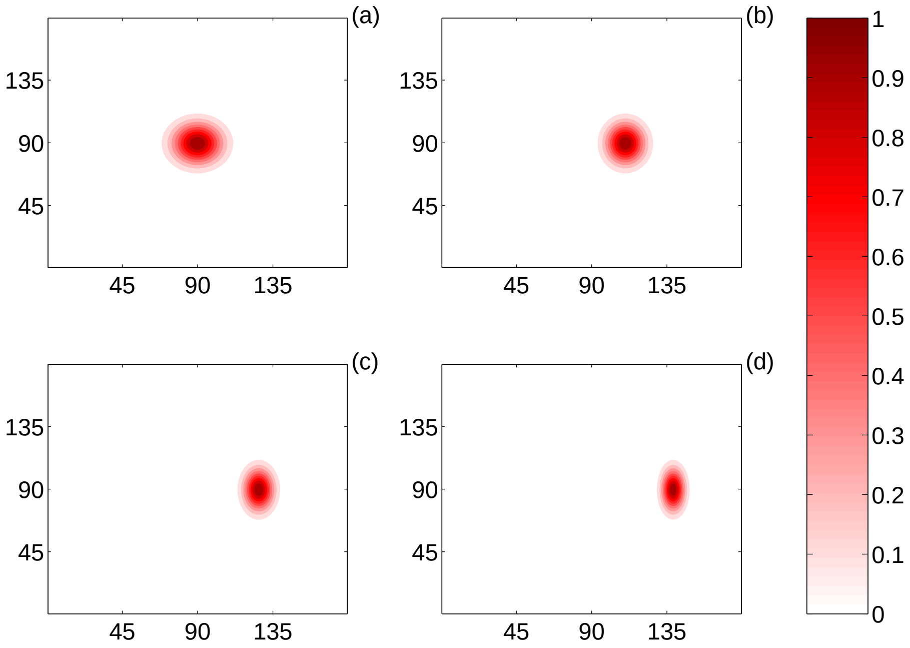


Figure B.1: Normalized ocean mixed layer temperature from the 1.25-layer model when the prescribed flow field is given by Equation (2.20) at (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

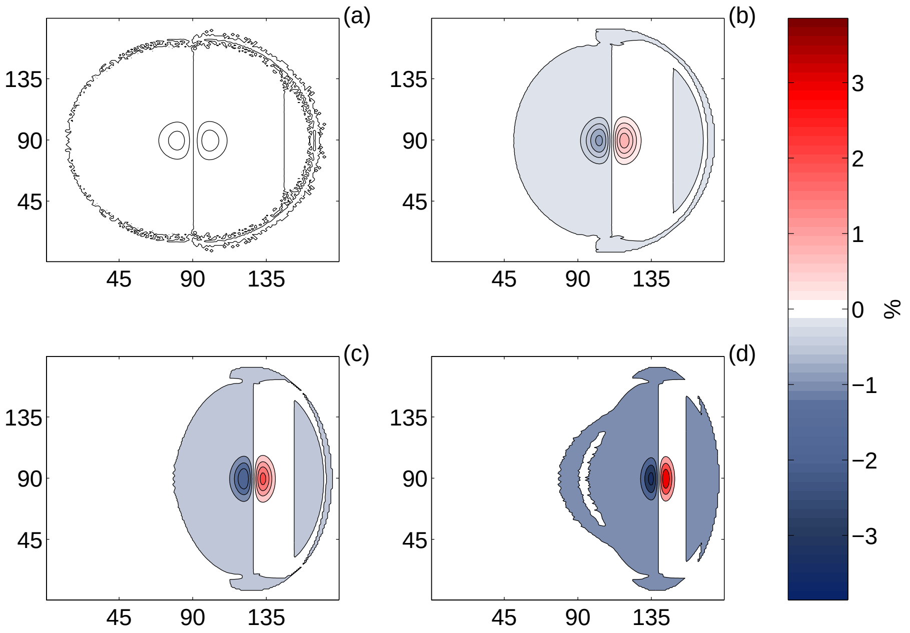


Figure B.2: The error in the numerical solution with respect to the analytical, measured in % of the magnitude of the initial anomaly, after (a) t=20 d, (b) t=1200 d, (c) t=2440 d and (d) t=3640 d.

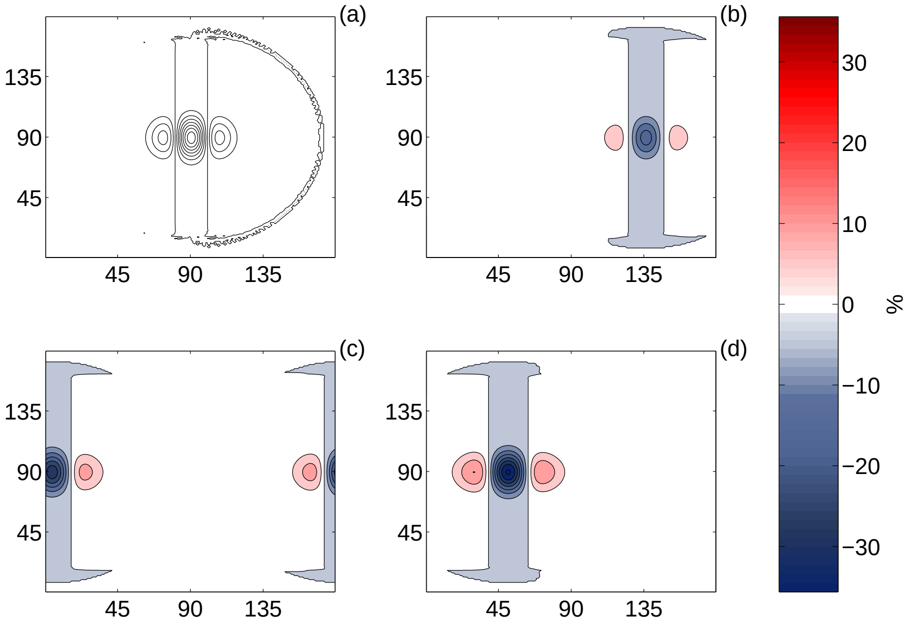


Figure B.3: The transport test with a homogeneous flow repeated for the first order upwind scheme.

