



CLIMATE CHANGE RESEARCH CENTRE  
UNIVERSITY OF NEW SOUTH WALES

---

# A DYNAMIC, EMBEDDED LAGRANGIAN MODEL FOR OCEAN CLIMATE MODELS

---

Michael L Bates

2011

Supervisors: Stephen M Griffies and Matthew H England

A thesis submitted in the fulfillment of the requirements for the degree of Doctor of  
Philosophy.



# **Originality Statement**

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgment is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the projects design and conception or in style, presentation and linguistic expression is acknowledged.



# Abstract

A framework is developed for embedding a Lagrangian sub-model into an ocean climate model that has a fixed horizontal Eulerian grid. The embedded Lagrangian model can be used to explicitly represent processes that are at the subgrid scale to the Eulerian. The Lagrangian framework has a number of potential applications. The embedded Lagrangian framework is applied to two different applications in level coordinate ocean climate models, namely open ocean deep convection and gravity driven downslope flows. Both of these processes are subgrid scale in the present generation of global scale level coordinate ocean models.

To embed the Lagrangian model in order to represent open ocean deep convection and gravity driven downslope flows, it is necessary to develop a strategy to partition the mass and momentum of the model into an Eulerian component and a Lagrangian component. The Lagrangian parcels, called “blobs,” can be moved around in three dimensions using a more appropriate set of dynamics and may also interact with the Eulerian model.

The technique is not a parameterisation, but rather a framework in which a multitude of parameterisations may be implemented. It is thus possible to develop parameterisations which are analogous to many existing and schemes. Initially, several existing parameterisations are emulated. Then, two schemes are developed, one which models open ocean deep convection, and the other (based on a streamtube model) models gravity driven downslope flows.

The framework is also tested in several well known idealised test cases. In the test case for deep convection, a 100m resolution non-hydrostatic model of a deep convective patch is used and it is shown that the use of the Lagrangian blobs improved the fidelity of the simulation of a 2km resolution hydrostatic simulation when compared to the use of convective adjustment. Two test cases are used to examine the performance of the scheme for gravity driven downslope flows. The Lagrangain scheme significantly improves the representation of the physics of a plume on a uniform slope and the resultant large scale circulation.



# Acknowledgements

This work was supported by an award under the Merit Allocation Scheme on the NCI National Facility at the ANU. This project also benefited greatly from the support of the ARC Centre of Excellence for Mathematics and Statistics of Complex Systems, The ARC Network for Earth System Science and The ARC Centre of Excellence for Climate System Science.

I am greatly indebted to Stephen Griffies for the time, effort and patience that he has shown towards me and this project. His very high scientific standards and fastidious nature have greatly influenced not only this project, but also my own development as a scientist. Both myself and the project have benefited untold amounts. His professional guidance and personal friendship have been a constant and unwavering part of my life for almost four years, and for this, I am grateful. Thank you Steve.

The sincerest of thanks also go to Matthew England. Without Matt giving me the academic freedom to pursue blobs, this project would never have become a reality. Matt's enthusiastic and encouraging demeanour, along with his astute scientific nature and friendship have had a large and positive influence on my own development and that of this project.

Particularly when the blobs were in their formative stages, I spoke to many people about them. The soundest initial advice that I received was from Alistair Adcroft, who also made himself available throughout the project to discuss algorithm development and debugging. I would also like to thank Darren Engwirda, who was always keen to discuss algorithm development, and suggested the point location scheme that was eventually implemented. Other people who I spoke to and gave me ideas and encouragement, particularly toward the beginning of the project were Jan Zika, Willem Sijp, Julien Le Sommer, Alex Sen Gupta, David Dritschel, Tim Leslie and the Ocean Model Development Team at GFDL.

Thanks go to the GFDL team, and, Trevor McDougall at CSIRO for hosting visits of mine at their respective institutions, which gave me invaluable time and personal contact with Steve. Adi Benito and Francisco were very welcoming hosts on several occasions in Princeton, and I greatly enjoyed their company when I was a long way from home.

I am very grateful to my parents, Pam and Bob Bates, who always encouraged me to pursue that which interested me. It was this spirit that lead me to take on this “high risk” project, and it was their values of perseverance and hard work that gave me the ability and wherewithal to get the project to the stage it has reached. Finally, I would like to thank Emily Shaw, whose love, companionship, good nature, sympathetic ear, sound advice and delicious vegan desserts has single-handedly maintained my sanity.



# Contents

<b>Originality Statement</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>I Background, Theory and Implementation</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Eulerian Ocean Models . . . . .	5
1.2 Lagrangian Ocean Models . . . . .	5
1.3 An Embedded Lagrangian Framework . . . . .	6
<b>2 Applications of an Embedded Lagrangian Framework</b>	<b>9</b>
2.1 Open Ocean Deep Convection . . . . .	9
2.2 Gravity-Driven Downslope Flows . . . . .	11
2.2.1 The Weddell Sea . . . . .	13
2.2.2 The Ross Sea . . . . .	14
2.2.3 The Faroe Bank Channel . . . . .	14
2.2.4 The Denmark Strait . . . . .	16
2.3 Convection in OGCM's . . . . .	16
2.3.1 Open Ocean Convection . . . . .	16
2.4 Bottom Boundary Layer Schemes in OGCM's . . . . .	18
2.5 Other Potential Applications of the Lagrangian Framework . . . . .	21
2.6 Summary . . . . .	23
<b>3 Formulation of the Lagrangian Framework</b>	<b>25</b>
3.1 What is a Blob? . . . . .	25
3.2 Partitioning an Ocean Model . . . . .	26
3.3 Seawater Mass and Tracer Mass . . . . .	27
3.4 Momentum . . . . .	32
3.5 Energy . . . . .	35
3.5.1 Gravitational Potential Energy . . . . .	36
3.5.2 Kinetic Energy . . . . .	37

3.6	Summary . . . . .	40
<b>4</b>	<b>Dynamically Passive Schemes</b>	<b>41</b>
4.1	An “NCon”-like Exchange of Mass . . . . .	41
4.2	A “Diffusion-like” Scheme . . . . .	44
4.3	A “Swap” Exchange of Mass . . . . .	45
4.4	An Overflow Scheme . . . . .	46
4.5	Two Slightly Different Overflow Schemes . . . . .	49
4.6	Summary . . . . .	50
<b>5</b>	<b>Dynamically Active Schemes</b>	<b>51</b>
5.1	Blob Dynamics . . . . .	51
5.2	An Open Ocean Implementation . . . . .	53
5.2.1	Open Ocean Blob Initialisation . . . . .	54
5.2.2	Open Ocean Blob Entrainment, Detrainment and Destruction . . . . .	57
5.3	A Bottom Boundary Implementation . . . . .	58
5.3.1	Bottom Blob Initialisation . . . . .	62
5.4	Switching Between Dynamically Active Regimes . . . . .	64
5.5	Summary . . . . .	66
<b>6</b>	<b>Implementation of Grid Cell Partitioning in MOM</b>	<b>69</b>
6.1	General Numerical Considerations . . . . .	69
6.1.1	Numerical Representation of Blobs . . . . .	70
6.1.2	Order of Operations . . . . .	70
6.1.3	Parallelisation and Bitwise Reproducibility . . . . .	71
6.1.4	Restart and Diagnostic Files . . . . .	72
6.2	Implementation of Partitioning . . . . .	73
6.2.1	Calculation of Thickness . . . . .	74
6.2.2	Tracer Budgets . . . . .	74
6.2.3	Compatibility of Tracer Mass and Seawater Mass . . . . .	76
6.2.4	Momentum Budgets . . . . .	77
6.2.5	Energy Budgets . . . . .	79
6.3	Discretisation Considerations . . . . .	80
6.3.1	Vertical Coordinate Considerations . . . . .	80
6.3.2	Calculation of Total Density, Tracer Concentration and Buoyancy Frequency . . . . .	84
6.3.3	Calculation of Hydrostatic Pressure . . . . .	85
6.3.4	Calculation of Surface Height . . . . .	87
6.3.5	Calculation of Bottom Pressure . . . . .	88
6.3.6	Pressure at a Tracer Point . . . . .	89
6.3.7	Geopotential at a Tracer Point . . . . .	90
6.4	Horizontal Grid Considerations . . . . .	90
6.5	Time Stepping Considerations . . . . .	92

6.5.1	Some Further Vertical Coordinate Considerations . . . . .	96
6.5.2	Calculation of Blob Divergence . . . . .	100
6.5.3	Cell Thickness Considerations . . . . .	100
6.5.4	A Vertical CFL Constraint . . . . .	103
6.5.5	A Note Regarding the E System Vertical Velocity . . . . .	104
6.5.6	Further Time Stepping Considerations . . . . .	104
6.6	Summary . . . . .	107
<b>7</b>	<b>Space and Time Methods for Blobs</b>	<b>109</b>
7.1	Blobs and the Eulerian Grid . . . . .	109
7.1.1	Horizontal Projection Considerations . . . . .	109
7.1.2	Blob Grid Cell Search Algorithms in the Horizontal . . . . .	110
7.1.3	Cyclic Grids . . . . .	113
7.1.4	The Tripolar Grid . . . . .	115
7.1.5	Blob Grid Cell Search Algorithms in the Vertical . . . . .	116
7.1.6	Topography Considerations . . . . .	118
7.1.7	Grounding and Penetration of the Surface . . . . .	119
7.1.8	Interpolation to Blobs . . . . .	119
7.2	Blob List Sorting Algorithms . . . . .	123
7.3	Blob Trajectories . . . . .	125
7.3.1	Eulerian Grid Interaction and Step Size Considerations . . . . .	127
7.3.2	A Series of Explicit Runge-Kutta Schemes with Error Control . . . . .	128
7.3.3	The Open Ocean Implementation . . . . .	132
7.3.4	The Price-Baringer Dynamic Streamtube Implementation . . . . .	133
7.3.5	Reproducibility with an Adaptive Step Size . . . . .	133
7.3.6	Limiting Velocity and Position . . . . .	135
7.3.7	Namelist Variables . . . . .	136
7.4	Summary . . . . .	140
<b>8</b>	<b>Miscellanea</b>	<b>141</b>
8.1	Overall Algorithmic Structure and Flow . . . . .	141
8.2	Known Limitations . . . . .	143
8.2.1	GOTM . . . . .	143
8.2.2	Pressure Based coordinates . . . . .	143
8.2.3	Momentum Transfer Between the E and L systems . . . . .	143
8.2.4	The Evolution of Non-Conservative Tracers . . . . .	143
8.2.5	The Grid Cell Mass Constraint with Dynamically Active Bottom Blobs . . . . .	143
8.2.6	The Height and Width of Dynamically Active Bottom Blobs . . . . .	144
8.2.7	Code Optimisation . . . . .	144

<b>II Proof of Concept</b>	<b>145</b>
<b>9 Introduction</b>	<b>147</b>
9.1 Previous Studies . . . . .	147
9.1.1 Deep Convection . . . . .	147
9.1.2 Overflows . . . . .	148
9.2 Preliminaries . . . . .	149
<b>10 Dynamically Passive Blobs</b>	<b>151</b>
10.1 The NCon Scheme . . . . .	151
10.2 The Overflow Scheme . . . . .	151
<b>11 The Deep Convection Test Case</b>	<b>153</b>
11.1 Configuration Details . . . . .	153
11.1.1 The Hydrostatic Approximation and Model Effects . . . . .	154
11.1.2 Lagrangian Blob Parameters . . . . .	155
11.2 Results . . . . .	156
11.2.1 Hydrostatic Approximation . . . . .	156
11.2.2 Mean Depth of Passive Tracer . . . . .	161
11.2.3 Depth of Blobs . . . . .	166
11.2.4 Blob Speed . . . . .	171
11.2.5 Blob Mass and Density . . . . .	171
11.2.6 Number of Blobs . . . . .	174
11.2.7 Compute Time . . . . .	178
11.3 Summary . . . . .	179
11.3.1 The Lagrangian Blobs . . . . .	180
<b>12 The DOME Test Case</b>	<b>183</b>
12.1 Configuration Details . . . . .	183
12.2 Analysis Techniques . . . . .	186
12.3 Experimental Details . . . . .	187
12.3.1 Friction . . . . .	188
12.3.2 Lateral Diffusion and Bottom Drag . . . . .	189
12.3.3 Traditional Overflow Schemes . . . . .	189
12.3.4 Lagrangian Blob Schemes . . . . .	189
12.4 Results . . . . .	192
12.4.1 Preliminaries . . . . .	192
12.4.2 Plume Mass and Density . . . . .	204
12.4.3 Blob Mass and Density . . . . .	206
12.4.4 Plume Speed . . . . .	210
12.4.5 Blob Speed . . . . .	213
12.4.6 Plume Depth . . . . .	216
12.4.7 Blob Age and the Number of Blobs . . . . .	218
12.4.8 Compute Time . . . . .	222

12.5	Summary . . . . .	225
12.5.1	Friction Scheme . . . . .	225
12.5.2	Traditional Overflow Schemes . . . . .	225
12.5.3	The Lagrangian Blobs . . . . .	226
<b>13</b>	<b>The Bowl Test Case</b>	<b>229</b>
13.1	Configuration Details . . . . .	229
13.2	Experimental Details . . . . .	231
13.2.1	Friction . . . . .	231
13.2.2	Lateral Diffusion . . . . .	232
13.2.3	Traditional Overflow Schemes . . . . .	232
13.2.4	Lagrangian Blob Schemes . . . . .	233
13.3	Results . . . . .	234
13.3.1	Preliminaries . . . . .	235
13.3.2	System Properties . . . . .	246
13.3.3	Blob Age and the Number of Blobs . . . . .	249
13.3.4	Blob Mass and Density . . . . .	254
13.3.5	Compute Time . . . . .	257
13.4	Summary . . . . .	260
13.4.1	Friction Scheme . . . . .	260
13.4.2	Traditional Overflow Schemes . . . . .	260
13.4.3	The Lagrangian Blob Scheme . . . . .	260
<b>14</b>	<b>Conclusions</b>	<b>263</b>
<b>A</b>	<b>Calculation of <math>m^a</math> for the Diffusion-like Scheme</b>	<b>269</b>
<b>B</b>	<b>Surface Area and Radius of a Sphere</b>	<b>271</b>
<b>C</b>	<b>List of Symbols and Abbreviations</b>	<b>273</b>
C.1	Abbreviations . . . . .	273
C.2	Latin Symbols . . . . .	273
C.3	Greek Symbols . . . . .	277
C.4	Model Variables . . . . .	279
C.5	Operators . . . . .	280



# **Part I**

# **Background, Theory and Implementation**



# Chapter 1

## Introduction

For many applications of computational fluid dynamics, the main limiting factor is available computational resources. The limitation of computational resources and the very large scale of the domain in global climate modelling results in a large number of climatically important processes that are unresolved (and are unlikely to be fully resolved in the near future; Griffies et al., 2010). There has thus been significant effort in the ocean model development community to formulate subgridscale (SGS) parameterisations (Griffies, 2004; McWilliams, 2007; Griffies et al., 2010).

All global scale ocean models that are used for realistic coupled climate experiments use a fixed Eulerian grid in the horizontal. There are some models under development that are fully Lagrangian (Haertel and Randall, 2002), and there are others that have an adaptive mesh (Piggott et al., 2007), however, none of these other formulations have yet reached a stage where they are able to be used for realistic, coupled global scale studies. Lagrangian coordinates in the vertical (for example, isopycnal) are finding much utility for realistic studies, however, the most common class of ocean model used in realistic global scale studies remains the “level” model (which we take here to mean geopotential,  $z^*$ , pressure and  $p^*$  coordinates). A summary of level model vertical coordinates may be found in Chapter 6 of Griffies (2004).

Regardless of the coordinate system used, capturing the effects of unresolved or poorly represented physics on the larger scale grid should be the aim of all parameterisations. The success or otherwise of a parameterisation depends largely on what questions the modeller is asking. Tracer fields (including biogeochemistry), momentum and energetics are all affected directly or indirectly by parameterisations. As such, a parameterisation should aim to improve the realism of the properties of the simulation. There is a caveat, however. Model simulations are often “tuned” using various parameters to optimise model performance against a subset of observable metrics. The addition of more realistic physics or parameterisations may initially appear to degrade a model solution, however, that can mean that other aspects of the model physics need to be improved or different values for tunable parameters selected. Finally, a parameterisation should also aim to minimise the additional computational cost required to run a model with that parameterisation, bearing

in mind that any significant additional computational loading will be borne at the expense of otherwise increased resolution.

In terms of the assessment of the fidelity of parameterisations, this can be problematic. Observations that are used as a benchmark or ground truth are themselves imperfect and subject to their own errors. The fields used to force realistic simulations have their own errors and uncertainty, making the comparison and assessment of one solution versus another solution versus observations of nature a non-trivial matter.

The fundamental equations of any numerical ocean model are applied throughout the computational domain. There may be spatially and time varying parameters or parameterisations applied to the domain, however, the same basic equations are applied throughout. For global scale climate models, these equations are the primitive equations (see Chapter 12 of Griffies, 2004). One variation on this idea is, however, the super-parameterisation, first applied in the oceanographic context by Campin et al. (2011). The super-parameterisation embeds a two dimensional non-hydrostatic model into a hydrostatic model. The admission of non-hydrostatic dynamics is fundamentally different to the hydrostatic primitive equations of the main model. The application of a super-parameterisation to convection, and indeed to other problems, is potentially a fruitful avenue of enquiry.

Here, we propose a rather different means by which to admit fundamentally different dynamics of unresolved or poorly represented physics in ocean climate models. Specifically, we embed a Lagrangian model within an Eulerian model. The proposed framework most closely resembles the cloud-in-cell method (Christiansen, 1973; Mohammadian and Marshall, 2010), however it need not be restricted to two dimensional vorticity. The iceberg model of Martin and Adcroft (2010) is a precedent for our proposed framework within the global scale ocean modelling community. The present framework, however, generalises the ideas behind the iceberg model to be fully three dimensional and fully interactive with the Eulerian model. Note that here, when we say Eulerian model, we are referring to models in which we have a fixed horizontal coordinate system, but where the vertical coordinate system is not necessarily fixed. Thus, in addition to the geopotential coordinate system, the framework may possibly be applied to other depth based coordinates, terrain following coordinates, pressure based coordinates, or isopycnal based coordinate systems. In Chapter 5 we specifically develop and apply the framework to convection and down slope flows in level models.

It is contended in this thesis that certain physical processes are well suited to a Lagrangian representation and that it is viable, and indeed advantageous, to represent those processes within an Eulerian ocean climate model. This contention is tested by designing and implementing an embedded Lagrangian model for two physical processes, being gravity driven downslope flows and deep convection. The embedding of an interactive Lagrangian model that moves parcels of fluid within an Eulerian model is unique in the ocean

model community and, as far as the author is aware, is unique in the field of computational fluid dynamics.

## 1.1 Eulerian Ocean Models

There is a plethora of Eulerian ocean models that have been developed over a period of decades, with the most common class of Eulerian model being the geopotential coordinate (Bryan, 1969; Semtner, 1974; Cox, 1984). Four other common classes of model exist, being the pressure based coordinate model (Huang et al., 2001; de Szoeke and Samelson, 2002), the terrain following coordinate (Blumberg and Mellor, 1987) and isopycnal-coordinate models. Griffies (2004) provides a summary of the relative strengths and weaknesses of each class of Eulerian model, and Chapter 6 of Griffies (2009) provides a pedagogical comparison of pressure based and depth based coordinates.

Note that we refer here to “Eulerian models” as any model where the horizontal positioning of a discrete cell is fixed in time. Hence, the “quasi-Lagrangian” vertical coordinate models (e.g. pressure, isopycnal) are considered “Eulerian” for the purposes of this document.

A new class of adaptive, unstructured mesh, finite element ocean models is under development (Piggott et al., 2007). These models, while having powerful potential, are not yet developed to the stage where they are able to be incorporated into realistic coupled climate simulations (see Griffies et al., 2010, for a review of the current state of ocean climate models).

Models in which the grid elements are beyond a certain aspect ratio need to make the hydrostatic approximation in order to maintain numerical stability. It is this high aspect ratio regime that this thesis is concerned with. The embedded Lagrangian model is not the only example of embedding a model with non-hydrostatic dynamics; Campin et al. (2011) develops a “super-parameterisation” (discussed in Section 2.3.1) which embeds a 2-d non-hydrostatic model within a hydrostatic model (the super-parameterisation is also a common technique in atmospheric models). While there are implications for conservation of energy of having embedded non-hydrostatic models (discussed in Section 3.5.2), there are clear advantages to being able to explicitly represent such non-hydrostatic physics without the cost of having to run a full non-hydrostatic model.

## 1.2 Lagrangian Ocean Models

In contrast to Eulerian ocean models, Lagrangian ocean models are uncommon. The only Lagrangian ocean model known to the author is the Slippery Sacks Ocean Model (SSOM; Haertel and Randall, 2002; Haertel et al., 2004, 2009). The SSOM uses “slippery sacks,” so called because of the shape and properties of the Lagrangian water parcels.

The study by Haertel et al. (2004) used the parabolic lake configuration of Beletsky et al. (1997) to test the slippery sacks model and drew attention to some of the shortcomings of the formulation and suggests some potential improvements. Haertel et al. (2009) further develops the model by admitting the horizontal mixing of momentum, and showing that the formulation can reproduce the analytic solutions of Stommel (1948) and Munk (1950) for a western boundary current in an idealised North Atlantic domain.

A fully Lagrangian atmospheric model has also been proposed by Alam and Lin (2008). Recently, Mohammadian and Marshall (2010) have proposed a “vortex in cell” method for quasi-geostrophic oceanic flows. A vortex in cell method is a Lagrangian approach, where to improve the computational efficiency, properties are inverted to a grid, which is then used to increment the position of the Lagrangian particles.

In terms of the embedding of Lagrangian models into Eulerian models, there are only a limited number of applications that the author is aware of. One application is the iceberg models of Martin and Adcroft (2010), in which an iceberg model (where the bergs are represented as Lagrangian particles) are embedded on the surface of the Eulerian grid (iceberg models, such as Bigg et al., 1996, have been used before; however, they were not interactive with the Eulerian model). Another application in which Lagrangian and Eulerian models are coupled is wind wave interactions (e.g. Ardhuin et al., 2001).

### 1.3 An Embedded Lagrangian Framework

We begin formulating the Lagrangian blob<sup>1</sup> scheme by making some statements about the entire system, that is, the combined properties of the Eulerian model and the Lagrangian model.

1. The Eulerian model and the Lagrangian model operate within the same coordinate system (for example latitude, longitude and depth).
2. The mass of the system (in non-Boussinesq models) or volume of the system (in Boussinesq models) must evolve in a conservative manner.
3. Tracer content must evolve in a conservative manner.
4. Mass (in non-Boussinesq models) or volume (in Boussinesq models) and tracer are conservatively transferred between the Eulerian model and the Lagrangian model.

The embedded Lagrangian framework is an attempt to utilise the advantages of Lagrangian formulations and Eulerian formulations. It is not the first attempt in computational fluid dynamics at such an approach. One approach is the Arbitrary Lagrangian-Eulerian approach (ALE; Hirt et al., 1972), in which a mesh may be fixed in the Eulerian

---

<sup>1</sup>The term “blob” is chosen as the buoyancy blob technique described is, in many ways, analogous to the “vortex blob” technique (Chorin, 1978). Indeed, the Lagrangian framework and buoyancy blob technique was largely inspired by the author’s previous work on vortex blobs with Charlie Macaskill while at the University of Sydney.

sense or move in the Lagrangian sense. The ALE method is quite distinct from the embedded Lagrangian framework proposed here.

The present approach is more similar in approach with the cloud in cell method (Christiansen, 1973), and the Contour-Advection Semi-Lagrangian (CASL; Dritschel and Ambaum, 1997) algorithm. As mentioned above, there has recently been an application of the vortex in cell method to oceanic flows by Mohammadian and Marshall (2010), which is the method that is most similar to the present embedded scheme that has been applied to oceanic flows.

The Lagrangian blob approach proposed here results from the desire to treat fluid parcels with the most appropriate dynamics and applying the most convenient numerics to solve for those dynamics.

The structure of the rest of this thesis is as follows. The thesis is split into two parts. The first part provides the motivation, theory and implementation of an embedded Lagrangian model:

- Chapter 2 provides a discussion of the physics of the applications chosen for our particular implementation, as well as some other potential applications for the framework.
- Chapter 3 provides a formalism for the Lagrangian framework.
- Chapter 4 formulates some schemes for the Lagrangian framework that recover existing parameterisations.
- Chapter 5 formulates two new schemes for implementation in the Lagrangian framework.
- Chapter 6 describes the software engineering techniques and numerical algorithms used to implement the infrastructure required to accommodate the Lagrangian framework.
- Chapter 7 describes the software engineering techniques and numerical algorithms used to implement the Lagrangian framework and model itself.
- Chapter 8 covers some miscellaneous topics regarding the implementation of the model.

The second part is concerned with proof of concept, that is, testing the schemes formulated in the first part. It is divided into the following chapters:

- Chapter 9 gives an introduction and literature review of previous idealised studies of overflows and deep convection, as well as some concepts required for the analysis of the test cases.
- Chapter 10 describes the results of experiments using the dynamically passive blobs.

- Chapter 11 describes the results of experiments, including some using the dynamically active free blobs, in the deep convection test case of Jones and Marshall (1993).
- Chapter 12 describes the results of experiments, including some using the dynamically active bottom blobs, in the DOME test case.
- Chapter 13 describes the results of experiments, including some using the dynamically active bottom blobs, in the bowl test case.
- Chapter 14 is the conclusion.

## Chapter 2

# Applications of an Embedded Lagrangian Framework

The resolution of ocean general circulation models (OGCM's) is restricted by the computing power available to modellers. Large scale ocean model studies, with a grid resolution of  $O(10)$ km or larger, may reasonably employ the hydrostatic approximation, which assumes that the buoyancy force and gravity are balanced (see Sections 4.5 and 4.6 of Griffies, 2004). In order to resolve non-hydrostatic dynamics, a grid resolution of less than  $\sim 1$ km is required. In other words, until the horizontal to vertical grid aspect ratio is of order unity, hydrostatic dynamics are by far the most dominant terms in the vertical momentum balance (Marshall et al., 1997b).

Many small scale processes that are not in hydrostatic balance have a non-trivial projection onto larger scales, necessitating the parameterisation of subgrid-scale (SGS) processes in order to take into account their effect on the large scale system. Parameterisations are varied in their approaches, complexity and computational cost, but are necessary for realistic OGCM simulations.

Two climatically important processes that are not in perfect hydrostatic balance are open ocean deep convection and gravity-driven down-slope flows. These two processes have been chosen as applications of the embedded Lagrangian framework.

The rest of this section describes the physics of deep convection and downslope flows, as well as reviewing ocean modelling frameworks and existing SGS parameterisations. Some other potential applications for the embedded Lagrangian framework are also discussed.

### 2.1 Open Ocean Deep Convection

Open ocean deep convection occurs over limited regions of the ocean surface, where air-sea heat fluxes are generally very strong. This process causes the ocean's surface layer to loose buoyancy and become gravitationally unstable, creating a convective mixed layer. If these intense interactions continue for long enough, the mixed layer deepens and eventu-

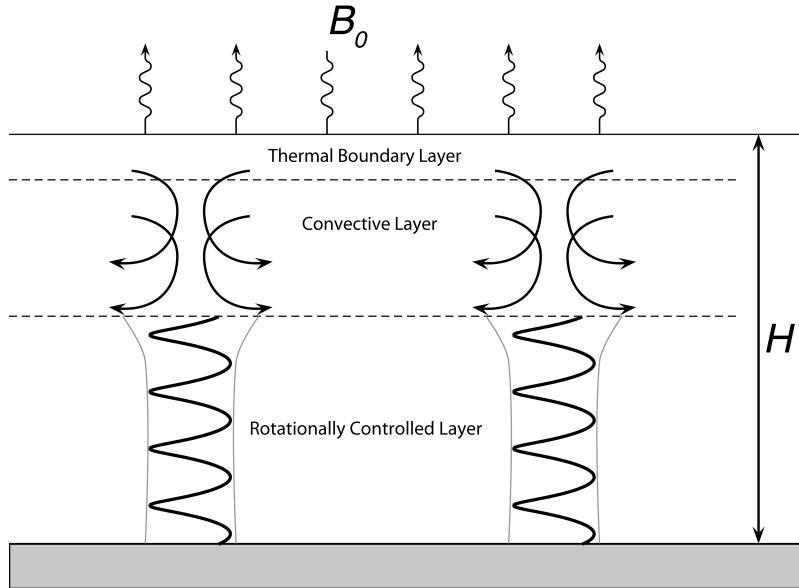


Figure 2.1: The fully developed second stage of open ocean deep convection – where small scale ( $\sim 1 \text{ km}$ ) plumes have developed and extend below the convective layer.  $B_0$  is the buoyancy loss and  $H$  is the depth of the ocean. Adapted from Figure 1 of Jones and Marshall (1993).

ally convective plumes, under the influence of rotation, reach out from the bottom of the convective mixed layer to the depths (Killworth, 1983; Jones and Marshall, 1993; Marshall and Schott, 1999). Open ocean deep convection is illustrated in Figure 2.1.

Deep convection is an intermittent process that involves a number of phases and scales. The first phase is large scale ( $\sim 100 \text{ km}$ ) “preconditioning” where isopycnals dome, bringing dense, weakly stratified waters to the surface. The second phase is the deep convective phase, where small scale ( $\sim 1 \text{ km}$ ) intense plumes sink. The convective plumes have vertical length scales that are the same order as the horizontal scale of motion. The third phase, which often occurs concurrently with the deep convective phase, is a lateral advective exchange between the convective site and the surrounding fluid. The lateral exchange has a scale  $\sim 10 \text{ km}$ .

There are three known major open ocean deep convection sites in the global ocean, namely, the Labrador Sea, the Greenland Sea and the northwestern Mediterranean (Marshall and Schott, 1999). Open ocean deep convection is also known to occur occasionally in the Weddell Sea (Gordon, 1982; Marshall and Schott, 1999). There are features that predispose a region to deep convection. For deep convection to occur, there must be strong buoyancy forcing, and thus, intense surface fluxes. Large sensible heat, latent heat and moisture fluxes induced by cold dry winds blowing over water can significantly contribute to the buoyancy forcing. Thus, deep convective sites tend to be close to a boundary, be it land or ice. Secondly, for deep convective plumes to be formed, the stratification must be relatively weak over that layer. And finally, there must be a mechanism to cause isopyc-

nals to dome. The final requirement is essential for the preconditioning phase mentioned above. The doming of isopycnals brings dense water close to the surface to be exposed to the surface buoyancy forcing. There can be some features of deep convection that are particularly important in a certain region. For instance, it is thought that thermobaric effects are important in the Greenland and Weddell Seas (Marshall and Schott, 1999).

Regions of deep convection are not associated with a large vertical mass flux (Send and Marshall, 1995). While they do communicate surface properties to the deep ocean, and are an important physical process in the meridional overturning circulation, most of the vertical mass flux in the overturning circulation occurs in boundary currents. The mass flux in the boundary currents and deep convection are, however, intimately linked (Spall, 2008).

Deep convection in the ocean is an intrinsically non-hydrostatic process, with relatively large vertical velocities. One thing that sets oceanic deep convection apart from its atmospheric analogue is that it has a slow enough time scale that it is influenced by rotation. The non-trivial vertical velocities, coupled with the timescale of deep convective processes means that the traditional practice of ignoring the horizontal component of rotation is questionable (Wirth and Barnier, 2008). The effect of the horizontal component of the Earth's rotation is to induce "tilted convection" that causes convecting plumes to deflect (Denbo and Skillingstad, 1996; Sheremet, 2004; Wirth and Barnier, 2006). Some process studies indicate that the deflection of plumes as a result of the horizontal component of the Earth's rotation (in addition to horizontal density gradients) can cause convective plumes to penetrate deeper than if a purely "upright convection" occurred (for example Straneo et al., 2002).

## 2.2 Gravity-Driven Downslope Flows

Near boundary convection occurs in marginal seas and near boundaries where the effect of topography can be influential. Intense air-sea fluxes play a large role, and a significant proportion of the near boundary convection that occurs in the ocean is from polynyas around Antarctica (Killworth, 1983).

Modification of dense, sinking waters in the oceanic bottom boundary layer plays an important role in setting the properties of many of the world's deep and bottom waters (Killworth, 1983). Termed "overflows", the waters of gravity-driven downslope flows initially form in reservoirs by brine rejection, evaporation or cooling. The reservoirs are generally marginal seas or coastal shelves. When dense water leaves the reservoir, they generally flow over some topographic feature such as a sill or strait, before flowing down the continental slope. As the waters descend along the continental slope, they are significantly modified by entrainment of water from the bulk ocean (Smith, 1975; Price and Baringer, 1994). The overflows do not merely act under the influence of gravity, but are also subject to geostrophy, bottom drag (Smith, 1975; Killworth, 1977; Price and Baringer,

1994; Özgökmen and Fischer, 2008) and topographic steering (Muench et al., 2009; Ilicak et al., 2011b). Overflow waters, modified by entrainment, play a large part in forming the water-masses of the deep and abyssal ocean, such as North Atlantic Deep Water (Dickson and Brown, 1994) and Antarctic Bottom Water (Gordon et al., 2001). In recent years, there has been considerable effort put into improving our understanding of overflow processes (for instance, Legg et al., 2009)

Due to the limitation on resolution dictated by the present availability of computational resources, ocean climate models used for global studies cannot explicitly resolve the small scale processes that are involved in the formation of deep and bottom water-masses. Nonetheless, a firm understanding of the real world processes should help to guide the modeller when selecting parameterizations, bathymetry and so forth.

Broadly speaking, it is postulated that the various ways that downslope flows occur are as follows (Baines and Condie, 1998):

1. If there is a sufficiently extensive amount of source water at the top of a slope, then the fluid descends the slope as a broad sheet with thickness much greater than the thickness of the Ekman layer. A combination of geostrophic dynamics and a continuous supply of fluid forces the water down the slope.
2. If the source water is not so large, then the descent will either begin as a largely geostrophic descent, but, will only reach intermediate depths, from which continued descent occurs as viscous drainage in the Ekman layer. Under some circumstances, this drainage occurs right from the top of the slope.
3. Submarine canyons and channel flow may act as conduits to greater depths, transporting water (sometimes relatively unaltered by entrainment or detrainment) relatively rapidly.

In the cases of the Weddell and Ross Seas, it would appear as if the main conduit for the very cold and/or salty source waters to reach the continental margin is via submarine canyons. These relatively unaltered waters then spill out onto the continental slope (in some instances, at depths greater than 1500m) and then are altered and mixed as they proceed under the influence of geostrophy. These shelf waters are thought to mix with Circumpolar Deep Water and Antarctic Surface Waters, which then combine to form what is generally described as Antarctic bottom water (AABW; although, the properties of AABW differ from source region to source region which sometimes makes an accurate definition of AABW difficult; Johnson, 2008).

In the case of the Nordic Seas, the main conduits into the North Atlantic is the Faroe Bank Channel and the Denmark Strait. Overflowed waters (modified by entrainment) from the Nordic Seas then make up a large portion of North Atlantic Deep Water (NADW).

### 2.2.1 The Weddell Sea

It is believed that approximately 60% of the flux of AABW comes from the Atlantic sector of AABW production (Orsi et al., 1999), with the majority originating in the Weddell Sea.

Along the margins of the Weddell Sea, from the Fenchner Depression to the tip of the Antarctic Peninsula, there are a number of cold, dense water masses formed including Weddell Sea Deep Water (WSDW; Fahrbach et al., 1995) and Weddell Sea bottom water (WSBW; Carmack and Foster, 1975). These water masses are transported along the eastern flank of the Antarctic Peninsula in the western boundary current whence they turn eastward, some of which flows down the continental margin to form AABW (Gordon et al., 2001). The general bathymetry of the region, as well as the pathways of various water masses is shown in Figure 2.2.

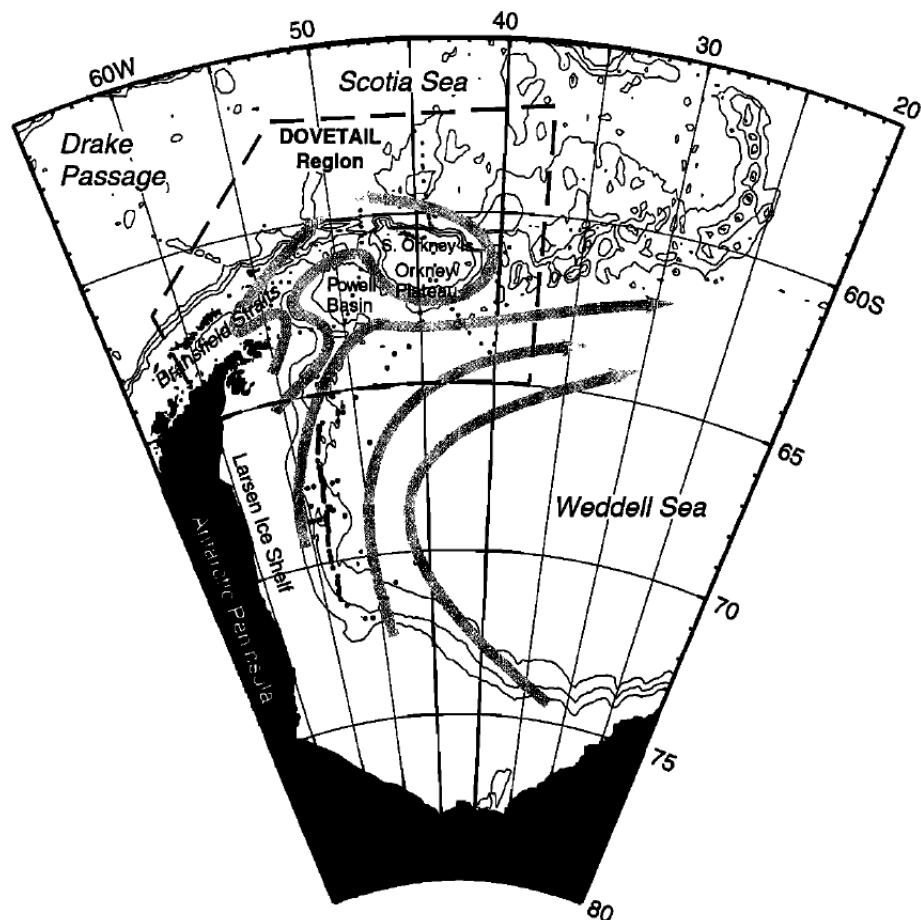


Figure 2.2: The Weddell Sea, taken from Gordon et al. (2001). The 1km, 2km and 3km isobaths are shown. The arrows indicate the direction of bottom water flow.