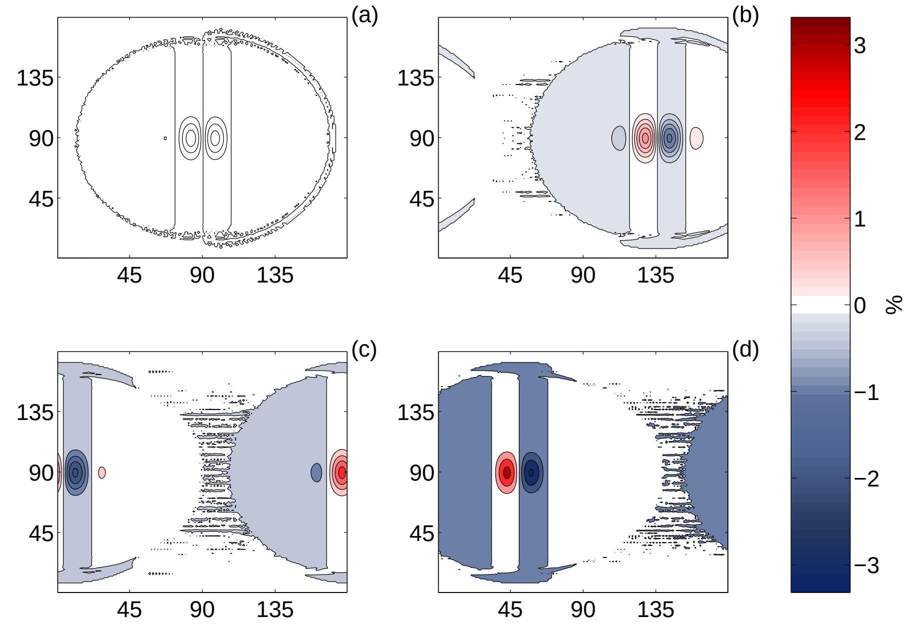


Figure B.4: The transport test with a homogeneous flow repeated for the second order upwind scheme.

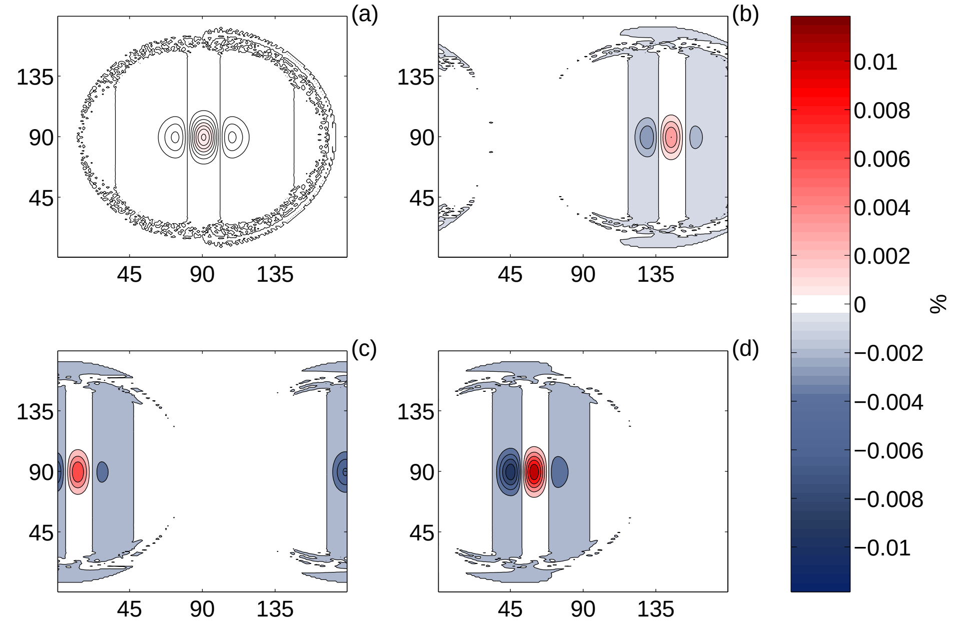


Figure B.5: The transport test with a homogeneous flow repeated for the fifth order upwind scheme.

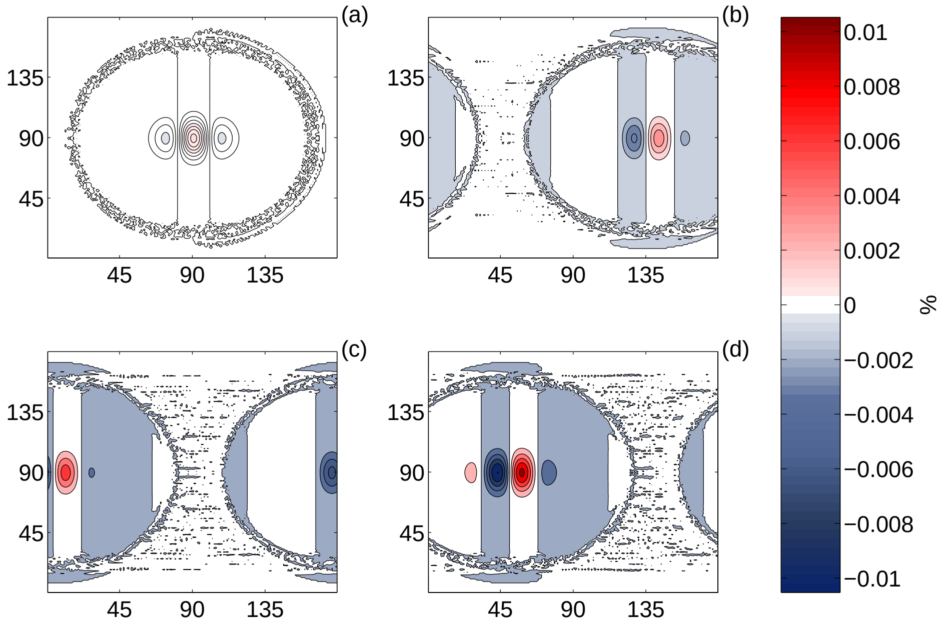


Figure B.6: The transport test with a homogeneous flow repeated for the sixth order upwind scheme.

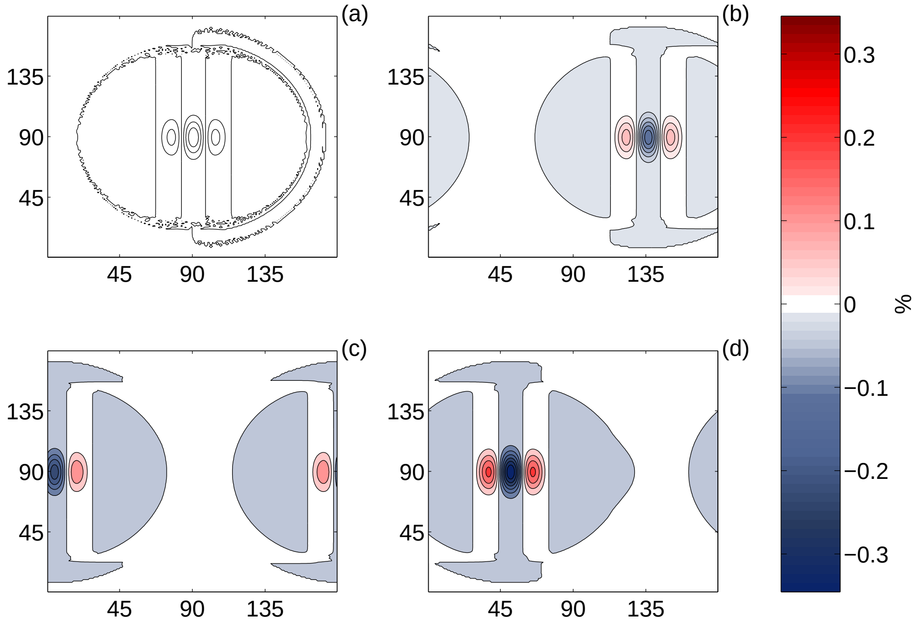


Figure B.7: The transport test with a homogeneous flow repeated with a Courant number of 0.0049.

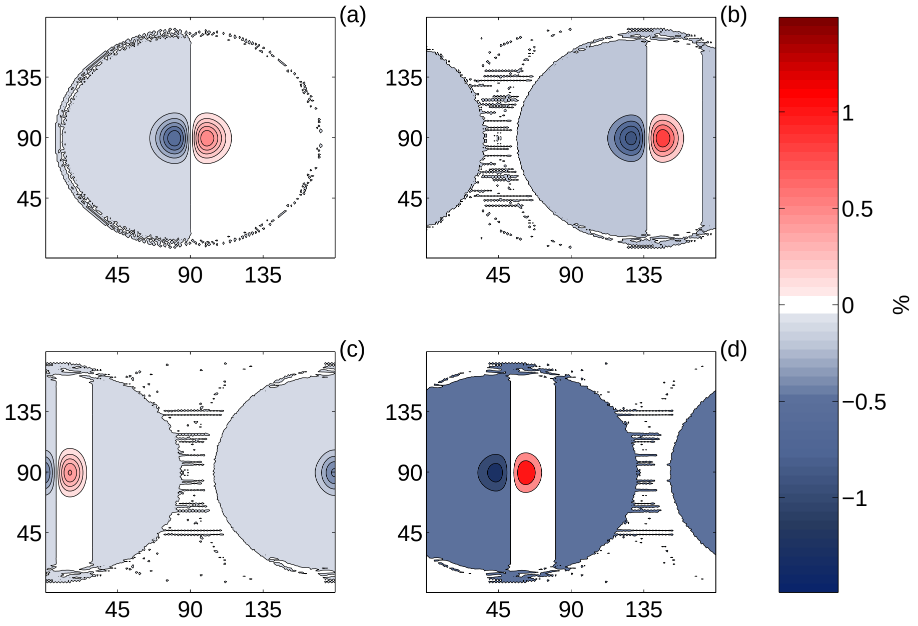


Figure B.8: The transport test with a homogeneous flow repeated with a Courant number of 0.22.

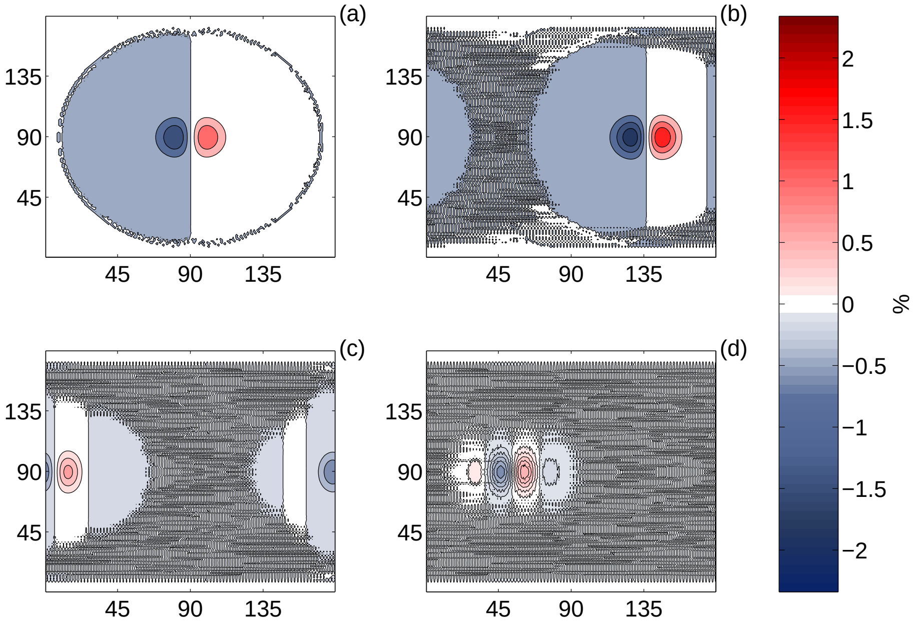


Figure B.9: The transport test with a homogeneous flow repeated with a Courant number of 0.34.

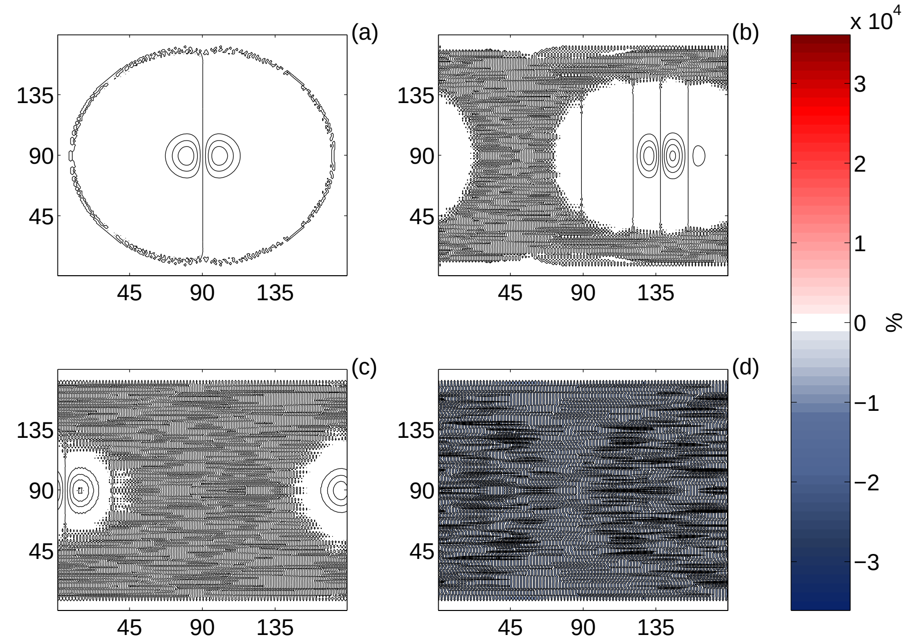


Figure B.10: The transport test with a homogeneous flow repeated with a Courant number of 0.36.