The main precursor for AABW is WSBW, the majority of which originates as high salinity shelf water. The high salinity shelf water is produced by loss of latent heat through polynyas which are maintained by a combination of offshore winds from the Filchner-Ronne Ice Shelf and tidal divergence at the edge of the same ice shelf.

The appearance of polynyas occur throughout most of the year, and result in large amounts of sea ice production (and hence, large amounts of brine rejection and intense oceanic heat loss). In addition to the polynyas, there are also intense atmosphere-ocean interactions from leads in sea ice to the north of the ice shelf margin.

The high salinity shelf water then takes one of two general paths, either entering the western boundary current, or flowing under the cavity beneath the Filchner-Ronne Ice Shelf. While the water is quite cold, it is still above freezing, and will tend to melt some of the ice on the bottom of the shelf, thus cooling and decreasing in salinity. The eventual fate of this water is to flow out via the Filchner Depression. From there, there are a number of processes which determine how rapidly the water sinks down the slope (versus how rapidly it progresses along the shelf under the influence of the earth's rotation) and how much it is modified by surrounding water masses (such as WSDW; Foldvik et al., 2004).

### 2.2.2 The Ross Sea

High salinity shelf water is produced in the western Ross Sea through broadly similar processes as that produced in the Weddell Sea, that is, intense air-sea interactions through polynyas and leads resulting in cold, salty water. Gordon et al. (2004) describe a particularly salty form of AABW originating in the Ross Sea. The water flows down the Drygalski Trough in the western Ross Sea and rapidly descends the continental slope, entraining lower circumpolar deep water as it descends.

There are two other troughs that provide a conduit for the high salinity shelf water to flow out into the open ocean, being the Glomar Challenger Trough (furthest to the east) and the Joides Trough, which lies between the other two. The water that flows out through the troughs becomes progressively less salty as one goes eastward (Gordon et al., 2009).

The majority of the energetic gravity currents flowing down the continental slope have their origins largely in waters that flow out of one of these three troughs (Gordon et al., 2009). Once onto the continental slope, the waters are mixed with circumpolar deep water and Antarctic surface waters through various processes, including intense tides (Whitworth and Orsi, 2006).

### 2.2.3 The Faroe Bank Channel

The Faroe Bank Channel (see Figure 2.3) is the deepest connection between the North Atlantic and the Nordic Seas, with a sill depth of approximately 850m. The main outlet for the Norwegian Sea is via the Faroe-Shetland Channel (which separates the Faroe Islands from the Shetland Islands. Southward flowing water encounters the Wyville-Thomson Ridge, which turns it westward (although, there can occasionally be water overflowing

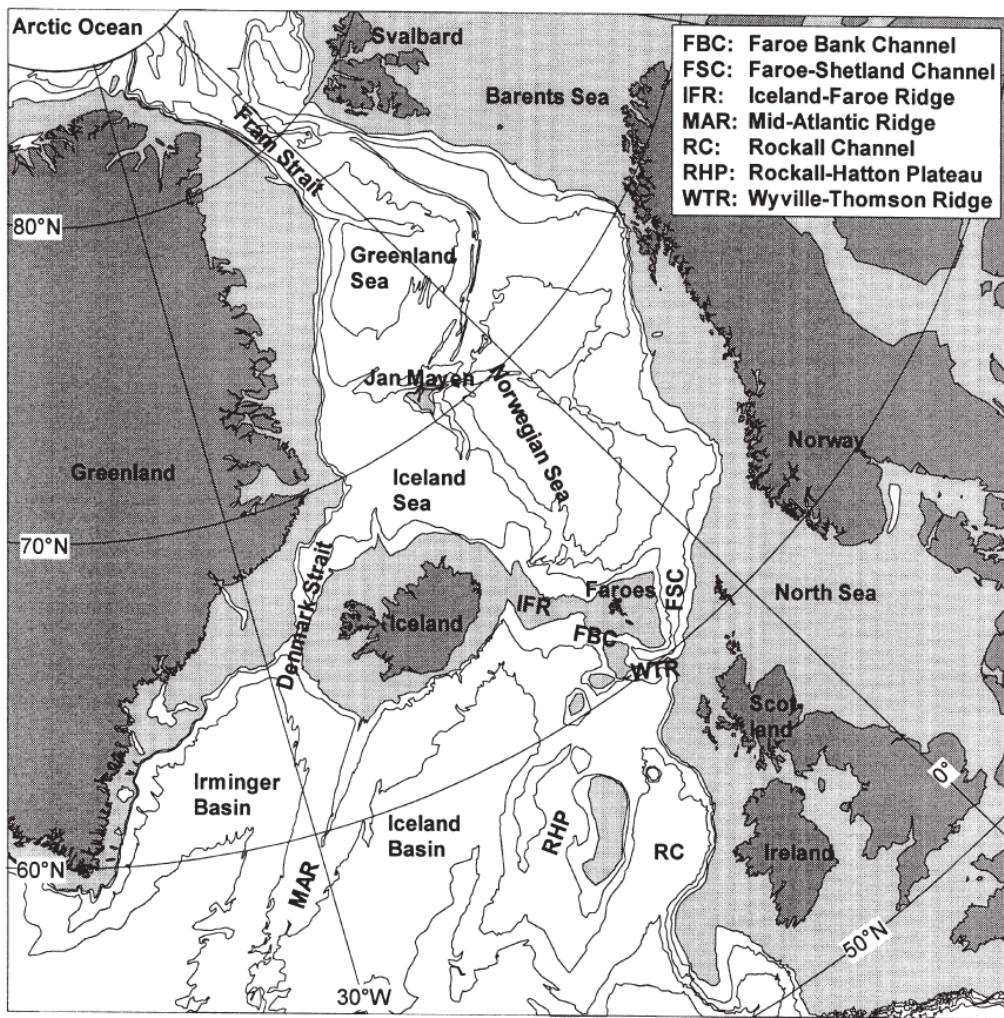


Figure 2.3: A map showing the major bathymetric features of the Nordic Seas and the North Atlantic. From Hansen and Østerhus (2000).

across this ridge). The water is then forced through the Faroe Bank Channel (to the south of the Faroe Islands) and over the sill. After passing the sill, the water continues in a roughly westward direction and spills out at about 2000m into the Iceland Basin. From the Iceland Basin, the water joins the deep western boundary current and eventually ends up at a depth of more than 4000m when it reaches the tropical Atlantic (Hansen and Østerhus, 2000).

Even though the sill in the Faroe Bank Channel is deeper than that of the Denmark Strait, the product waters are significantly warmer and lighter. It is hypothesised that the waters travelling through the Faroe Bank Channel are altered significantly by mixing due to resonant internal waves during the sharp turn when exiting the Faroe-Shetland Channel (Mauritzen et al., 2005).

### 2.2.4 The Denmark Strait

Water from the Iceland Sea flows through the Denmark Strait and into the Irminger Basin (see Figure 2.3). It has a sill at a depth of approximately 600m and carries approximately half of the water that flows from the Nordic seas to the North Atlantic (Girton and Sanford, 2003). The overflow waters undergo significant modification in the first 160km downstream of the sill, but, comparatively little for the subsequent 160km. It is then that the Denmark Strait waters merge with other Nordic overflow waters to form NADW (Dickson and Brown, 1994).

## 2.3 Convection in OGCM's

As the embedded Lagrangian framework is formulated to overcome problems associated with the representation of convection in depth based and pressure based ocean models, the following discussion focuses purely on these two classes of model.

### 2.3.1 Open Ocean Convection

There are a number of open ocean convection schemes presently employed in coarse resolution Eulerian ocean models. Typically, convective adjustment schemes mix both active and passive tracers. The commonly used parameterisations broadly fall into one of two categories,<sup>1</sup>

1. instantaneous homogenisation of an unstable water column, and
2. enhanced vertical diffusivity in unstable water columns.

The former method includes a method implemented by Cox (1984), in which the water column is tested for instability and vertically adjacent grid cells are homogenised. This methodology was subsequently modified by several authors, including Marotzke (1991), Madec et al. (1991) and Rahmstorf (1993). The latter formulations generally decreased computational cost and homogenise the entire unstable water column, whereas the original Cox (1984) scheme was prone to leaving some grid boxes unstable.

The second method is the prescription of an enhanced vertical diffusion in regions of static instability (i.e. the square of the buoyancy frequency is less than zero). Importantly this formulation does not remove the instability instantaneously. The utility of an enhanced vertical diffusivity is motivated by the fact that convection has a finite timescale. It is therefore not unphysical, and indeed it may in fact be desirable for instabilities to be removed over a finite period, rather than instantaneously (Klinger et al., 1996).

---

<sup>1</sup>The homogenisation method can be mathematically considered an extrema of the diffusion method (that is, instantaneous homogenisation is equivalent to having an infinite diffusion coefficient). However, we delineate these two methods as the methodology for numerical implementation is somewhat different.

While it has been shown that these approaches capture much of the bulk effects of open ocean convection (Send and Marshall, 1995), there are several drawbacks that are unsatisfying from a physical perspective.

1. They do not capture the effects of tilted convection (see Section 2.1), and as such, may underestimate the depth of penetration.
2. In many realistic ocean models the deep/bottom water mass properties tend to be too cold and too fresh, with improvements occurring through ad hoc, heuristic methods, for example, the method employed by Duffy and Caldeira (1997).
3. Only tracer concentration is mixed in these schemes and the advection of seawater mass is ignored.
4. The horizontal smallness of the physical process is not captured.
5. In most convective adjustment schemes, momentum is ignored (with the exceptions being the schemes of Palusziewicz and Romea, 1997; Campin et al., 2011)

Whether capturing mass flux is a desirable property of a convection scheme, as is suggested in item (3), warrants some discussion. While the net mass flux in convection is approximately zero and the bulk effect of convection is captured by mixing tracer concentration only (Send and Marshall, 1995; Marshall and Schott, 1999), from a physical process perspective, it is more satisfying if the movement of mass can also be captured, particularly if it can be done so in a computationally efficient manner. It has also recently been suggested by Hughes et al. (2009) that both homogenisation and enhanced vertical mixing do not accurately represent the energetics of convection. The energetics is not represented accurately because convective adjustment acts as a sink for available potential energy without first being converted to kinetic energy. The net result is that convective adjustment schemes rapidly dilute density anomalies, which markedly reduces the available potential energy. Hughes et al. (2009) indicate that caution should be exercised when drawing conclusions about the overturning circulation when using traditional convective adjustment schemes.

A third, less common method is the ocean penetrative plume scheme (OPPS) proposed by Palusziewicz and Romea (1997). In this parameterisation, unsteady water columns are adjusted with the use of a one dimensional plume scheme. An “ensemble” of plumes, whose physics is based on Turner (1973), transport mass and tracer to the deep ocean. This scheme has been shown to improve deep water characteristics, particularly in the Southern Ocean (Kim and Stössel, 2001), but is computationally expensive (Marsland et al., 2003).

A new method has recently been proposed by Campin et al. (2011), in which a two dimensional (vertical slice) non-hydrostatic model is embedded within a three dimensional grid box. Comparisons have been made with non-hydrostatic models and with hydrostatic

models using the traditional convection parameterisations, and it appears that the super-parameterisation shows an improvement in realism, which addresses some of the issues mentioned above.

## 2.4 Bottom Boundary Layer Schemes in OGCM's

Due to the coarse resolution of OGCM's, there is some difficulty in representing bottom flows. Winton et al. (1998) show that bottom boundary layers in level coordinate models are unable to be resolved until the vertical resolution is such that there are a number of points within the bottom boundary layer, and that the horizontal resolution is such that,

$$\Delta x \approx \Delta z / |\partial_x H|, \quad (2.1)$$

where  $\Delta x$  and  $\Delta z$  are, respectively, the horizontal and vertical grid spacing, and  $\partial_x H$  is the slope of the bottom topography,  $H$ . Until those resolution criteria are satisfied, level coordinate models are unable to explicitly represent the dynamical processes that are occurring in overflowing waters.

How dense shelf water makes its way down a slope in a level coordinate model without any parameterisation is illustrated in Figure 2.4. There may be dense water on a shelf and less dense water at the same level off the shelf. This arrangement will cause water to flow off the shelf and into the deep water column at the same level, thereby increasing the density of the deep water column at the depth of the shelf. If the density increases sufficiently that the deep ocean water column becomes unstable, the model then invokes its convective parameterisation, thereby diluting the density signal of the original shelf waters. The resultant dilution of the source waters is generally much greater than the dilution that occurs due to entrainment in the physical system. The over-dilution of the signal generally compromises the integrity of simulated overflow waters.

It is hypothesised that improvements in the representation of overflows in realistic models should significantly improve the properties of deep and bottom waters. Unrealistic AABW formation rates and incorrect AABW properties are ubiquitous in the present generation of ocean and coupled climate models. As an example, Downes et al. (2011) examines the AABW production in different configurations of the GFDL coupled models, and ECCO products. They show for transport across 30S, the highest transport of the models examined is less than half the lowest AABW transport estimates from observations.

There have been a number of attempts to improve the representation of overflows in level models. Examples include a sigma advection-diffusion scheme (Beckmann and Döscher, 1997), an overflow scheme (Campin and Goosse, 1999) and a slab BBL (Killworth and Edwards, 1999; Nakano and Sugino, 2002). Each of these schemes has its own strengths and weaknesses.

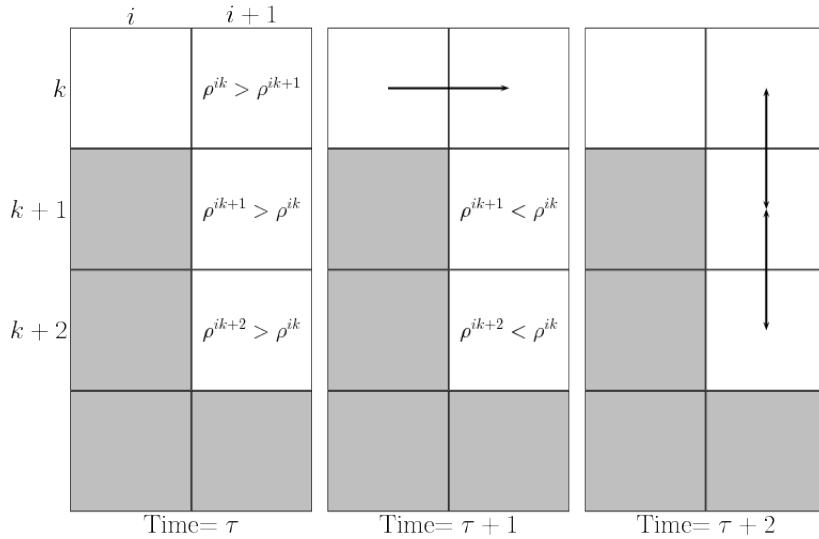


Figure 2.4: The process of dilution of source waters in level models when the bottom boundary layer is not resolved and there is no bottom boundary layer parameterisation. The dense shelf water moves into the deep ocean water column, where it is convectively adjusted, thereby rapidly diluting the density signal.

The Beckmann and Döscher (1997) sigma advection-diffusion scheme creates a terrain following bottom boundary layer that does not capture all instances where an enhanced downslope flow would improve model realism, since this parameterisation only deals with the bottom grid cells. In the instance of a high “cliff”, where the horizontally adjacent cell has a higher density than the on shelf grid cell, but the density of the grid cell at the bottom of the cliff has a density greater than that of the on shelf grid cell, then the sigma advection-diffusion scheme will not prescribe an enhanced downslope flow.

On the other hand, the Campin and Goosse (1999) overflow scheme does transport the dense shelf properties to its neutral depth. This parameterisation is, however, somewhat limited because the BBL thickness is not resolved and as a result, properties rapidly diffuse. Additionally, most implementations of the parameterisation do not allow transport beyond the adjacent column, although, this problem is somewhat overcome in MOM (see Chapter 19.4 of Griffies, 2009).

Both the sigma advection-diffusion scheme, and the overflow scheme deal with the tracer equations only, while the Killworth and Edwards (1999) slab BBL alters both the momentum and tracer equations. The relatively simple scheme of Nakano and Sugino (2002) is prone to non-trivial truncation errors arising from the calculation of the horizontal pressure gradient when using sigma coordinates. The truncation error is particularly large at low latitudes due to the smallness of the Coriolis parameter. Even still, Nakano and Sugino (2002) show significant improvement in water mass properties formed in the northern North Atlantic, the Ross Sea and the Weddell Sea. In addition, the improved representation of overflows reduces spurious density gradients in those regions, which in

turn, reduces spurious effects associated with the Gent and McWilliams (1990) scheme acting on those density gradients.

Bottom boundary layer currents are often important in regions where topography restricts flow (for instance, in the Greenland-Iceland-Norway Basin), with much of the outflow taking place through canyons that the present generation of ocean climate models are unable to resolve. One method widely used in coarse resolution models to capture such outflows is to unrealistically deepen topography. While unrealistically deeper topography may partly overcome the problems associated with not being able to represent outflows on the coarse grid topography, there are a number of undesirable effects associated with such a technique (see, for example, Roberts and Wood, 1997). To maintain a realistic topography while also capturing much of the effect of outflows, Kösters et al. (2005) diagnose the hydraulic transport over overflows from the large-scale conditions and impose a value of the exchange across a sill. They find some improvement in model realism. Born et al. (2009) conduct similar experiments using the same parameterisation and find that there is a large impact on the Atlantic meridional overturning stability to freshwater pulses, as well as a realistic improvement in the sub-polar gyre strength.

Some more recent developments are being incorporated in the present generation of ocean climate models. Modifications to boundary conditions to represent subgrid-scale straits and channels (e.g. the Strait of Gibraltar) include cross-land mixing (Griffies et al., 2005), as well as thin and partial barriers (A.J. Adcroft, personal communication, 2010). Having straits that are artificially widened, or completely closed, spuriously affects the properties of the source waters. Methods for improving the topographic representation of features works well for narrow channels and straits, however, they are unable to be utilised to represent undersea canyons and other subsurface topographic features, such as sea mounts.

Another approach has been implemented in the CCSM3 for the Mediterranean overflow (Wu et al., 2007), and a more sophisticated version in the CCSM4 for the Denmark Strait (Danabasoglu et al., 2010), the Mediterranean, the Weddell Sea and the Ross Sea (Briegleb et al., 2010). The approach is to define a subsurface source region and a number of predefined subsurface injection locations. Water is taken from the source region and instantaneously transported to the injection location. These regions are illustrated for the Nordic Overflows in Figure 2.5. The water may be modified based on the properties of a predefined input region and entrainment region, which are intermediate regions between the source region and the injection locations. The amount of source water transported from the source region, how much the water is modified and the choice of injection location (where the product water is placed) are all based on a physical model and calculated. The procedure requires the modification (raising) of topography (Figure 2.6) in order to stop the grid scale model from carrying out the procedure shown in Figure 2.4.

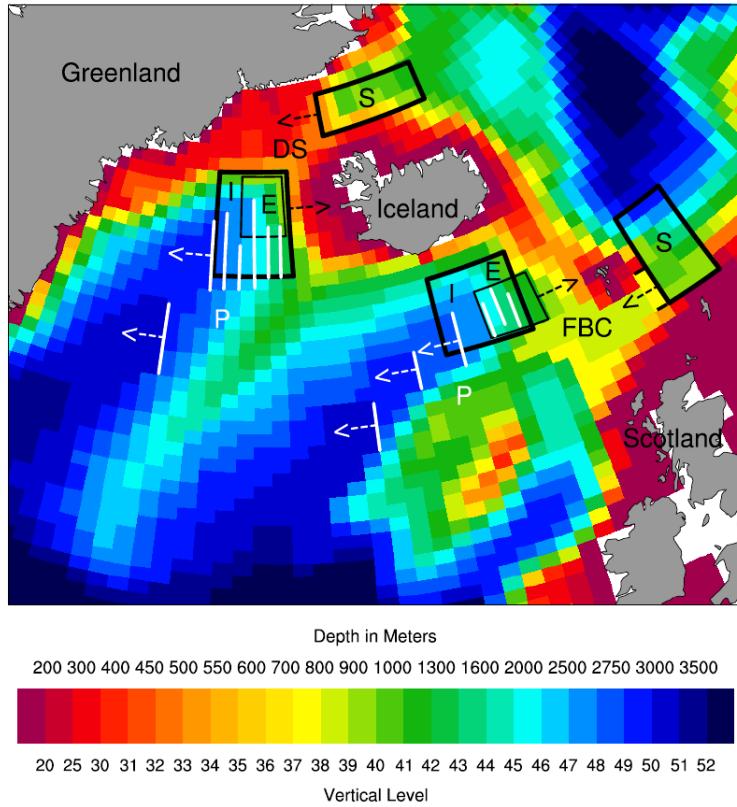


Figure 2.5: The CCSM model set up for the Denmark Strait (DS) and the Faroe Bank Channel (FBC), indicating the source water at sill depth (S), entrainment region (E), interior waters at sill depth (I) and the product water injection site (P) for the Denmark Strait (DS). From Danabasoglu et al. (2010).

The implementation of the above parameterisation in the Nordic Sea overflows has had important non-local effects when implemented in the CCSM. The largest effect has been the removal of a spurious poleward deep western boundary current below about 2600m, which almost universally improves the properties of the North Atlantic (Danabasoglu et al., 2010).

## 2.5 Other Potential Applications of the Lagrangian Framework

As discussed previously, a Lagrangian iceberg model was implemented by Martin and Adcroft (2010) and is another example of how an embedded Lagrangian model can be used. There are many other potential applications of an embedded Lagrangian model, of which a few will be mentioned here.

- One idea is a passive “Argo blob,” that is, a three dimensional drifter that has similar properties to an Argo float. Using much of the infrastructure and techniques developed for the dynamically active blobs, an Argo blob would be a passive (zero mass) blob, whose density is a function of time and varies at some predetermined

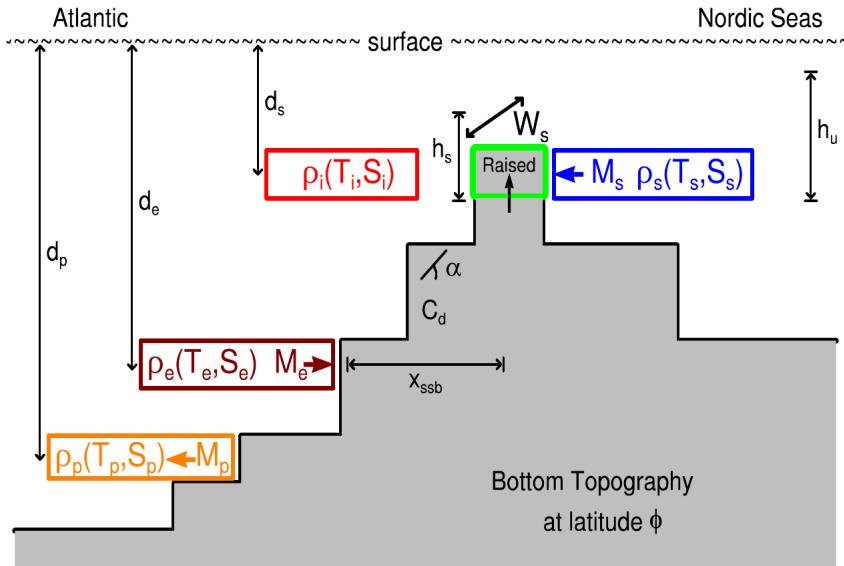


Figure 2.6: The relative depth of where different properties are taken. The subscripts follow from the definitions in Figure 2.5, with  $\rho$  being density,  $T$  temperature,  $S$  salinity and  $M$  being mass. The subscripts refer to the regions defined in Figure 2.5. From Danabasoglu et al. (2010).

rate, similar to that of an Argo float. Diagnostics analogous to that of an Argo float could then be conducted. The use of such blobs would potentially be helpful in comparisons of models with observations. Historical floats could be released in realistic simulations and their trajectories tracked and compared to the trajectory of the real float. Kamenkovich et al. (2011) used Lagrangian drifters to simulate an array of Argo floats, with the floats instantaneously moving between 1500m and the surface. Due to the instantaneous nature of the profiling, the floats only took vertical profiles. One advantage of an Argo blob is that the float would park at a particular density surface, rather than a particular depth (as is the case for real floats). Using the Argo blobs would also give non-vertical profiles as it would take a finite amount of time for the float to travel between the float's neutral density and the surface of the ocean, which is a property of real Argo floats.

- Another approach could be to embed a vortex blob model to represent SGS eddies. Eddy fluxes could then be explicitly calculated and represented with the model and projected back to the Eulerian model (much like the vortex in cell method of Mommadian and Marshall, 2010).
- Another, rather different application would be a Lagrangian gravity wave ray tracing model. Internal gravity waves are thought to be a non-trivial source of non-local mixing, but, their explicit representation is absent from the present generation of ocean climate models. If it is possible to derive a condition for internal gravity wave

creation and destruction based on the resolved flow, a Lagrangian “wave packet” could be used to trace the path of internal gravity waves from a region of gravity wave generation, carrying energy with it and prescribing some sort of enhanced mixing coefficient to parameterise the effect of the wave. There exist some ray tracing methods (e.g. Rainville and Pinkel, 2006) that may be suitable for implementation.

## 2.6 Summary

There are a number of physical processes that are poorly represented in the present generation of ocean climate models. Some of these processes naturally lend themselves to a Lagrangian representation. Two of these processes, open ocean deep convection and gravity driven downslope flows, are chosen in this thesis to be represented and implemented in a Lagrangian model that is embedded into an Eulerian ocean climate model.

Both open ocean deep convection and gravity driven downslope flows have parameterisations of varying complexity that are common in the present generation of ocean climate models. It can be reasonably argued, particularly in the case of overflows, that the parameterisations are lacking many important characteristics.

In the following sections, we shall formulate, implement and demonstrate a range of schemes for the Lagrangian model. The most complex are fully dynamic and interactive with the Eulerian model and attempt to include much of the essential physics, without becoming overly complicated.



# Chapter 3

## Formulation of the Lagrangian Framework

To treat parcels of mass within a grid cell separately, we must firstly derive a theoretical framework that allows for the partitioning of mass and other properties into an Eulerian (E) component and a Lagrangian (L) component. Furthermore we require that the framework is able to transfer properties between the two systems, while maintaining local and global conservation of properties, where appropriate.

For all of the other potential applications of an embedded Lagrangian model mentioned in Section 2.5, much of the following is not necessary as the other applications mentioned do not involve the transport of mass and tracer properties by the Lagrangian model. The manipulations that follow in this chapter relate only to applications where mass is transported, necessitating the partitioning of the total model into E and L components.

### 3.1 What is a Blob?

In Lagrangian computational fluid dynamics literature, the term blob refers to a desingularised point vortex (Chorin, 1978). It is from the votex blob literature that the embedded Lagrangian blob model drew its inspiration, and so, has been named accordingly. A blob, in the present context, refers to a fluid parcel whose properties are concentrated at a single point in space. The prognostic variables of a blob are its mass, tracer content, position in space-time and velocity. A Lagrangian discretisation is chosen because Lagrangian models are generally relatively computationally cheap, which offers an attractive way to represent certain important physical processes without a large increase in cost.

In the context of the technology required by MOM to calculate the thickness of a grid cell using generalised level coordinates, it is required that the L system's contribution to the total thickness be calculated. So, even though all of the properties of a blob are concentrated at a point, they are also assumed to fill a certain volume. The relation between volume and mass can be found using density which is found using an equation of state for seawater (IOC et al., 2010). There is nothing intrinsic in the formulation that requires a

blob to have a particular shape in order to calculate its volume, or other properties. It should be added that one of the advantages of considering all of a blob's properties to reside at a point in space, is that a blob may only reside entirely within one grid, and may not partially reside in two or more. Such an approach significantly reduces the complexity of the algorithm required to admit the embedded Lagrangian model.

As stated above, a blob has its properties concentrated at a discrete point in space and there is thus no assumed shape for the blob. Properties such as mass, volume, tracer content, and so on. all contribute to the total system's properties without having to make any statement regarding the shape of a blob. However, to parameterise certain interactions between the E system and the L system (e.g. entrainment) we may be motivated to prescribe an assumed shape in order to formulate the parameterisation. In the case of an entrainment parameterisation, we want to be able to relate the volume of a blob to the surface area of the blob in order to figure out the mass of the entrained material from an entrainment velocity. The precise details are parameterisation specific and shall be discussed in due course.

## 3.2 Partitioning an Ocean Model

To illustrate the partition, we integrate some arbitrary scalar,  $\Psi$ , over a grid cell. The types of grids under consideration are those that have a fixed horizontal mesh, i.e. the lateral walls are fixed in time and thus the horizontal area is fixed in time. This perspective is encapsulated in the following integral,

$$\iiint_{V_T} \Psi_T dV = \iint_A dA \int_T \Psi_T dz \quad (3.1a)$$

$$\iiint_{V_E} \Psi_E dV + \iiint_{V_L} \Psi_L dV = \iint_A \left( \int_E \Psi_E dz + \int_L \Psi_L dz \right), \quad (3.1b)$$

where T represents the total system, E represents the Eulerian system, L represents the Lagrangian system.  $x$  is the nominal<sup>1</sup> zonal direction and  $y$  is the nominal meridional direction and  $dA = dx dy$  is the horizontal area of an Eulerian cell (which, for any given cell, is fixed in time and space). The value of  $\Psi$  of blobs in a grid cell is thus calculated as

$$\iint_A dA \int_L \Psi_L dz = \sum_{q=1}^Q \Psi^q V^q, \quad (3.2)$$

where  $Q$  is the total number of blobs in the grid cell under consideration, and  $V^q$  is the volume of the  $q^{\text{th}}$  blob.

For brevity, the use of multiple integral signs will be avoided where it is clear that the integral involves more than one dimension. The surface integral symbol  $\oint$  shall be used

---

<sup>1</sup>It is the nominal direction because when using a curvilinear grid, such as the tripolar grid of Murray (1996), the  $x$  direction may not be purely zonal and the  $y$  direction may not be purely meridional.

to indicate surface integrals. For brevity, the horizontal area of a grid cell shall be written as

$$\mathcal{A} = \oint dA. \quad (3.3)$$

For instance Equation (3.1a) can be written as

$$\int_{V_T} \Psi_T dV = \mathcal{A} \int_T \Psi_T dz.$$

### 3.3 Seawater Mass and Tracer Mass

To begin with, we note that seawater mass must be conserved both locally and globally in a non-Boussinesq formulation. The Boussinesq case may be recovered from the non-Boussinesq case by setting the density to a reference density,  $\rho_0$ , everywhere except when coupled to the acceleration due to gravity,  $g$  (Spiegel and Veronis, 1960). The Boussinesq case is volume conserving, rather than mass conserving. We shall only consider the non-Boussinesq case unless the Boussinesq case cannot be easily recovered.

A corollary of seawater mass conservation is that tracer mass must also be conserved locally and globally, and that there exists a compatibility condition between seawater mass conservation and tracer mass conservation (Griffies et al., 2001). The partitioning of tracer mass and seawater mass may be written as,

$$(m\mathcal{C})_T = (m\mathcal{C})_E + (m\mathcal{C})_L \quad (3.4a)$$

$$m_T = m_E + m_L, \quad (3.4b)$$

where  $m$  is seawater mass,  $\mathcal{C}$  is tracer concentration. We define the  $n^{\text{th}}$  tracer's material concentration as,

$$\mathcal{C}_n = \frac{m_n}{m}, \quad (3.5)$$

where  $m$  is the mass of the seawater parcel being considered and  $m_n$  is the mass of the constituent, with units of kilograms (kg). The density of the constituent is defined as,

$$\rho_n = \frac{m_n}{V}, \quad (3.6)$$

where  $V$  is the volume of the seawater parcel being considered, with units of cubic metres ( $\text{m}^3$ ). Seawater density,  $\rho$  is thus defined as the sum of  $N$  constituents,

$$\rho = \sum_{n=1}^N \rho_n. \quad (3.7)$$

Some thermodynamic tracers may be defined in a similar fashion to material tracer concentration as in Equation (3.5). In particular, the use of conservative temperature (Mc-

Dougall, 2003) leads to the relation,

$$\Theta = \frac{\mathcal{H}}{mC_p^0} \quad (3.8)$$

where  $\mathcal{H}$  is the potential enthalpy of the seawater parcel being considered (with units of Joules; J) and  $C_p^0$  is the heat capacity, with units of  $\text{JK}^{-1}\text{kg}^{-1}$ , where K is Kelvin. The effects resulting from the non-linear equation of state for seawater are not discussed because there is no need for special treatment of the equation of state in the embedded Lagrangian model.

The continuous tracer conservation and compatible mass conservation equations are written as,

$$\partial_t(\rho\mathcal{C}) + \nabla \cdot (\rho\mathbf{v}\mathcal{C}) = \rho\mathcal{S}^{(\mathcal{C})} - \nabla \cdot (\rho\mathbf{F}) \quad (3.9a)$$

$$\partial_t\rho + \nabla \cdot (\rho\mathbf{v}) = \rho\mathcal{S}^{(M)}, \quad (3.9b)$$

where  $t$  is time,  $\mathcal{S}^{(\mathcal{C})}$  is a tracer source,  $\mathcal{S}^{(M)}$  is a mass source,  $\mathbf{v} = (u, v, w)$  is the three dimensional velocity,  $\mathbf{F}$  is the subgridscale tracer concentration flux and  $\nabla$  is the three dimensional gradient operator. The compatibility condition means that we may obtain the mass continuity Equation (3.9b) from the tracer continuity Equation (3.9a) by setting the tracer concentration to a uniform constant, noting that subgridscale tracer concentration flux is zero when the tracer is a uniform constant,  $\mathbf{F}(\mathcal{C} = \text{const}) = 0$ . Any partitioning of the total system must respect the compatibility condition.

Application of Green's Theorem (see, for instance, Section 3.31 of Aris, 1962) over a finite volume (e.g. a grid cell) allows us to compute a budget for that volume based on fluxes across the surface of that volume,

$$\partial_t \int_V \Psi dV = \int_V \mathcal{S}^{(\Psi)} dV - \oint_S \hat{\mathbf{n}} \cdot \mathbf{J}^{(\Psi)} dS, \quad (3.10)$$

where  $\mathbf{J}$  is the scalar flux,  $\hat{\mathbf{n}}$  is the outward pointing normal vector to the surface being considered and  $S$  is the surface of the volume,  $V$ , being considered. A more detailed discussion and derivation may be found in Chapter 5 of Griffies (2004).

If we consider the volume,  $V$ , to be a grid cell, and we volume integrate Equation (3.9a), we obtain,

$$\int_V \partial_t(\rho\mathcal{C}) dV = \int_V \rho\mathcal{S}^{(\mathcal{C})} dV - \int_V \nabla \cdot (\rho(\mathbf{v}\mathcal{C} + \mathbf{F})) dV. \quad (3.11)$$

In the grid cells we are considering the side walls (i.e. those that are parallel with gravity) are fixed for all time. The top and bottom faces, however, may vary with time with respect to geopotential surfaces. To take the partial derivative with respect to time outside of the integral (in order to apply Green's Theorem), it would require the application of Leibnitz's rule. For notational convenience, we define the dia-surface velocity component

$w^{(s)} = z_{,s}ds/dt$  (Section 6.7 of Griffies, 2004), where  $\frac{d}{dt}$  is the material derivative, and the comma notation (Aris, 1962) has been used to indicate partial derivatives,  $z_{,s} = \frac{\partial z}{\partial s}$ , and  $s = s(t, x, y, z)$  is a generalised vertical coordinate surface. Use of the dia-surface velocity allows us to take the derivative with respect to time outside of the integrals through an implicit application of Leibnitz's rule,

$$\partial_t \int_V (\rho C) dV = \int_V \rho S^{(C)} dV - \int_V \nabla \cdot \rho [\mathbf{v}^{(s)} C + \mathbf{F}] dV, \quad (3.12)$$

where  $\mathbf{F} = (F^{(x)}, F^{(y)}, F^{(s)})$  and  $\mathbf{v}^{(s)} = (u, v, w^{(s)})$  is the velocity relative to the surface,  $S$ . Applying Green's Theorem, Equation (3.10), to the tracer Equation (3.12) yields,

$$\begin{aligned} \partial_t \int (\rho C) dV &= \int \rho S^{(C)} dV - \oint_{x_1} \rho (u C + F^{(x)}) dy dz + \oint_{x_2} \rho (u C + F^{(x)}) dy dz \\ &\quad - \oint_{y_1} \rho (v C + F^{(y)}) dx dz + \oint_{y_2} \rho (v C + F^{(y)}) dx dz \\ &\quad - \oint_{s_1} \rho (w^{(s)} C + F^{(s)}) dA + \oint_{s_0} \rho (w^{(s)} C + F^{(s)}) dA, \end{aligned} \quad (3.13)$$

where  $x_1$  and  $x_2$  are the nominal western and eastern faces,  $y_1$  and  $y_2$  are the nominal southern and northern faces, and  $s_0$  and  $s_1$  are the bottom and top faces. In Equation (3.13), the first term on the right hand side is a source term and the remaining terms are the convergence due to resolved advection and subgrid-scale fluxes.

Applying the budget, Equation (3.10), to the tracer content of the Lagrangian system, we ask how does the Lagrangian system change the tracer content of the cell that it resides in? We simplify the derivation if we consider blobs to have all of their properties concentrated at a single point, negating any requirement for calculations where a blob may straddle two or more grid cells (see Section 3.1).

For discrete particles (such as blobs), it is sufficient to count the number of blobs entering and leaving a grid cell in order to calculate the change of properties. In the context of a numerical ocean model, we take a time average over some time,  $\Delta\tau$  to find the rate of change of properties of the L system,

$$\partial_t \int_{V_L} (\rho C) dV = \int_{V_L} \rho S^{(C)} dV + \sum_{q=1}^{Q_{(\text{in})}} \frac{(mC)_{(\text{in})}^q}{\Delta\tau} - \sum_{q=1}^{Q_{(\text{out})}} \frac{(mC)_{(\text{out})}^q}{\Delta\tau}, \quad (3.14)$$

where  $Q_{(\text{in})}$  and  $Q_{(\text{out})}$  are the number of blobs entering and leaving a grid cell in the time interval  $\Delta\tau$ .

Using the chain rule, an expression for the rate of change of tracer content may be found

$$\frac{d}{dt} (mC) = C \frac{dm}{dt} + m \frac{dC}{dt}, \quad (3.15)$$

where the first term on the right hand side is the change in tracer content due to a change

in mass of the blob. The second term is the change in tracer content due to a change in the tracer concentration. In the context of the blobs, this arises due to non-conservative tracers, such as age tracers or biogeochemical tracers. In other systems, this term can also represent processes such as diffusion. in the present context all of the exchange of properties between the E and L system occurs via a transfer of seawater mass. Thus, the second term on the right hand side is exclusively associated with non-conservative tracers and is consequently zero for conservative tracers such as conservative temperature and salt.

With regard to the first term in Equation (3.15) the change in tracer content due to the change in mass of a blob may be written as

$$\mathcal{C} \frac{dm}{dt} = \mathcal{C}_E \left( \frac{dm}{dt} \right)_{(E2L)} + \mathcal{C}_L \left( \frac{dm}{dt} \right)_{(L2E)}, \quad (3.16)$$

where the subscript (E2L) indicates the transfer of properties from the E system to the L system and the subscript (L2E) represents the transfer of properties from the L system to the E system. Here, the change of mass is taken from the blob's perspective, thus, the rate of change of mass due to entrainment is positive ( $(\frac{dm}{dt})_{(E2L)} \geq 0$ ) and the rate of change of mass due to detrainment is negative ( $(\frac{dm}{dt})_{(L2E)} \leq 0$ ).

New blobs or blobs that are destroyed also represent a transfer of tracer between the E system and the L system. New blobs represent a tracer sink for the E system and a tracer source for the L system of  $(m\mathcal{C})_{\text{new}}$ , while destroyed blobs represent a tracer source for the E system and a tracer sink for the L system of  $(m\mathcal{C})_{\text{dstry}}$ .

Combining the above gives the following source terms for the E and L systems respectively,

$$\begin{aligned} \mathcal{A} \int_E \rho \mathcal{S}_E^{(\mathcal{C})} dz = & - \sum_{q=1}^{Q_{\text{(new)}}} \underbrace{\frac{(m\mathcal{C})_{\text{(new)}}^q}{\Delta\tau}}_{\text{New blobs}} + \sum_{q=1}^{Q_{\text{(dstry)}}} \underbrace{\frac{(m\mathcal{C})_{\text{(dstry)}}^q}{\Delta\tau}}_{\text{Destroyed blobs}} \\ & - \sum_{q=1}^Q \underbrace{\mathcal{C}_E^q \left( \frac{dm}{dt} \right)_{(E2L)}^q}_{\text{Entrainment}} - \sum_{q=1}^Q \underbrace{\mathcal{C}_L^q \left( \frac{dm}{dt} \right)_{(L2E)}^q}_{\text{Detrainment}} + \underbrace{\int_E \rho \tilde{\mathcal{S}}_E^{(\mathcal{C})} dz}_{\text{Other Sources}} \end{aligned} \quad (3.17a)$$

$$\begin{aligned} \mathcal{A} \int_E \rho \mathcal{S}_L^{(\mathcal{C})} dz = & \sum_{q=1}^{Q_{\text{(new)}}} \underbrace{\frac{(m\mathcal{C})_{\text{(new)}}^q}{\Delta\tau}}_{\text{New blobs}} - \sum_{q=1}^{Q_{\text{(dstry)}}} \underbrace{\frac{(m\mathcal{C})_{\text{(dstry)}}^q}{\Delta\tau}}_{\text{Destroyed blobs}} \\ & + \sum_{q=1}^Q \underbrace{\mathcal{C}_E^q \left( \frac{dm}{dt} \right)_{(E2L)}^q}_{\text{Entrainment}} + \sum_{q=1}^Q \underbrace{\mathcal{C}_L^q \left( \frac{dm}{dt} \right)_{(L2E)}^q}_{\text{Detrainment}} + \sum_{q=1}^Q \underbrace{m^q \left( \frac{d\mathcal{C}}{dt} \right)^q}_{\text{Non-Conservation}} \end{aligned} \quad (3.17b)$$

where we sum over all blobs,  $Q$ , all new blobs,  $Q_{\text{(new)}}$  and all destroyed blobs  $Q_{\text{(dstry)}}$  in the volume under consideration. Here  $\tilde{\mathcal{S}}_E^{(\mathcal{C})}$  is sources for the E system that are not associated with the transfer of properties with the L system.

Importantly, the compatibility condition is satisfied by Equations (3.17) (which can be confirmed by setting  $\mathcal{C} = \text{const}$ ). By summing Equations (3.17) it can be seen that the transfer of properties between the E and L system is consistent and conservative,

$$\int_{V_T} \rho S_T^{(\mathcal{C})} dV = \int_{V_E} \rho \tilde{S}_E^{(\mathcal{C})} dV + \sum_{q=1}^Q m^q \left( \frac{d\mathcal{C}}{dt} \right)^q. \quad (3.18)$$

Furthermore, summing the divergence and source terms for the L system, we can write the semi-discrete equation for the change of tracer content for the L system in a given time interval,  $\Delta\tau$  as

$$\begin{aligned} \mathcal{A} \partial_t \int_L (\rho \mathcal{C})_L dz = & \sum_{q=1}^{Q_{(\text{new})}} \frac{(m\mathcal{C})_{(\text{new})}^q}{\Delta\tau} - \sum_{q=1}^{Q_{(\text{dstry})}} \frac{(m\mathcal{C})_{(\text{dstry})}^q}{\Delta\tau} + \sum_{q=1}^{Q_{(\text{in})}} \frac{(m\mathcal{C})_{(\text{in})}^q}{\Delta\tau} \\ & - \sum_{q=1}^{Q_{(\text{out})}} \frac{(m\mathcal{C})_{(\text{out})}^q}{\Delta\tau} + \sum_{q=1}^Q \left[ C_E \left( \frac{dm}{dt} \right)_{(\text{E2L})}^q + C^q \left( \frac{dm}{dt} \right)_{(\text{L2E})}^q \right] \end{aligned} \quad (3.19)$$

For the E system budget, we write it as shown in Equation (3.13), along with the additional source and sink terms arising from the exchange of properties with the L system, as described by Equation (3.17a).

Summing the E system and L system contributions gives the rate of change of tracer content of the combined system,

$$\begin{aligned} \partial_t \mathcal{A} \int_T (\rho \mathcal{C})_T dz = & - \int_{x_1} \int_E \rho_E (u\mathcal{C} + F^{(x)})_E dz dy + \int_{x_2} \int_E \rho_E (u\mathcal{C} + F^{(x)})_E dz dy \\ & - \int_{y_1} \int_E \rho_E (v\mathcal{C} + F^{(y)})_E dz dx + \int_{y_2} \int_E \rho_E (v\mathcal{C} + F^{(y)})_E dz dx \\ & + \oint_{s_0} \rho_E (w^{(s)}\mathcal{C} + F^{(s)})_E dA - \oint_{s_1} \rho_E (w^{(s)}\mathcal{C} + F^{(s)})_E dA \\ & + \frac{1}{\Delta\tau} \left( \sum_{q=1}^{Q_{(\text{in})}} (m\mathcal{C})_{(\text{in})}^q - \sum_{q=1}^{Q_{(\text{out})}} (m\mathcal{C})_{(\text{out})}^q \right) \\ & + \int_{V_E} \rho \tilde{S}_E^{(\mathcal{C})} dV + \sum_{q=1}^Q m^q \left( \frac{d\mathcal{C}}{dt} \right)^q, \end{aligned} \quad (3.20)$$

where any other source terms not associated with the L system have been ignored. Importantly, Equation (3.20) satisfies the compatibility condition.

### 3.4 Momentum

The equations for the continuum linear momentum per unit volume of a rotating fluid, based on Newton's second and third laws, is given by

$$\rho \frac{d\mathbf{v}}{dt} + \rho(2\boldsymbol{\Omega} + \mathcal{M}\hat{\mathbf{z}}) \wedge \mathbf{v} = \mathcal{S}^{(\mathbf{v})} - \nabla p - \rho g \hat{\mathbf{z}} + \nabla \cdot \boldsymbol{\tau}, \quad (3.21)$$

where where  $\mathcal{S}^{(\mathbf{v})}$  is a momentum source,  $\boldsymbol{\tau}$  is the stress tensor (a second order symmetric tensor),  $p$  is pressure,  $\hat{\mathbf{z}}$  is the vertical unit vector,  $2\boldsymbol{\Omega} = (0, f^*, f)$  is the rotation vector of the earth and  $\mathcal{M}$  is the advection metric frequency. Here  $\Omega = |\boldsymbol{\Omega}|$  is the angular speed of the Earth's rotation,  $f = 2\Omega \sin \phi$  and  $f^* = 2\Omega \cos \phi$ , where  $\phi$  is latitude. For a derivation and discussion of the linear momentum budget, the reader is referred to Chapter 4 of Griffies (2004).

The evolution of linear momentum for an individual blob is given by

$$\frac{d}{dt}(m\dot{\mathbf{x}}) = m\frac{d\dot{\mathbf{x}}}{dt} + \dot{\mathbf{x}}\frac{dm}{dt}, \quad (3.22)$$

where Newton's notation has been used to indicate the material derivative, which helps to emphasise the point-particle like nature of the blobs. In Newton's notation, an overdot indicates the material derivative. Thus the velocity components are  $\dot{\mathbf{x}} = (\dot{x}, \dot{y}, \dot{z})$  and the acceleration components are  $\ddot{\mathbf{x}} = (\ddot{x}, \ddot{y}, \ddot{z})$ . The first term on the right hand side of Equation (3.22) represents the change in momentum due to the change in velocity and the second term is the change in momentum due to the change in mass.

Multiplying the volume of a blob by the continuum linear momentum Equation (3.21) gives the change in momentum of a blob due to the change in velocity

$$m\frac{d\dot{\mathbf{x}}}{dt} = -2\boldsymbol{\Omega} \wedge m\dot{\mathbf{x}} + V\rho_L \mathbf{g} - V\nabla p + V\nabla \cdot \boldsymbol{\tau}, \quad (3.23)$$

where,  $\mathbf{g} = (0, 0, -g)$  is acceleration due to gravity.

The change in momentum of a blob due to the change in mass can be written as

$$\dot{\mathbf{x}}\frac{dm}{dt} = \dot{\mathbf{x}}\frac{d}{dt}m_{(L2E)} + \mathbf{u}_E \frac{d}{dt}m_{(E2L)}, \quad (3.24)$$

where the first term is the mass exchanged from the L system to the E system (e.g. via detrainment) and the second term is the exchange of mass from the E to the L system (e.g. via entrainment). Waters that are transferred from the E system to the L system are done so at the momentum per unit mass of the E system. In the following discussion Equation (3.24) is included via the source term.

Returning to Equation (3.21) and integrating over a grid cell using the standard no-

tation yields

$$\int_V \partial_t(\rho\mathbf{v})dV = \int_V \mathcal{S}^{(\mathbf{v})}dV - \int_V (\mathbf{v} \cdot \nabla)(\rho\mathbf{v})dV + \int_V (\nabla \cdot \boldsymbol{\tau} - \nabla p)dV - \int_V \rho[\hat{\mathbf{z}}g + (2\boldsymbol{\Omega} + \mathcal{M}\hat{\mathbf{z}}) \wedge \mathbf{v}]dV, \quad (3.25)$$

from which a grid cell momentum budget may be obtained. Similarly to mass and tracer, the dia-surface velocity can be used to apply Green's Theorem and formulate a grid cell budget,

$$\partial_t \int_V \rho\mathbf{v}dV = \int_V \mathcal{S}^{(\mathbf{v})}dV - \oint_S (\mathbf{v}^{(s)} \cdot \hat{\mathbf{n}})(\rho\mathbf{v})dS + \oint_S (\hat{\mathbf{n}} \cdot \boldsymbol{\tau} - \hat{\mathbf{n}}p)dS - \int_V \rho[\hat{\mathbf{z}}g + (2\boldsymbol{\Omega} + \mathcal{M}\hat{\mathbf{z}}) \wedge \mathbf{v}]dV. \quad (3.26)$$

Within this budget there is a source term for the transfer between the E and L systems. A blob that is created is a source for the L system and a sink for the E system. The converse is true when a blob is destroyed. The momentum exchange described by Equation (3.24) is also a part of the source terms.

Vertical velocity is not a prognostic variable in the E system and, therefore, there is no vertical component of momentum, consequently the E system's momentum source is purely horizontal,  $\mathcal{S}^{(\mathbf{u})}$ ,

$$\begin{aligned} \mathcal{A} \int_{\Delta\tau} \int_E \mathcal{S}_E^{(\mathbf{u})} dz dt &= - \sum_{q=1}^{Q_{(\text{new})}} [m(\dot{x}, \dot{y})]_{(\text{new})}^q + \sum_{q=1}^{Q_{(\text{dstry})}} [m(\dot{x}, \dot{y})]_{(\text{dstry})}^q \\ &\quad + \sum_{q=1}^Q \left\{ -d[m(\dot{x}, \dot{y})]_{(\text{E2L})}^q + d[m(\dot{x}, \dot{y})]_{(\text{L2E})}^q \right\}. \end{aligned} \quad (3.27a)$$

The L system's vertical velocity is prognostic, and therefore makes up part of the momentum budget,

$$\begin{aligned} \mathcal{A} \int_{\Delta\tau} \int_L \mathcal{S}_L^{(\mathbf{v})} dz dt &= \sum_{q=1}^{Q_{(\text{new})}} [m(\dot{x}, \dot{y}, \dot{z})]_{(\text{new})}^q - \sum_{q=1}^{Q_{(\text{dstry})}} [m(\dot{x}, \dot{y}, \dot{z})]_{(\text{dstry})}^q \\ &\quad + \sum_{q=1}^Q \left\{ d[m(\dot{x}, \dot{y}, \dot{z})]_{(\text{E2L})}^q - d[m(\dot{x}, \dot{y}, \dot{z})]_{(\text{L2E})}^q \right\}. \end{aligned} \quad (3.27b)$$

The second term on the right hand side of Equation (3.26) represents the transport of momentum across a cell boundary. There may be the advection across the cell face, which may arise from a diasurface velocity of the E system or from blobs crossing the surface. Application of the hydrostatic approximation to the E system means that only the horizontal velocities contribute to the momentum budget,

$$-\int_{\Delta\tau} \oint_{S_T} (\mathbf{v}^{(s)} \cdot \hat{\mathbf{n}})(\rho\mathbf{v})_T dS dt = -\int_{\Delta\tau} \oint_{S_E} (\mathbf{v}^{(s)} \cdot \hat{\mathbf{n}})(\rho\mathbf{u})_E dS dt + \sum_{q=1}^{Q_{(\text{in})}} (m\dot{\mathbf{x}})_{(\text{in})}^q - \sum_{q=1}^{Q_{(\text{out})}} (m\dot{\mathbf{x}})_{(\text{out})}^q, \quad (3.28)$$

where  $\mathbf{u} = (u, v)$  is the horizontal velocity vector. The E system's advection velocity term,

the first term on the right hand side of Equation (3.28), may be given by,

$$\begin{aligned} - \oint_{S_E} \hat{\mathbf{n}} \cdot \mathbf{v}^{(s)} (\rho \mathbf{u})_E dS &= \int_E \int_{x_1} u_E (\rho \mathbf{u})_E dy dz - \int_E \int_{x_2} u_E (\rho \mathbf{u})_E dy dz + \int_E \int_{y_1} v_E (\rho \mathbf{u})_E dx dz \\ &\quad - \int_E \int_{y_2} v_E (\rho \mathbf{u})_E dx dz - \oint_{s_0} w^{(s)} (\rho \mathbf{u})_E dA + \oint_{s_1} w^{(s)} (\rho \mathbf{u})_E dA. \end{aligned} \quad (3.29)$$

The change of momentum due to frictional stresses and pressure, the third term on the right hand side of Equation (3.26), may be computed by evaluating the pressure and friction at the surface of the volume,  $V$ . In addition to the friction and pressure on the surface,  $S$ , there may be surface stress and pressure at the interface between a blob and the E system. Recall, however, that all of a blob's properties are conceived to be concentrated at a single point in space (see the discussion in Section 3.1). Therefore, the "surface stress" between the E and L systems must be parameterised. Here, the net force as a result of the parameterised surface stress shall be denoted as  $\mathbf{G}$ . Considering the blobs as point particles simplifies the calculation of pressure gradients considerably, as the combined system's horizontal pressure gradient may be taken as (for more details, see Section 6.3.3),

$$\oint_S \hat{\mathbf{n}} \cdot \boldsymbol{\tau} - \hat{\mathbf{n}} p dS = \oint_{S_E} \hat{\mathbf{n}} \cdot \boldsymbol{\tau}_E - \hat{\mathbf{n}} p dS + \sum_{q=1}^Q [\mathbf{G}^q - V^q (\hat{\mathbf{x}} \partial_x, \hat{\mathbf{y}} \partial_y) p^q + g V^q \rho_E \hat{\mathbf{z}}] , \quad (3.30)$$

where  $(\partial_x, \partial_y) p^q = (\partial_x, \partial_y) p(x^q, y^q, z^q)$  is the hydrostatic pressure gradient at the position of the  $q^{\text{th}}$  blob and  $\hat{\mathbf{x}}$  is the eastward pointing unit vector and  $\hat{\mathbf{y}}$  northward pointing unit vector.

It should be noted that in this formulation, the blobs do not push back against the E system, while they can feel a body force resulting from interfacial drag with the E system. A mechanism by which the momentum of the E system can be altered due to the interfacial drag with a blob could be incorporated using the E system's momentum source term,  $\mathcal{S}_E^{(\mathbf{u})}$ .

The stress and pressure terms for the E system can be expanded using Green's Theorem

$$\begin{aligned} \oint_{S_E} \hat{\mathbf{n}} \cdot \boldsymbol{\tau}_E - \hat{\mathbf{n}} p dS dt &= - \int_E \int_{x_1} (\hat{\mathbf{x}} \cdot \boldsymbol{\tau}_E - p \hat{\mathbf{x}}) dy dz + \int_E \int_{x_2} (\hat{\mathbf{x}} \cdot \boldsymbol{\tau}_E - p \hat{\mathbf{x}}) dy dz \\ &\quad - \int_E \int_{y_1} (\hat{\mathbf{y}} \cdot \boldsymbol{\tau}_E - p \hat{\mathbf{y}}) dx dz + \int_E \int_{y_2} (\hat{\mathbf{y}} \cdot \boldsymbol{\tau}_E - p \hat{\mathbf{y}}) dx dz \\ &\quad + \oint_{s_0} z_s (\nabla \cdot \boldsymbol{\tau}_E - p \nabla s) dA - \oint_{s_1} z_s (\nabla \cdot \boldsymbol{\tau}_E - p \nabla s) dA \end{aligned} \quad (3.31)$$

The final term in Equation (3.26) is made up of gravity, rotation and the advection metric frequency. Eulerian models generally do not include the horizontal component of rotation (in part, due to the hydrostatic approximation), while it may be admitted by the

L system since hydrostatic balance is not generally imposed on the L model, such that

$$\begin{aligned}
-\int_{\Delta\tau} \int_V \rho(g\hat{\mathbf{z}} + (2\boldsymbol{\Omega} + \mathcal{M}\hat{\mathbf{z}}) \wedge \mathbf{v}) dV dt &= -\mathcal{A} \int_{\Delta\tau} \int_E \rho_E g\hat{\mathbf{z}} dz dt - \sum_{q=1}^Q m^q g\hat{\mathbf{z}} \\
&\quad - \mathcal{A} \int_{\Delta\tau} \int_E \rho_E (f + \mathcal{M})\hat{\mathbf{z}} \wedge \mathbf{u}_E dz dt \\
&\quad - \sum_{q=1}^Q m^q (2\boldsymbol{\Omega} + \mathcal{M}\hat{\mathbf{z}}) \wedge \dot{\mathbf{x}}^q. \tag{3.32}
\end{aligned}$$

Under the hydrostatic approximation, the vertical pressure terms involving  $p\nabla s$  in Equation (3.31) are assumed to be balanced perfectly by gravity. Thus, in the E system, terms involving  $p\nabla s$  in Equation (3.31) cancel with the gravity term,  $\rho g\hat{\mathbf{z}}$ , in Equation (3.32).

Combining all of the above, the overall semi-discrete momentum budget for a grid cell can be written as

$$\begin{aligned}
\partial_t \int_{V_T} \rho v dV &= - \oint_{S_E} (\mathbf{v}^{(s)} \cdot \hat{\mathbf{n}})(\rho \mathbf{u})_E dS + \frac{1}{\Delta\tau} \left( \sum_{q=1}^{Q_{(\text{in})}} (m\dot{\mathbf{x}})_E^q - \sum_{q=1}^{Q_{(\text{out})}} (m\dot{\mathbf{x}})_E^q \right) \\
&\quad + \oint_{S_E} \hat{\mathbf{n}} \cdot \boldsymbol{\tau}_E dS + \frac{1}{\Delta\tau} \sum_{q=1}^Q [\mathbf{G}^q - V^q(\hat{\mathbf{x}} \partial_x, \hat{\mathbf{y}} \partial_y) p^q] \\
&\quad + \sum_{q=1}^Q \frac{gV^q \rho_E}{\Delta\tau} \hat{\mathbf{z}} - \sum_{q=1}^Q \frac{g(V\rho)^q}{\Delta\tau} \hat{\mathbf{z}} - \mathcal{A} \int_E \rho_E (f + \mathcal{M})\hat{\mathbf{z}} \wedge \mathbf{u}_E dz \\
&\quad + \oint_{y_1} \hat{\mathbf{x}} p dx dz - \oint_{y_2} \hat{\mathbf{x}} p dx dz + \oint_{x_1} \hat{\mathbf{x}} p dy dz - \oint_{x_2} \hat{\mathbf{x}} p dy dz \\
&\quad - \frac{1}{\Delta\tau} \sum_{q=1}^Q m^q (2\boldsymbol{\Omega} + \mathcal{M}\hat{\mathbf{z}}) \wedge \dot{\mathbf{x}}^q + \sum_{q=1}^{Q_{(\text{new})}} \frac{(m\dot{z})_E^q}{\Delta\tau} - \sum_{q=1}^{Q_{(\text{dstry})}} \frac{(m\dot{z})_E^q}{\Delta\tau} \\
&\quad + \frac{1}{\Delta\tau} \sum_{q=1}^Q \left\{ d[m\dot{z}]_{(\text{E2L})}^q - d[m\dot{z}]_{(\text{L2E})}^q \right\}. \tag{3.33}
\end{aligned}$$

Equation (3.33) indicates that there are significant alterations to the momentum budget of a grid cell, particularly via the inclusion of vertical terms (via the L system) in the momentum budget. In particular the pseudo-nonhydrostatic term,  $\frac{gV^q}{\Delta\tau}(\rho_E - \rho^q)$ , can potentially include a non-trivial vertical component to the momentum budget.

### 3.5 Energy

The energetics of a turbulent stratified fluid may be partitioned into kinetic energy, gravitational potential energy and internal energy, with the gravitational potential energy able to be partitioned into background potential energy and available potential energy (Winters et al., 1995; Hughes et al., 2009). Ocean models, and particularly Boussinesq models, are missing certain energetic conversions (Hughes et al., 2009; Tailleux, 2010), however, anal-

ysis of ocean model energetics remains an important tool. Here, we shall ignore internal energy, as most global scale ocean models do not capture the conversion from turbulent kinetic energy to internal energy, nor do most capture the conversion of internal energy to background potential energy.

Gravitational potential energy per unit mass,  $\Phi$ , and kinetic energy per unit mass,  $\mathcal{K}$  ( $\text{m}^2 \text{s}^{-2}$ ), are given by,

$$\Phi = gz \quad (3.34\text{a})$$

$$\mathcal{K} = \frac{1}{2}\mathbf{v} \cdot \mathbf{v}. \quad (3.34\text{b})$$

We now look at the alterations to the combined system's energy when an embedded Lagrangian blob model is introduced.

### 3.5.1 Gravitational Potential Energy

To examine the evolution of gravitational potential energy, we start by stating the non-Boussinesq continuity equation (see Equation (8.1) of Griffies, 2004),

$$\partial_t \rho + \nabla \cdot (\mathbf{v} \rho) = \rho \mathcal{S}^{(M)}, \quad (3.35)$$

where  $\mathcal{S}^{(M)}$  is a mass source, with units of  $\text{s}^{-1}$ . Multiplying by  $\Phi$ , defined in Equation (3.34a), and integrating over the volume of a grid cell, we obtain,

$$\int_V \Phi \partial_t \rho dV + \int_V \Phi \nabla \cdot (\mathbf{v} \rho) dV = \int_V \Phi \rho \mathcal{S}^{(M)} dV. \quad (3.36)$$

Use of the chain rule yields,

$$\int_V \partial_t (\rho \Phi) dV = \int_V \rho \partial_t \Phi dV - \int_V \Phi \nabla \cdot (\mathbf{v} \rho) dV + \int_V \Phi \rho \mathcal{S}^{(M)} dV \quad (3.37)$$

Using Leibnitz's rule in conjunction with the diastrophic velocity,  $w^{(s)}$ , and applying Green's Theorem yields,

$$\partial_t \int_V \rho \Phi dV = \int_V \rho \partial_t \Phi dV - \oint_S \Phi \hat{\mathbf{n}} \cdot (\mathbf{v} \rho) dS + \int_V \Phi \rho \mathcal{S}^{(M)} dV. \quad (3.38)$$

As we did previously, we focus on the source term for the E and L systems.

The L system will have a change in potential energy given by a source term (the exchange with the E system), the convergence of blobs in/out of the grid cell, and, the change of potential of blobs that already reside in the grid cell. We express this mathematically

as,

$$\int_{\Delta\tau} \int_V \rho \Phi dV dt = \int_{\Delta\tau} \int_V \Phi \rho \mathcal{S}^{(M)} dV dt + \sum_{q=1}^Q (m d\Phi)^q + \sum_{q=1}^{Q_{(\text{in})}} (m\Phi)_{(\text{in})}^q - \sum_{q=1}^{Q_{(\text{out})}} (m\Phi)_{(\text{out})}^q, \quad (3.39)$$

where  $d\Phi$  is the change in potential of a blob over the time period  $\Delta\tau$ . Focussing on the source term, we note that new blobs and the transfer of mass from the E to the L system represent potential energy sinks for the E system and sources for the L system. Conversely, destroyed blobs and the transfer of mass from the L to the E system represents a potential energy source for the E system and a sink for the L system,

$$\int_{\Delta\tau} \int_{V_E} (\Phi \rho \mathcal{S}^{(M)})_E dV dt = - \sum_{q=1}^{Q_{(\text{new})}} [m\Phi]^q + \sum_{q=1}^Q \Phi^q \left\{ -(dm)_{(E2L)}^q + (dm)_{(L2E)}^q \right\} + \sum_{q=1}^{Q_{(\text{dstry})}} [m\Phi]^q \quad (3.40a)$$

$$\int_{\Delta\tau} \int_{V_L} (\Phi \rho \mathcal{S}^{(M)})_L dV dt = \sum_{q=1}^{Q_{(\text{new})}} [m\Phi]^q + \sum_{q=1}^Q \Phi^q \left\{ (dm)_{(E2L)}^q - (dm)_{(L2E)}^q \right\} - \sum_{q=1}^{Q_{(\text{dstry})}} [m\Phi]^q \quad (3.40b)$$

Combining Equations (3.38) to (3.40) and including specific labels for the E and L system components, we obtain the semi-discrete equation,

$$\begin{aligned} \partial_t \int_{V_T} (\rho\Phi)_T dV &= \int_{V_E} \rho_E \partial_t \Phi_E dV - \oint_{S_E} \Phi_E \hat{\mathbf{n}} \cdot (\mathbf{v}_E \rho_E) dS \\ &\quad + \sum_{q=1}^Q (m \partial_t \Phi)^q + \frac{1}{\Delta\tau} \left( \sum_{q=1}^{Q_{(\text{in})}} (m\Phi)_{(\text{in})}^q - \sum_{q=1}^{Q_{(\text{out})}} (m\Phi)_{(\text{out})}^q \right). \end{aligned} \quad (3.41)$$

Similarly to previous results for tracer and momentum, we can see that there are additional, non-trivial terms in Equation (3.41) that are not present in a purely Eulerian system.

### 3.5.2 Kinetic Energy

The Eulerian model is assumed to be in hydrostatic balance. The relevant formulation for the kinetic energy per unit mass of a hydrostatic fluid is given by (Griffies, 2004)

$$\mathcal{K}^{(h)} = \mathbf{u}^2 / 2. \quad (3.42)$$

The justification for ignoring the vertical component of velocity arises because the Hamiltonian of a hydrostatic fluid has kinetic energy only associated with horizontal motions (Bokhove, 2000).

The Lagrangian model, on the other hand, is not hydrostatic and the vertical component of the velocity cannot be ignored, so the form of the kinetic energy per unit volume uses the full three dimensional velocity,  $\mathbf{v}$ , as per Equation (3.34b). It is this difference

in the formulation of the measurement of kinetic energy that requires a slightly different approach to the formulation and analysis of kinetic energy. Rather than starting with the full system and partitioning it into an E and L system and then studying the two systems individually, the two systems shall instead be studied individually and combined at the end to show the full system.

Starting with the E system's kinetic energy, the hydrostatic momentum equation (see Section 5.3.1 of Griffies, 2004) is

$$\partial_t(\rho\mathbf{u}) + (\mathbf{v} \cdot \nabla)(\rho\mathbf{u}) + (f + \mathcal{M})\hat{\mathbf{z}} \wedge (\rho\mathbf{u}) = -\nabla p + \rho\mathbf{F}^{(\mathbf{u})} + \mathcal{S}^{(\mathbf{u})}, \quad (3.43)$$

where  $\mathbf{F}^{(\mathbf{u})}$  is the horizontal frictional force. Taking the dot product of the horizontal velocity with this momentum equation, the evolution equation for the hydrostatic kinetic energy per unit volume is obtained

$$\partial_t(\rho\mathcal{K}^{(h)}) + (\mathbf{v} \cdot \nabla)(\rho\mathcal{K}^{(h)}) + \mathcal{K}^{(h)}\nabla \cdot (\rho\mathbf{v}) = \mathbf{u} \cdot (-\nabla p + \rho\mathbf{F}^{(\mathbf{u})}) + \mathcal{S}^{(\mathcal{K})}. \quad (3.44)$$

The third term on the left hand side is due to the compressibility of seawater, and, in a Boussinesq system  $\nabla \cdot (\rho_0\mathbf{v}) \equiv 0$ . Section 10.1 of Griffies (2009) conducts manipulations of this equation to show, for instance, how kinetic energy is converted to and from other forms of energy. Here, the focus is on how kinetic energy is transferred between the E and L systems.

Integrating Equation (3.44), and using the identity (from Equation (10.3) of Griffies, 2009),

$$\int_V \mathbf{u} \cdot \nabla p = \oint_S p(\hat{\mathbf{n}} \cdot \mathbf{v})dS + \int_V \rho \frac{d\Phi}{dt} dV \quad (3.45)$$

gives an expression for the kinetic energy budget of a grid cell for the E system,

$$\begin{aligned} \partial_t \int_{V_E} \rho_E \mathcal{K}_E^{(h)} dV &= - \oint_{S_E} (\mathbf{v}_E \cdot \hat{\mathbf{n}}) \rho_E \mathcal{K}_E^{(h)} dS - \int_{V_E} \mathcal{K}_E^{(h)} \nabla \cdot (\rho \mathbf{v})_E dV - \oint_{S_E} p(\hat{\mathbf{n}} \cdot \mathbf{v}_E) dS \\ &\quad - \int_{V_E} \rho_E \frac{d\Phi_E}{dt} dV + \int_{V_E} \rho_E \mathbf{u} \cdot \mathbf{F}^{(\mathbf{u})}_E dV + \int_{V_E} \mathcal{S}_E^{(\mathcal{K})} dV \end{aligned} \quad (3.46)$$

where Green's Theorem and the diastrophic velocity has been used to reach this result.

An individual blob's kinetic energy is given by

$$(m\mathcal{K})^q = \frac{1}{2} m \dot{\mathbf{x}} \cdot \dot{\mathbf{x}}, \quad (3.47)$$

and the rate of change of kinetic energy is given by,

$$\frac{d}{dt}(m\mathcal{K})^q = m \frac{d\mathcal{K}}{dt} + \mathcal{K} \frac{dm}{dt}. \quad (3.48)$$

For the L system, it is known that the change in kinetic energy must be equal to the

change in kinetic energy of blobs that reside in that grid cell vis-a-vis Equation (3.48), less those that leave, plus those that enter, so that

$$\int_{\Delta\tau} \partial_t \int_{V_L} \rho \mathcal{K} dV dt = \int_{\Delta\tau} \sum_{q=1}^Q (m \frac{d}{dt} \mathcal{K})^q dt + \sum_{q=1}^{Q_{(in)}} (m \mathcal{K})^q - \sum_{q=1}^{Q_{(out)}} (m \mathcal{K})^q + \int_{\Delta\tau} \int_{V_L} \mathcal{S}^{(\mathcal{K})} dV dt. \quad (3.49)$$

The second term on the right hand side of Equation (3.48) is included in the source term of Equation (3.49). To find the evolution of kinetic energy of an individual blob, the dot product of the velocity with the blob momentum Equation (3.23) is taken

$$m \frac{d}{dt} \mathcal{K} = -m \dot{z} g - \dot{\mathbf{x}} \cdot V (\nabla p + \nabla \cdot \boldsymbol{\tau}). \quad (3.50)$$

Noting that the vertical pressure gradient is that of the E system (and thus the hydrostatic approximation,  $\partial_z p = -g\rho_E$ ), and that the interfacial force is parameterised as  $\rho \mathbf{G}$ ,

$$m \frac{d}{dt} \mathcal{K} = g V \dot{z} (\rho_E - \rho_L) - \dot{x} V \partial_x p - \dot{y} V \partial_y p + \dot{\mathbf{x}} \cdot (m \mathbf{G}). \quad (3.51)$$

The final piece of the puzzle is the source terms.