Appendix C

The slab ocean module of the Bedymo source code

```fortran
module ocean_slab
  use kind
  use consts, only: pi, cp_atm => cp, cp_ocn
  use slab_config, only:
  use grid
  use ncout, only: register_var_yxt
  use derivatives
  use bdr
  use timeint
  use coupler, only: reg_coup_var_yx, reg_coup_var_yxt, req_coup_var_2d, &
                              & check_coup_var_available
  !
  implicit none
  !
  ! Variables
  real(kind=nr), allocatable :: rho_ocn(:,:), rho_atm(:,:), Hm(:,:), Hr(:,:),                     &
                             &  Cbl(:,:), Crl(:,:), Cdr(:,:), Csh(:,:),                           &
                             &  tempdiff(:,:), Te(:,:), Ta(:,:), F(:,:),                          &
                             &  ue(:,:), ve(:,:), ub(:,:), vb(:,:),                               &
                             &  u_ocn(:,:), v_ocn(:,:), w_ocn(:,:), windspeed(:,:)
                             !
  real(kind=nr), allocatable :: Tm(:,:,:,:), Tm_tend(:,:,:), Tm_tend_adv(:,:), Tm_tend_rlx(:,:),  &
                             &  Tm_tend_hfl(:,:), Tm_flxx_adv(:,:), Tm_flxy_adv(:,:),             &
                             &  Tr(:,:,:), Tr_tend_adv(:,:), Tr_flxx_adv(:,:), Tr_flxy_adv(:,:)
                             !
  real(kind=nr), allocatable :: utest(:,:,:), vtest(:,:,:)
  real(kind=nr)              :: eps
  integer(kind=ni)           :: tln, tli, tlo, tls
                             !
  logical                    :: l050layermodel, l100layermodel, l125layermodel, lekm, lrlx
  integer(kind=ni)           :: cvarid_atm_xwind, cvarid_atm_ywind, cvarid_atm_temp
  !-------------------------------------------------------------------------------------------------
  !
```

```
contains
  !
  ! Initialization
  subroutine init_ocean_slab()
    use slab_config, only: l050layermodel_c => l050layermodel, l100layermodel_c => l100layermodel,
        &    l125layermodel_c => l125layermodel, lekmantransp, lrelaxation,
        &    relaxation_coeff, drag_coeff, sh_exchange_coeff, mixed_layer_depth
        !
    allocate(rho_ocn(axy:zy,axy:zx), rho_atm(axy:zy,axy:zx), Hm(axy:zy,axy:zx), Hr(ny,nx),
        &    Cbl(axy:zy,axy:zx), Crl(ny,nx), Cdr(axy:zy,axy:zx), Csh(ny,nx),
        &    tempdiff(ny,nx), Te(ny,nx), Ta(axy:zy,axy:zx), F(ny,nx),
        &    ue(ny,0_ni:nx), ve(0_ni:ny,nx), ub(ny,0_ni:nx), vb(0_ni:ny,nx),
        &    u_ocn(ny,0_ni:nx), v_ocn(0_ni:ny,nx), w_ocn(ny,nx), windspeed(axy:zy,axy:zx))
        !
    allocate(Tm(1_ni,axy:zy,axy:zx,0_ni:nt-1_ni), Tm_tend(1_ni,ny,nx),
        &    Tm_tend_adv(ny,nx), Tm_tend_rlx(ny,nx),
        &    Tm_tend_hfl(ny,nx), Tm_flxx_adv(ny,0_ni:nx), Tm_flxy_adv(0_ni:ny,nx),
        &    Tr(axy:zy,axy:zx,0_ni:nt-1_ni), Tr_tend_adv(ny,nx),
        &    Tr_flxx_adv(ny,0_ni:nx), Tr_flxy_adv(0_ni:ny,nx))
        !
    tlo = 0_ni                               !Index of "old" time step
    tli = 0_ni                               !Index of "intermediate" time step (used for advection & f
    tln = 0_ni                               !Index of "new" time step
    tls = -1_ni                              !Index of a "saved" tendency, used in multi-step integrato
    !
    rho_ocn(:,:) = 1.026e3_nr                !Ocean density
    rho_atm(:,:) = 1.0383510273868126_nr !Atm density TODO: Get from atmosphere
    Hm(:,:)      = mixed_layer_depth       !Mixed layer depth
    !Hr(:,:)     = 150.0_nr                !Depth of return flow in 2-layer model
    Crl(:,:)     = relaxation_coeff        !Relaxation coefficient
    Cdr(:,:)     = drag_coeff              !Exchange coefficient for momentum
    Csh(:,:)     = sh_exchange_coeff       !Exchange coefficient for sensible heat
    !Tr(:,:,:)   = 273.15_nr               !Temperature of return flow in 2-layer model
    !
    Te(:,:)          = 0.0_nr                !Equilibrium temperature
    Tm(:,:,:,:)      = 0.0_nr                !Mixed layer temperature
    Ta(:,:)          = 0.0_nr                !Atm temperature
    eps              = 1.0e-5_nr             !Parameter representing various dissipative effects
    Cbl(:,:)         = 0.0_nr                !Bulk "coefficient" used in the calculation of Ekman curre
    tempdiff(:,:)    = 0.0_nr                !Atm-ocean temperature difference
    windspeed(:,:)   = 0.0_nr                !Atm wind speed at lower boundary
    F(:,:)           = 0.0_nr                !Sensible heat flux at atm-ocean interface
    Tm_tend_adv(:,:) = 0.0_nr                !Advective tendency of mixed layer temp
    Tm_tend_rlx(:,:) = 0.0_nr                !Relaxation tendency ------------------
    Tm_tend_hfl(:,:) = 0.0_nr                !Heat flux tendency --------------------
    ue(:,:)          = 0.0_nr                !Ekman current EW component
    ve(:,:)          = 0.0_nr                !Ekman current NS component
    ub(:,:)          = 0.0_nr                !Background current EW component
    vb(:,:)          = 0.0_nr                !Background current NS component
    u_ocn(:,:)       = 0.0_nr                !Total ocean current EW component
    v_ocn(:,:)       = 0.0_nr                !Total ocean current NS component
    !
    l050layermodel   = l050layermodel_c  !Switch for 0.5-layer model
    l100layermodel   = l100layermodel_c  !Switch for 1-layer model
    l125layermodel   = l125layermodel_c  !Switch for 1.25-layer model
    lekm             = lekmantransp      !Switch for Ekman currents
    lrlx             = lrelaxation       !Switch for relaxation
    !
    ! Prescribed wind field for test runs
    !allocate(utest(1_ni,0_ni:ny+1_ni,0_ni:nx), vtest(1_ni,0_ni:ny,0_ni:nx+1_ni))
    !
    !call set_flowfield(utest(1_ni,1_ni:ny,0_ni:nx), vtest(1_ni,0_ni:ny,1_ni:nx), 0.0_nr, 0.0_nr,
              !&              nx/2.0_nr, ny/2.0_nr, 0.0_nr, 0.0_nr, 1_ni)
    !
    ! Initial Tm profile
    call set_temp(10.0_nr, 3.0_nr*nx/4.0_nr, ny/2.0_nr, 5.0_nr, 5.0_nr, &
           &      Tm(1_ni,1_ni:ny,1_ni:nx,0_ni), 1_ni)
    ! Equilibrium temperature
    call set_temp(10.0_nr, 3.0_nr*nx/4.0_nr, ny/2.0_nr, 5.0_nr, 5.0_nr, Te, 1_ni)
    !
    ! Boundary condition for Tm
    call set_latbdr_Tm(Tm(:,:,:,0_ni))
    !
    call register_var_yxt(Tm, tln, 'sea_temperature', 'oceanic_mixed_layer_temperature', 'K')
    !
    call reg_coup_var_yx(F, 'ocean', 'surface heating')
    !
  end subroutine
  !
  ! Initialization of coupling
  subroutine init_slab_coupling()
    !
    call check_coup_var_available('atmos', 'surface x-wind', cvarid_atm_xwind)
    write(*,*) cvarid_atm_xwind
    call check_coup_var_available('atmos', 'surface y-wind', cvarid_atm_ywind)
    write(*,*) cvarid_atm_ywind
    call check_coup_var_available('atmos', 'surface temperature', cvarid_atm_temp)
    write(*,*) cvarid_atm_temp
    !
    !TODO: This will not work in the restart case
    call cal_ocean_diag(0_ni)
    !
  end subroutine
  !
  ! Clean up
  subroutine term_ocean_slab()
    !
    deallocate(rho_ocn, rho_atm, Hm, Hr, Cbl, Crl, Cdr, Csh,             &
        &      Te, Ta, tempdiff, F, ue, ve, ub, vb,                      &
        &      u_ocn, v_ocn, w_ocn, windspeed)
    !
    deallocate(Tm, Tm_tend_adv, Tm_tend_rlx, Tm_tend_hfl,                &
        &      Tm_flxx_adv, Tm_flxy_adv, Tr, Tr_tend_adv, Tr_flxx_adv,   &
        &      Tr_flxy_adv)
    !
    !deallocate(utest, vtest)
    !
  end subroutine
  !
  !
  ! The main function that actually does stuff
  subroutine run_ocean_slab(t, ts)
    !
    integer(kind=ni), intent(in) :: t, ts
    !
    real(kind=nr)    :: tstep, savefac
    integer(kind=ni) :: substep
    logical          :: ts_finished
    ! -----------------------------------------------------------------
    !
    ts_finished = .false.
    substep = 0_ni
    do while ( .not. ts_finished )
       ! Update time levels
       substep = substep + 1_ni
       call int_time%p(ts_finished, tstep, savefac, tlo, tli, tln, tls)
       !
       ! Prepare a save time level
       if ( substep == 1_ni .and. tls >= 0_ni ) then
          Tm(:,:,:,tls) = 0.0_nr
       end if
       !
       ! Calculating tendencies
       call cal_Tm_tendency()
       !
       ! Saving and applying tendencies
       if ( tls >= 0_ni ) then
          call apply_tend(Tm(:,1_ni:ny,1_ni:nx,:), Tm_tend, -savefac, tls, tls)
       end if
       if ( .not. ts_finished .or. tls < 0_ni ) then
          call apply_tend(Tm(:,1_ni:ny,1_ni:nx,:), Tm_tend, tstep, tlo, tln)
       else
          call apply_tend(Tm, Tm(:,:,:,tls), tstep, tlo, tln)
       end if
       !
       ! Set boundary conditions for prognostic variables
       call set_latbdr_Tm(Tm(:,:,:,tln))
       !
       ! Update diagnostics
       call cal_ocean_diag(t)
    end do
    !
  end subroutine
```

```
!
! Calculate time tendency of ocean mixed layer temperature
subroutine cal_Tm_tendency()
  !
  !    0.5-layer model
  if     (l050layermodel) then
     call cal_tend_hfl()
     call cal_tend_rlx()
     !
     ! 1-layer model
  elseif (l100layermodel) then
     call cal_tend_hfl()
     call cal_tend_adv1()
     call cal_tend_rlx()
     !
     ! 1.25-layer model
  elseif (l125layermodel) then
     call cal_tend_hfl()
     call cal_tend_adv2()
     call cal_tend_rlx()
     !
  end if
  !
  Tm_tend(1_ni,:,:) = Tm_tend_hfl(:,:) + Tm_tend_adv(:,:)    &
          &         + Tm_tend_rlx(:,:)
  !
end subroutine
!
! Temp tendency due to heat flux
subroutine cal_tend_hfl()
  !
  Tm_tend_hfl(:,:) = F(:,:)/(rho_ocn(1_ni:ny,1_ni:nx)*cp_ocn*Hm(1_ni:ny,1_ni:nx))
  !
end subroutine
!
! Temp tendency due to relaxation
subroutine cal_tend_rlx()
  !
  if (lrlx) then
     Tm_tend_rlx(:,:) = Crl(:,:)*(Tm(1_ni,1_ni:ny,1_ni:nx,tli) - Te(:,:))
  else
     Tm_tend_rlx(:,:) = 0.0_nr
  end if
  !
end subroutine
!
! Heat transport for 1-layer model: dT/dt ~ -u*grad(T)
subroutine cal_tend_adv1()
  !
  real(kind=nr) :: div_ocn(ny,nx)
  integer(kind=ni) :: i,j
```

```
    !---------------------------------------------------
    !
    ! Horizontal fluxes
    call cal_advx_2d(Tm(1_ni,:,:,tli), u_ocn, Tm_flxx_adv)
    call cal_advy_2d(Tm(1_ni,:,:,tli), v_ocn, Tm_flxy_adv)
    !
    !TODO: x,y (and z) for the ocean?
    div_ocn(:,:) = 0.0_nr
    call ddx_stag_2d(u_ocn, div_ocn, x_x(1_ni,:,:), y_x(1_ni,:,:))
    call ddy_stag_2d(v_ocn, div_ocn, x_y(1_ni,:,:), y_y(1_ni,:,:))
    !
    call cal_adv_2d(Tm(1_ni,1_ni:ny,1_ni:nx,tli), Tm_flxx_adv,                    &
             &      Tm_flxy_adv, div_ocn, Tm_tend(1_ni,:,:),                      &
             &      x_x(1_ni,:,:), y_x(1_ni,:,:), x_y(1_ni,:,:), y_y(1_ni,:,:))
    !
  end subroutine
  !
  ! Heat transport for 1.25-layer model: dT/dt ~ -div(u*T) + (max(w,0)*T_re + min(w,0)*T)/Hm
  subroutine cal_tend_adv2()
    !
    integer(kind=ni) :: i,j
    real(kind=nr)    :: T_re ! Reservoir temperature
    !----------------------------------------------
    !
    T_re = 283.15_nr !TODO:Make an array, so that not only a constant is possible. Probably redefin
    !                !TODO:Consider making reservoir temperature configurable.
    ! Horizontal fluxes
    call cal_advx_2d(Tm(1_ni,:,:,tli), u_ocn, Tm_flxx_adv)
    call cal_advy_2d(Tm(1_ni,:,:,tli), v_ocn, Tm_flxy_adv)
    !
    ! Horizontal flux divergence
    Tm_tend_adv = 0.0_nr
    call ddx_stag_2d(Tm_flxx_adv, Tm_tend_adv, x_x(1_ni,:,:), y_x(1_ni,:,:))
    call ddy_stag_2d(Tm_flxy_adv, Tm_tend_adv, x_y(1_ni,:,:), y_y(1_ni,:,:))
    !
    ! Upwelling/downwelling
    do i=1_ni,nx
       do j=1_ni,ny
          Tm_tend_adv(j,i) = Tm_tend_adv(j,i)                                    &