The source and sink terms for kinetic energy account for the energy in the mass transferred between the two systems. The kinetic energy source term acts as a sink for the E system when kinetic energy is transferred from the E system to the L system. This can be when a blob is formed, or, via some process such as entrainment. Similarly, when a blob is destroyed, or, momentum is transferred from the L system to the E system (e.g. via detrainment), it acts as a kinetic energy source term for the E system. We express this mathematically as

$$\int_{V_E} \mathcal{S}_E^{(\mathcal{K})} dV = - \sum_{q=1}^{Q_{(new)}} [m(\dot{x}^2 + \dot{y}^2)]_{new}^q + \sum_{q=1}^Q [m(\dot{x}^2 + \dot{y}^2)]_{(L2E)}^q - \sum_{q=1}^Q [m(\dot{x}^2 + \dot{y}^2)]_{(E2L)}^q + \sum_{q=1}^{Q_{(dstry)}} [m(\dot{x}^2 + \dot{y}^2)]_{dstry}^q, \quad (3.52a)$$

noting that there is no contribution from the vertical component of velocity because the E system is hydrostatic. In contrast, the L system is pseudo-non-hydrostatic, and thus the vertical component of velocity is included,

$$\int_{V_L} \mathcal{S}_L^{(\mathcal{K})} dV = + \sum_{q=1}^{Q_{(new)}} [m\dot{x}^2]_{new}^q - \sum_{q=1}^Q [m\dot{x}^2]_{(L2E)}^q + \sum_{q=1}^Q [m\dot{x}^2]_{(E2L)}^q - \sum_{q=1}^{Q_{(dstry)}} [m\dot{x}^2]_{dstry}^q, \quad (3.52b)$$

We may now combine Equations (3.46) and (3.49), substituting in the appropriate expressions to find the semi-discrete equation for the total kinetic energy budget of the combined E and L systems,

$$\begin{aligned} \partial_t \int_{V_T} (\rho \mathcal{K})_T dV &= - \oint_{S_E} (\mathbf{u}_E \cdot \hat{\mathbf{n}}) \rho_E \mathcal{K}_E^{(h)} dS - \int_{V_E} \mathcal{K}_E^{(h)} \nabla \cdot (\rho \mathbf{v})_E dV - \oint_{S_E} p (\hat{\mathbf{n}} \cdot \mathbf{v}_E) dS - \int_{V_E} \rho_E \frac{d\Phi_E}{dt} dV \\ &\quad + \int_{V_E} (\rho \mathbf{u} \cdot \mathbf{F}^{(u)})_E dV + \frac{1}{\Delta\tau} \sum_{q=1}^Q \left\{ V^q \left[ g \dot{z} (\rho_E - \rho_L) - \dot{x} \partial_x p - \dot{y} \partial_y p + \dot{\mathbf{x}} \cdot \rho_L \mathbf{G} \right]^q - [m\dot{z}^2]_{(L2E)}^q \right\} \\ &\quad + \frac{1}{\Delta\tau} \left\{ \sum_{q=1}^{Q_{(in)}} (m\dot{x}^2)^q - \sum_{q=1}^{Q_{(out)}} (m\dot{x}^2)^q + \sum_{q=1}^{Q_{(new)}} [m\dot{x}^2]_{new}^q - \sum_{q=1}^{Q_{(dstry)}} [m\dot{x}^2]_{dstry}^q \right\}. \end{aligned} \quad (3.53)$$

As can be seen, there are significant alterations to the kinetic energy budget of the total

system, as when compared to a purely Eulerian system. Most significantly, there is now the addition of terms involving the vertical velocity.

### 3.6 Summary

The term “blob” is derived from the term vortex blob, in the Lagrangian computational fluid dynamics literature. In the present applications, where the blobs we are working with transport mass, we are required to partition the mass of the ocean model into an Eulerian (E) system and a Lagrangian (L) system.

How properties are transferred between those two systems needs to be well defined in order for the total system to conserve tracer mass, seawater mass, momentum and energy. This is particularly true for mass and tracer. Experience has shown that if careful attention is not paid to the mass and tracer budgets, the system very easily becomes locally and/or globally non-conservative.

A derivation of momentum and mechanical energy budgets has been presented. Equations that describe the evolution of momentum and energy under a combined E-L system were shown. It is important to note, however, that there are some differences between the theory presented here and the implementation, as will be described later in Section 6.2.4.

In Section 6.2 we examine the implementation of the above theory in MOM. All of the terms in the mass and tracer budgets are implemented to the budgets in MOM, however, as will be discussed there are a number that are neglected in the momentum and energetic budgets.

## Chapter 4

# Dynamically Passive Schemes

The first step in the implementation of the Lagrangian blob framework is to recover existing deep convection and overflow schemes within the Lagrangian framework. It is important to note that it is not possible to bitwise recover existing schemes, as the fundamental framework (and therefore the order of operations) is different.

Dynamically passive schemes do not directly affect the momentum budget of the model, while dynamically active schemes do (i.e. dynamically passive blobs do not have momentum or velocity, they are created moved and destroyed instantaneously). The dynamically passive schemes are designed to emulate existing Eulerian schemes. The emulation of existing schemes serves two main purposes. First, as a developmental step for the implementation of the embedded Lagrangian framework. Second as a conceptual tool to show how the framework works without the additional complexity of active dynamics.

Since the dynamically passive schemes are not developed for their physical integrity but rather as a development tool, the fidelity of their physics is not considered in detail. The dynamically active schemes introduce more realistic physics, and are a more complete representation of the processes under examination and therefore more time is spent in Chapter 5 on discussing detailed physics of the schemes being developed.

There is little to be gained for the effort of recovering schemes that are based not only on a varying diffusivity, but also on the second derivative of the tracer concentration. As such, schemes like the KPP scheme of Large et al. (1994), or the sigma advection-diffusion scheme of Beckmann and Döscher (1997), are not included.

Not all of the schemes presented in this chapter have been implemented in MOM.

### 4.1 An “NCon”-like Exchange of Mass

The original “NCon” scheme was implemented by Cox (1984). The basic principle of the scheme is that if the upper of two vertically adjacent grid cells is more dense than the lower, the tracer properties of the two grid cells are homogenised. By iteration, a deep

mixed layer may result when a surface dense water cell cascades, grid-box by grid-box, to significant depth. This general principle can be recovered with the Lagrangian scheme. The procedure being that the two adjacent grid cells instantaneously do the following.

1. Create Lagrangian blobs, that is seawater mass and tracer is transferred to the Lagrangian model.
2. The blobs positions are swapped, that is, the blob from the lower grid cell is placed in the upper grid cell, and vice versa.
3. The mass and tracer is transferred from the Lagrangian model back to the Eulerian model.

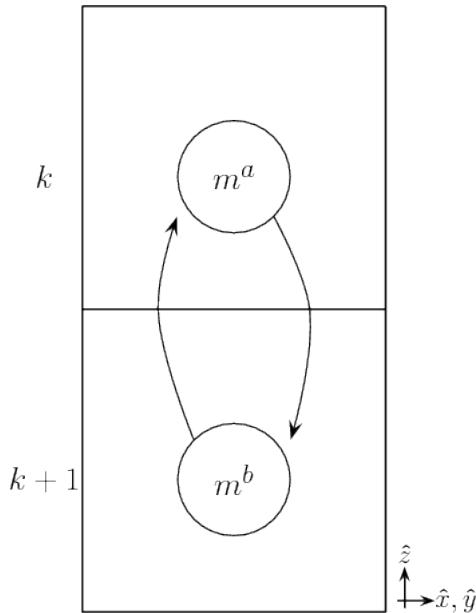


Figure 4.1: Illustration of the transfer of mass used to derive the Lagrangian “NCon”-like scheme. If the density of gridbox  $k$  is greater than that of gridbox  $k + 1$ . Seawater mass and tracer mass is transferred from grid cell  $k$  to the Lagrangian blob  $a$  and from grid cell  $k + 1$  to the Lagrangian blob  $b$ . The blobs are then instantaneously transported from one cell to the other, where the seawater mass and tracer mass is transferred back to the Eulerian model, thus homogenising the properties of the two grid cells.

It is instructive to derive the mass of these blobs that is exchanged. Firstly, we assume mass conservation

$$m_i^k + m_i^{k+1} = m_f^k + m_f^{k+1}, \quad (4.1)$$

where superscripts  $k$  and  $k + 1$  are upper and lower adjacent grid cells and subscripts  $i$  and  $f$  indicate initial and final states with  $m$  being the mass of the respective grid cell at the respective time. We also assume that the total tracer content is conserved so that

$$m_i^k C_i^k + m_i^{k+1} C_i^{k+1} = m_f^k C_f^k + m_f^{k+1} C_f^{k+1}, \quad (4.2)$$

where  $C$  is the tracer concentration. See Section 3.3 for a definition of tracer concentration, including the treatment of the thermodynamic tracer, conservative temperature.

With reference to the schematic in Figure 4.1, we can see that the final tracer content of each grid cell will be the initial content, less that transported out of the cell by the Lagrangian scheme, plus that which is transported into the cell by the Lagrangian scheme,

$$m_f^k \mathcal{C}_f^k = m_i^k \mathcal{C}_i^k - m^a \mathcal{C}_i^k + m^b \mathcal{C}_i^{k+1} \quad (4.3a)$$

$$m_f^{k+1} \mathcal{C}_f^{k+1} = m_i^{k+1} \mathcal{C}_i^{k+1} + m^a \mathcal{C}_i^k - m^b \mathcal{C}_i^{k+1}. \quad (4.3b)$$

It is necessary for the initial and final mass of each cell to be constant due to the order of operations. Since surface forcing is usually a major source of vertical instability, convective adjustment schemes should follow the surface forcing. The surface forcing is often applied implicitly in time. The calculation of the mass of a grid cell (in MOM, this is equivalent to calculating the density weighted thickness) is done explicitly in time. Thus, convective adjustment is often applied after the mass of grid cells has been calculated. So as not to create any unnecessary additional computational burden (to recalculate cell thicknesses), it is desirable for cells to have a final mass equal to their initial mass when convectively adjusting the cells. Furthermore, as discussed in Section 2.1, the integral effect of open ocean deep convection is approximately a zero net mass flux (Send and Marshall, 1995), which means that from a physical perspective, it is desirable for the mass of grid cells to remain constant and for the mass of blobs to be equal. If the mass in each grid cell remains constant, the mass transferred between the two cells must be equal,

$$m_i^k = m_f^k \quad (4.4a)$$

$$m_i^{k+1} = m_f^{k+1} \quad (4.4b)$$

$$\Rightarrow m^a = m^b. \quad (4.4c)$$

Dropping the subscripts for the grid cell mass, and substituting  $m^b$  with  $m^a$  we get the following expressions for the final tracer concentration from Equations (4.3a) and (4.3b),

$$\mathcal{C}_f^k = \mathcal{C}_i^k - \mathcal{C}_i^k \frac{m^a}{m^k} + \mathcal{C}_i^{k+1} \frac{m^a}{m^k} \quad (4.5a)$$

$$\mathcal{C}_f^{k+1} = \mathcal{C}_i^{k+1} + \mathcal{C}_i^k \frac{m^a}{m^{k+1}} - \mathcal{C}_i^{k+1} \frac{m^a}{m^{k+1}}. \quad (4.5b)$$

After the swapping occurs, we want the tracer concentrations to be homogenised between the two grid cells

$$\mathcal{C}_f^k = \mathcal{C}_f^{k+1} \quad (4.6)$$

We therefore set Equations (4.5) to be equal, eliminating  $\mathcal{C}_f^k$  and  $\mathcal{C}_f^{k+1}$  and solve for  $m^a$ ,

$$\mathcal{C}_i^k - \mathcal{C}_i^k \frac{m^a}{m^k} + \mathcal{C}_i^{k+1} \frac{m^a}{m^k} = \mathcal{C}_i^{k+1} + \mathcal{C}_i^k \frac{m^a}{m^{k+1}} - \mathcal{C}_i^{k+1} \frac{m^a}{m^{k+1}} \quad (4.7a)$$

$$\frac{m^a}{m^k} (\mathcal{C}_i^{k+1} - \mathcal{C}_i^k) + \frac{m^a}{m^{k+1}} (\mathcal{C}_i^{k+1} - \mathcal{C}_i^k) = (\mathcal{C}_i^{k+1} - \mathcal{C}_i^k) \quad (4.7b)$$

$$\Rightarrow m^a = \frac{m^k m^{k+1}}{m^k + m^{k+1}}. \quad (4.7c)$$

Thus, the mass transferred from gridbox  $k$  to  $k+1$  by blob  $a$ , and the return mass transferred from gridbox  $k+1$  to gridbox  $k$  by blob  $b$  is given by Equation (4.7c).

The result in Equation (4.7c) is the exchange of mass required to achieve the equivalent tracer concentrations obtained by the original NCon scheme. The net result from the exchange of mass is a mass weighted average of the two cell's tracer concentrations. It can also be seen that, in the limit where  $m^k = m^{k+1}$  the mass transferred from one cell to another is equal to half the mass of a cell.

## 4.2 A “Diffusion-like” Scheme

The purpose of the diffusion-like scheme is not to recover an existing scheme, but instead to illustrate in a relatively simple manner how a diffusive process may be represented within the Lagrangian framework. When reading this section, it is important to recall that the vertical transport of mass and tracer effected by the scheme is done by the Lagrangian system.

In the diffusion-like scheme the same procedure is performed as for the NCon-like scheme (Section 4.1 and Figure 4.1), however, we use a different formulation for calculating the mass of the blobs. We seek a formulation where the mass of the blobs is proportional to the density difference between adjoining grid cells, and will act to restabilise the water column over a finite time, rather than instantaneously.

In order to achieve this goal we firstly assume mass conservation,

$$\rho_i^k V_i^k + \rho_i^{k+1} V_i^{k+1} = \rho_f^k V_f^k + \rho_f^{k+1} V_f^{k+1}. \quad (4.8)$$

If we assume that the mass of blob  $a$  is proportional to the difference in density between the two grid cells, then

$$m^a = V_i^k (\rho_i^k - \rho_i^{k+1}),$$

where  $m^a$  indicates the mass of blob  $a$ . We add a non-dimensional scaling parameter  $\gamma$  in front of the right hand side (the reason for doing so is discussed below)

$$m^a = \gamma V_i^k (\rho_i^k - \rho_i^{k+1}). \quad (4.9a)$$

Assuming that  $m^b = m^a$  it can be seen that the mass of blob  $b$  is

$$m^b = \gamma V_i^k (\rho_i^k - \rho_i^{k+1}). \quad (4.9b)$$

We can write down the final tracer concentration for each grid cell, recalling that the initial and final mass of the grid cell are equal, and that the mass of blobs  $a$  and  $b$  are also equal,

$$C_f^k = C_i^k + \frac{m^a}{\rho_f^k V_f^k} (C_i^{k+1} - C_i^k) \quad (4.10a)$$

$$C_f^{k+1} = C_i^{k+1} + \frac{m^a}{\rho_f^{k+1} V_f^{k+1}} (C_i^k - C_i^{k+1}). \quad (4.10b)$$

We have not constrained the tracer concentration to be homogenised in this scheme, however, we can see from the second term in each of the above equations, the concentration will tend towards homogenisation as  $t \rightarrow \infty$ . The rate of convergence of the tracer will largely depend on the chosen value of  $\gamma$ .

A calculation is presented in Appendix A with  $\gamma = 1$  showing that the mass of the blob,  $m^a$  represents a very small fraction of the entire grid box – only  $O(0.1\%)$ . The value of  $\gamma$  should depend on the time step and the mixing timescale, which scales like (Send and Marshall, 1995; Klinger et al., 1996)

$$t_{\text{mix}} \sim \frac{h_{\text{conv}}^2}{\kappa}, \quad (4.11)$$

where  $h_{\text{conv}}$  is the depth of convection,  $t_{\text{mix}}$  is the timescale of mixing and  $\kappa$  is the equivalent vertical diffusivity. Using typical values for the depth of convection and the diffusivity we can find a typical timescale of mixing, which we are then able to compare to the timestep. The size of the timescale of mixing compared to the timestep can then help us decide what the value for  $\gamma$  should be. Typical values would be  $h_{\text{conv}} = 1000\text{m}$  and  $\kappa = 50\text{m}^2\text{s}^{-1}$ . As can be seen, the choice of  $\gamma$  should be related to the time step,  $\Delta t$ . If  $\Delta t \sim t_{\text{mix}}$ , then our choice of  $\gamma$  should be such that  $\gamma \gg 1$ . On the other hand, if  $\Delta t \ll t_{\text{mix}}$ , then a reasonable choice of  $\gamma$  would be  $\gamma \sim 1$ .

Another way to consider the relationship between  $\gamma$  and  $m^a$ , is to allow  $\gamma$  to vary freely, such that the ratio  $m^a/m^k$  remains constant. The impact of this constraint is considered further in Section 4.3.

### 4.3 A “Swap” Exchange of Mass

Following from the discussion in Section 4.2 about the scaling parameter  $\gamma$ , if the ratio of  $m^a/m^k$  is a constant, we ensure that a constant proportion of a particular grid box is convected whenever an instability exists.

Using Equation (4.9b) in the limit of  $m^a/m^k = 1$ , the parameter  $\gamma$  is found by

$$\gamma = \frac{\rho_i^k}{\rho_i^k - \rho_i^{k+1}}. \quad (4.12a)$$

in the non-Boussinesq case, or by

$$\gamma = \frac{\rho_0}{\rho_i^k - \rho_i^{k+1}}, \quad (4.12b)$$

in the Boussinesq case.

We consider a picture similar to that of Figure 4.1 with a sinking blob and a return blob, except that the sinking blob  $a$  is the entire volume/mass of the grid cell  $k$ ,

$$m^a = m_i^k \quad (4.13a)$$

$$C_f^k = C_i^{k+1}. \quad (4.13b)$$

We can therefore say that the final density and tracer properties of the upper grid cell are the same as the initial density and tracer properties of the lower grid cell, noting that in virtually all ocean models, the grid cell thickness increases with depth (with the exception of partial bottom cells, see below), that is  $V^k \leq V^{k+1}$ .

Now using the equation for tracer conservation, Equation (4.3b), we also assume that the final mass of each grid box is the same as the initial mass (we can therefore drop the  $i$  and  $f$  subscripts from the mass). This last assumption requires that the mass of blob  $a$  and  $b$  be equal,  $m^a = m^b$ . We therefore find that the final concentration of the bottom  $k + 1$  gridbox is

$$C_f^{k+1} = \left(1 - \frac{m^k}{m^{k+1}}\right) C_i^{k+1} + \frac{m^k}{m^{k+1}} C_i^k. \quad (4.14)$$

Note, that care must be taken in the event of partial bottom grid cells (Adcroft et al., 1997), which can mean that  $m^k > m^{k+1}$ . In this instance, we use  $m^a = m_i^{k+1}$  instead of Equation (4.13a).

## 4.4 An Overflow Scheme

A common overflow scheme is that of Campin and Goosse (1999). The scheme does a search of four adjacent grid cells to test if the shallow ocean water is more dense than the deep ocean (see Figure 4.2). When the deep ocean density is more dense a downslope velocity is calculated based on a number of parameters.

The scheme calculates a downslope velocity,  $\mathbf{u}^{\text{slope}}$ , that depends on a prescribed frictional dissipation  $\mu$  (with units  $\text{s}^{-1}$ ), the density difference  $\Delta\rho = \rho^{ijk} - \rho^{lk}$ , the fraction of fluid within the cell participating in the flow  $0 \leq \delta \leq 1$  and the gradient of the topography,

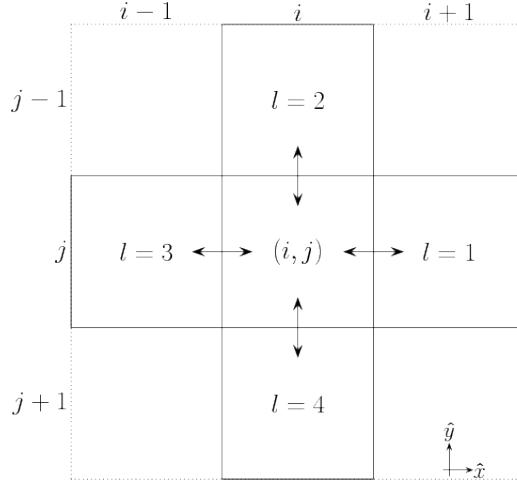


Figure 4.2: The search pattern for finding shallow ocean-deep ocean instability in the scheme of Campin and Goosse (1999). The four adjacent grid cells (solid outline and indicated by  $l$ ) to grid cell  $(i, j)$  are compared only if the  $l$  water column is deeper than the  $i, j$  water column by one or more full grid cells. If an instability is detected (see Figure 4.3), an overflow transport is calculated. The diagonal grid cell are not tested for instability.

$H$ . This leads to

$$\mathbf{u}^{\text{slope}} = \left( \frac{g\delta}{\rho_0 \mu} \right) (\partial_x H, \partial_y H) \Delta \rho, \quad (4.15)$$

where  $\Delta \rho = \rho^{ijk} - \rho^{lk}$ . From this velocity calculation, the volume transport is

$$\mathbf{U}^{\text{slope}} = \begin{cases} (u^{\text{slope}} dz^{ijk} dy^{ij}, 0) & \text{if } l = 1 \\ (0, v^{\text{slope}} dz^{ijk} dx^{ij}) & \text{if } l = 2 \\ (u^{\text{slope}} dz^{ijk} dy^{i-1j}, 0) & \text{if } l = 3 \\ (0, v^{\text{slope}} dz^{ijk} dx^{ij-1}) & \text{if } l = 4 \end{cases}, \quad (4.16)$$

where  $dx^{ij}$  is the zonal width of the north face of a grid cell,  $dy^{ij}$  is the meridional width of the eastern face of a grid cell and  $dz^{ijk}$  is the thickness of a grid cell (note that  $dx$  and  $dy$  are fixed in time and depth, whereas  $dz$  may vary in both time and space). There are special considerations when making the volume calculations in MOM due to the partial bottom grid cells, however, these issues will not form part of our discussion here – for further information, see Chapter 19 of Griffies (2009). As indicated by Figure 4.2 this parameterisation is applied in either the  $x$  or the  $y$  direction, not diagonally. It can, however, operate on more than one of the four surrounding grid cells in a given time step.

In the present blob scheme  $n + 2$  blobs are formed, where  $n$  is the difference in grid cell index of the shallow ocean cell and the target deep ocean cell, as shown in Figure 4.3. Blob  $a$  transports dense shelf water to its neutral depth in the deep ocean, or in the absence of the neutral surface in the adjacent water column, to the bottom grid cell. Blob 0, which is of equal mass to  $a$ , is formed and transported from the adjacent deep ocean grid cell to the shallow ocean grid cell. The remaining  $n$  blobs are transported up one cell each. Note

that  $n$  may take a value of zero in which case only two blobs are formed ( $a$  and 0), and they are exchanged between grid cells  $(i, j, k)$  and  $(l, k)$ .

From Equation (4.16), we can see that the mass of blob  $a$  should be

$$m^a = \int_{\Delta\tau} \mathbf{U}^{\text{slope}} \rho_i^{ijk} dt, \quad (4.17)$$

where  $\Delta\tau$  is the tracer time step and  $m^a$  is the mass of blob  $a$ . We set all blob masses to be the same,

$$m^a = m^0 = \dots = m^n. \quad (4.18)$$

The final tracer content of the shallow ocean grid box,  $(i, j, k)$  will be,

$$\mathcal{C}_f^{ijk} = \mathcal{C}_i^{ijk} + \frac{m^a}{m^{ijk}} (\mathcal{C}_i^{lk} - \mathcal{C}_i^{ijk}), \quad (4.19)$$

where subscript f indicates final and subscript i indicates initial concentration. The final tracer concentration for gridboxes in the deep ocean water column for  $k, \dots, k+n$  can be recovered with similar expressions, although the form for the final concentrations for  $(l, k)$  will depend on whether  $n$  is greater than zero or not.

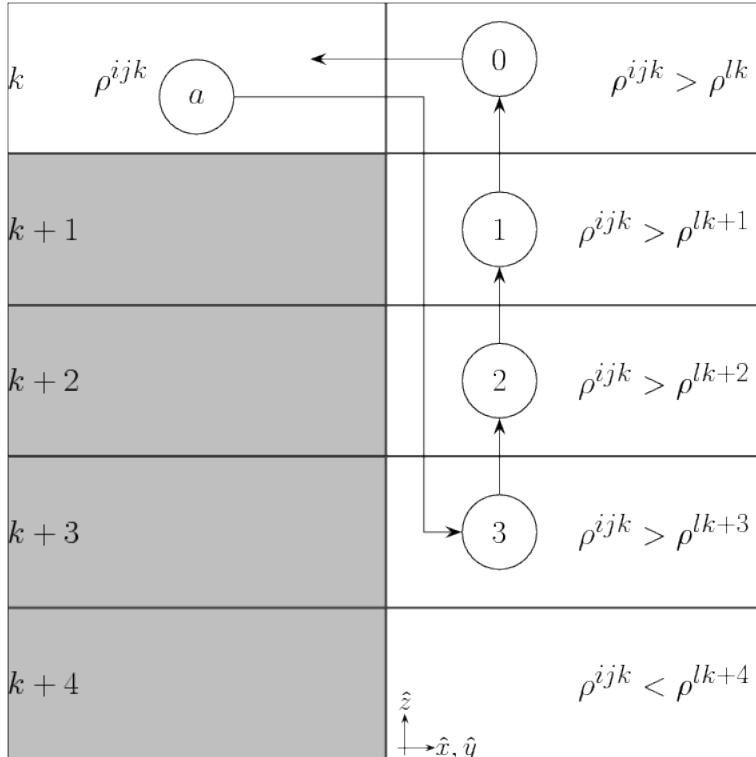


Figure 4.3: Adapted from Figure 1 of Campin and Goosse (1999). The density of the shallow ocean grid cell  $(i, j, k)$  is compared with that of the deep ocean grid cell  $(l, k)$ , where  $l$  has the same meaning as in Figure 4.2.

## 4.5 Two Slightly Different Overflow Schemes

Before beginning the formulation of the dynamically active blobs, we wish to formulate a scheme that will easily test the stability of transporting mass over the ubiquitous “cliffs” of level coordinate models without explicitly specifying a return of mass. That is, we wish to transport mass and allow the prognostic equations of the Eulerian model to respond and evolve accordingly. Two schemes are developed based on the original Campin and Goosse (1999) overflow scheme, developed in Section 4.4.

The first formulation is that two blobs are formed (the “one return blob overflow scheme”); one shallow ocean blob at  $(i, j, k)$  and an open ocean blob sitting over the deep water column at  $(l, k)$  (blob  $a$  and 0 respectively in Figure 4.4). The shallow ocean blob is moved to its neutral level (or the bottom grid cell, whichever is the shallower) in the open ocean column, and the deep ocean blob overlying deep water is moved into the shallow ocean cell. Note that the one return blob scheme is identical to the full return blob scheme in the instances where  $n = 0$ .

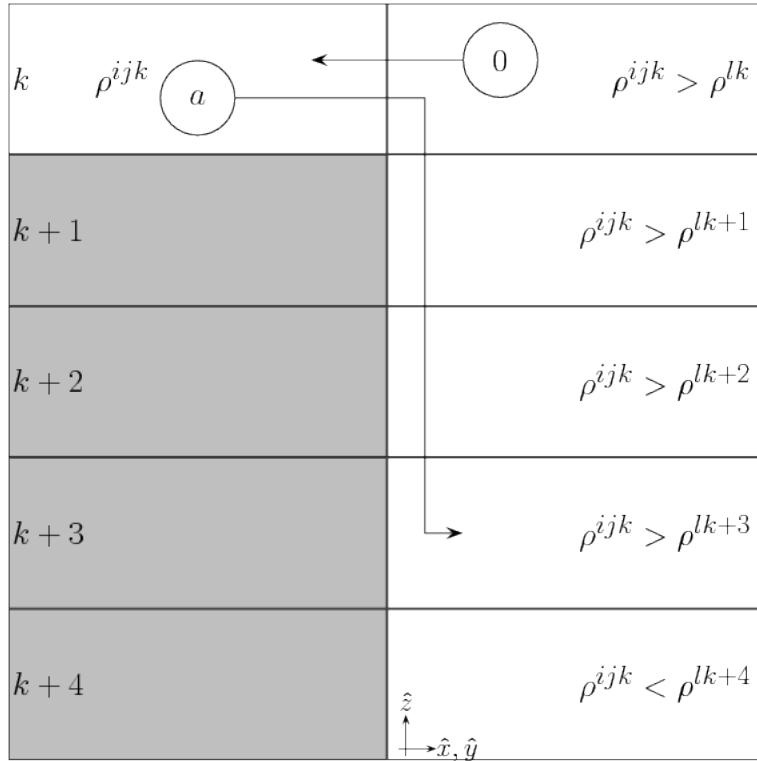


Figure 4.4: The “one return blob” overflow scheme. Only the on-shelf blob and the open ocean blob at  $k$  are formed and transported. The no return blob scheme is the same except blob 0 is not formed. These two schemes are used to test the stability of downslope flow schemes that transport mass down the slope while only handling part or none of the return of mass.

The second formulation is that only a single shallow ocean blob is formed (the “no return blob overflow scheme”), and no open ocean blob is formed. The no return blob

overflow scheme is equivalent to Figure 4.4 without blob 0.

## 4.6 Summary

Several dynamically passive schemes have been formulated, and, some of them implemented in MOM. By dynamically passive scheme, it is meant that the blobs within these schemes are created, moved, and then destroyed instantaneously (i.e. they do not integrate a momentum equation to solve for position, velocity, etc.).

The schemes that have thus far been implemented are designed to replicate existing non-blob schemes, or, a variation on those schemes.

# Chapter 5

## Dynamically Active Schemes

We now turn to further developing the momentum aspect of the Lagrangian framework. The dynamical properties of the Lagrangian system and how the Lagrangian and Eulerian systems interact is developed. As described in Chapter 3 we conceptually partition a model grid cell into two sections: E and L for the Eulerian and Lagrangian systems respectively.

For the momentum describing the combined Eulerian and Lagrangian systems, we assume there to be particular forms of frictional stresses acting on E and L, with details of the friction a function of the subgrid scale process considered. Buoyancy forces occur in the form of Archimedes' Principle, where the buoyancy force acting on L is equal to the weight of E displaced by L. This feature of the dynamics manifests in the vertical momentum equation for the L fluid.

### 5.1 Blob Dynamics

Chapter 3 focussed on theory required to admit an embedded Lagrangian model, as well as the ramifications to the overall system. Here, we focus on developing the dynamics of individual blobs. For completeness, the momentum Equation (3.23) for an individual blob is restated in terms of density

$$\rho_L \ddot{\mathbf{x}} + 2\boldsymbol{\Omega} \wedge \rho_L \dot{\mathbf{x}} = \rho_L \mathbf{g} - \nabla p + \nabla \cdot \boldsymbol{\tau} + \mathcal{S}_L^{(\mathbf{v})}, \quad (5.1)$$

where subscript L indicates a Lagrangian blob quantity (i.e.  $\rho_L$  is the density of a blob).

In Equation (5.1), interfacial contact forces are encapsulated in the momentum equation via the divergence of the stress tensor,  $\boldsymbol{\tau}$ , and can include bottom (topographical) drag and drag generated over the surface that interacts with the Eulerian model. Momentum source/sink terms are encapsulated in the  $\mathcal{S}_L^{(\mathbf{v})}$  term, and these can arise from mass exchange between the Eulerian and Lagrangian models.

When embedded in a hydrostatic Eulerian model, the Lagrangian blobs are pseudo-

non-hydrostatic.<sup>1</sup> This fact warrants some discussion about the nature of the pressure gradient term. If we consider a blob to be a point particle for the purpose of calculating the pressure gradient term, the pressure gradient reduces to that of the Eulerian model. In the vertical, the pressure gradient of the E system is the hydrostatic pressure  $\frac{\partial p}{\partial z} = -g\rho_E$ . In order to remain internally consistent, the Eulerian model needs to be able to calculate the total depth integrated pressure, including the contribution by the Lagrangian system. As such, the average depth integrated pressure for a particular water column (for example, the bottom of a grid cell) is calculated as

$$p = p_A + g \int_E \rho_E dz + \frac{g}{dA} \sum_{q=1}^Q m_L^q, \quad (5.2)$$

where the integral is for the E system water column from a depth of  $z$  to  $\eta = \eta(x, y, t)$ , the sea surface height deviation from the geoid. The first term,  $p_A$ , is the applied pressure at the ocean surface, due to atmospheric and sea ice loading. The second term is the hydrostatic pressure calculated from the mass in the water column of the Eulerian model and the third term represents the sum of the weight of  $Q$  Lagrangian blobs between the depth  $z$  and the surface  $\eta$  divided by the cross-sectional area of the water column  $dA$ .

For the vertical pressure gradient, we consider a blob to be a point particle and the pressure force acting upon it is the hydrostatic pressure. For the horizontal pressure gradients we take the horizontal pressure gradient terms in the blob momentum equations to be the pressure gradient calculated by the Eulerian model, interpolated to the position of the blob. These methods to calculate the pressure gradients are simple and convenient, however, may be too simple to adequately capture non-hydrostatic dynamics (see results in Chapter 11).

If we take Equation (5.1), the momentum equation for the blobs, and split it up into components and write  $\frac{1}{\rho_L} \nabla \cdot \boldsymbol{\tau} = (F^{(x)}, F^{(y)}, F^{(z)})$ ,

$$\rho_L \ddot{x} = \rho_L \dot{y} f + \rho_L \dot{z} f^* - \partial_x p + \rho_L F^{(x)} \quad (5.3a)$$

$$\rho_L \ddot{y} = -\rho_L \dot{x} f - \partial_y p + \rho_L F^{(y)} \quad (5.3b)$$

$$\rho_L \ddot{z} = -\rho_L \dot{x} f^* - g(\rho_L - \rho_E) + \rho_L F^{(z)}. \quad (5.3c)$$

We shall refer to the second term on the right hand side of Equation (5.3c) as the pseudo-non-hydrostatic term.<sup>2</sup>

If we then say that the blobs can entrain with an entrainment velocity of  $\mathcal{E}$  and detrain with a detrainment velocity of  $\mathcal{D}$  (both of which have units of  $\text{m s}^{-1}$ ), we can use

<sup>1</sup>Quasi-non-hydrostatic (Marshall et al., 1997b) is possibly a more apt description, however, this term commonly refers to a technique used in atmospheric modelling (for example White and Bromley, 1995; MacDonald et al., 2000). In order to avoid confusion, we use the term pseudo-non-hydrostatic.

<sup>2</sup>While it is recognised that this term is also the reduced gravity term, its description as “the pseudo-non-hydrostatic term” more accurately describes its role in the Lagrangian blob framework.

the entrainment and detrainment velocities to define the rate of change of properties of the blobs, and the properties that are transferred between the Eulerian model and the Lagrangian model (and vice versa),

$$\frac{d(m_L \mathcal{C}_L)}{dt} = (\rho_E \mathcal{C}_E \mathcal{E} - \rho_L \mathcal{C}_L \mathcal{D}) A_{\text{interface}} \quad (5.4a)$$

$$\frac{dm_L}{dt} = (\rho_E \mathcal{E} - \rho_L \mathcal{D}) A_{\text{interface}}, \quad (5.4b)$$

where  $A_{\text{interface}}$  is the surface area of the interface between the blob and the Eulerian model.

In order to find  $A_{\text{interface}}$ , some assumption about the shape of a blob must be made. As discussed in Section 3.1, there is nothing intrinsic in the blob formulation that requires us to specify the shape of a blob. However, when adapting parameterisations for entrainment and detrainment to the blobs, it may be necessary to specify the surface area over which the entrainment and detrainment is occurring. In some formulations, it may be desirable to partition the surface area of the blob so that there are varying rates of entrainment across the interface, however, we do not consider this case in this thesis, except where part of the blobs “surface” does not entrain at all (see Section 5.3).

It is common to use a large number of particles when representing a fluid parcel with passive Lagrangian floats. Sometimes a random-walk term is added to represent diffusive processes (e.g. Hunter et al., 1993) or simply to provide reliable statistics. This is not necessary for the dynamically active blobs as we are not trying to represent diffusive processes. Furthermore, the horizontal pathways of overflows are dominated by topographic effects rather than chaotic flows. On the other hand, open ocean deep convection involves chaotic motion, but, this occurs in small regions and is not a dispersive process. Thus, random-walk processes do not need to be included in the open ocean implementation.

## 5.2 An Open Ocean Implementation

It is intended that the open ocean implementation is a discrete, Lagrangian representation of the convective plumes depicted in Figure 2.1 and described in Section 2.1. The formation conditions described in Section 5.2.1 likely forms blobs in other circumstances as well, the formation conditions chosen are simple and easy to implement. There is also no direct consequences of forming blobs in other regions, aside from some additional computations.

Beginning by taking the momentum equation for the blobs, Equations (5.3), assuming that the frictional interaction between a blob and the E system takes the form of a Rayleigh drag

$$F^{(x)} = -\alpha(\dot{x} - u_E) \quad (5.5a)$$

$$F^{(y)} = -\alpha(\dot{y} - v_E) \quad (5.5b)$$

$$F^{(z)} = -\alpha(\dot{z} - w_E), \quad (5.5c)$$

where,  $\alpha$  is a coefficient of drag that may vary like  $\alpha = \alpha(t, x, y, z)$  with units of  $s^{-1}$ . We thus obtain the following dynamical equations for the blobs,

$$\ddot{x} = \dot{y}f - \dot{z}f^* - \frac{\partial_x p}{\rho_L} - \alpha(\dot{x} - u_E) \quad (5.6a)$$

$$\ddot{y} = -\dot{x}f - \frac{\partial_y p}{\rho_L} - \alpha(\dot{y} - v_E) \quad (5.6b)$$

$$\ddot{z} = \dot{x}f^* - g\frac{\rho_L - \rho_E}{\rho_L} - \alpha(\dot{z} - w_E). \quad (5.6c)$$

Equations (5.6) form the basis for the dynamics of this implementation of Lagrangian blobs. We now focus our attention on implementation issues.

With regards to the entrainment rate, an intuitive approach would be to relate the entrainment of a blob to the interfacial drag experienced. However, the relative importance of entrainment in open ocean deep convection remains an unanswered question (Marshall and Schott, 1999). Entrainment of point plumes has been demonstrated and quantified in non-rotating laboratory experiments by a number of authors, including Turner (1973, section 6.1.2). The equations arising from these experiments are, however, functions of depth and buoyancy input, quantities that do not lend themselves to the current methodology. On the other hand, Canuto et al. (2007) present a non-local mixing model for deep convection and it is beyond the scope of this study to attempt to incorporate such a parameterisation. In the interest of maintaining simplicity, the following implementation assumes no entrainment and a simple detrainment. As stated in Section 1.3, the aim of these notes is not to present the most physically comprehensive parameterisation, but rather as a proof of concept for the Lagrangian framework.

### 5.2.1 Open Ocean Blob Initialisation

The formation criterion is a vertical stability criterion. The vertical stability of the water column is tested using the square of the buoyancy frequency (also known as the Brunt-Väisälä frequency),

$$N^2 = -\frac{g}{\rho} \left( \frac{\partial \rho}{\partial \Theta} \frac{\partial \Theta}{\partial z} + \frac{\partial \rho}{\partial S} \frac{\partial S}{\partial z} \right), \quad (5.7)$$

For further information, see Gill (1982) or Pedlosky (1987). When square of the buoyancy frequency of the the water column is less than a reference value,  $N_0^2$ , a blob is formed. How to treat the square of the buoyancy frequency in the combined L and E system is discussed in Section 6.5.1.  $N_0^2$  should be a value less than or equal to zero, otherwise the water column is statically stable. The threshold condition for instability is not necessarily the most physically sound means for initiating deep convection (e.g. Campin et al., 2011), but, was chosen here for reasons of simplicity. A static instability threshold, however, is not an intrinsic part of the framework and more complex or physically based conditions could be implemented.

When defining the initial conditions of each blob, it is assigned the tracer concentra-

tions of its grid cell of origin, and therefore, by definition, the blob density is the same as grid cell of origin. If the blob does not leave the grid cell that it is formed in, the second term on the right hand side of the blob's vertical momentum Equation (5.6c) is zero, and the blob originating in the  $k$  cell has no impetus to sink. On the other hand the  $k + 1$  grid cell must have a lesser density than the blob, otherwise the instability condition would not have been satisfied. So, by initially placing a blob in the  $k + 1$  grid cell, the second term of Equation (5.6c) is negative and the blob will accelerate downwards.

We must also take into consideration that a major source for vertical instability to develop is due to surface buoyancy fluxes. It is thus desirable that a blob is formed after the surface forcing has been applied (see Section 6.1.2). In MOM, the surface forcing is generally applied implicitly in time. Therefore new blobs resulting from vertical instability should be formed implicitly in time (after the application of surface forcing) as a vertical adjustment.

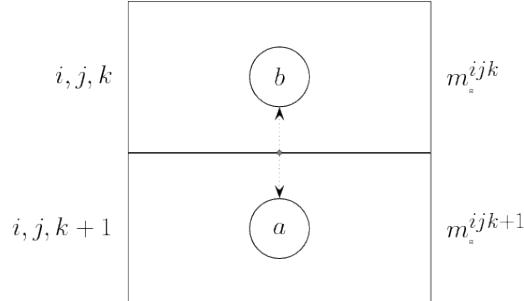


Figure 5.1: The properties and placement of a dynamical open ocean blob. Note that no assumption has been made about the shape of the blobs, the spherical shape in this schematic is illustrative only. Blob  $a$  obtains its scalar properties from the grid cell  $(i, j, k)$  and is initially placed inside  $(i, j, k + 1)$ , while blob  $b$  obtains its scalar properties from  $(i, j, k + 1)$  and is initially placed in the grid cell  $(i, j, k)$ .

Two blobs are formed for each instability in the same fashion as the NCon-like scheme (see Section 4.1). The blobs then swap grid boxes as depicted in Figure 5.1. For operations that are implicit in time, we must maintain a zero net mass flux, due to order of operation constraints (see Section 6.1.2). Thus, we require at least two blobs to be formed in order to balance the mass flux. The upper blob's mass and tracer content is given by,

$$m^a = \Lambda \frac{m_{\text{E}}^{ijk} m_{\text{E}}^{ijk+1}}{m_{\text{E}}^{ijk} + m_{\text{E}}^{ijk+1}}, \quad (5.8a)$$

$$\mathcal{C}_{\text{L}}^a m^a = \mathcal{C}_{\text{E}}^{ijk} m^a, \quad (5.8b)$$

and the lower blob's mass and tracer content given by,

$$m^b = \Lambda \frac{m_{\text{E}}^{ijk} m_{\text{E}}^{ijk+1}}{m_{\text{E}}^{ijk} + m_{\text{E}}^{ijk+1}} \quad (5.9a)$$

$$\mathcal{C}_{\text{L}}^b m^b = \mathcal{C}_{\text{E}}^{ijk+1} m^b. \quad (5.9b)$$

We note that  $\Lambda$  is a dimensionless scaling variable that is given a value  $0 < \Lambda \leq 1$  and is a heuristic way to control the size of the blobs produced. If we take the limit in which the two cell thicknesses are equal,  $dz^k = dz^{k+1}$ , the resultant blobs will occupy half of the mass of the cells if  $\Lambda = 1$ . Another way to view  $\Lambda$  is that it is analogous to the scaling parameter,  $\delta$ , of the Campin and Goosse (1999) scheme discussed in Section 4.4, which specified the fraction of a grid cell participating in the overflow scheme.

There are two possible ways that we can proceed. One is to form both blobs as dynamically active blobs, and the other is to form the downward blob,  $a$ , as a dynamically active blob and the upward blob,  $b$  as a dynamically passive blob. Forming one dynamically active and one dynamically passive blob has the advantage of only needing to integrate half as many blobs at subsequent time steps as if both are dynamically active, thereby halving the memory and computational costs. Under such an arrangement, blob  $b$  is treated identically to blob  $b$  in the NCon scheme of Section 4.1.

There is also a physical motivation for only creating one dynamically active blob. The present scheme is designed to model plumes in deep convection events (see Section 2.1), in which the mass flux of narrow descending plumes ( $\sim 1\text{km}$ ) is balanced by a broad upwelling. The effect of shifting mass upwards (which is essentially what blob  $b$  is doing) is more akin to a broad upwelling than if the blob were to be a dynamically active blob, which would imply that it is an ascending plume.

Another implication of considering blobs to be a representation of descending plumes is that the present scheme does not necessarily replace other convective adjustment schemes, or, boundary layer schemes, but, is a supplement to better represent deep convection.

As the blobs are formed in the implicit part of the time step, we prescribe the initial position of blob  $a$  at time  $\tau + \Delta\tau$ . We choose the initial horizontal position to be the centre of the tracer cell,  $(x_{\mathcal{T}}, y_{\mathcal{T}})$ , where  $\mathcal{T}$  denotes the tracer point (see the discussion in Section 6.4 for more details on the various points defined on the Arakawa-B grid in MOM). The velocity is that given by the tracer cell of origin. To this end, we take the unweighted average of the four surrounding velocity grid cells, and assign this as the blob's initial horizontal velocity.

For the initial vertical velocity and position, we are motivated by energetics arguments to give the initial velocity as the vertical velocity of the grid cell of origin plus a linear acceleration over time  $\Delta\tau$  resulting from the pseudo-non-hydrostatic term. We find the

initial vertical velocity to be

$$\dot{z}^a = w_{\mathcal{W}}^{ijk} - \Delta\tau g \frac{\rho_L^a - \rho_E^{ijk+1}}{\rho_L^a}. \quad (5.10)$$

Note that here  $\mathcal{W}$  is the label for the vertical velocity grid point (see the discussion in Section 6.4). The distance travelled over the time  $\Delta\tau$  is given by  $\Delta\tau(w_{\mathcal{W}}^{ijk} - \frac{1}{2}\Delta\tau g \frac{\rho_L^a - \rho_E^{ijk+1}}{\rho_L^a})$ .

The initial position and velocity for the blob  $a$  is written mathematically as,

$$\xi^a(\tau + \Delta\tau) = \left( \xi_1^{ij} + \frac{\Delta\tau \dot{x}^a}{h_1^a}, \xi_2^{ij} + \frac{\Delta\tau \dot{y}^a}{h_2^a} \right) \quad (5.11a)$$

$$z^a(\tau + \Delta\tau) = z_{\mathcal{W}}^{ijk} + \Delta\tau w_E^{ijk} - (\Delta\tau)^2 g \frac{\rho_L^a - \rho_E^{ijk+1}}{2\rho_L^a} \quad (5.11b)$$

$$\dot{\mathbf{x}}^a(\tau + \Delta\tau) = \left( u_E^{ijk}, v_E^{ijk}, w_E^{ijk} - \Delta\tau g \frac{\rho_L^a - \rho_E^{ijk+1}}{\rho_L^a} \right), \quad (5.11c)$$

where  $\xi = (\xi_1, \xi_2)$  is the native coordinate of the Eulerian grid (e.g. longitude and latitude) and  $\mathbf{h} = (h_1, h_2)$  is the stretching function (see Section 7.1.1). We note that for a blob to be formed we require that  $\rho_L^a - \rho_E^{ijk+1} > 0$ , giving the correct sign for the position that is indicated in Figure 5.1.

The formation criterion discussed here is by no means the only possible criterion. As an example of another possible formation criterion, let us imagine that we have a situation where the surface forcing data is a finer resolution than the ocean model. We may find it desirable to create a blob located at or near the surface, directly under the area of strong negative buoyancy flux with the initial properties based on that buoyancy flux, rather than the average properties of the grid cell of origin.

### 5.2.2 Open Ocean Blob Entrainment, Detrainment and Destruction

We may wish to destroy blobs by saying that once they reach their neutral surface their properties are returned to the E system. An alternative is to have a background detrainment. Thus, a detrainment velocity is sought that is near zero detrainment when the density difference with the ambient grid box is large (the blob is not near its neutral density surface), but rapidly detrains and transfers its properties to the Eulerian model when the density difference is small (when the blob is in the vicinity of its neutral density surface). A form that suits this condition well is if the detrainment velocity is inversely proportional to the density difference between the Lagrangian blob and the surrounding Eulerian grid box,

$$\mathcal{E}^q = 0 \quad (5.12)$$

$$\mathcal{D}^q = -\frac{\Gamma}{|\rho_L^q - \rho_E^{ijk}|}, \quad (5.13)$$

where  $\Gamma$  is a tunable proportionality parameter with units  $\text{kg m}^{-2} \text{s}^{-1}$  that is specified *a priori*. We place an upper bound on the detrainment velocity,  $|\mathcal{D}^q| \leq \mathcal{D}_{\max}$ .

In order to find the rate of change of properties of blob  $q$ , an interfacial surface area  $A_{\text{interface}}^q$  must be assumed between the Lagrangian blob and the Eulerian grid box (see Equations (5.4a) and (5.4b) and the ensuing discussion). For the present implementation, we assume the blob is spherical, and that rate of detrainment is uniform across  $A_{\text{interface}}^q$ . We know that the volume of the blob must be  $V_L^q = m_L^q / \rho_L^q$ , therefore the area is

$$A_{\text{interface}}^q = (36\pi)^{\frac{1}{3}} \left( \frac{m_L^q}{\rho_L^q} \right)^{\frac{2}{3}}. \quad (5.14)$$

See Appendix B for details of the relation between surface area and volume of a sphere.

There are a number of possible destruction conditions:

- When the mass of the blob goes below a certain threshold (specified in the namelist) and the remaining properties are returned to the E system, the blob is erased from the model’s memory.
- If a blob’s vertical position means that it will rise above the surface, then the blob is immediately destroyed and its properties returned to the surface grid cell.
- It can also be possible to have too much mass in the L partition of a grid cell, in which case we also need to destroy blobs (see Section 6.5.3 for a detailed discussion).
- It is occasionally possible for a blob to become grounded, which is also a reason to destroy the blob. By “grounded” it is meant that a blob has crossed from a wet ocean water column to a dry land column (not necessarily at the surface). The properties of the blob are returned to the last wet grid cell that the blob resided in before moving into the dry column.

### 5.3 A Bottom Boundary Implementation

The bottom boundary layer dynamical regime for this implementation is chosen to have bottom stress, interfacial stress and entrainment based on the Price and Baringer (1994) steady-state streamtube model. The Price and Baringer (1994) model uses several of the streamtube approximations of Smith (1975) to model and estimate various properties for four overflows. They utilise a Froude number dependent parameterisation for entrainment and an Ekman number dependent parameterisation for broadening.

We begin by restating the original momentum and entrainment equations and then reformulate the Price and Baringer (1994) equations with reference to the blob dynamics discussion in Section 5.1. Finally, we will discuss implementation issues in a fashion similar to the previous section.

Respectively, the steady state momentum, bottom stress, entrainment, Richardson number, continuity and spreading of width equations used by the Price and Baringer (1994) streamtube model are

$$\mathbf{u} \cdot \nabla \mathbf{u} + (0, 0, f) \wedge \mathbf{u} = g \frac{\rho_{st} - \rho_{ocn}}{\rho_0} \nabla H - \frac{\tau_{bot}}{\rho_0 h_{st}} - \frac{\mathcal{E} \mathbf{u}}{h_{st}} \quad (5.15a)$$

$$\tau_{bot} = \rho_0 C_d |\mathbf{u}| \mathbf{u} \quad (5.15b)$$

$$\mathcal{E} = \begin{cases} |\mathbf{u}|^{\frac{0.08-0.1Ri}{1+5Ri}} & \text{if } Ri \leq 0.8 \\ 0 & \text{if } Ri > 0.8 \end{cases} \quad (5.15c)$$

$$Ri = g \frac{(\rho_{st} - \rho_{ocn}) h_{st}}{\rho_0 |\mathbf{u}|^2} \quad (5.15d)$$

$$\mathbf{u} \cdot \nabla h_{st} = \mathcal{E} - \frac{h_{st} \mathbf{u} \cdot \nabla W}{W} - h_{st} \nabla \cdot \mathbf{u} \quad (5.15e)$$

$$(\mathbf{u} \cdot \nabla W)/|\mathbf{u}| = \beta, \quad (5.15f)$$

where,  $\rho_{st}$  is the density of the middle of the streamtube,  $\rho_{ocn}$  is the density of the ocean near the streamtube,  $H$  is the depth of the ocean,  $\tau_{bot}$  is the bottom surface stress,  $h_{st}$  is the height of the streamtube,  $C_d$  is the dimensionless coefficient of drag,  $Ri$  is the Richardson number (which expresses the ratio of potential to kinetic energy),  $W$  is the width of the streamtube and  $\beta$  is a dimensionless parameter that is specified *a priori*. The dimensions of the streamtube are illustrated in Figure 5.2.

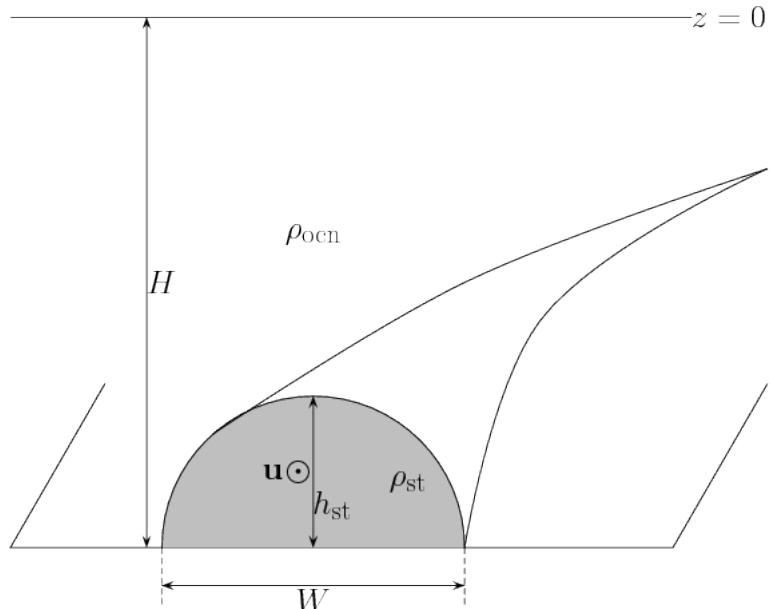


Figure 5.2: Schematic illustration of a streamtube model. The shaded area is a cross-section of the streamtube. The velocity of the streamtube,  $\mathbf{u}$  is perpendicular to the shaded section.  $H$  is the depth of the topography,  $h_{st}$  is the height of the streamtube from the bottom,  $W$  is the width of the streamtube,  $\rho_{st}$  is the density of the streamtube and  $\rho_{ocn}$  is the density of the ambient ocean.

Firstly, we discuss the dynamics by recognising the properties of Equation (5.15a).

- This is a steady-state, Boussinesq, Cartesian Eulerian two dimensional along slope coordinate. We seek a non-steady, non-Boussinesq, three dimensional Cartesian Lagrangian formulation.
- Only the vertical component of the Earth's rotation vector is used, which is reasonable, as the horizontal components of velocity are generally much larger than the vertical components for bottom boundary layers.
- The first term on the right hand side includes the gradient of the topography. We wish to generalise this term to three dimensions, and make this our pseudo-non-hydrostatic term.
- The second and third terms on the right hand side are stress due to bottom drag and stress due to interfacial drag with the bulk ocean respectively.
- Aside from buoyancy, there are no pressure terms.
- There are no acceleration sources or sinks.

We are now in a position to reformulate the Lagrangian momentum Equations (5.3) with inspiration from the Price-Baringer streamtube model.

Firstly, we note that the Price and Baringer (1994) formulation is a two dimensional along-slope coordinate system. To make a three dimensional formulation, we must take into account the normal force from the ocean floor. The normal force is given by multiplying the pseudo-non-hydrostatic term by the normal vector,

$$\rho_L \mathbf{F}_N = -g(\rho_L - \rho_E) \hat{\mathbf{n}}, \quad (5.16)$$

where  $\mathbf{F}_N$  is the normal force and  $\hat{\mathbf{n}} = -\frac{\nabla(H+z)}{|\nabla(H+z)|}$  is the outward normal vector. An outward pointing normal vector is the convention in the literature. Here, the normal force is anti-parallel to the outward normal vector, necessitating a negative sign in Equation (5.16). The horizontal component of rotation is ignored. The same form for the detrainment velocity as for the dynamically active free blobs is used (see Section 5.2.2). A modified version of the Price and Baringer (1994) formulation for bottom stress and entrainment is used. The modifications are to formulate the equations in three dimensional Cartesian coordinates, and, to make the velocity used in the Richardson number and entrainment

velocity calculations relative to the E system velocity,

$$\tau_{\text{bot}} = \rho_L C_d |\dot{\mathbf{x}}| \dot{\mathbf{x}} \quad (5.17a)$$

$$\mathcal{E} = \begin{cases} |\dot{\mathbf{x}} - \mathbf{u}_E|^{\frac{0.08-0.1Ri}{1+5Ri}} & \text{if } Ri \leq 0.8 \\ 0 & \text{if } Ri > 0.8 \end{cases} \quad (5.17b)$$

$$\mathcal{D} = -\frac{\Gamma}{|\rho_L - \rho_E|} \quad (5.17c)$$

$$Ri = g \frac{(\rho_L - \rho_E) h_L}{\rho_L |\dot{\mathbf{x}} - \mathbf{u}_E|^2}, \quad (5.17d)$$

where  $\dot{\mathbf{x}} - \mathbf{u}_E = (\dot{x} - u_E, \dot{y} - v_E, \dot{z})$ . Using all of this information, the basic Equations (5.3) are modified to yield time dependent momentum equations for a blob travelling along the bottom boundary, with drag terms based on the formulation of Price and Baringer (1994)

$$\ddot{x} = f\dot{y} + \partial_x H \left( \frac{g(\rho_L - \rho_E)}{\rho_L \sqrt{(\partial_x H)^2 + (\partial_y H)^2 + 1}} \right) - \frac{\tau_{\text{bot}}^x}{\rho_L h_L} - \frac{\mathcal{E}(\dot{x} - u_E)}{h_L} \quad (5.18a)$$

$$\ddot{y} = -f\dot{x} + \partial_y H \left( \frac{g(\rho_L - \rho_E)}{\rho_L \sqrt{(\partial_x H)^2 + (\partial_y H)^2 + 1}} \right) - \frac{\tau_{\text{bot}}^y}{\rho_L h_L} - \frac{\mathcal{E}(\dot{y} - v_E)}{h_L} \quad (5.18b)$$

$$\ddot{z} = g \frac{\rho_L - \rho_E}{\rho_L} \left( \frac{1}{\sqrt{(\partial_x H)^2 + (\partial_y H)^2 + 1}} - 1 \right) - \frac{\tau_{\text{bot}}^z}{\rho_L h_L} - \frac{\mathcal{E}\dot{z}}{h_L}, \quad (5.18c)$$

where  $h_L$  is the height of a blob, which we define in Equation (5.19b). The bottom stress term,  $\tau_{\text{bot}}/(\rho_L h_L)$  is independent of the properties of the surrounding Eulerian grid cell.

There is no  $w_E$  term in the vertical component of the interfacial drag term in Equation (5.18c) because we have used the small slope assumption. The bottom boundary conditions dictate that there is no normal flow,  $-\mathbf{v} \cdot \hat{\mathbf{n}} = 0$ . The small slope assumption says that there is very little of the normal component of the flow is made up by horizontal velocities,  $-\mathbf{v}_E \cdot \hat{\mathbf{n}} \approx w_E$  and that the greater majority of the lateral fluxes are made up of horizontal velocities. This assumption is valid for the E system because the horizontal length scales of a slope are typically much greater than its vertical scale.

An upper bound is placed on the magnitude of the entrainment and detrainment velocities,  $|\mathcal{E}| \leq \mathcal{E}_{\max}$  and  $|\mathcal{D}| \leq \mathcal{D}_{\max}$ . It is assumed that the height to width ratio  $h_L/W_L$  of the blob remains constant, that is, we do not use the Price and Baringer (1994) parameterisation for spreading. Furthermore, we take  $h_L$  to be the diameter of a sphere (see appendix B) and recall again that  $V = m/\rho$ , so that

$$A_{\text{interface}} = \left( \frac{9\pi}{4} \right)^{\frac{1}{3}} \left( \frac{m_L^q}{\rho_L^q} \right)^{\frac{2}{3}} \quad (5.19a)$$

$$h_L = \left( \frac{6}{\pi} \frac{m_L^q}{\rho_L^q} \right)^{\frac{1}{3}}. \quad (5.19b)$$

For the rate of change of properties due to entrainment and detrainment, we take the surface area of the blob to be half the surface area of a sphere.

Equations (5.18) form the governing equations for the dynamically active bottom blob formulation. Similar to the free blobs, there are a number of ways that blobs can be destroyed. The destruction criteria for bottom blobs are:

- when a blob’s mass is less than a small mass (specified as a namelist option),
- when the amount of L system mass contained within a grid cell reaches some critical point (see Section 6.5.3), or
- a blob is grounded, that is, when a blob moves into a dry water column.

With regards to topography, the Lagrangian framework is not intrinsically restrictive of the topography that is used. It is conceivable that the L system’s topography may be at a finer resolution than the E system’s topography, thereby allowing better resolution of important bathymetric features, such as ridges, canyons and sills. On the other hand, it is not possible for the scheme to work if there is “stepped” topography, as the pseudo-non-hydrostatic term will not be able to take effect without a slope. For the purposes of these notes we will only consider a simple topography based on the depth information of the E system grid. We discuss the details of the topography in Section 7.1.6.

### 5.3.1 Bottom Blob Initialisation

A dynamically active bottom blob is formed when the difference in density of a shallow ocean grid box  $(i, j, k)$  and the bottom grid cell in the deep column is greater than some threshold

$$\Delta\rho_{\text{th}} > \rho^{ijk} - \rho^{lk_{\text{bot}}}, \quad (5.20)$$

where  $\Delta\rho_{\text{th}}$  is the threshold density difference for a blob to be formed. The initial position of a bottom blob is schematically illustrated in Figure 5.3. In this sense, the condition is similar to that of Beckmann and Döscher (1997) which uses the density gradient along the E system’s topography, whereas the Campin and Goosse (1999) condition (illustrated in Section 4.4) tests the density at the same level in the two columns.

To obtain the initial properties for the blob we make use of a modified form of the formulation of Campin and Goosse (1999). Instead of the formulation given in equations (4.15) and (4.17) which calculates the initial horizontal velocity based on the density contrast between the grid cells  $(i, j, k)$  and  $(l, k)$ , we use the difference in density between the grid cell  $(i, j, k)$  and the grid cell in the deep ocean column that is at the same level as the shallow ocean cell,  $(l, k)$ .

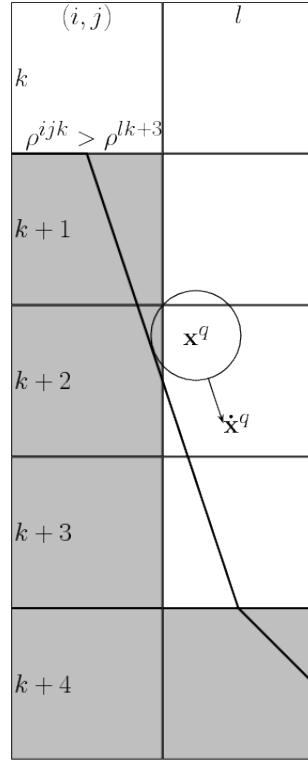


Figure 5.3: Initial position and schematic of the dynamically active bottom blob formation criterion. Topography as the bottom blobs “feel” it, is also illustrated with a thick black line. Shaded boxes are solid earth grid boxes.  $l$  has the same meaning as in Figure 4.2. A blob that is formed in gridcell  $(i, j, k)$  is placed on the topography in the neighbouring water column with the initial conditions given by equations Equation (5.21)

Mathematically, we state the initial position and velocity as,

$$(\xi^q, z^q)(\tau + \Delta\tau) = \begin{cases} \left( \xi_{1\mathcal{E}}^{ijk} + \Delta\tau \frac{\dot{x}^q}{h_1}, \xi_{2\mathcal{E}}^{ijk}, -H_{\mathcal{E}}^{ij} \right) & \text{if } l = 1 \\ \left( x_{\mathcal{N}}^{ijk}, y_{\mathcal{N}}^{ijk} + \Delta\tau \frac{\dot{y}^q}{h_2}, -H_{\mathcal{N}}^{ij} \right) & \text{if } l = 2 \\ \left( x_{\mathcal{E}}^{i-1jk} + \Delta\tau \frac{\dot{x}^q}{h_1}, y_{\mathcal{E}}^{i-1jk}, -H_{\mathcal{E}}^{i-1j} \right) & \text{if } l = 3 \\ \left( x_{\mathcal{N}}^{ij-1k}, y_{\mathcal{N}}^{ij-1k} + \Delta\tau \frac{\dot{y}^q}{h_2}, -H_{\mathcal{N}}^{ij-1} \right) & \text{if } l = 4 \end{cases} \quad (5.21a)$$

$$\dot{x}^q(\tau + \Delta\tau) = \begin{cases} u_{\mathcal{E}}^{ijk} + \frac{g\delta}{\rho_E^{ijk}\mu} (\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}) \partial_x H_{\mathcal{E}}^{ij} & \text{if } l = 1 \\ u_{\mathcal{N}}^{ijk} + \frac{g\delta}{\rho_E^{ijk}\mu} (\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}) \partial_x H_{\mathcal{N}}^{ij} & \text{if } l = 2 \\ u_{\mathcal{E}}^{i-1jk} + \frac{g\delta}{\rho_E^{ijk}\mu} (\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}) \partial_x H_{\mathcal{E}}^{i-1j} & \text{if } l = 3 \\ u_{\mathcal{N}}^{ij-1k} + \frac{g\delta}{\rho_E^{ijk}\mu} (\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}) \partial_x H_{\mathcal{N}}^{ij-1} & \text{if } l = 4 \end{cases} \quad (5.21b)$$

$$\dot{y}^q(\tau + \Delta\tau) = \begin{cases} v_{\mathcal{E}}^{ijk} & \text{if } l = 1 \\ v_{\mathcal{N}}^{ijk} + \frac{g\delta}{\rho_E^{ijk}\mu} (\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}) \partial_y H_{\mathcal{N}}^{ij} & \text{if } l = 2 \\ v_{\mathcal{E}}^{i-1jk} & \text{if } l = 3 \\ v_{\mathcal{N}}^{ij-1k} + \frac{g\delta}{\rho_E^{ijk}\mu} (\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}) \partial_y H_{\mathcal{N}}^{ij-1} & \text{if } l = 4 \end{cases} \quad (5.21c)$$

$$\dot{z}^q(\tau + \Delta\tau) = \begin{cases} g \frac{\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}}{\rho_E^{ijk}} \left( \left[ (\partial_x H_{\mathcal{E}}^{ij})^2 + 1 \right]^{-\frac{1}{2}} - 1 \right) & \text{if } l = 1 \\ g \frac{\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}}{\rho_E^{ijk}} \left( \left[ (\partial_y H_{\mathcal{N}}^{ij})^2 + 1 \right]^{-\frac{1}{2}} - 1 \right) & \text{if } l = 2 \\ g \frac{\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}}{\rho_E^{ijk}} \left( \left[ (\partial_x H_{\mathcal{E}}^{i-1j})^2 + 1 \right]^{-\frac{1}{2}} - 1 \right) & \text{if } l = 3 \\ g \frac{\rho_E^{ijk} - \rho_E^{lk_{\text{bot}}}}{\rho_E^{ijk}} \left( \left[ (\partial_y H_{\mathcal{N}}^{ij-1})^2 + 1 \right]^{-\frac{1}{2}} - 1 \right) & \text{if } l = 4 \end{cases}, \quad (5.21d)$$

where a subscripted  $\mathcal{E}$  indicates the value of a variable at the centre of the eastern face of a grid cell and  $\mathcal{N}$  indicate the value of a variable at the centre of the northern face of a grid cell (see Section 6.4).  $\delta$  is the fraction of a grid cell participating in the overflow event and  $\mu$  is a prescribed frictional dissipation ( $s^{-1}$ ). We may wish to make  $\mu = \mu(\xi_1, \xi_2)$  to more accurately account for local topography. The form for the vertical velocity is taken from Equation (5.18).

Using the volume transport defined in Equation (4.16), we may find the initial mass and tracer content of the blob,

$$m_L^q = \begin{cases} g\delta\mu^{-1}|\partial_x H|(\rho^{ijk} - \rho^{lk})\Delta\tau dz_E^{ijk} dy^{ij} & \text{if } l = 1 \\ g\delta\mu^{-1}|\partial_y H|(\rho^{ijk} - \rho^{lk})\Delta\tau dz_E^{ijk} dx^{ij} & \text{if } l = 2 \\ g\delta\mu^{-1}|\partial_x H|(\rho^{ijk} - \rho^{lk})\Delta\tau dz_E^{ijk} dy^{i-1j} & \text{if } l = 3 \\ g\delta\mu^{-1}|\partial_y H|(\rho^{ijk} - \rho^{lk})\Delta\tau dz_E^{ijk} dx^{ij-1} & \text{if } l = 4 \end{cases} \quad (5.22a)$$

$$\mathcal{C}_L^q m_L^q = \mathcal{C}_E^{ijk} m_L^q, \quad (5.22b)$$

where  $dx$  is the horizontal length of the eastern face of a tracer grid cell,  $dy$  is the horizontal length of the northern face of a tracer grid cell.

Theoretically, a grid cell could form up to four blobs. To ensure that a grid cell does not empty due to the formation of blobs, a condition is enforced that a blob may not be larger than one quarter the mass of the grid cell

$$m^a \leq \frac{(dA(\rho dz)_E)^{ijk}}{4}. \quad (5.23)$$

## 5.4 Switching Between Dynamically Active Regimes

In order to smoothly and realistically represent the shift of dynamical regimes when a blob interacts with topography we must define how the momentum and velocity are changed by the interaction with topography. This process is depicted schematically in Figure 5.4, with the “open ocean” regime depicted with a dotted outline and dashed pathway, while the “bottom boundary” regime is depicted with a solid outline and pathway.

The kinematic bottom boundary condition is one of no normal flow. In order to satisfy the boundary condition when a free blob interacts with topography, the momentum vector of the blob  $m_L \dot{\mathbf{x}}$ , must be rotated to be parallel with the topography,  $m_L \dot{\mathbf{x}}_{\text{new}}$ .

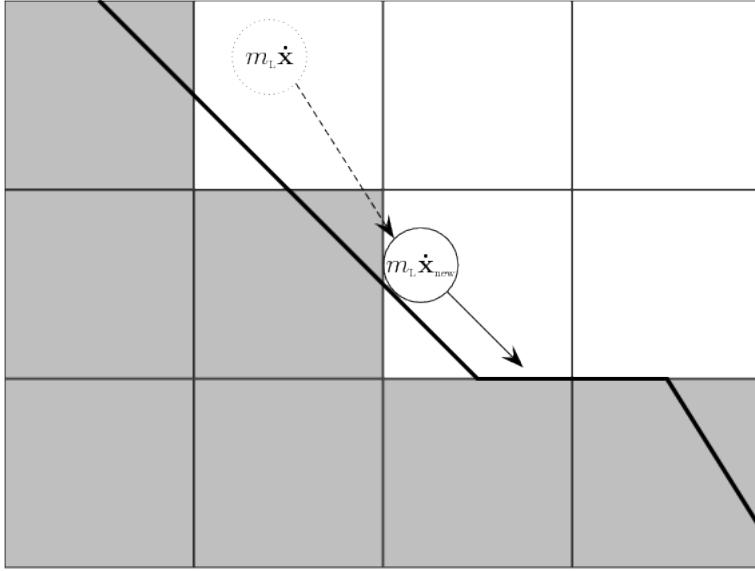


Figure 5.4: The process by which an open ocean blob becomes a bottom blob. The pathway and position prior to interacting with the topography are indicated by the dotted circle and dashed arrow. The pre-interaction momentum is also indicated,  $m_L \dot{\mathbf{x}}$ . The position at interaction is indicated by the solid outlined circle and is labelled with the momentum at the instant after the interaction,  $m_L \dot{\mathbf{x}}_{\text{new}}$ . The post-interaction pathway is indicated by the solid arrow.

To rotate the momentum vector we begin by stating the equations for the velocity parallel and perpendicular to the topography,  $\dot{\mathbf{x}}_{\parallel}$  and  $\dot{\mathbf{x}}_{\perp}$  respectively (the form of which comes from vector calculus),

$$m_L \dot{\mathbf{x}}_{\perp} = m_L (\dot{\mathbf{x}} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}} \quad (5.24a)$$

$$m_L \dot{\mathbf{x}}_{\parallel} = m_L (\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\perp}). \quad (5.24b)$$

It is assumed that the collision is perfectly elastic,  $|m_L \dot{\mathbf{x}}_{\text{new}}| = |m_L \dot{\mathbf{x}}|$ , unless  $\dot{\mathbf{x}}_{\parallel} = 0$ , in which case, a perfectly inelastic collision is assumed,  $|m_L \dot{\mathbf{x}}_{\text{new}}| = 0$ . An expression for the new velocity that satisfies these conditions is

$$m_L \dot{\mathbf{x}}_{\text{new}} = \begin{cases} \varepsilon m_L |\dot{\mathbf{x}}| \frac{\dot{\mathbf{x}}_{\parallel}}{|\dot{\mathbf{x}}_{\parallel}|} & \text{if } |\dot{\mathbf{x}}_{\parallel}| > 0 \\ 0 & \text{if } |\dot{\mathbf{x}}_{\parallel}| = 0 \end{cases} \quad (5.25)$$

$\varepsilon$  is a parameter that defines how elastic the interaction is. For a perfectly elastic collision

$$\varepsilon = 1. \quad (5.26)$$

This is by no means the only possible (and probably not the most realistic) formulation for this elasticity parameter, however, in the interest of developmental simplicity it is the formulation that is chosen. Some other possible formulations may include the following:

- a constant that is between zero and one,<sup>3</sup>  $0 \leq \varepsilon < 1$ , for  $\dot{\mathbf{x}}_{\parallel} \neq 0$  (this ensures all interactions are either inelastic,  $0 < \varepsilon < 1$ , or perfectly inelastic,  $\varepsilon = 0$  ),
- a spatially varying parameter, such that  $\varepsilon = \varepsilon(x, y)$  with values chosen to represent the effects of local topography,
- related to the angle of impact, such that  $\varepsilon = \cos(2\varphi)$ , where  $\varphi$  is the angle that the blobs trajectory makes with the topography. So if  $\varphi = 0$  the interaction is perfectly elastic,  $\varepsilon = 1$  and if  $\varphi = \pi/2$  the interaction is perfectly inelastic,  $\varepsilon = 0$ .

It has been previously discussed how it is possible for a blob to shift dynamic regimes from open ocean dynamics to bottom boundary dynamics. Bottom boundary layers in the real ocean separate from topography when they reach their neutral level. To capture the process of boundary layer separation with the Lagrangian framework some conditions must be formulated in order to transfer a blob from the bottom boundary dynamics to the open ocean dynamics.

The condition for separation is

$$\rho_L^q < \rho_E^{ijk}. \quad (5.27)$$

The process of separation is depicted diagrammatically in Figure 5.5. To avoid a blob frequently changing between the bottom and free regimes, a condition of no negative vertical velocity is placed on all newly separated blobs.

## 5.5 Summary

Two formulations of dynamically active schemes in the embedded Lagrangian framework have been derived and presented. The formulations allow for water parcels to be treated with more appropriate dynamics than the dynamics available in the present generation of level coordinate ocean climate models.