                           ! Adding vertical flux if w nonzero
                  &        - max(w_ocn(j,i), 0.0_nr)*T_re/Hm(j,i)                 &
                  &        - min(w_ocn(j,i), 0.0_nr)*Tm(1_ni,j,i,tli)/Hm(j,i)
       end do
    end do
    !
  end subroutine
  !
  ! Updates all ocean diagnostics
  subroutine cal_ocean_diag(t)
    use consts, only: T0
    !
    integer(kind=ni), intent(in) :: t
    !
    real(kind=nr) :: u_atm(axy:zy,axy:zx-1_ni), v_atm(axy:zy-1_ni,axy:zx)
    real(kind=nr) :: atm_temp_ptb(axy:zy,axy:zx)
    !--------------------------------------------------------------------
    !
    if (cvarid_atm_xwind >= 0_ni) then
       call req_coup_var_2d(cvarid_atm_xwind, u_atm)
    end if
    if (cvarid_atm_ywind >= 0_ni) then
       call req_coup_var_2d(cvarid_atm_ywind, v_atm)
    end if
    !
    call cal_windspeed(u_atm, v_atm)
    call cal_ekm_current(u_atm, v_atm)
    call set_background_current(t)
    !
    u_ocn(:,:) = ue(:,:) + ub(:,:)
    v_ocn(:,:) = ve(:,:) + vb(:,:)
    !
    call cal_w_ocn()
    !
    if (cvarid_atm_temp >= 0_ni) then
       call req_coup_var_2d(cvarid_atm_temp, atm_temp_ptb)
       !
       Ta(:,:)       = T0 + atm_temp_ptb(:,:)
       tempdiff(:,:) = Ta(1_ni:ny,1_ni:nx) - Tm(1_ni,1_ni:ny,1_ni:nx,tln)
       !
    else
       tempdiff(:,:) = 0.0_nr
       !
    end if
    !
    F(:,:) = -rho_atm(1_ni:ny,1_ni:nx)*cp_atm*Csh(:,:)*windspeed(1_ni:ny,1_ni:nx)*tempdiff(:,:)
    !
  end subroutine
  !
  ! Calculates the wind speed
  subroutine cal_windspeed(u, v)
    !
    real(kind=nr), intent(in) :: u(axy:,axy:), v(axy:,axy:)
    !
    integer(kind=ni) :: i,j
    !-------------------------------------------------------
    !
    do i=axy+1_ni,zx-1_ni
       do j=axy+1_ni,zy-1_ni
          windspeed(j,i) = (u(j,i) + u(j,i-1_ni))**2_ni   &
              &          + (v(j,i) + v(j-1_ni,i))**2_ni
       end do
    end do
    !
    windspeed(:,:) = max(sqrt(windspeed(:,:)/4.0_nr), 1.0_nr)
    !
  end subroutine
  !
  ! Ekman currents
  subroutine cal_ekm_current(u, v)
    !
    real(kind=nr), intent(in) :: u(axy:,axy:), v(axy:,axy:)
    real(kind=nr) :: f_m(axy:zy,axy:zx)
    !
    integer(kind=ni) :: i,j
    !------------------------------------------------------
    !
    if (lekm) then
       call cal_coriolis_parameter_mgrid(f_m)
       !
       Cbl(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_ni) = (  rho_atm(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_
                                        &      *        Cdr(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_ni)
                                        &      *  windspeed(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_ni))
                                        &      / (  rho_ocn(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_ni)
                                        &      *         Hm(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_ni)
                                        &      *  (     f_m(axy+1_ni:zy-1_ni,axy+1_ni:zx-1_ni)**
                                        &      +        eps**2_ni                              ))
       !----------------------------------------------------------------------------------------
       do i=0_ni,nx
          do j=1_ni,ny
             !           Linearly interpolating to u points.
             ue(j,i) =   (Cbl(j,i)         + Cbl(j,i+1_ni))                &
                &    *   (f_m(j,i)         + f_m(j,i+1_ni))                &
                &    *   (v(j-1_ni,i)      + v(j,i)                        &
                &    +    v(j-1_ni,i+1_ni) + v(j,i+1_ni)  )/16.0_nr        &
                !
                &    +   (Cbl(j,i)         + Cbl(j,i+1_ni))                &
                &    *   u(j,i)*eps/2.0_nr
          end do
       end do
       !
       do i=1_ni,nx
          do j=0_ni,ny
             !           Linearly interpolating to v points.
             ve(j,i) = - (Cbl(j,i)         + Cbl(j+1_ni,i))                &
                &    *   (f_m(j,i)         + f_m(j+1_ni,i))                &
                &    *   (u(j,i-1_ni)      + u(j,i)                        &
                &    +    u(j+1_ni,i-1_ni) + u(j+1_ni,i)    )/16.0_nr      &
                !
                &    +   (Cbl(j,i)         + Cbl(j+1_ni,i))                &
                &    *    v(j,i)*eps/2.0_nr
          end do
       end do
       !
    else
       !
       ue(:,:) = 0.0_nr
       ve(:,:) = 0.0_nr
       !
    end if
    !
  end subroutine
  !
  ! Any wanted background current should be specified here.
  subroutine set_background_current(t)
    !
    integer(kind=ni), intent(in) :: t
    !-------------------------------
    !
    call set_flowfield(ub, vb, 0.0_nr, 0.0_nr, nx/2.0_nr, ny/2.0_nr, 0.0_nr, 0.0_nr, 1_ni)
    !
  end subroutine
  !
  ! Ocean vertical velocity at grid cell bottoms
  subroutine cal_w_ocn()
    !
    integer(kind=ni) :: i,j
    !----------------------
    !
    w_ocn = 0.0_nr
    call ddx_stag_2d(u_ocn, w_ocn, x_x(1_ni,:,:), y_x(1_ni,:,:))
    call ddy_stag_2d(v_ocn, w_ocn, x_y(1_ni,:,:), y_y(1_ni,:,:))
    !
    w_ocn(:,:) = w_ocn(:,:)*Hm(1_ni:ny,1_ni:nx)
    !
  end subroutine
  !
  ! Coriolis parameter on main grid
  subroutine cal_coriolis_parameter_mgrid(f)
    use pe_diag, only: fcor, betacor
    !
    real(kind=nr), intent(inout) :: f(axy:,axy:)
    !---------------------------------------------
    !
    f(axy:zy,axy:zx) = fcor + betacor*y_m(1_ni,axy:zy,axy:zx)
    !
  end subroutine
  !
  ! Boundaries for ocean mixed layer temp
  subroutine set_latbdr_Tm(a)
    use base_config, only : latbdrtyp_ew, latbdrtyp_sn
    !
    real(kind=nr), intent(inout) :: a(:,axy:,axy:)
    !
    integer(kind=ni) :: i,j, obdr_ew, obdr_sn !-1=periodic, 0=zeroflux