The first is a scheme to represent SGS plumes in open ocean deep convection. The second is to represent gravity currents from overflows. Details of how to calculate the creation conditions, initial properties and destruction conditions of Lagrangian blobs were also presented. In addition, methods for transferring from one dynamic regime to the other were presented.

---

<sup>3</sup>A constant that is greater than one,  $\varepsilon > 1$ , should not be chosen, as this will be a super elastic interaction, and the blob will have more kinetic energy than prior to the interaction.

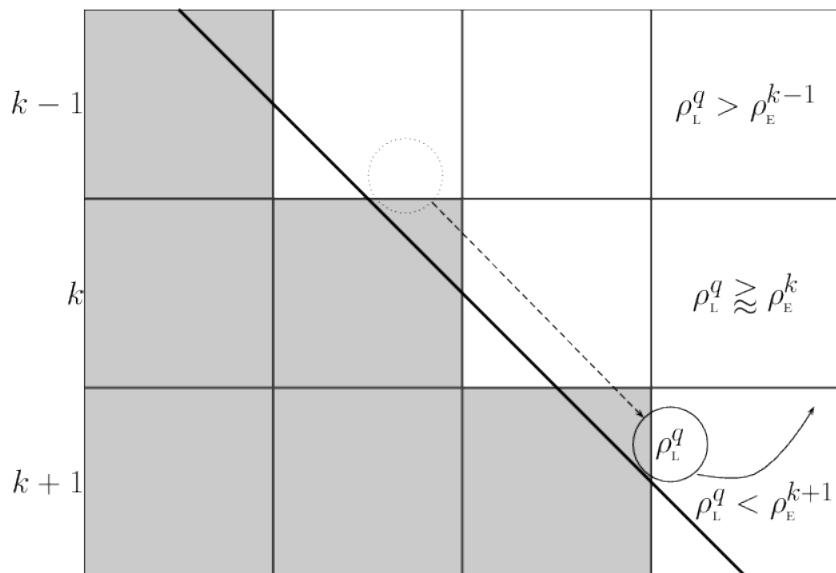


Figure 5.5: The process of boundary layer separation. The dashed circle indicates the blob's position at time  $\tau$ . The dashed line indicates the blob's pathway along the slope until some time later  $\tau + \Delta\tau$ . At this point, the separation criterion is met and the blob separates, changing dynamical regimes to the open ocean. The pseudo-non-hydrostatic term will thus give impetus to separate from the bottom, as indicated by the solid line arrow. Whether the blob continues to rise, or, stays at a relatively stable depth depends on the density structure and flow field of the E system. As with previous figures, a circle to represent a blob is only illustrative.



# Chapter 6

## Implementation of Grid Cell Partitioning in MOM

MOM has six supported vertical coordinate systems. There are three depth-based Boussinesq coordinate systems: geopotential,  $z^*$ ,  $\sigma$ , and three pressure-based non-Boussinesq coordinate systems: pressure,  $p^*$  and  $\sigma^{(p)}$ . For the purposes of these notes, we refer to geopotential,  $z^*$ , pressure and  $p^*$  as level coordinates. Chapter 6 of Griffies (2009) provides a detailed discussion of the supported coordinate systems in MOM.

One very important variable that allows MOM to have multiple coordinate systems is  $z_s$ , which has different values for each kind of vertical coordinate. While MOM has six different coordinate systems available, the embedded Lagrangian model has only been implemented in three of them ( $z^*$ , pressure and  $p^*$ ). It has not been implemented in geopotential,  $\sigma$  or  $\sigma^{(p)}$  coordinates, however in principle, there is no particular reason why the embedded Lagrangian framework should not work in any of the other coordinate systems.

Blobs moving from one grid cell to another affect the mass budget of a grid cell. A corollary is that the depth of vertical coordinate surfaces will be affected (except for geopotential coordinates at the bottom and in the interior). The calculation of the depth of the Eulerian system's coordinate surfaces with the embedded Lagrangian system presents an interesting order of operations problem when using a split time stepping scheme. This section also addresses how to numerically handle grid cell mass and tracer budgets, and, the order of operations required to do so.

### 6.1 General Numerical Considerations

There are some techniques that are not often employed in numerical ocean modelling which are used to implement the Lagrangian framework. The present section details the techniques used to implement the Lagrangian framework.

### 6.1.1 Numerical Representation of Blobs

The MOM is written in Fortran using the more modern syntax of Fortran 90. Some elements of the code use properties of the Fortran 95 standard (Metcalf et al., 2004). The ability to use pointers in Fortran 90 and later standards means that the natural numerical representation of the blobs is in a linked list (Newell and Shaw, 1957; Ellis et al., 1994), as linked lists are much more convenient than arrays for handling an arbitrary and potentially highly variable number of nodes.

Linked lists can have nodes added to them indefinitely, without having to undergo the costly process of increasing the size of an array. Similarly, nodes can easily be deleted from a linked list without then having the computational cost of resizing an array, or having the unnecessary memory overheads of maintaining very sparse arrays. In this way, the linked list is an elegant way to program the blobs, whose numbers may have great spatio-temporal variance both within individual simulations, but also between different simulations.

One disadvantage of using linked lists is that nodes may only be accessed sequentially and not randomly, thus increasing the number of operations required to find a particular node within the list. Another disadvantage, which can impact performance, is that the nodes will not generally be allocated sequentially in memory, unlike arrays where memory can be accessed sequentially. Accessing memory sequentially improves performance, as doing so reduces the number of times the processor must load memory into its cache. Despite the drawbacks of using linked lists, it is deemed that linked lists are suitable for our purposes.

In the MOM implementation, a doubly-linked list is used. While it is slightly more memory intensive, a doubly linked list is more efficient and convenient for inserting and deleting blobs from an arbitrary position within the list. The ability to efficiently insert or delete blobs in an arbitrary place in the list becomes particularly important for code parallelisation (see Section 6.1.3 for further details).

### 6.1.2 Order of Operations

Operator splitting is a numerical technique for solving partial differential equations. The technique splits the equations up into smaller well behaved parts to make them easier to solve. Splitting is achieved by taking an equation that can be described as the sum of a number of (not necessarily linear) operators. Each of these additive operators is solved for a fractional time-step, with the variable being solved for updated at the end of each fractional time-step.

Operator splitting is used in MOM, and while it is a convenient and efficient method to solve large and complex equations, there can be some sensitivity to the order of operation. For instance, surface forcing is usually handled implicitly in time. Since surface forcing is

a major factor in destabilising the water column, convective adjustment schemes should be applied after the surface forcing has been applied.

A consequence of the application of the surface forcing is that the formation and initial position of dynamically active open ocean blobs from the vertical instability formation criterion (see Section 5.2) must occur in MOM during the implicit tracer calculations as an adjustment. The movement of the dynamically active blobs at subsequent time steps need not necessarily occur during the implicit part of the tracer calculations, and indeed, it is more convenient to calculate their trajectories and effect on tracer fields explicitly in time. On the other hand, the on-shelf/off-shelf stability criterion is much less influenced by surface forcing and it is not necessary to form the resultant blobs during the implicit part of the tracer time step. Both the formation and the subsequent calculation of dynamically active blob trajectories and the exchange of properties with the Eulerian system is handled explicitly in time.

### 6.1.3 Parallelisation and Bitwise Reproducibility

Parallelisation of linked lists, while maintaining bitwise agreement of results across an arbitrary number of processors, increases the computational burden of maintaining the linked lists. MOM partitions the entire domain into smaller compute domains, with one compute domain assigned to each processor (see Chapter 4 of Griffies et al., 2004, for details of grids and halos used in MOM). When blobs can move laterally from one grid cell to another, there is the possibility of blobs then moving from one compute domain to another. The possibility of more than one blob coexisting in any given grid cell can then lead to reproducibility issues arising from machine error (see Chapter 1 of Atkinson, 1989, for a pedagogical discussion on different types of numerical errors). To illustrate, when working with finite precision, we have the situation that,

$$(a + b) + c \neq a + (b + c).$$

To eliminate differences in machine error across an arbitrary number of processors, we must ensure that when properties are exchanged between the Eulerian and Lagrangian system, numerical operations are always done in the same relative order.

To ensure that terms are always done in the same relative order, blobs must be processed in the same relative order. The most convenient way for blobs to be processed in the same order is to ensure that they always appear in the linked list in the same relative order. Each domain has its own list of blobs, and the number of blobs (and therefore, the order that blobs may appear in a list) may vary depending on the number of processors (and thus domains) used.

When blobs move from one domain to another, they must be packed into a buffer (an

array or series of arrays containing the blob’s properties), unpacked on the neighbouring compute domain and then inserted into the linked list for that domain. The requirements for bitwise agreement for parallel code adds significant complexity. We shall return to the issue of parallelisation with respect to the time stepping scheme in Section 7.3.5.

It is not, however, only bitwise reproduction that motivates us to maintain a strict ordering of the blob linked lists. As is discussed in Section 6.5.1, it is necessary, on occasion to destroy blobs to ensure that local mass conservation is obeyed. Blobs should be destroyed in a systematic manner, which assists in keeping simulations across different processors as similar as possible when not insisting on strict bitwise reproduction. We are thus required to order the blobs in a consistent manner and so the same list sorting algorithm is applied whether the user requires bitwise reproduction or not.

#### 6.1.4 Restart and Diagnostic Files

In MOM, the file format used for restart files, as well as diagnostic output is the NetCDF format (Rew and Davis, 1990). NetCDF is very well suited to the accurate, platform independent storage of multidimensional data. The gridded nature of the storage necessitates the use of two separate files to store model restart and diagnostic data associated with the Lagrangian blob framework. One file is required for data that naturally occurs on the model grid (for instance, the grid cell counter variable discussed in Section 7.2), and the other required to store individual blob data.

There are presently two NetCDF standards in use, the NetCDF3 standard and the NetCDF4 standard (Rew et al., 2010). The NetCDF3 standard allows for one unlimited dimension. The NetCDF4 standard offers some additional features such as multiple unlimited dimensions, groups, ragged arrays and user defined types. In theory, these additional features should be helpful in the logical storage and presentation of blob data. In practice, these new features remain poorly tested with a very small user base and the more convenient features for the blobs remain unreliable. It was thus deemed impractical to implement diagnostics using some of the new features of NetCDF4. We are thus restricted to using the NetCDF3 standard.

The file that stores blob data has two forms. One form stores individual blob data. The NetCDF3 allows for one unlimited dimension. For regular, gridded data, this dimension is usually time. For irregular data, such as individual blob data makes for some difficulty when attempting to write efficient diagnostics for individual blobs. The iceberg model of Martin and Adcroft (2010) was written under the NetCDF3 standard. The approach taken by the iceberg model is to store iceberg diagnostic data until the end of the model run. The diagnostics are then written sequentially, by iceberg, along the one unlimited dimension with empty entries separating the data from each iceberg. The disadvantage of such an approach is that it is not a user friendly format for storage and interpretation of data.

Despite the unfriendliness of the data formatting, pragmatism leads us to follow a technique similar to that of the iceberg model. Blobs are written in a single, unlimited dimension. Unlike the iceberg model, the blob data is written unordered, that is, data from a single blob at different times do not necessarily occur sequentially in the file. An unordered diagnostic file was chosen to save on memory and computational expense. Ordering of blob diagnostic files can be done as a post-processing step.

Another question concerns the frequency of output. The frequency of output is written at a certain frequency of Eulerian time steps, which is defined by the user in the diagnostic table (as is presently the case). Note that a blob's position is sub-cycled relative to the Eulerian time step, as discussed in Sections 7.3.1 and 7.3.2.

Similar to the existing Flexible Modelling System diagnostic manager is written based on regular gridded arrays of the NetCDF3 standard, the blobs have their own diagnostic manager to read, interpret and use the relevant diagnostic table data.

Aside from the diagnostics associated with individual blobs, we may wish to diagnose properties on the Eulerian grid. Some data, such as the blob counter, or, the blob grid cell convergence, can be handled by the existing diagnostic manager for MOM. The interpretation of some other types of diagnostics becomes troublesome, however, as blobs are not always present in any given grid cell. To illustrate, imagine we are taking an average of salinity. In one time step, there is a blob in the grid cell with a salinity of  $34\text{g kg}^{-1}$ . Then, the next time step, there is no blob in the grid cell. On average, the salinity of the Lagrangian system in that grid cell will be  $17\text{g kg}^{-1}$ .

The approach favoured is to instead take mass weighted averages of the combined system, as well as diagnostics for the Eulerian system. In this way, differences between the total system and the Eulerian system may be viewed as the Lagrangian system's contribution to the total system. By default, the standard variable names in the diagnostic table refer to the combined system, while an E is appended to the variable name to obtain the E system diagnostic value.

## 6.2 Implementation of Partitioning

In Chapter 3 a full derivation of mass, tracer, momentum and energy budgets of the individual and combined systems was done. Here, we discuss how these budgets are implemented in MOM. Notably, some terms in the momentum and energy budgets are not included.

### 6.2.1 Calculation of Thickness

Of central importance to the Lagrangian framework is the requirement that the combined Eulerian-Lagrangian system remains internally consistent as well as locally and globally conservative of seawater mass and tracer mass. Pressure gradient calculations and vertically integrated mass budgets must take into account material that is contained within the Lagrangian system, while still respecting the local partition of mass between the Lagrangian and the Eulerian systems.

Since the ocean model is now partitioned into an E and L region (see discussion in Section 3.2), not all of the material contained within a grid box will have the same properties. We illustrate this partition with the grid cell height. In the continuum, we write,

$$\int_T dz = \int_E dz + \int_L dz, \quad (6.1a)$$

while the equivalent discrete equations are,

$$dz_T^{ijk} = dz_E^{ijk} + dz_L^{ijk}, \quad (6.1b)$$

where we note that,

$$dz_L^{ijk} = \sum_{q=1}^{\mathcal{Q}^{ijk}} dz^q \quad (6.1c)$$

$$= (dA^{ij})^{-1} \sum_{q=1}^{\mathcal{Q}^{ijk}} \frac{m_L^q}{\rho_L^q}, \quad (6.1d)$$

where  $dz_E^{ijk}$  is the thickness contribution by the E partition,  $dz_L^{ijk}$  is the thickness contribution from the L partition,  $dz_T^{ijk}$  combined thickness of the E and L partitions,  $m_L^q$  is the mass of the  $q^{\text{th}}$  blob,  $\rho_L^q$  is the density of the  $q^{\text{th}}$  blob, where we recall that  $V = m/\rho$ , and  $dA^{ij}$  is the horizontal area of grid boxes located in the  $(i, j)$  water column. The thickness contribution by the L partition is made up of the sum of contributions from  $\mathcal{Q}$  individual blobs in the grid box  $(i, j, k)$ . The physical interpretation of  $dz_L$  is illustrated in the right hand side of Figure 6.5.

### 6.2.2 Tracer Budgets

The discrete equation for the total tracer content of a grid cell can be written as the sum of the tracer content of the E partition and the content from  $\mathcal{Q}$  Lagrangian blobs that exist in that tracer cell

$$(dA^{ij} dz_T^{ijk}) (\rho_T^{ijk} C_T^{ijk}) = (dA^{ij} dz_E^{ijk}) (\rho_E^{ijk} C_E^{ijk}) + \sum_{q=1}^{\mathcal{Q}^{ijk}} [(dA^{ij} dz_L^q) (\rho_L^q C_L^q)]. \quad (6.2)$$

The horizontal area of the grid cell  $dA^{ij}$  is independent of time and depth. It can thus be cancelled out as a common factor to give the tracer content per unit area of a grid cell

$$dz_T^{ijk} \left( \rho_T^{ijk} \mathcal{C}_T^{ijk} \right) = dz_E^{ijk} \left( \rho_E^{ijk} \mathcal{C}_E^{ijk} \right) + \sum_{q=1}^{\mathcal{Q}^{ijk}} [dz_L^q (\rho_L^q \mathcal{C}_L^q)]. \quad (6.3)$$

As seen from this equation, we can weight variables in the Eulerian model by the E system's thickness,  $dz_E^{ijk}$ . By setting  $\mathcal{C}$  to a constant globally it can be seen that compatibility with the thickness weighted density is also maintained.

The rate of change for the thickness weighted tracer budget for interior, bottom and surface grid cells of the total system is, respectively (from equations (3.165), (3.183) and (3.194) of Griffies, 2009),

$$\partial_t (dz \rho \mathcal{C})_T^{ijk} = (dz \rho \mathcal{S}^{(C)})_T^{ijk} - \nabla \cdot [dz \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_T^{ijk} - [\rho(w \mathcal{C} + F^{(s)})]_T^{ijk-1} + [\rho(w \mathcal{C} + F^{(s)})]_T^{ijk} \quad (6.4a)$$

$$\partial_t (dz \rho \mathcal{C})_T^{ijk_{\text{bot}}} = (dz \rho \mathcal{S}^{(C)})_T^{ijk_{\text{bot}}} - \nabla \cdot [dz \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_T^{ijk_{\text{bot}}} - [\rho(w \mathcal{C} + F^{(s)}) + z_s \nabla \cdot \mathbf{F}]_T^{ijk_{\text{bot}}-1} \quad (6.4b)$$

$$\partial_t (dz \rho \mathcal{C})_T^{ij1} = (dz \rho \mathcal{S}^{(C)})_T^{ij1} - \nabla \cdot [dz \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_T^{ij1} + [\rho(w \mathcal{C} + z_s \nabla \cdot \mathbf{F})]_T^{ij1} + (\mathcal{Q}_W \mathcal{C}_W - \mathcal{Q}_{(C)}^{(\text{turb})})_T^{ij}, \quad (6.4c)$$

where  $k_{\text{bot}}$  is the bottom ocean grid cell. Here,  $\mathbf{F}$  is the horizontal tracer concentration flux and  $F^{(z)}$  is the vertical concentration flux. Note that we have ignored the possibility of tracer entering through the bottom boundary (for example, geothermal sources),  $\mathcal{Q}_{(C)}^{(\text{turb})}$  is the tracer due to turbulent fluxes,  $\mathcal{Q}_W$  is the mass flux of water per unit time crossing the ocean surface ( $\text{kg m}^{-2}\text{s}^{-1}$ ),  $\mathcal{C}_W$  is the tracer concentration of water crossing the ocean surface,  $\mathbf{F}$  is the subgrid-scale (SGS) tracer concentration flux vector introduced in Equation (3.12) and we have taken  $k = 2, \dots, k_{\text{bot}} - 1$  to denote vertical levels.

Given the discussion in Section 3.3, we must include the exchange of properties between the E and L system, as well as blob convergence into the discrete grid cell seawater mass and tracer mass budgets. We rewrite the semi-discrete tracer budget Equation (3.20) in discrete form for an interior grid cell, vis-à-vis Equation (6.4a),

$$\begin{aligned} \partial_t (dz \rho \mathcal{C})_T^{ijk} &= - \nabla \cdot [dz \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_E^{ijk} - [\rho(w \mathcal{C} + F)]_E^{ijk-1} + [\rho(w \mathcal{C} + F)]_E^{ijk} \\ &\quad - \sum_{q=1}^{\mathcal{Q}_{(out)}^{ijk}} \frac{(dz \rho \mathcal{C})_L^q}{dt} + \sum_{q=1}^{\mathcal{Q}_{(in)}^{ijk}} \frac{(dz \rho \mathcal{C})_L^q}{dt}. \end{aligned} \quad (6.5a)$$

Similarly, the new budgets for bottom and surface grid cells are, respectively,

$$\begin{aligned} \partial_t (\mathrm{d}z \rho \mathcal{C})_{\mathrm{T}}^{ijk_{\mathrm{bot}}} &= -\nabla \cdot [\mathrm{d}z \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_{\mathrm{E}}^{ijk_{\mathrm{bot}}} - [\rho (w \mathcal{C} + F)]_{\mathrm{E}}^{ijk_{\mathrm{bot}}-1} \\ &\quad + (z_s \nabla \cdot \mathbf{F})_{\mathrm{E}}^{ijk_{\mathrm{bot}}-1} - \sum_{q=1}^{\mathcal{Q}_{(\mathrm{out})}^{ijk_{\mathrm{bot}}}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}t} + \sum_{q=1}^{\mathcal{Q}_{(\mathrm{in})}^{ijk_{\mathrm{bot}}}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}t} \end{aligned} \quad (6.5\mathrm{b})$$

$$\begin{aligned} \partial_t (\mathrm{d}z \rho \mathcal{C})_{\mathrm{T}}^{ij1} &= -\nabla \cdot [\mathrm{d}z \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_{\mathrm{E}}^{ij1} + [\rho (w \mathcal{C} + F)]_{\mathrm{E}}^{ij1} + (\mathcal{Q}_{\mathrm{W}} \mathcal{C}_{\mathrm{W}})_{\mathrm{E}}^{ij} \\ &\quad - (\mathcal{Q}_{(\mathrm{C})}^{(\mathrm{turb})})_{\mathrm{E}}^{ij} - \sum_{q=1}^{\mathcal{Q}_{(\mathrm{out})}^{ij1}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}t} + \sum_{q=1}^{\mathcal{Q}_{(\mathrm{in})}^{ij1}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}t}. \end{aligned} \quad (6.5\mathrm{c})$$

Thus, when implementing the total tracer budget of a grid cell we are required to create a variable for the blob convergence of tracer.

### 6.2.3 Compatibility of Tracer Mass and Seawater Mass

The continuous non-Boussinesq tracer conservation equation is

$$\partial_t (\rho \mathcal{C}) + \nabla \cdot (\rho \mathbf{v} \mathcal{C}) = \rho \mathcal{S}^{(\mathcal{C})} - \nabla \cdot (\rho \mathbf{F}). \quad (6.6)$$

In the continuous equations, conservation of mass reduces from the tracer conservation Equation (6.6) by setting  $\mathcal{C}$  to a non-zero constant globally

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = \mathcal{S}^{(M)}. \quad (6.7)$$

These equations can be discretised, and from them local seawater mass and tracer mass budgets can be built. While the compatibility between the tracer mass and seawater mass conservation equations follows naturally in the continuous equations, it is very important that compatibility is maintained in the discrete equations (Griffies et al., 2001). Formulating discrete equations that respect the compatibility condition ensures that global and local conservation of tracer mass and seawater mass is maintained.

While the column integrated tracer is not time stepped, vertically integrating the tracer concentration equation over an ocean column is an important test of compatibility. If we combine and sum equations (6.5a) to (6.5) from  $k = 1$  to  $k = k_{\mathrm{bot}}$ , we get

$$\begin{aligned} \partial_t \left( \sum_{k=1}^{k_{\mathrm{bot}}} [\mathrm{d}z \rho \mathcal{C}]_{\mathrm{T}}^{ijk} \right) &= -\nabla \cdot \left( \sum_{k=1}^{k_{\mathrm{bot}}} [\mathrm{d}z \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_{\mathrm{E}}^{ijk} \right) + \sum_{q=1}^{\mathcal{Q}_{(\mathrm{in})}^{ij}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}t} \\ &\quad - \sum_{q=1}^{\mathcal{Q}_{(\mathrm{out})}^{ij}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}t} + (\mathcal{Q}_{\mathrm{W}} \mathcal{C}_{\mathrm{W}} - \mathcal{Q}_{(\mathrm{C})}^{(\mathrm{turb})})_{\mathrm{E}}^{ij}. \end{aligned} \quad (6.8)$$

where  $Q^{ij}$  denotes all blobs in the water column, that is,

$$\sum_{q=1}^{Q^{ij}} \equiv \sum_{k=1}^{k_{\text{bot}}} \sum_{q=1}^{Q^{ijk}} . \quad (6.9)$$

We can see that compatibility is maintained when setting  $\mathcal{C} \equiv \text{const}$  globally (recalling that  $\mathbf{F}(\mathcal{C} = \text{const}) = 0$  and  $F(\mathcal{C} = \text{const}) = 0$  and noting that  $\mathcal{Q}_{(\text{C})}^{(\text{turb})}(\mathcal{C} = \text{const}) = 0$ ),

$$\partial_t \left( \sum_{k=1}^{k_{\text{bot}}} [\mathrm{d}z \rho]_{\text{T}}^{ijk} \right) = -\nabla \cdot \left( \sum_{k=1}^{k_{\text{bot}}} [\mathrm{d}z \rho \mathbf{u}]_{\text{E}}^{ijk} \right) + \sum_{q=1}^{Q^{(\text{in})}} \frac{(\mathrm{d}z \rho)_{\text{L}}^q}{\mathrm{d}t} - \sum_{q=1}^{Q^{(\text{out})}} \frac{(\mathrm{d}z \rho)_{\text{L}}^q}{\mathrm{d}t} + (\mathcal{Q}_{\text{W}})^{ij} . \quad (6.10)$$

With this formulation for the embedding of the L system the compatibility between the budgets for seawater mass and tracer mass is maintained, thus satisfying a necessary condition for maintaining conservation.

#### 6.2.4 Momentum Budgets

In Section 3.4 it was shown that the momentum budget of the combined hydrostatic Eulerian system and the psuedo-nonhydrostatic Lagrangian system cannot be closed due to the lack of vertical momentum in the Eulerian system. There are several other terms in the momentum budget that are not included in the implementation in MOM. The terms that are excluded in the prognostic momentum equations is that of the change in momentum due to the change in mass of a blob, Equation (3.24). As is detailed below, there are technical issues that must be overcome to solve the full equations, and, there is also some physical motivation for neglecting these terms.

Solving the full momentum Equation (3.24) for a blob gives the momentum ( $m\dot{\mathbf{x}}$ ), rather than position,  $\mathbf{x}$ . Position is an important variable for the interpolation schemes and point locations required by the L system. In order to solve for position from the momentum equation requires a division by mass

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = (m\dot{\mathbf{x}})/m . \quad (6.11)$$

Experience has shown that  $m$  is frequently  $O(10^{15})\text{kg}$  or more and  $\dot{\mathbf{x}}$  is frequently  $O(10^{-2})\text{m s}^{-1}$  or less. In the momentum equation for an individual blob, the alteration of the momentum of the blob due to changes in mass is thus ignored, such that

$$\frac{\mathrm{d}}{\mathrm{d}t}(m\dot{\mathbf{x}}) \approx m \frac{\mathrm{d}\dot{\mathbf{x}}}{\mathrm{d}t} . \quad (6.12)$$

Such an approximation easily allows the momentum per unit mass to be readily solved (and thus, the position), rather than the full momentum. This provides an easy means to overcome the division problem mentioned above, as the change in momentum due to the change in velocity can be divided through by mass to yield a convenient equation for

solving for position:

$$\frac{d\dot{\mathbf{x}}}{dt} = -2\boldsymbol{\Omega} \wedge \dot{\mathbf{x}} + \mathbf{g} - \rho^{-1}\nabla p + \rho^{-1}\nabla \cdot \boldsymbol{\tau}. \quad (6.13)$$

A corollary of this assumption is that, from the blob's perspective, mass that is transferred out of, and into, the blob is transferred at the momentum per unit mass (i.e. the velocity) of the blob. Such an assumption about the entrainment is consistent with the turbulent entrainment assumption for gravity currents in a homogenous environment at rest, as outlined in Section 6 of Turner (1986). It is also consistent with the streamtube formulation of Price and Baringer (1994).

From the perspective of the E system, however, the fluid that is entrained by a blob is considered to have been transferred to the L system at the momentum per unit mass of the E system. The converse assumption is made with the detrained mass. From the L system's perspective mass that is detrained is assumed to be detrained at the momentum per unit mass of the blob, while it is assumed to have the momentum per unit mass of the grid cell of the surrounding E cell from the E system's perspective.

The assumptions about the momentum per unit mass are made for all mass transferred between the E system and L system, including for new blobs and destroyed blobs. For the purpose of formulating momentum budgets for grid cells and indeed, the model as a whole, these assumptions about the transfer of momentum lead to source terms given by the semi-discrete equations

$$dA^{ij} \int_{\Delta\tau} \mathcal{S}_E^{(v)ijk} dz^{ijk} dt = \mathbf{u}^{ijk} \left( - \sum_{q=1}^{Q_{(\text{new})}^{ijk}} m_{(\text{new})}^q + \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} m_{(\text{dstry})}^q + \sum_{q=1}^{Q^{ijk}} \left\{ -dm_{(\text{E2L})}^q + dm_{(\text{L2E})}^q \right\} \right), \quad (6.14a)$$

while the L system's equations remain the same,

$$dA^{ij} \int_{\Delta\tau} \mathcal{S}_L^{(v)ijk} dz^{ijk} dt = \sum_{q=1}^{Q_{(\text{new})}^{ijk}} [m\dot{\mathbf{x}}]_{(\text{new})}^q - \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} [m\dot{\mathbf{x}}]_{(\text{dstry})}^q + \sum_{q=1}^{Q^{ijk}} \left\{ d[m\dot{\mathbf{x}}]_{(\text{E2L})}^q - d[m\dot{\mathbf{x}}]_{(\text{L2E})}^q \right\}. \quad (6.14b)$$

The alteration to the source terms significantly modifies the grid cell momentum budget

Equation (3.33),

$$\begin{aligned}
 dA^{ij} \partial_t (\rho v)_T^{ijk} dz_T^{ijk} = & - \oint_{S_E^{ijk}} (\mathbf{v}^{(s)} \cdot \hat{\mathbf{n}})^{ijk} (\rho \mathbf{u})_E^{ijk} dS + \frac{1}{\Delta \tau} \left( \sum_{q=1}^{Q_{(\text{in})}^{ijk}} (m \dot{\mathbf{x}})_{(\text{in})}^q - \sum_{q=1}^{Q_{(\text{out})}^{ijk}} (m \dot{\mathbf{x}})_{(\text{out})}^q \right) \\
 & + \frac{1}{\Delta \tau} \sum_{q=1}^{Q^{ijk}} [\mathbf{G}^q - V^q (\hat{\mathbf{x}} \partial_x, \hat{\mathbf{y}} \partial_y) p^q] + \sum_{q=1}^{Q^{ijk}} \frac{g V^q (\rho_E - \rho_L^q)}{\Delta \tau} \hat{\mathbf{z}} + \oint_{S_E^{ijk}} (\hat{\mathbf{n}} \cdot \boldsymbol{\tau})^{ijk} dS \\
 & - dA^{ij} dz^{ijk} \rho_E (f + \mathcal{M}) \hat{\mathbf{z}} \wedge \mathbf{u}_E - \frac{1}{\Delta \tau} \sum_{q=1}^{Q^{ijk}} m^q (2\Omega + \mathcal{M} \hat{\mathbf{z}}) \wedge \dot{\mathbf{x}}^q + \oint_{S_E^{ijk}} (\hat{\mathbf{n}} p)^{ijk} dS \\
 & + \sum_{q=1}^{Q_{(\text{new})}^{ijk}} \frac{m_{(\text{new})}^q}{\Delta \tau} (\dot{\mathbf{x}}_{(\text{new})}^q - \mathbf{u}^{ijk}) + \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} \frac{m_{(\text{dstry})}^q}{\Delta \tau} (\mathbf{u}^{ijk} - \dot{\mathbf{x}}_{(\text{dstry})}^q) \\
 & + \frac{1}{\Delta \tau} \sum_{q=1}^{Q^{ijk}} \left\{ dm_{(\text{E2L})}^q (\dot{\mathbf{x}}^q - \mathbf{u}^{ijk}) + dm_{(\text{L2E})}^q (\mathbf{u}^{ijk} - \dot{\mathbf{x}}^q) \right\}. \tag{6.15}
 \end{aligned}$$

It is important to note that  $\dot{\mathbf{x}}$  is a three dimensional velocity and  $\mathbf{u}$  is a two dimensional velocity. So, a major difference between a purely E system and a combined E and L system is the addition of vertical velocity components. The implementation of the momentum budget in MOM, Equation (6.15), has some non-conservative momentum terms when it comes to the exchange of momentum between the two systems.

### 6.2.5 Energy Budgets

There is no alteration, in the implementation of the Lagrangian blobs in MOM, to the gravitational potential energy budget described by Equation (3.41).

Unlike the potential energy budget, there are modifications to the kinetic energy which are analogous to the modifications made to the momentum budget, discussed in Section 6.2.4. Firstly, the source terms become,

$$\mathcal{S}_{E^{ijk}}^{(\mathcal{K})} = - \sum_{q=1}^{Q_{(\text{new})}^{ijk}} m^q (\mathbf{u}^{ijk})^2 + \sum_{q=1}^{Q^{ijk}} (\mathbf{u}^{ijk})^2 \left( dm_{(\text{L2E})}^q - dm_{(\text{E2L})}^q \right) + \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} m^q (\mathbf{u}^{ijk})^2 \tag{6.16a}$$

$$\mathcal{S}_{L^{ijk}}^{(\mathcal{K})} = \sum_{q=1}^{Q_{(\text{new})}^{ijk}} m^q (\dot{\mathbf{x}}^q)^2 + \sum_{q=1}^{Q^{ijk}} (\dot{\mathbf{x}}^q)^2 \left( dm_{(\text{E2L})}^q - dm_{(\text{L2E})}^q \right) + \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} m^q (\dot{\mathbf{x}}^q)^2. \tag{6.16b}$$

Rewriting the kinetic energy budget Equation (3.53) in a semi-discrete form that reflects

the implementation of the Lagrangian system in MOM,

$$\begin{aligned}
 dA^{ij} \partial_t (\rho \mathcal{K})_T^{ijk} dz_T^{ijk} = & - \oint_{S_E^{ijk}} (\mathbf{u}_E \cdot \hat{\mathbf{n}})^{ijk} (\rho_E \mathcal{K}_E^{(h)})^{ijk} dS - \oint_{S_E^{ijk}} p^{ijk} (\hat{\mathbf{n}} \cdot \mathbf{v}_E)^{ijk} dS \\
 & - (dAdz_E)^{ijk} \rho_E^{ijk} \frac{d\Phi_E^{ijk}}{dt} + (dAdz_E)^{ijk} (\rho \mathbf{u} \cdot \mathbf{F}^{(u)})_E^{ijk} \\
 & + \frac{1}{\Delta\tau} \sum_{q=1}^{Q^{ijk}} \left\{ V^q \left[ g\dot{z}(\rho_E - \rho_L) - \dot{x}\partial_x p - \dot{y}\partial_y p + \dot{\mathbf{x}} \cdot \mathbf{G} \right]^q \right\} \\
 & + \frac{1}{\Delta\tau} \left\{ \sum_{q=1}^{Q_{(in)}^{ijk}} (m\dot{\mathbf{x}}^2)^q - \sum_{q=1}^{Q_{(out)}^{ijk}} (m\dot{\mathbf{x}}^2)^q \right\} + \sum_{q=1}^{Q_{(dstry)}^{ijk}} m^q ((\mathbf{u}^{ijk})^2 - (\dot{\mathbf{x}}^q)^2) \\
 & + \sum_{q=1}^{Q_{(new)}^{ijk}} m^q ((\dot{\mathbf{x}}^q)^2 - (\mathbf{u}^{ijk})^2) + \sum_{q=1}^{Q^{ijk}} ((\mathbf{u}^{ijk})^2 - (\dot{\mathbf{x}}^q)^2) (dm_{(L2E)}^q - dm_{(E2L)}^q)
 \end{aligned} \tag{6.17}$$

## 6.3 Discretisation Considerations

There are a number of discretisation considerations when embedding the Lagrangian model into an Eulerian model. Detailed here are some of the specific considerations that necessitate code changes from a purely Eulerian model.

### 6.3.1 Vertical Coordinate Considerations

The density weighted thickness of a grid cell is extremely important to the way that MOM handles generalised level coordinates (see Chapters 4–7 of Griffies, 2009). Since we are partitioning grid cells into E and L sections, the technology used by MOM to handle the thickness weighted density, as well as the depth of grid cell centres and bottom of grid cells must be modified.

The thickness of a grid cell is given by

$$dz = z_{,s} ds, \tag{6.18}$$

where  $s$  is a generalised level coordinate surface. The thickness Equation (6.18) is reformulated as

$$dz_E + dz_L = z_{,s} ds. \tag{6.19}$$

In order to implement the partition of thickness contributions from the E and L systems, we must first partition the dimensions of a tracer grid cell. The different variables that define the vertical dimensions of a grid cell are illustrated in Figure 6.1.  $dz_{WT}$  is the distance between tracer points for  $0 < k < k_{bot}$ , where  $k_{bot}$  is the bottom ocean grid cell. For  $k = 0$ ,  $dz_{WT}$  is the distance between the surface boundary and the tracer point,

while for  $k = k_{\text{bot}}$  it is the distance between the tracer point and the bottom boundary.  $\text{dzwtT}$  is made up of contributions from the E and the L partition,

$$\text{dzwtT} = \text{dzwtE} + \text{dzwtL}. \quad (6.20)$$

$\text{dzwtT}$  can also be interpreted as the total thickness of vertical velocity grid cells.

$\text{dztT}$  is the thickness of a tracer grid cell, with  $1 \leq k \leq k_{\text{bot}}$ , and it is also comprised of contributions from the E and L systems,

$$\text{dztT} = \text{dztE} + \text{dztL}. \quad (6.21)$$

Similarly  $\text{dztupT}$  is the distance from the top of a grid cell to the tracer point and  $\text{dztloT}$  is the distance from a tracer point to the bottom of a grid cell. They also have contributions from the E and L systems

$$\text{dztupT} = \text{dztupE} + \text{dztupL} \quad (6.22a)$$

$$\text{dztloT} = \text{dztloE} + \text{dztloL}. \quad (6.22b)$$

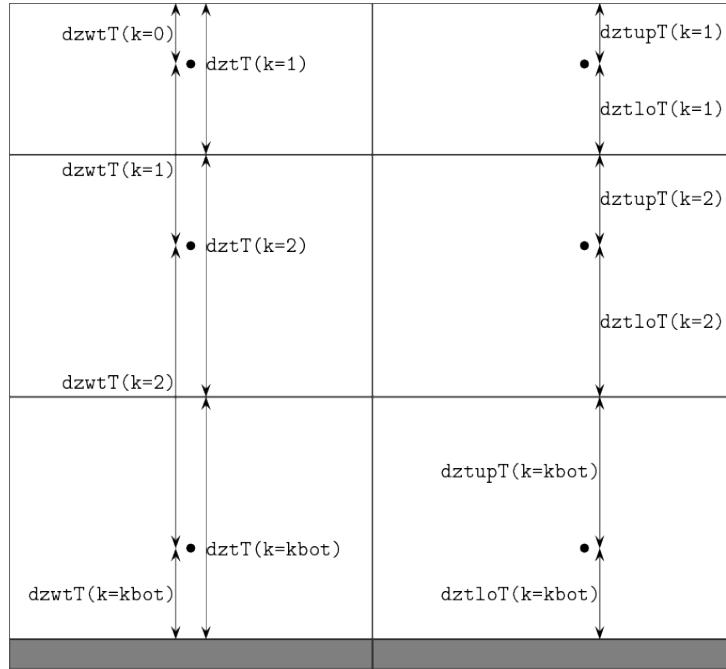


Figure 6.1: A “vertical slice” of tracer cell dimensions using the example of three grid cells (adapted from Figure 7.1 of Griffies, 2009). The solid black dots represent the tracer grid point, the boxes represent the grid box and the shaded section are the solid Earth. The various dimensions of the tracer cell is indicated. See text for definitions of variables.

All E partition variables should be treated as they are in a purely Eulerian system. Thus dropping the E suffix from model variables representing the E partition makes the-

oretical sense, and, also minimises trivial changes to the existing code base, so hereafter

$$\begin{aligned}
 \text{dzt} &\equiv \text{dztE} \\
 \text{dzwt} &\equiv \text{dzwtE} \\
 \text{dztup} &\equiv \text{dztupE} \\
 \text{dztlo} &\equiv \text{dztloE} \\
 \text{rho\_dzt} &\equiv \text{rho\_dztE},
 \end{aligned} \tag{6.23}$$

where  $\text{rho\_dzt}$  is the thickness weighted density,  $(\rho dz)$ . The same notation is adopted for variables on the velocity grid (see Section 6.4).

We now have three types of thickness variables. The total thickness and L system thickness are prognostic, while the E system thickness is diagnostic. The total thickness needs to be calculated in the same way that the grid cell thickness is calculated in an unpartitioned model,

$$\text{dztT}^{ijk} = z_{,s}^{ijk} \text{dst}^{ijk}, \tag{6.24a}$$

where  $\text{dst}$  is the grid cell's thickness in the model's native vertical coordinate system. The Lagrangian thickness is calculated as the sum of blob volumes divided by the area of the horizontal grid cell of residence,

$$\text{dztL}^{ijk} = \sum_{q=1}^{Q^{ijk}} \frac{m^q}{\rho_L^q dA^{ij}}. \tag{6.24b}$$

From the above two equations, we may then find the remaining Eulerian contribution to the total thickness,

$$\text{dzt}^{ijk} = \text{dztT}^{ijk} - \text{dztL}^{ijk}. \tag{6.24c}$$

The upper and lower portions of a grid cell thickness are calculated in the same manner

$$\text{dztlo}^{ijk} = \text{dztloT}^{ijk} - \text{dztloL}^{ijk} \tag{6.25a}$$

$$\text{dztup}^{ijk} = \text{dztupT}^{ijk} - \text{dztupL}^{ijk} \tag{6.25b}$$

noting that this requires us to differentiate between whether a blob is shallower or deeper than a tracer point

$$\text{dztloL}^{ijk} = \sum_{q=1}^{Q_{(1o)}^{ijk}} \frac{m^q}{\rho^q dA^{ij}} \tag{6.26a}$$

$$\text{dztupL}^{ijk} = \sum_{q=1}^{Q_{(up)}^{ijk}} \frac{m^q}{\rho^q dA^{ij}}, \tag{6.26b}$$

where  $Q_{(1o)}^{ijk}$  is the number of blobs in a grid cell below the tracer point and  $Q_{(up)}^{ijk}$  is the

number of blobs in a grid cell above the tracer point.

For calculating  $\text{dzwtT}$ , there are two methods available, the energetic method and the finite volume method. We review these two methods, which are detailed in Section 7.3.2 of Griffies (2009). The energetic method takes the form,

$$ds = \overline{s}_z dz, \quad (6.27)$$

where we note that  $\overline{()^z}$  indicates an unweighted average vertically over two cells,

$$\overline{a}^z = \frac{a^k + a^{k+1}}{2}.$$

Using the energetic approach, we write

$$\text{dzwtT}^k = \begin{cases} \text{dswt}^0 z_s^1 & \text{for } k = 0 \\ \text{dswt}^k \overline{z_s^k}^z & \text{for } 0 < k < k_{\text{bot}} \\ \text{dswt}^{k_{\text{bot}}} z_s^{k_{\text{bot}}} & \text{for } k = k_{\text{bot}}, \end{cases} \quad (6.28)$$

where  $\text{dswt}$  is the vertical distance between tracer grid cells in the native vertical coordinate of the model. It is not possible to calculate  $\text{dzwtL}$  using the energetic approach.

The finite volume approach, however, is calculated as,

$$\text{dzwtT}^k = \begin{cases} \text{dztupT}^1 & \text{for } k = 0 \\ \text{dztloT}^k + \text{dztupT}^{k+1} & \text{for } 0 < k < k_{\text{bot}} \\ \text{dztloT}^{k_{\text{bot}}} & \text{for } k = k_{\text{bot}}. \end{cases} \quad (6.29)$$

For the calculation of the Lagrangian contribution for the distance between tracer points, there is no equivalent procedure to the energetic method for the total thickness. As such, it may only be calculated in a finite volume sense,

$$\text{dzwtL}^k = \begin{cases} \text{dztupL}^1 & \text{for } k = 0 \\ \text{dztloL}^k + \text{dztupL}^{k+1} & \text{for } 0 < k < k_{\text{bot}} \\ \text{dztloL}^{k_{\text{bot}}} & \text{for } k = k_{\text{bot}}. \end{cases} \quad (6.30)$$

Finally, we calculate the Eulerian contribution as,

$$\text{dzwt}^k = \text{dzwtT}^k - \text{dzwtL}^k \quad \text{for } 0 \leq k \leq k_{\text{bot}}. \quad (6.31)$$

This is the only instance in the Lagrangian formulation that is incompatible with the energetic approach. As such, one may wish to proceed using an energetic approach for all variables except for the calculation of  $\text{dzwtL}$ . Experience indicates that stability and conservation are maintained when using this approach, however, if the objective is a complete and consistent treatment of variables, then the finite volume method should be the preferred choice.

The calculation of `rho_dzt` is done in a similar manner,

$$\text{rho\_dzt}^{ijk} = \text{rho\_dztT}^{ijk} - \text{rho\_dztL}^{ijk}. \quad (6.32)$$

The calculation of `rho_dztT`<sup>*ijk*</sup> must be considered in a slightly different manner, however, to the way that it is calculated in standard MOM. Before we can understand why it must be done differently, we must understand the timestepping and order of operations of the standard release of MOM, which is not discussed until Section 6.5. We shall thus postpone discussing the calculation of `rho_dztT`<sup>*ijk*</sup> until Section 6.5.1. We shall also return to a discussion on the calculation of  $z_s$  in Section 6.5.1, also after the discussion on timestepping in MOM given in Section 6.5.

### 6.3.2 Calculation of Total Density, Tracer Concentration and Buoyancy Frequency

The E system's density,  $\rho_E$ , is a function of E system variables,

$$\rho_E^{ijk} = \rho(\Theta_E^{ijk}, S_E^{ijk}, p^{ijk}), \quad (6.33)$$

where the density and pressure (given at a tracer point) are diagnostic variables, while the temperature and salinity are prognostic variables. The density of individual blobs is similarly given,

$$\rho_L^q = \rho(\Theta_L^q, S_L^q, p^{ijk}), \quad (6.34)$$

where density and tracer concentrations are diagnostic variables. The pressure is a reference pressure, which we take to be the pressure at the tracer point in the grid cell where the blob resides.

To find the combined density of the system, we sum the mass of the two systems and divide by the combined volume,

$$\begin{aligned} \rho_T^{ijk} &= \frac{m_T^{ijk}}{V_T^{ijk}} \\ &= \frac{m_E^{ijk} + m_L^{ijk}}{V_E^{ijk} + V_L^{ijk}} \\ &= \frac{(\rho V)_E^{ijk} + \sum_{q=1}^Q m_L^q}{V_E^{ijk} + \sum_{q=1}^Q (m/\rho)_L^q} \end{aligned} \quad (6.35a)$$

As shall be seen later, the density of the combined system is required for the calculation of hydrostatic pressure.

Similarly, to find the total tracer content of a grid cell with  $Q$  blobs residing in it, we sum the tracer content of the E system and all of the blobs and divide by the combined

mass,

$$\mathcal{C}_{\text{T}}^{ijk} = \frac{dA^{ij}(\rho dz)_{\text{E}}^{ijk} \mathcal{C}_{\text{E}}^{ijk} + \sum_{q=1}^{Q^{ijk}} (m\mathcal{C})^q}{dA^{ij}(\rho dz)_{\text{E}}^{ijk} + \sum_{q=1}^{Q^{ijk}} m^q}. \quad (6.36)$$

We use the combined tracer concentration to calculate the square of the buoyancy frequency of the combined system. The square of the buoyancy frequency is used as the formation condition for new dynamically active free blobs. For the calculation of the square of the buoyancy frequency of the combined system,  $N_{\text{T}}^2$ , we use the equation of state with the thickness-density weighted average of temperature and salinity as the input values. We write the square of the buoyancy frequency, and the combined temperature and salinity as,

$$N_{\text{T}}^2 = -\frac{g}{\rho_0} \left( \frac{\partial \rho}{\partial \Theta} \frac{\partial \Theta}{\partial z} + \frac{\partial \rho}{\partial S} \frac{\partial S}{\partial z} \right)_{\text{T}}, \quad (6.37)$$

where  $\Theta_{\text{T}}$  and  $S_{\text{T}}$  are calculated using Equation (6.36).

### 6.3.3 Calculation of Hydrostatic Pressure

Use of generalised vertical coordinates complicates the calculation of horizontal pressure gradients in a hydrostatic model by introducing geopotential gradients along coordinate surfaces (see Section 3.7.2 of Griffies, 2009) when taking the horizontal pressure gradient

$$\rho^{-1} \nabla_z p = \rho^{-1} \nabla_s p + \nabla_s \Phi, \quad (6.38)$$

where the hydrostatic balance,

$$\partial_z p = -\rho g, \quad (6.39a)$$

geopotential,

$$\Phi = gz, \quad (6.39b)$$

and the following operator transformation,

$$\nabla_z = \nabla_s - (\nabla_s z) \partial_z, \quad (6.39c)$$

have been used. The non alignment of geopotential surfaces and coordinate surfaces is illustrated in Figure 6.2.

To calculate the total hydrostatic pressure from the combined E and L systems we require a method for combining the contributions from both systems. We begin by looking at the hydrostatic pressure at a point in depth based vertical coordinates in a purely Eulerian system (from Equation (4.12) of Griffies, 2009)

$$p = p_A + \rho_0 g \eta - \rho_0 \Phi + p', \quad (6.40a)$$

where  $p_A$  is the applied pressure at the ocean surface,  $\rho_0$  is a reference density and  $p'$  is the anomalous pressure,

$$p' = g \int_z^\eta \rho' dz, \quad (6.40b)$$

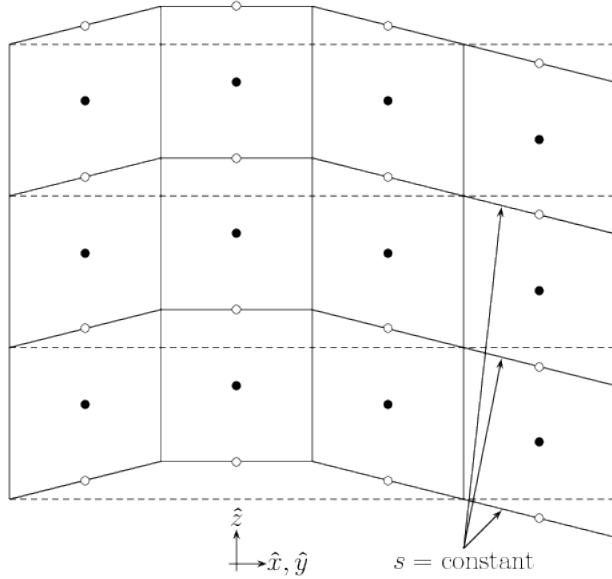


Figure 6.2: Adapted from Figure 4.1 of Griffies (2009). A vertical slice is taken to illustrate the shape of grid cells (the boxes) relative to their tracer grid points ( $\mathcal{T}$ , filled circles) and vertical velocity grid points ( $\mathcal{W}$ , unfilled circles) when using generalised level vertical coordinates. This figure also illustrates how constant geopotential surfaces (dashed lines) and coordinate surfaces are not generally aligned when using vertical coordinates other than geopotential coordinates.

where  $\rho'$  is the anomalous density, defined by  $\rho' = \rho - \rho_0$ . The second term on the right hand side of Equation (6.40a) is a result of the free surface and the third term is the pressure using the average ocean density referenced to  $z = 0$ .

We require an expression for  $\rho'$  which takes into account the fact that we may have an arbitrary number of blobs within a grid cell, each with a different density. We write  $\rho_T = \rho_0 + \rho'_T$ , and use Equation (6.35a),

$$\begin{aligned} \rho_T^{ijk} + \rho_0 &= \frac{dA^{ij} (\rho dz)_E^{ijk} + \sum_{q=1}^{\mathcal{Q}^{ijk}} m_L^q}{dA^{ij} dz_E^{ijk} + dA^{ij} \sum_{q=1}^{\mathcal{Q}^{ijk}} dz_L^q} \\ \rho_T^{ijk} &= \frac{(\rho dz)_E^{ijk} + \sum_{q=1}^{\mathcal{Q}^{ijk}} m_L^q / dA^{ij}}{dz_T^{ijk}} - \rho_0 . \end{aligned} \quad (6.41)$$

The discrete equation for the anomalous pressure at a grid cell  $1 \leq K \leq k_{\text{bot}}$  is

$$p^{ijkK} = g \sum_{k=1}^K \rho_T^{ijk} dz_T^{ijk} . \quad (6.42)$$

Substituting Equation (6.41) into (6.42) gives us the discrete equation for anomalous

pressure at the bottom of a grid cell in terms of E and L variables

$$p'^{ijkK} = g \sum_{k=1}^K \left[ (\rho dz)_E^{ijk} + \sum_{q=1}^{Q^{ijk}} m_L^q / dA^{ij} - \rho_0 dz_T^{ijk} \right] \quad (6.43)$$

In pressure-based vertical coordinates, the hydrostatic pressure at a point is (see Equation (4.24) of Griffies, 2009)

$$p = p_b - \rho_0 \Phi + \rho_0 (\Phi_b + \Phi') \quad (6.44)$$

where  $p_b$  is the bottom pressure,  $\Phi_b = -gH$  is the bottom geopotential and  $\Phi'$  is the anomalous geopotential, defined by

$$\rho_0 \Phi' = -g \int_{-H}^z \rho' dz. \quad (6.45)$$

Noting the isomorphism with Equation (6.40b), the discretised expression for the anomalous geopotential is

$$\rho_0 \Phi'_T^{ijkK} = -g \sum_{k=k_{\text{bot}}}^K \rho_T^{ijk} dz_T^{ijk}, \quad (6.46)$$

where  $1 \leq K \leq k_{\text{bot}}$ . Substituting the expression for the total anomalous density, Equation (6.41), into the equation for the total anomalous geopotential, Equation (6.46), yields

$$\rho_0 \Phi'_T^{ijkK} = -g \sum_{k=k_{\text{bot}}}^K \left[ (\rho dz)_E^{ijk} + \sum_{q=1}^{Q^{ijk}} m_L^q / dA^{ij} - \rho_0 dz_T^{ijk} \right]. \quad (6.47)$$

### 6.3.4 Calculation of Surface Height

In MOM depth based vertical coordinate systems solve prognostically for the sea surface height which in turn, is used to update grid cell thickness. As a result of the partition of thickness between the E and L systems discussed in Section 6.2.1, we need to adjust expressions for the depth of the water column to take into account the L partition,

$$(H + \eta)^{ij} = \sum_{k=1}^{k_{\text{bot}}} dz_T^{ijk} \quad (6.48a)$$

$$= \sum_{k=1}^{k_{\text{bot}}} dz_E^{ijk} + (dA^{ij})^{-1} \sum_{q=1}^{Q^{ij}} V^q \quad (6.48b)$$

$$= \sum_{k=1}^{k_{\text{bot}}} dz_E^{ijk} + \sum_{k=1}^{k_{\text{bot}}} dz_L^{ijk}, \quad (6.48c)$$

where the volume of a blob may be found as  $V^q = m^q / \rho^q$ .

The prognostic equation for the surface height, is given by (from Equation (7.63) of

Griffies, 2009),

$$\partial_t \eta^{ij} = -\nabla \cdot \mathbf{U}^{ij} + \mathcal{Q}_W / \rho_0 \quad (6.49)$$

where we have ignored volume sources. Here,

$$\mathbf{U}(x, y) \equiv \int_{-H}^{\eta} \mathbf{u} dz \quad (6.50a)$$

is the depth integrated horizontal volume transport. In discrete form, the transport is given by,

$$\mathbf{U}^{ij} \equiv \sum_{k=1}^{k_{\text{bot}}^{ij}} dz^{ijk} \mathbf{u}^{ijk}. \quad (6.50b)$$

In the partitioned system, we have the convergence of the depth integrated velocity of the E system, the volume flux across the free surface, and the convergence of blobs,

$$\partial_t \eta^{ij} = -\nabla \cdot \mathbf{U}_E^{ij} + \mathcal{Q}_W / \rho_0 + \sum_{q=1}^{Q_{(\text{in})}^{ij}} \frac{dz_L^q}{dt} - \sum_{q=1}^{Q_{(\text{out})}^{ij}} \frac{dz_L^q}{dt}. \quad (6.51)$$

Details of how to time step this equation are discussed in Section 6.5.

On the other hand, models using pressure based vertical coordinates diagnose sea surface height. The diagnostic expression for surface height in pressure based coordinates is (from Equation (7.61) of Griffies, 2009),

$$\eta^{ij} = -H^{ij} + \sum_{k=1}^{k_{\text{bot}}^{ij}} dz_T^{ijk}. \quad (6.52)$$

### 6.3.5 Calculation of Bottom Pressure

The bottom pressure is prognostically solved in pressure based vertical coordinates and diagnostically in depth-based coordinates. The equation for the vertically integrated mass budget for the combined E and L system is (from Equation (7.72) of Griffies, 2009)

$$\partial_t (p_b - p_A)^{ij} = -g \nabla \cdot (\mathbf{U}^\rho)^{ij} + g \mathcal{Q}_W^{ij}, \quad (6.53)$$

where  $p_b$  is the bottom pressure and  $p_A$  is the applied pressure at the surface. Here,

$$\mathbf{U}^\rho(x, y) \equiv \int_{-H}^{\eta} \rho \mathbf{u} dz, \quad (6.54a)$$

is the depth integrated horizontal mass transport. In discrete form, the transport is given by,

$$(\mathbf{U}^\rho)^{ij} \equiv \sum_{k=1}^{k_{\text{bot}}^{ij}} \rho^{ijk} dz^{ijk} \mathbf{u}^{ijk}. \quad (6.54b)$$

The partitioned equation for the vertically integrated mass budget is given by the convergence of the horizontal mass flux, the mass flux crossing the free surface and the blob convergence,

$$\partial_t(p_b - p_A)^{ij} = -g\nabla \cdot (\mathbf{U}^\rho)_E^{ij} + g\mathcal{Q}_W^{ij} + g \sum_{q=1}^{Q_{(in)}^{ij}} \frac{m_L^q}{dt dA^{ij}} - g \sum_{q=1}^{Q_{(out)}^{ij}} \frac{m_L^q}{dt dA^{ij}}. \quad (6.55)$$

How the time stepping is managed is discussed further in Section 6.5.

For a model using depth based coordinates, the bottom pressure may be diagnosed as (from Equation (7.68) of Griffies, 2009),

$$(p_b - p_a)^{ij} = g \sum_{k=1}^{k_{bot}^{ij}} \rho_E^{ijk} dz_E^{ijk} + g(dA^{ij})^{-1} \sum_{q=1}^{Q_{(up)}^{ij}} m_L^q. \quad (6.56)$$

### 6.3.6 Pressure at a Tracer Point

MOM has two possible methods to find the hydrostatic pressure at a tracer point. The first method is the finite volume method, motivated by the dimensions that are shown in Figure 6.1. We note that unlike the purely Eulerian system, the combined E and L system may have different averaged densities between the upper part of a tracer grid cell and the lower part. As such, we use Equation (6.41), the density anomaly for the combined system to define two distinct density anomaly quantities,

$$\rho_{T(up)}^{ijk} = \frac{(\rho dz)_E^{ijk} + \sum_{q=1}^{Q_{(up)}^{ijk}} (\rho dz)_L^q}{dz t u p T^{ijk}} - \rho_0 \quad (6.57a)$$

$$\rho_{T(1o)}^{ijk} = \frac{(\rho dz)_E^{ijk} + \sum_{q=1}^{Q_{(1o)}^{ijk}} (\rho dz)_L^q}{dz t l o T^{ijk}} - \rho_0, \quad (6.57b)$$

one for the upper part of the grid cell,  $\rho_{T(up)}^{ijk}$ , and lower part,  $\rho_{T(1o)}^{ijk}$ . Here,  $Q_{(up)}^{ijk}$  and  $Q_{(1o)}^{ijk}$  are the number of blobs in the upper and lower part respectively. Using the anomalous pressure Equation (6.40b), we can see that it is reasonable to calculate anomalous pressure at a tracer point as

$$p_T^{ij1} = g d z t u p T^{ij1} \rho_{T(up)}^{ij1} \quad (6.58a)$$

$$p_T^{ijK} = p_T^{ij1} + g \sum_{k=2}^K \left( \rho_{T(up)}^{ijk} d z t u p T^{ijk} + \rho_{T(1o)}^{ijk-1} d z t l o T^{ijk-1} \right), \quad (6.58b)$$

where  $2 \leq K \leq k_{bot}$ .

The second method is the energetic method, formulated by Bryan (1969), and is equivalent to the finite volume method when tracer cells are of equal thickness. The total pressure

anomaly using the energetic method is given by

$$p_T'^{ijk} = p_T'^{ij1} + g \sum_{k=2}^K dzwt T^{ijk-1} \overline{\rho_T'^{ijk}}^z \quad (6.59a)$$

$$\overline{\rho_T'^{ijk}}^z = (\rho_{T(1o)}'^{ijk-1} + \rho_{T(up)}'^{ijk})/2, \quad (6.59b)$$

where  $2 \leq K \leq k_{\text{bot}}$  and  $p_T'^{ij1}$  is given by Equation (6.58a).

### 6.3.7 Geopotential at a Tracer Point

Similarly to the pressure at a tracer point, MOM can use either the finite volume method or the energetic method for calculating the geopotential anomaly. Beginning with the finite volume method, recalling the Equation (6.41) for the density anomaly,

$$\Phi'^{ijk}_{\text{bot}} = -\frac{g}{\rho_0} \rho_{T(1o)}'^{ijk_{\text{bot}}} dztlo T^{ijk_{\text{bot}}} \quad (6.60a)$$

$$\Phi'^{ijk} = \Phi'^{ijk}_{\text{bot}} - \frac{g}{\rho_0} \sum_{k=k_{\text{bot}}-1}^K \left( \rho_{T(1o)}'^{ijk} dztlo T^{ijk} + \rho_{T(up)}'^{ijk+1} dztup T^{ijk+1} \right), \quad (6.60b)$$

where the summation is in increments of  $-1$ , and that  $k_{\text{bot}} > K \geq 1$ .

Similarly to the energetic method for the calculation of hydrostatic pressure, the energetic method is equivalent to the finite volume method when cell thicknesses are equal. The geopotential anomaly using the energetic method is given by

$$\Phi'^{ijk} = \Phi'^{ijk}_{\text{bot}} - \frac{g}{\rho_0} \sum_{k=k_{\text{bot}}-1}^K dzwt T^{ijk} \overline{\rho_T'^{ijk+1}}^z, \quad (6.61)$$

where  $\Phi'^{ijk}_{\text{bot}}$  is calculated using Equation (6.60a) and  $\overline{\rho_T'^{ijk+1}}^z$  is calculated using Equation (6.59b)

## 6.4 Horizontal Grid Considerations

The horizontal grid used in MOM is an Arakawa B-grid (Arakawa, 1966). Horizontal velocity points,  $\mathcal{U}$ , “live” on the corner of tracer cells as shown in Figure 6.3. Thus, at least four adjacent tracer points,  $\mathcal{T}$ , are required before one velocity point can exist.

The grid specification for MOM includes two additional grid points, the “northern” point of a tracer cell,  $\mathcal{N}$ , and the “eastern” point of a tracer cell,  $\mathcal{E}$ .<sup>1</sup> All four horizontal grid points are depicted in Figure 6.3.

---

<sup>1</sup>The terms northern and eastern in this context do not necessarily correspond to the geographical north or east of a tracer cell, but instead, are terms used to uniquely identify two of the walls of a six sided finite volume.

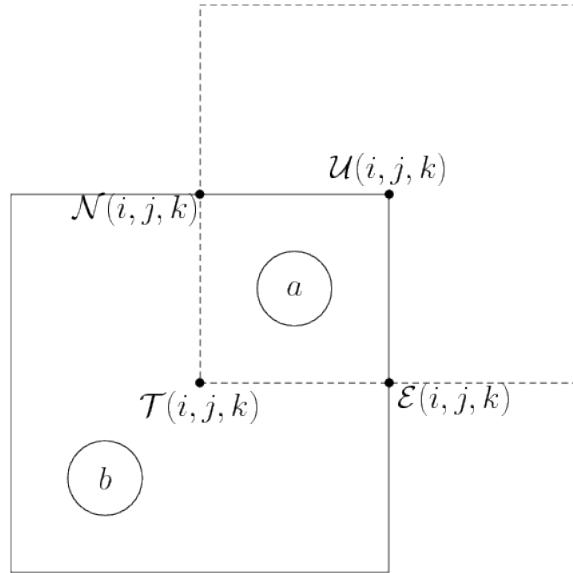


Figure 6.3: Illustration of an Arakawa (1966) B-grid, based on Figure 4.1 and 4.2 of Griffies et al. (2004).  $\mathcal{T}$  is a tracer point and  $\mathcal{U}$  is a velocity point. In addition to the tracer and velocity points two other points are defined, the  $\mathcal{N}$  and  $\mathcal{E}$  points. The circles represent two blobs,  $a$  and  $b$  which are both in the  $\mathcal{T}$  cell, but only  $a$  is in the  $\mathcal{U}$  cell.

The same model variable notation for the velocity grid as for the tracer grid is adopted, that is, in the partitioned system, the suffix  $\mathbf{E}$  is omitted. For example, the Eulerian contribution to thickness is written as  $\mathbf{dzu}$  while the Lagrangian contribution is  $\mathbf{dzuL}$  and the total thickness is  $\mathbf{dzuT}$ . To reduce the complexity of operating on two grids, we interpolate salient  $\mathbf{E}$  system values to a blob. Details of the interpolation scheme are given in Section 7.1.8.

In addition to the four horizontal grid points, there is a vertical velocity grid point,  $\mathcal{W}$ . The vertical velocity point is illustrated in Figure 6.2 as unfilled circles, while tracer grid points are illustrated as filled circles.

Figure 6.3 also illustrates that there can be a difference in magnitude of the L partition between the tracer and velocity grid, where, it can be seen that for the tracer cell

$$\mathbf{dzt}^{ijk} = \mathbf{dzt}^{ijk} + \frac{V_L^a}{\mathbf{dat}^{ij}} + \frac{V_L^b}{\mathbf{dat}^{ij}}, \quad (6.62)$$

where  $\mathbf{dat}^{ij}$  is the horizontal area of the grid cell (which is independent of depth). On the other hand, the total thickness of the velocity cell is

$$\mathbf{dzuT}^{ijk} = \mathbf{dzu}^{ijk} + \frac{V_L^a}{\mathbf{dau}^{ij}}, \quad (6.63)$$

where  $\mathbf{dzuT}$  is the total thickness of a velocity grid cell and  $\mathbf{dau}^{ij}$  is the horizontal area of the velocity cell (which is also independent of depth). Comparing Equations (6.62) and (6.63), it can be seen that the contribution to the thickness of tracer and velocity grid

cells from the L system differ for corresponding  $\mathcal{T}$  and  $\mathcal{U}$  grid cells, since the tracer and velocity points are not co-located.

In MOM the thickness of a  $\mathcal{U}$  cell is calculated as the minimum of the thickness of the four surrounding  $\mathcal{T}$  cells,

$$dzu^{ijk} = \min(dzt^{ijk}, dzt^{i+1jk}, dzt^{i+1j+1k}, dzt^{ij+1k}) \quad (6.64)$$

where  $\min$  is the minimum function. There are analogous expressions for  $dzwu$  and  $\text{rho\_dzu}$ . The logic for this method of calculating the velocity cell thicknesses was originally applied to the calculation of velocity cell thicknesses for partial bottom cells (see Chapter 26 of Pacanowski and Griffies, 1999) and has been generalised for use with the types of vertical coordinates used by MOM.

As the method remains empirically useful in purely Eulerian models, we extend the method when partitioning the grid cells into E and L sections,

$$dzuT^{ijk} = \min(dztT^{ijk}, dztT^{i+1jk}, dztT^{i+1j+1k}, dztT^{ij+1k}) \quad (6.65a)$$

$$dzuL^{ijk} = \min(dztL^{ijk}, dztL^{i+1jk}, dztL^{i+1j+1k}, dztL^{ij+1k}) \quad (6.65b)$$

$$dzu^{ijk} = dzuT^{ijk} - dzuL^{ijk}, \quad (6.65c)$$

with analogous expressions for each of  $dzwu$  and  $\text{rho\_dzu}$ .

## 6.5 Time Stepping Considerations

In the present section, the staggered time stepping scheme used by MOM is reviewed. The older leap-frog time stepping schemes are ignored. A discussion on time stepping schemes can be found in Sections 8.4 to 8.9 of Griffies (2009) and Chapter 12 of Griffies (2004).

To time step the seawater mass and tracer mass budgets in a generalised level coordinate system, the rate of change of seawater mass and tracer mass per unit area must be calculated. To calculate the tracer mass and seawater mass per unit area, knowledge of  $z_s$  is required

$$\partial_t(\rho C dz)_T^{ijk} = \partial_t(\rho C z_s ds)_T^{ijk} \quad (6.66a)$$

$$\partial_t(\rho dz)_T^{ijk} = \partial_t(\rho z_s ds)_T^{ijk}. \quad (6.66b)$$

The sea surface height or bottom pressure are prognostic variables that are used to calculate  $z_s$ . A requirement of the staggered time stepping method is the value of the sea surface height (for depth-based coordinates) or bottom pressure (for pressure-based coordinates) at integer time steps,  $\tau$  and half integer time steps  $\tau + \Delta\tau/2$ . The split is required to couple velocity variables, which are calculated at integer time steps, to tracer variables,

which are calculated at half integer time steps.

The dynamics of MOM are split into a “fast” barotropic component and a “slow” baroclinic component, with the split typically being  $\sim 80$ . The fast barotropic component is solved in MOM by using a predictor-corrector method (see Section 12.8 of Griffies, 2004). The scheme is run from  $\tau$  to  $\tau + 2\Delta\tau$ , with the number of steps being  $N$ . The prognostic variable at time  $\tau + \Delta\tau$  is then found by taking the average value over the two integer steps,

$$\eta(\tau + \Delta\tau) = \frac{1}{N+1} \sum_{n=0}^N \eta^{(b)}(\tau, n) \quad (6.67a)$$

$$(p_b - p_A)(\tau + \Delta\tau) = \frac{1}{N+1} \sum_{n=0}^N (p_b - p_A)^{(b)}(\tau, n) \quad (6.67b)$$

where  $p_b$  is the bottom pressure and the superscript  $(b)$  indicates the barotropic predicted-corrected value at step  $n$ . The prognostic variable at half integer steps is found by averaging at steps  $\tau$  and  $\tau + \Delta\tau$ ,

$$\eta(\tau + \Delta\tau/2) = \frac{\eta(\tau) + \eta(\tau + \Delta\tau)}{2} \quad (6.68a)$$

$$(p_b - p_A)(\tau + \Delta\tau/2) = \frac{(p_b - p_A)(\tau) + (p_b - p_A)(\tau + \Delta\tau)}{2}. \quad (6.68b)$$

As detailed in Sections 8.9.1 and 8.9.2 of Griffies (2009), we may calculate the rate of change at integer time steps for depth based and pressure based coordinates, respectively using

$$\frac{\eta(\tau + \Delta\tau/2) - \eta(\tau - \Delta\tau/2)}{\Delta\tau} = -\nabla \cdot \mathbf{U}(\tau) + \mathcal{Q}_w(\tau) \quad (6.69a)$$

$$\frac{(p_b - p_A)(\tau + \Delta\tau/2) - (p_b - p_A)(\tau - \Delta\tau/2)}{\Delta\tau} = -g\nabla \cdot \mathbf{U}^\rho(\tau) + g\mathcal{Q}_w(\tau). \quad (6.69b)$$

where we have omitted source terms for brevity. The contribution of the blobs to the “fast” barotropic component is held constant for the barotropic steps.

For brevity, we shall henceforth utilise the following notation,

$$\tau - 1 \equiv \tau - \Delta\tau, \quad \tau - 1/2 \equiv \tau - \Delta\tau/2, \quad \tau + 1/2 \equiv \tau + \Delta\tau/2, \quad \tau + 1 \equiv \tau + \Delta\tau,$$

and so on for other full and fractional time-steps. We also subscript the time index for a convenient short hand.

The compatible vertically integrated tracer Equations (6.69) are given by,

$$\begin{aligned} \sum_{k=1}^{k_{\text{bot}}} \frac{(dz\mathcal{C})_{\tau+1/2}^{ijk} - (dz\mathcal{C})_{\tau-1/2}^{ijk}}{\Delta\tau} = & \mathcal{Q}_W/\rho_0 - \sum_{k=1}^{k_{\text{bot}}} \nabla \cdot \left[ (dz\mathbf{u})_{\tau}^{ijk} \mathcal{C}_{\tau-1/2}^{ijk} + dz^{ijk} \mathbf{F}^{ijk} \right]^{ijk} \\ & + \sum_{k=1}^{k_{\text{bot}}-1} \left[ w_{\tau} \mathcal{C}_{\tau-1/2} + F^{(s)} \right]^{ijk} - \sum_{k=2}^{k_{\text{bot}}} \left[ w_{\tau} \mathcal{C}_{\tau-1/2} + F^{(s)} \right]^{ijk-1} \end{aligned} \quad (6.70a)$$

$$\begin{aligned} \sum_{k=1}^{k_{\text{bot}}} \frac{(\rho dz\mathcal{C})_{\tau+1/2}^{ijk} - (\rho dz\mathcal{C})_{\tau-1/2}^{ijk}}{g\Delta\tau} = & \mathcal{Q}_W - \sum_{k=1}^{k_{\text{bot}}} \nabla \cdot \left[ (dz\rho\mathbf{u})_{\tau}^{ijk} \mathcal{C}_{\tau-1/2}^{ijk} + (dz\rho)^{ijk} \mathbf{F}^{ijk} \right]^{ijk} \\ & + \sum_{k=1}^{k_{\text{bot}}-1} \left[ (w\rho)_{\tau} \mathcal{C}_{\tau-1/2} + \rho F^{(s)} \right]^{ijk} - \sum_{k=2}^{k_{\text{bot}}} \left[ (w\rho)_{\tau} \mathcal{C}_{\tau-1/2} + \rho F^{(s)} \right]^{ijk-1}, \end{aligned} \quad (6.70b)$$

for depth- and pressure-based coordinates respectively. Equations (6.70) reduce to Equations (6.69) when  $\mathcal{C}$  is set to a non-zero constant globally, meaning that the compatibility condition is satisfied. The lateral and vertical SGS fluxes are conspicuously absent of time labels. The compatibility condition does not place restrictions upon these fluxes, and we are thus free to choose convenient time labels. The SGS concentration flux time labels shall be discussed further in Section 6.5.6.