    !-------------------------------------------------
    !
    obdr_ew = latbdrtyp_ew
    obdr_sn = latbdrtyp_sn
    !
    ! East-West boundaries
    select case(obdr_ew)
      case(-1_ni)
        call set_latbdr_periodic_ew(a)
      case( 0_ni)
        call set_latbdr_zerograd_ew(a)
      case( 1_ni)
        call set_latbdr_zerograd_ew(a)
    end select
    !
    ! North-South boundaries
    select case(obdr_sn)
      case(-1_ni)
        call set_latbdr_periodic_sn(a)
      case( 0_ni)
        call set_latbdr_zerograd_sn(a)
      case( 1_ni)
        call set_latbdr_zerograd_sn(a)
    end select
    !
  end subroutine
  !
  ! Boundaries for u_ocn. As implied by the atm winds except for wall condition.
  subroutine set_latbdr_u_ocn(u)
    use base_config, only : latbdrtyp_ew, latbdrtyp_sn
    !
    real(kind=nr), intent(inout) :: u(:,0_ni:)
    !
    integer(kind=ni) :: i,j, obdr_ew, obdr_sn !-1=periodic, 0=zeroflux
    !-----------------------------------------------------------------
    !
    obdr_ew = latbdrtyp_ew
    obdr_sn = latbdrtyp_sn
    !
    ! East-West boundaries
    select case(obdr_ew)
      case(-1_ni)
        !do nothing
      case( 0_ni)
        call set_latbdr_zero_u_ocn(u)
      case( 1_ni)
        !do nothing
    end select
    !
    ! North-South boundaries
    select case(obdr_sn)
```

```
      case(-1_ni)
        !do nothing
      case( 0_ni)
        !do nothing
      case( 1_ni)
        !do nothing
    end select
    !
  end subroutine
  !
  ! Boundaries for v_ocn.
  subroutine set_latbdr_v_ocn(v)
     use base_config, only : latbdrtyp_ew, latbdrtyp_sn
    !
    real(kind=nr), intent(inout) :: v(0_ni:,:)
    !
    integer(kind=ni) :: i,j, obdr_ew, obdr_sn !-1=periodic, 0=zeroflux
    !-----------------------------------------------------------------
    !
    obdr_ew = latbdrtyp_ew
    obdr_sn = latbdrtyp_sn
    !
    ! East-West boundaries
    select case(obdr_ew)
      case(-1_ni)
        !do nothing
      case( 0_ni)
        !do nothing
      case( 1_ni)
        !do nothing
    end select
    !
    ! North-South boundaries
    select case(obdr_sn)
      case(-1_ni)
        !do nothing
      case( 0_ni)
        call set_latbdr_zero_v_ocn(v)
      case( 1_ni)
        !do nothing
    end select
    !
  end subroutine
  !
  ! Set lateral boundaries to zero
  subroutine set_latbdr_zero_u_ocn(u)
    real(kind=nr), intent(inout) :: u(:,0_ni:)
    !
    integer(kind=ni) :: i,j
    ! ---------------------------------------
    !
    do j = 1_ni,ny
       u(j,0_ni) = 0._nr
       u(j,nx) = 0._nr
    end do
    !
  end subroutine
  !
  !
  subroutine set_latbdr_zero_v_ocn(v)
    real(kind=nr), intent(inout) :: v(0_ni:,:)
    !
    integer(kind=ni) :: i,j
    ! --------------------------------------
    !
    do i = 1_ni,nx
       v(0_ni,i) = 0._nr
       v(ny,i) = 0._nr
    end do
    !
  end subroutine
  !
  !
  ! Set some Gaussian temperature profile where the background temp equals T0 from const module
  subroutine set_temp(amp, cx, cy, sx, sy, T, sel)
    use consts, only: T0
    !
    integer(kind=ni), intent(in) :: sel
    real(kind=nr), intent(in)    :: amp, cx, cy, sx, sy
    real(kind=nr), intent(inout) :: T(:,:)
    !
    integer(kind=ni) :: i,j
    real(kind=nr)    :: arg
    !-----------------------------------------------------
    !
    select case(sel)
       case(1_ni)
         ! 2d Gaussian anomaly
         do i=1_ni,nx
            do j=1_ni,ny
               arg = ((i - cx)**2_ni)/(2.0_nr*sx**2_ni)  &
               &   + ((j - cy)**2_ni)/(2.0_nr*sy**2_ni)
               !
               T(j,i) = T0 + amp*exp(-arg)
            end do
         end do
         !
    end select
    !
  end subroutine
  !
  ! Set some idealised flow field
```

```fortran
subroutine set_flowfield(u, v, uspeed, vspeed, cx, cy, sx, sy, sel)
  !
  real(kind=nr), intent(in)    :: uspeed, vspeed, cx, cy, sx, sy
  integer(kind=ni), intent(in) :: sel
  real(kind=nr), intent(inout)   :: u(:,0_ni:), v(0_ni:,:)
  !
  integer(kind=ni) :: i,j
  real(kind=nr)    :: arg
  !----------------------------------------------------------
  !
  select case(sel)
    case(1_ni)
      ! Uniform fields
      u(:,:) = uspeed
      v(:,:) = vspeed
      !
    case(2_ni)
      ! u linearily increasing towards the North
      do j=1_ni,ny
         u(j,:) = (uspeed/ny)*j
      end do
      !
      v(:,:) = vspeed
      !
    case(3_ni)
      ! u linearily increasing towards the West
      do i=0_ni,nx
         u(:,i) = uspeed*(1.0_nr - (1.0_nr/nx)*i)
      end do
      !
      v(:,:) = vspeed
      !
    case(0_ni)
      ! u linearily increasing towards the East
      do i=0_ni,nx
         u(:,i) = (uspeed/nx)*i
      end do
      !
      v(:,:) = vspeed
      !
    case(4_ni)
      ! Gaussian y-dependence in v
      do j=0_ni,ny
         arg = ((j - cy)**2_ni)/(2.0_nr*sy**2_ni)
         v(j,:) = vspeed*exp(-arg)
      end do
      !
      u(:,:) = uspeed
      !
    case(5_ni)
      ! Gaussian x-dependence in v
```

```
        do i=1_ni,nx
           arg = ((i - cx)**2_ni)/(2.0_nr*sx**2_ni)
           v(:,i) = vspeed*exp(-arg)
        end do
        !
        u(:,:) = uspeed
        !
      case(6_ni)
        ! Anticyclonic solid body rotation
        do j=1_ni,ny
           u(j,:) = (uspeed/(0.5_nr*ny))*(j - cy)
        end do
        !
        do i=1_ni,nx
           v(:,i) = -(vspeed/(0.5_nr*nx))*(i - cx)
        end do
        !
      case(7_ni)
        ! Cyclonic solid body rotation
        do j=1_ni,ny
           u(j,:) = -(uspeed/(0.5_nr*ny))*(j - cy)
        end do
        !
        do i=1_ni,nx
           v(:,i) = (vspeed/(0.5_nr*nx))*(i - cx)
        end do
        !
    end select
    !
  end subroutine
  !
  !
end module ocean_slab
```

Bibliography

Boyce, W. E. & DiPrima, R. C. (2005), *Elementary Differential Equations and Boundary Value Problems*, eight edn, Jon Wiley and Sons.

Codron, F. (2012), 'Ekman Heat Transport for Slab Oceans', *Climate Dynamics* **38**(1-2), 379–389.

Collins, W. D., Rasch, P. J., Boville, B. A., Hack, J. J., McCaa, J. R., Williamson, D. L., Kiehl, J. T., Briegleb, B., Bitz, C. & Lin, S. (2004), 'Description of the NCAR Community Atmosphere Model (CAM 3.0)', *NCAR Tech. Note NCAR/TN-464+ STR* .

Donnadieu, Y., Pierrehumbert, R., Jacob, R. & Fluteau, F. (2006), 'Modelling the Primary Control of Paleogeography on Cretaceous Climate', *Earth and Planetary Science Letters* **248**(1), 426–437.

Fairall, C., Bradley, E. F., Hare, J., Grachev, A. & Edson, J. (2003), 'Bulk Parameterization of Air-Sea Fluxes: Updates and Verification for the COARE Algorithm', *Journal of Climate* **16**(4), 571–591.

Farlow, S. J. (1993), *Partial Differential Equations for Scientists and Engineers*, Dover Publications.

Gill, A. E. (1980), 'Some Simple Solutions for Heat-Induced Tropical Circulation', *Quarterly Journal of the Royal Meteorological Society* **106**(449), 447–462.

Held, I. M. (2005), 'The Gap Between Simulation and Understanding in Climate Modeling', *Bulletin of the American Meteorological Society* **86**(11), 1609–1614.

Holton, J. (2004), *An Introduction to Dynamic Meteorology*, fourth edn, Elsevier Academic Press.

Kara, A. B., Hurlburt, H. E. & Wallcraft, A. J. (2005), 'Stability-Dependent Exchange Coefficients for Air-Sea Fluxes*', *Journal of Atmospheric and Oceanic Technology* **22**(7), 1080–1094.

Kara, A. B., Rochford, P. A. & Hurlburt, H. E. (2000), 'Efficient and Accurate Bulk Parameterizations of Air-Sea Fluxes for use in General Circulation Models', *Journal of Atmospheric and Oceanic Technology* **17**(10), 1421–1438.

Liu, W. T., Katsaros, K. B. & Businger, J. A. (1979), 'Bulk Parameterization of Air-Sea Exchanges of Heat and Water Vapor Including the Molecular Constraints at the Interface', *Journal of the Atmospheric Sciences* **36**(9), 1722–1735.

Marshall, J. & Molteni, F. (1993), 'Toward a Dynamical Understanding of Planetary-Scale Flow Regimes', *Journal of the Atmospheric Sciences* **50**(12), 1792–1818.

Matsuno, T. (1966), 'Quasi-Geostrophic Motions in the Equatorial Area', *J. Meteor. Soc. Japan* **44**(1), 25–43.

Maze, G., D'andrea, F., Colin de Verdiere, A. & Klein, P. (2011), 'Stationary Atmospheric Responses to an Idealized Sea Surface Temperature Anomaly in the Southern Ocean', *Journal of Climate* **24**(14), 3686–3704.

Maze, G., D'Andrea, F. & De Verdiere, A. C. (2006), 'Low-Frequency Variability in the Southern Ocean Region in a Simplified Coupled Model', *Journal of geophysical research* **111**(C5), C05010.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992), *Numerical Recipes in Fortran 77, The Art of Scientific Computing*, Vol. 1, 2 edn, Cambridge University Press.

Spensberger, C. (2012), 'Bedymo: Concept and Realisation, draft version 1-d5'.

Stephens, M. Y., Oglesby, R. J. & Maxey, M. (2005), 'A One-Dimensional Mixed Layer Ocean Model for use in Three-Dimensional Climate Simulations: Control Simulation Compared to Observations', *Journal of Climate* **18**(13), 2199–2221.

Tremback, C. J., Powell, J., Cotton, W. R. & Pielke, R. A. (1987), 'The forward-in-time Upstream Advection Scheme: Extension to Higher Orders', *Monthly Weather Review* **115**(2), 540–555.

Williams, P. D. (2011), 'The RAW Filter: An Improvement to the Robert-Asselin Filter in Semi-Implicit Integrations', *Monthly Weather Review* **139**(6), 1996–2007.