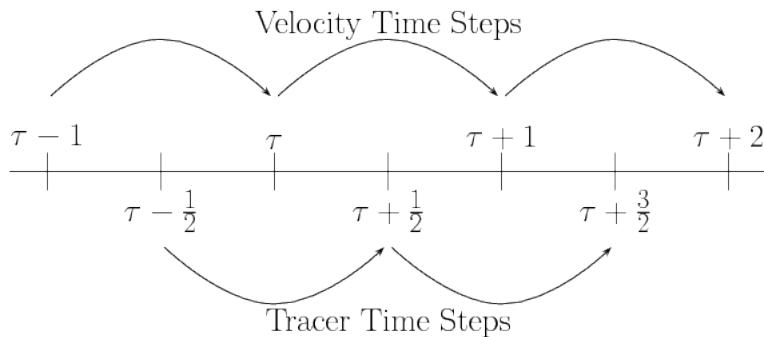


Figure 6.4: The second order, split time stepping scheme for tracer and velocity in MOM.

The staggered time stepping is the temporal analogue of the staggered spatial grid discussed in Section 6.4. The staggered time scheme is shown schematically in Figure 6.4. In order to time step the E system with the partitioned thickness, we need to further the discussion in Sections 6.2.1 and 6.2.3 on the calculation of thickness and the compatibility condition between tracer mass budgets and the seawater mass budget. How the barotropic equations are time stepped is discussed first. The barotropic equations are required to calculate the vertical coordinate surfaces. Recalling the prognostic Equation (6.51) for

depth-based coordinates and the prognostic Equation (6.55) for pressure-based coordinates

$$\begin{aligned}\partial_t \eta^{ij} &= \mathcal{D}_W^{ij}/\rho_0 - \nabla \cdot \mathbf{U}_E^{ij} + \sum_{q=1}^{Q_{(in)}^{ij}} \frac{m_L^q}{\rho_0 \Delta \tau dA^{ij}} - \sum_{q=1}^{Q_{(out)}^{ij}} \frac{m_L^q}{\rho_0 \Delta \tau dA^{ij}} \\ \partial_t (p_b - p_A)^{ij} &= g \mathcal{D}_W^{ij} - g \nabla \cdot (\mathbf{U}^\rho)_E^{ij} + g \sum_{q=1}^{Q_{(in)}^{ij}} \frac{m_L^q}{\Delta \tau dA^{ij}} - g \sum_{q=1}^{Q_{(out)}^{ij}} \frac{m_L^q}{\Delta \tau dA^{ij}}.\end{aligned}$$

Exposing the time labels on the various components of these semi-discrete equations, the discrete rate of change of sea surface height at time  $\tau$ , as a finite difference equation is written as

$$\partial_t \eta_\tau^{ij} = \frac{\eta_{\tau+1/2}^{ij} - \eta_{\tau-1/2}^{ij}}{\Delta \tau}.$$

The equivalent expression for pressure-based coordinates follows naturally. Another way of interpreting the equation is in a finite volume sense, in which the time  $\tau$  is a temporal grid cell centre, and the half integer time steps are the temporal grid cell faces. We discuss how to interpret the blob divergence in Section 6.5.2.

To examine the compatibility with the tracer equations the tracer budget for an interior grid cell for the total system is restated (for brevity, the budgets for a surface or bottom cell are not written, but can be regained from an interior cell by applying the appropriate boundary conditions), as written in Equation (6.5a)

$$\begin{aligned}\partial_t (dz \rho \mathcal{C})_T^{ijk} &= -\nabla \cdot [dz \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_E^{ijk} - [\rho(w \mathcal{C} + F^{(s)})]_E^{ijk-1} + [\rho(w \mathcal{C} + F^{(s)})]_E^{ijk} \\ &\quad - \sum_{q=1}^{Q_{(out)}^{ijk}} \frac{(dz \rho \mathcal{C})_L^q}{\Delta \tau} + \sum_{q=1}^{Q_{(in)}^{ijk}} \frac{(dz \rho \mathcal{C})_L^q}{\Delta \tau}.\end{aligned}\tag{6.71}$$

The respective E and L tracer budgets that make up the total budget can be written as,

$$\begin{aligned}\partial_t (dz \rho \mathcal{C})_E^{ijk} &= -\nabla \cdot [dz \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_E^{ijk} - [\rho(w \mathcal{C} + F^{(s)})]_E^{ijk-1} + [\rho(w \mathcal{C} + F^{(s)})]_E^{ijk} \\ &\quad - \sum_{q=1}^{Q_{(out)}^{ijk}} \frac{[(\rho \mathcal{C})_E^{ijk} \mathcal{E}^q - (\rho \mathcal{C})_L^q \mathcal{D}^q] A_{\text{interface}}^q}{dA^{ij}} - \sum_{q=1}^{Q_{(\text{new})}^{ijk}} \frac{m_L^q \mathcal{C}_E^{ijk}}{\Delta \tau dA^{ij}} + \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} \frac{m_L^q \mathcal{C}_E^{ijk}}{\Delta \tau dA^{ij}}\end{aligned}\tag{6.72a}$$

$$\begin{aligned}\partial_t (dz \rho \mathcal{C})_L^{ijk} &= - \sum_{q=1}^{Q_{(out)}^{ijk}} \frac{(dz \rho \mathcal{C})_L^q}{\Delta \tau} + \sum_{q=1}^{Q_{(in)}^{ijk}} \frac{(dz \rho \mathcal{C})_L^q}{\Delta \tau} + \sum_{q=1}^{Q_{(\text{new})}^{ijk}} \frac{m_L^q \mathcal{C}_E^{ijk}}{\Delta \tau dA^{ij}} - \sum_{q=1}^{Q_{(\text{dstry})}^{ijk}} \frac{m_L^q \mathcal{C}_E^{ijk}}{\Delta \tau dA^{ij}} \\ &\quad + \sum_{q=1}^{Q_{(out)}^{ijk}} \frac{[(\rho \mathcal{C})_E^{ijk} \mathcal{E}^q - (\rho \mathcal{C})_L^q \mathcal{D}^q] A_{\text{interface}}^q}{dA^{ij}}.\end{aligned}\tag{6.72b}$$

All of these tracer budget equations satisfy the compatibility condition, and it can be seen that summing Equations (6.72) will give Equation (6.71).

As discussed later in Sections 7.3.1 and 7.3.2, a blob's position is sub-cycled and must coincide with the E system's steps. One of the results of the above discussion is that, similarly to the spatially staggered grid, we must carefully consider how we interpret the temporal "location" of a blob.

Before interpreting what time levels the fluxes between the L and E systems are updated, a more fundamental question should be examined: how is grid cell thickness calculated and at what time?

### 6.5.1 Some Further Vertical Coordinate Considerations

Resuming the discussion on the calculation of  $z_{,s}$  first broached in Section 6.3.1, details of the various coordinate systems are given in Table 6.1. Paying particular attention to the rightmost column, each of the three coordinate systems is considered in turn.

Coordinate	Definition	Range	$z_{,s}$
zstar	$z^* = H(z - \eta)/(H + \eta)$	$-H \leq z^* \leq 0$	$1 + \eta/H$
pressure	$p$	$p_A \leq p \leq p_b$	$-1/g\rho$
pstar	$p^* = p_b^0(p - p_A)/(p_b - p_A)$	$0 \leq p^* \leq p_b^0$	$-(p_b - p_A)/(g\rho p_b^0)$

Table 6.1: Details of the various coordinate systems implemented in MOM. Based on Tables 6.1 and 6.2 of Griffies (2009).

The depth based coordinate system that the blobs have been implement in is  $z^*$ , and is calculated as

$$z_{\tau+1/2,s}^{ijk} = 1 + \frac{\eta_{\tau+1/2}^{ij}}{H^{ij}}, \quad (6.73)$$

where the time labels have been written explicitly. As detailed in Section 8.1 the value for  $\eta_{\tau+1/2}^{ij}$  is known when calculating  $z_{\tau+1/2,s}^{ijk}$  for depth based coordinates.

For the pressure based coordinates, there are some special procedures required for the initialisation of the coordinates, and the reader is referred to Section 7.3.3 of Griffies (2009) for those details. For pressure coordinates

$$z_{\tau+1/2,s}^{ijk} = - \left( g\rho_{\tau-1/2}^{ijk} \right)^{-1}, \quad (6.74)$$

noting that the time for the calculation of density is at  $\tau - 1/2$ . For  $p^*$  coordinates

$$z_{\tau+1/2,s}^{ijk} = - \left( \frac{(p_b - p_A)_{\tau+1/2}^{ij}}{gp_b^0 \rho_{\tau-1/2}^{ijk}} \right), \quad (6.75)$$

where  $p_b^0$  is a reference bottom pressure. In analogy with  $\eta$  in depth based coordinates,  $(p_b - p_A)$  has been updated to time  $\tau + 1/2$ . Similarly to pressure coordinates, however,

the density is taken at the previous time step.

There are two separate issues pertaining to pressure based coordinates. The first is the fact that density from  $\tau - 1/2$  is used for calculating  $z_s$  at  $\tau + 1/2$ . The bottom pressure ( $p_b - p_A$ ) has been updated using the methods described in Section 6.5. Density, however, is not available at the new time step (details as to why are given later in this section) and it is an order of operations problem which prevents us from being able to use density at the desired time step. It has, nevertheless, been empirically shown that use of density at time  $\tau - 1/2$  is useful and preserves local and global conservation of mass and tracer (S.M. Griffies, personal communication, 2010).

The second issue is to choose which value for density to use, the total density or the E system density. The Eulerian density has empirically been shown to be conservative and stable. A more physically consistent and satisfying value for density would be the combined density of the two systems. The combined system's density is calculated using Equation (6.35a). Thus, the method for calculating the total density and the calculation of  $z_s$  with explicit time labels for the combined system is given by,

$$\rho_{T\tau-1/2}^{ijk} = \frac{\rho_{E\tau-1/2}^{ijk} dz_{E\tau-1/2}^{ijk} + (dA^{ij})^{-1} \sum_{q=1}^{Q_{\tau-1/2}^{ijk}} m_{\tau-1/2}^q}{dz_{T\tau-1/2}^{ijk}} \quad (6.76a)$$

$$\partial_p z_{\tau+1/2}^{ijk} = - \left( g \rho_{T\tau-1/2}^{ijk} \right)^{-1} \quad (6.76b)$$

$$\partial_{p^*} z_{\tau+1/2}^{ijk} = - \frac{(p_b - p_A)_{\tau+1/2}^{ij}}{g p_b^0 \rho_{T\tau-1/2}^{ijk}}. \quad (6.76c)$$

As discussed at the end of Section 6.3.1, the calculation of the model variable `rho_dztT` deserves special attention. In the standard release of MOM the thickness weighted density in non-Boussinesq models is calculated as,

$$(\rho dz)_{\tau+1/2}^{ijk} = \rho_{\tau-1/2}^{ijk} dz_{\tau+1/2}^{ijk}. \quad (6.77)$$

The reason why the time label on density is for the previous time step is that the density at time  $\tau + 1/2$  is unknown until the tracer concentrations are known, and the tracer concentrations require knowledge of  $(\rho dz)$ , leading to an order of operations problem. The formulation in Equation (6.77) is a compromise but has been empirically shown to preserve local and global conservation of seawater mass and tracer mass (analysis not shown).

One might then be tempted, when considering the partitioning required to admit the embedded Lagrangian model, to calculate the total density weighted thickness as

$$(\rho dz)_{T\tau+1/2}^{ijk} = \rho_{E\tau-1/2}^{ijk} dz_{E\tau+1/2}^{ijk} + (dA^{ijk})^{-1} \sum_{q=1}^{Q_{\tau+1/2}^{ijk}} m_{\tau+1/2}^q.$$

For non-Boussinesq models, however, this formulation has been empirically shown to not conserve seawater mass. Instead, we make use of the rate of change of the total density weighted thickness,  $\partial_t(\rho dz)_{\text{T}\tau}^{ijk}$ , which is known at the time that the density weighted thickness is calculated. See Equations (6.69) to (6.70) and the surrounding discussion regarding how the rate of change of thickness is calculated. A scheme, which has been shown empirically to be conservative in both the Boussinesq and non-Boussinesq case is

$$(\rho dz)_{\text{T}\tau+1/2}^{ijk} = (\rho dz)_{\text{T}\tau-1/2}^{ijk} + \Delta\tau \partial_t(\rho dz)_{\text{T}\tau}^{ijk} \quad (6.78a)$$

$$(\rho dz)_{\text{L}\tau+1/2}^{ijk} = (\text{d}A^{ij})^{-1} \sum_{q=1}^{\mathcal{Q}_{\tau+1/2}^{ijk}} m_{\tau+1/2}^q \quad (6.78b)$$

$$(\rho dz)_{\text{E}\tau+1/2}^{ijk} = (\rho dz)_{\text{T}\tau+1/2}^{ijk} - (\rho dz)_{\text{L}\tau+1/2}^{ijk}. \quad (6.78c)$$

Next, the evolution of the partitioning of thickness is examined. For the purposes of the following discussion entrainment and detrainment are ignored without loss of generality. For these purposes, the blobs are viewed as an immiscible bubble of fluid moving through a continuum E system. Such a view is shown diagrammatically in Figure 6.5(a), where the thickness partition is also represented on the right. Recall that the shape of the blob is arbitrary, and that all of its properties are concentrated at a discrete point in space – the representation of a blob as spherical is for illustrative purposes only. The thicknesses represented on the right are also schematic only, noting that  $\text{dztL}$  is the blob's contribution to the total thickness,  $\text{dztT}$ . We emphasise again that the diagram is schematic and that any difference between the area represented by the circle and the area represented by the thickness is of no importance.

Let us say that the blob evolves such that at the next time step,  $\tau+1$ , it has crossed the coordinate surface between the cell  $k$  and the cell  $k+1$ , with the corresponding diagram appearing in Figure 6.5(b). Without loss of generality, we ignore all other effects such that there is no change in the prognostic barotropic variables,  $\partial_t\eta = 0$  in depth-based coordinates and  $\partial_t(p_b - p_A) = 0$  in pressure-based coordinates. This condition implies that the total density weighted thickness does not change,  $\partial_t(\rho dz)_T = 0$ . The partitioned density weighted thicknesses, however, will change,

$$\text{rho\_dzt}_{\tau-1/2}^{k+1} < \text{rho\_dzt}_{\tau+1/2}^{k+1} \quad (6.79a)$$

$$\text{rho\_dzt}_{\tau-1/2}^k > \text{rho\_dzt}_{\tau+1/2}^k \quad (6.79b)$$

$$\text{rho\_dztL}_{\tau-1/2}^{k+1} < \text{rho\_dztL}_{\tau+1/2}^{k+1} \quad (6.79c)$$

$$\text{rho\_dztL}_{\tau-1/2}^k > \text{rho\_dztL}_{\tau+1/2}^k. \quad (6.79d)$$

The incoming mass flux in the L system is balanced by an outgoing vertical mass flux in the E system, as can be shown by rearranging the seawater mass per unit area budget

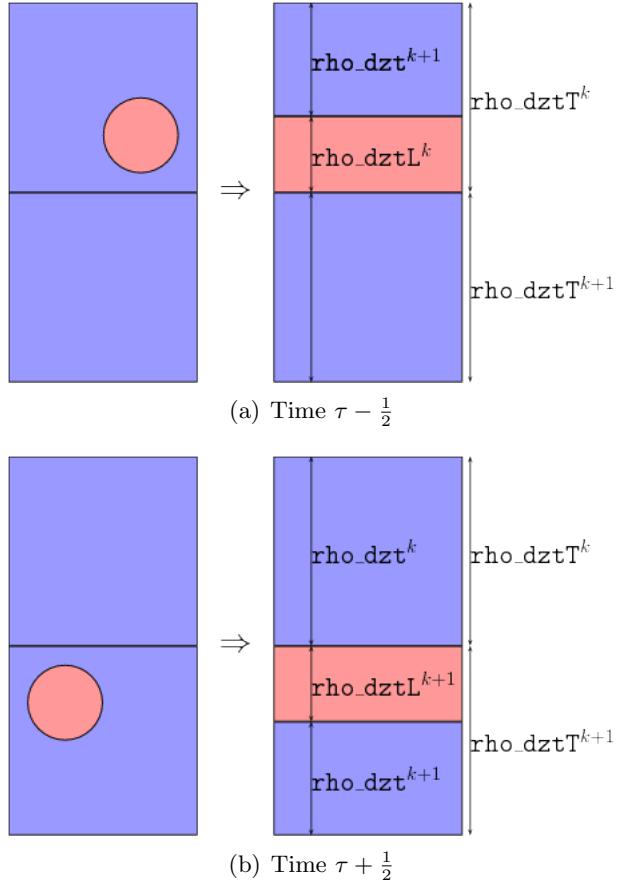


Figure 6.5: Schematic representation of a blob (left) and the resulting partition of thickness (right) at two separate times.

(from Equation 6.71)

$$(\rho w)_E^{ijk} = \partial_t(\rho dz)_T^{ijk} + \nabla \cdot (\rho dz \mathbf{u})_E^{ijk} + (\rho w)_E^{ijk-1} + \sum_{q=1}^{\mathcal{Q}_{\text{(out)}}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau} - \sum_{q=1}^{\mathcal{Q}_{\text{(in)}}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau}. \quad (6.80)$$

All of the terms are evaluated at time  $\tau$ . Next, the calculation of the thickness used in the E system divergence term,  $[\nabla \cdot (\rho dz \mathbf{u})_E^{ijk}]_\tau$ , is examined.

Recalling from Section 6.4 that the thickness weighted density on the  $\mathcal{U}$  grid is calculated as the minimum of the surrounding  $\mathcal{T}$  grid cells. In MOM the  $\mathcal{U}$  thickness is updated for the time  $\tau + 1$  directly from the  $\mathcal{T}$  values at  $\tau + \frac{1}{2}$

$$\text{rho\_dzu}_{\tau+1}^{ijk} = \min(\text{rho\_dzt}_{\tau+1/2}^{ijk}, \text{rho\_dzt}_{\tau+1/2}^{i+1jk}, \text{rho\_dzt}_{\tau+1/2}^{ij+1k}, \text{rho\_dzt}_{\tau+1/2}^{i+1j+1k}). \quad (6.81)$$

While this approach is ad hoc, it does provide a conservative and closed time stepping scheme. When the embedded Lagrangian system is introduced the density weighted thick-

ness equations become

$$\text{rho\_dzuT}_{\tau+1}^{ijk} = \min(\text{rho\_dztT}_{\tau+1/2}^{ijk}, \text{rho\_dztT}_{\tau+1/2}^{i+1jk}, \text{rho\_dztT}_{\tau+1/2}^{ij+1k}, \text{rho\_dztT}_{\tau+1/2}^{i+1j+1k}) \quad (6.82a)$$

$$\text{rho\_dzuL}_{\tau+1}^{ijk} = \min(\text{rho\_dztL}_{\tau+1/2}^{ijk}, \text{rho\_dztL}_{\tau+1/2}^{i+1jk}, \text{rho\_dztL}_{\tau+1/2}^{ij+1k}, \text{rho\_dztL}_{\tau+1/2}^{i+1j+1k}) \quad (6.82b)$$

$$\text{rho\_dzu}_{\tau+1}^{ijk} = \text{rho\_dzuT}_{\tau+1}^{ijk} - \text{rho\_dzuL}_{\tau+1}^{ijk}. \quad (6.82c)$$

Now, the vertical mass flux per unit area, Equation (6.80), can be written with all of the E system time labels revealed

$$[(\rho w)_E^{ijk}]_\tau = [\partial_t(\rho dz)_T^{ijk}]_\tau + [\nabla \cdot (\rho dz \mathbf{u})_E^{ijk} + (\rho w)_E^{ijk-1}]_\tau - \sum_{q=1}^{\mathcal{Q}_{(in)}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau} + \sum_{q=1}^{\mathcal{Q}_{(out)}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau}. \quad (6.83)$$

The blob divergence time labels are discussed in the next section.

### 6.5.2 Calculation of Blob Divergence

There are a couple of different ways in which we can calculate the blob divergence. The most conceptually simple method is to have a condition that if a blob moves from one grid cell, then the blob is added to the  $\mathcal{Q}_{(in)}$  list of the cell it is entering, and is added to the  $\mathcal{Q}_{(out)}$  list for the cell it is leaving. Such is the convention that has been used up until this point,

$$\nabla \cdot (\rho dz)_L^{ijk} \equiv \sum_{q=1}^{\mathcal{Q}_{(in)}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau} - \sum_{q=1}^{\mathcal{Q}_{(out)}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau} \quad (6.84)$$

An alternative approach is to subtract all blobs at time  $\tau - \frac{1}{2}$  and add all blobs at time  $\tau + \frac{1}{2}$ , after they have been moved. Taking the end points is not sufficient, however. We must also take into account material that is entrained and detrained. We may thus write the alternative formulation as,

$$\nabla \cdot (\rho dz)_L^{ijk} = \sum_{q=1}^{\mathcal{Q}_{\tau+1/2}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau} - \sum_{q=1}^{\mathcal{Q}_{\tau-1/2}^{ijk}} \frac{m_L^q}{dA^{ij} \Delta \tau} - \sum_{q=1}^{\mathcal{Q}_{\tau}^{ijk}} \frac{[(\rho C)_E^{ijk} \mathcal{C}^q - (\rho C)_L^q \mathcal{D}^q] A_{\text{interface}}^q}{dA^{ij}}. \quad (6.85)$$

The approach described by Equation (6.84) is the one that is implemented since it is conceptually simpler and easier to implement.

### 6.5.3 Cell Thickness Considerations

There can be situations in which a large proportion of the mass of the combined E and L model lies within the L system. The combined E and L mass in a grid cell may not exceed the total mass of a grid cell, where the total mass of a grid cell is dictated by the vertical coordinate. If there is a high enough concentration of blobs, the sum of all the blobs in a grid cell can exceed the total mass of the grid cell. How might one go about ensuring that the combined mass of the blobs in a grid cell does not exceed the total mass of that grid cell?

Restating Equations (6.24), the model thicknesses are calculated by

$$dztT^{ijk} = z_{,s}^{ijk} dst^{ijk} \quad (6.86a)$$

$$dztL^{ijk} = \sum_{q=1}^{Q^{ijk}} \frac{m^q}{\rho_L^q dA^{ij}} \quad (6.86b)$$

$$dst^{ijk} = dztT^{ijk} - dztL^{ijk}. \quad (6.86c)$$

The situation in which  $dztL < dztT$  at time  $\tau - \frac{1}{2}$  evolves into a situation where  $dztL > dztT$  at time  $\tau + \frac{1}{2}$  is depicted schematically in Figure 6.6.

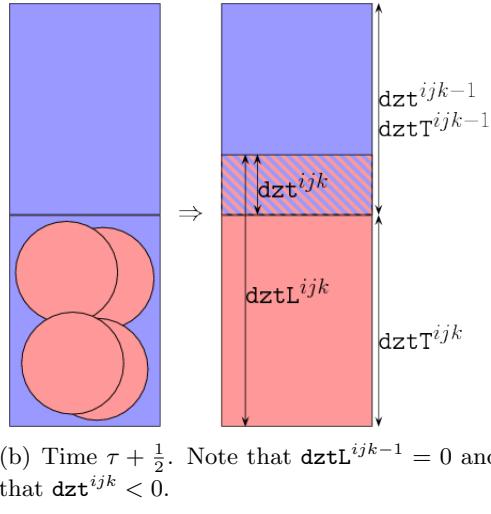
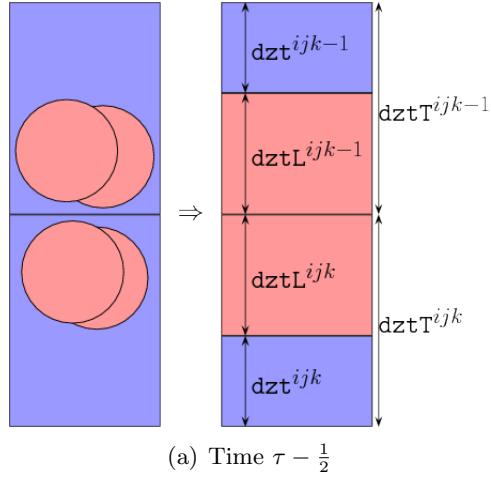


Figure 6.6: A schematic representation of a situation when the L system contribution to thickness is greater than the total thickness of a grid cell. Symbols and model variables are as defined in the text and are listed in Appendix C.

If the L system thickness is greater than the T system thickness then, by Equation (6.86c), the E system thickness will be negative. A negative E system thickness implies a negative mass, which is clearly unphysical. Equation (6.80) is rewritten slightly

to examine the grid cell mass budget

$$(\rho w)_E^{ijk} - (\rho w)_E^{ijk-1} = \partial_t(\rho dz)_T^{ijk} + \nabla \cdot (\rho dz \mathbf{u})_E^{ijk} + \sum_{q=1}^{\mathcal{Q}_{(in)}^{ijk}} \frac{(dz \rho)_L^q}{\Delta \tau} - \sum_{q=1}^{\mathcal{Q}_{(out)}^{ijk}} \frac{(dz \rho)_L^q}{\Delta \tau}. \quad (6.87)$$

All of the terms on the left hand side are diagnostic and are required to balance the terms on the right hand side. Without loss of generality, it is assumed that there are no barotropic changes ( $\partial_t(\rho dz)_T^{ijk} = 0$ ). Thus,  $dztL_{\tau+1}^{ijk} > dztT_{\tau+1}^{ijk}$  is equivalent to

$$\left[ (\rho w)_E^{ijk-1} - (\rho w)_E^{ijk} - \nabla \cdot (\rho dz \mathbf{u}_E)^{ijk} \right]_\tau \Delta \tau > \left[ (dz \rho)_E^{ijk} \right]_\tau, \quad (6.88)$$

which states that the E system material leaving a grid cell over a time step is greater than the material contained within the E partition of the grid cell. In the circumstances under consideration, such a situation can arise for sufficiently large values of  $\sum_{q=1}^{\mathcal{Q}_{(in)}^{ijk}} \frac{(dz \rho)_L^q}{dt}$ .

Clearly, something must be done to ensure that the scenario depicted in Figure 6.6(b) does not eventuate. There are three general strategies that could be employed to avoid the L system thickness of a grid cell exceeding the total thickness of a grid cell.

1. Prevent a blob from entering a grid cell if it causes  $dztL^{ijk} > dztT^{ijk}$ , and keep it in the grid cell that it would be leaving.
2. “Teleport” a blob around a grid cell. For example, if a blob is in grid cell  $k-1$  and enters cell  $k$  such that the condition  $dztL^{ijk} > dztT^{ijk}$  is satisfied, then we instead place it in grid cell  $k+1$ , so long as it being placed there does not cause the condition  $dztL^{ijk+1} > dztT^{ijk+1}$  to be satisfied.
3. If, at the end of an E system time step, it is found that  $dztL^{ijk} > dztT^{ijk}$  blobs in the grid cell in question are destroyed one by one, with their properties returned to the E system, until such time as  $dztL^{ijk} < dztT^{ijk}$  is enforced.

With reference to the algorithmic structure, Section 8.1, strategies 1 and 2 would be difficult to implement, because it cannot be said *a priori* whether a blob entering a grid cell will cause the condition  $dztL^{ijk} > dztT^{ijk}$  to be satisfied, because the total thickness of a grid cell is not calculated until after blob trajectories have been calculated.

In terms of energetics, all three strategies lead to inconsistencies. Strategies 1 and 2 spuriously alter the potential and kinetic energy of the system, while strategy 3 acts as a kinetic energy sink.

A further potential problem with strategies 1 and 2 is the potential to cause “traffic jams”, where given the right forcing, a significant part of the water column could be filled with blobs that are unable to move. This is particularly the case for sill like geometries such as those found over the Antarctic continental shelf, wherein dense waters accumulate for some time, before spilling over the shelf break and cascading to the abyssal ocean. It

is believed on balance that while strategy 3 is still less than ideal, it presents the least problematic implementation.

In order to implement strategy 3, a blob's trajectory is calculated without consideration to the L partition's contribution to thickness. When the routine `update_L_thickness` is called, the condition  $dztL^{ijk} > dztT^{ijk}$  is tested for. If the condition is satisfied, then blobs are destroyed, with those appearing last in the linked list being destroyed first. That is, blobs with a lower hash number (see Section 7.2) are destroyed preferentially to those with a higher hash number. Of blobs that have the same hash number, older blobs are destroyed preferentially to newer blobs (i.e. blobs with a lower counter number are destroyed preferentially to those with a higher counter number).

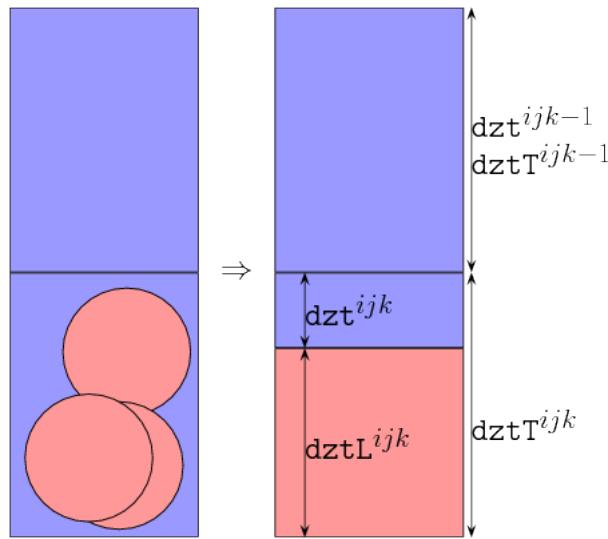


Figure 6.7: As for Figure 6.6(b), except that one blob has been destroyed in order to ensure that  $dzt^{ijk} > 0$ .

Blobs that are destroyed will cause a change in the tracer concentration of the E system partition of a grid cell by,

$$\Delta(dz\rho C)_E^{ijk} = \sum_{q=1}^{Q_{(dstry)}^{ijk}} \frac{(mC)_L^q}{dA^{ij}}. \quad (6.89)$$

Giving seawater mass and tracer mass back to the E system ensures that the seawater mass and tracer mass of the E system remains positive, thereby ensuring conservation is satisfied.

#### 6.5.4 A Vertical CFL Constraint

As a result of the potentially large mass convergence and divergence described in Section 6.5.3, it is possible for large vertical velocities to result, as indicated by the vertical mass flux Equation (6.83). The potentially larger vertical velocities makes the vertical CFL condition more likely to be violated than in a purely E system where all components

are in perfect hydrostatic balance.

A physical interpretation of a CFL condition is that the speed at which information travels in a time step must not be greater than the grid spacing (Durran, 1999). In the context of the vertical velocity, we say that the appropriate CFL condition is that the time step multiplied by the vertical velocity must not be greater than the grid cell thickness. That is,

$$dz_T^{ijk} < w_E^{ijk} \Delta\tau, \quad (6.90)$$

must be satisfied to maintain the vertical CFL condition.

### 6.5.5 A Note Regarding the E System Vertical Velocity

The vertical velocity,  $w$ , in level coordinate models is a diagnostic variable. In MOM, the vertical velocity is not explicitly calculated, and so we must derive it from the density weighted dia-coordinate-surface velocity,  $(\rho w^{(s)})$ . The dia-surface velocity of the water relative to the coordinate surface has two components, the vertical velocity of the water and the vertical velocity of the coordinate consideration, as illustrated by Equations (6.73) of Griffies (2004),

$$w^{(s)} = z_{,s} \frac{ds}{dt} \quad (6.91a)$$

$$= w - (\partial_t + \mathbf{u} \cdot \nabla_s)z. \quad (6.91b)$$

In MOM,  $(\rho w^{(s)})$  is used for calculating the vertical tracer fluxes in the grid cell budgets, Equations (6.4), and is not required anywhere else. For the dynamically active free blobs, we require  $w_E$  to correctly calculate the drag – see Equation (5.6c). To reduce the computational cost, it is assumed that the contribution to  $w^{(s)}$  from changes in depth relative to the geoid of coordinate surfaces is small, and that thus the Eulerian vertical velocity is approximated as,

$$w_E \approx \frac{(\rho w^{(s)})_E}{\rho_E}. \quad (6.92)$$

### 6.5.6 Further Time Stepping Considerations

The purpose of the present section is to interpret the time stepping scheme and assign the appropriate time levels for the E and L systems. The time stepping scheme must be closed, that is, it must satisfy the compatibility condition, and it must also not require an inappropriate or impossible order of operations. The following discussion is based on the staggered time stepping scheme (Section 6.5), which allows for two different algorithms to be selected in MOM. The first is the algorithm of Griffies (2004) and the second is based on the barotropic predictor-corrector (see Section 8.9 of Griffies, 2009). For completeness we restate the tracer budgets (which may be used by either of the aforementioned algorithms),

Equations (6.72) with the addition of the term for the destroyed blobs,

$$\begin{aligned}\partial_t(\mathrm{d}z \rho \mathcal{C})_{\mathrm{E}}^{ijk} &= -\nabla \cdot [\mathrm{d}z \rho (\mathbf{u} \mathcal{C} + \mathbf{F})]_{\mathrm{E}}^{ijk} - [\rho(w \mathcal{C} + F^{(s)})]_{\mathrm{E}}^{ijk-1} + [\rho(w \mathcal{C} + F^{(s)})]_{\mathrm{E}}^{ijk} \\ &\quad - \sum_{q=1}^{\mathcal{Q}_{\mathrm{E}}^{ijk}} \frac{[(\rho \mathcal{C})_{\mathrm{E}}^{ijk} \mathcal{E}^q - (\rho \mathcal{C})_{\mathrm{L}}^q \mathcal{D}^q] A_{\mathrm{interface}}^q}{\mathrm{d}A^{ij}} - \sum_{q=1}^{\mathcal{Q}_{\mathrm{new}}^{ijk}} \frac{m_{\mathrm{L}}^q \mathcal{C}_{\mathrm{E}}^{ijk}}{\Delta \tau \mathrm{d}A^{ij}} + \sum_{q=1}^{\mathcal{Q}_{\mathrm{dstry}}^{ijk}} \frac{(m \mathcal{C})_{\mathrm{L}}^q}{\mathrm{d}A^{ij}} \\ \partial_t(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^{ijk} &= - \sum_{q=1}^{\mathcal{Q}_{\mathrm{out}}^{ijk}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\Delta \tau} + \sum_{q=1}^{\mathcal{Q}_{\mathrm{in}}^{ijk}} \frac{(\mathrm{d}z \rho \mathcal{C})_{\mathrm{L}}^q}{\Delta \tau} + \sum_{q=1}^{\mathcal{Q}_{\mathrm{new}}^{ijk}} \frac{m_{\mathrm{L}}^q \mathcal{C}_{\mathrm{E}}^{ijk}}{\Delta \tau \mathrm{d}A^{ij}} \\ &\quad + \sum_{q=1}^{\mathcal{Q}_{\mathrm{dstry}}^{ijk}} \frac{[(\rho \mathcal{C})_{\mathrm{E}}^{ijk} \mathcal{E}^q - (\rho \mathcal{C})_{\mathrm{L}}^q \mathcal{D}^q] A_{\mathrm{interface}}^q}{\mathrm{d}A^{ij}} - \sum_{q=1}^{\mathcal{Q}_{\mathrm{dstry}}^{ijk}} \frac{(m \mathcal{C})_{\mathrm{L}}^q}{\Delta \tau \mathrm{d}A^{ij}}.\end{aligned}$$

The time labels for the mass budgets for E and L system divergence has been discussed in Sections 6.5.1 and 6.5.2.

Writing the compatible tracer version of the total mass budget,

$$\begin{aligned}[\partial_t(\mathcal{C} \rho \mathrm{d}z)]_{\tau}^{ijk} &= -\nabla \cdot [(\mathrm{d}z \rho \mathbf{u})_{\tau} \mathcal{C}_{\tau-1/2} + (\mathrm{d}z \rho) \mathbf{F}]_{\mathrm{E}}^{ijk} - [(\rho w)_{\tau} \mathcal{C}_{\tau-1/2} + \rho F^{(s)}]_{\mathrm{E}}^{ijk-1} \\ &\quad + [(\rho w)_{\tau} \mathcal{C}_{\tau-1/2} + \rho F^{(s)}]_{\mathrm{E}}^{ijk} + \sum_{q=1}^{\mathcal{Q}_{\mathrm{in}}^{ijk}} \frac{(\mathcal{C} m)_{\mathrm{L}}^q}{\Delta \tau \mathrm{d}A^{ij}} - \sum_{q=1}^{\mathcal{Q}_{\mathrm{out}}^{ijk}} \frac{(\mathcal{C} m)_{\mathrm{L}}^q}{\Delta \tau \mathrm{d}A^{ij}}. \quad (6.93)\end{aligned}$$

MOM handles the vertical SGS tracer concentration fluxes,  $F^{(s)}$ , implicitly in time. It is noted that the compatibility between tracer and mass does not place restrictions upon which time levels are used for the lateral SGS fluxes because the SGS fluxes vanish when  $\mathcal{C}$  is set to a constant globally (see Section 6.2.3). Following the discussion in Section 12.6.2 of Griffies (2004), the time label for the lateral fluxes can be written as

$$(\mathrm{d}z \rho) \mathbf{F} = (\mathrm{d}z \rho)_{\tau-1/2} \mathbf{F}_{\tau-1/2}. \quad (6.94)$$

As discussed in Chapter 21 of Griffies (2009), a slightly different approach is taken for the implicit vertical fluxes,  $F$  than to the lateral fluxes. Since the vertical fluxes are calculated implicitly in time, the updated tracer is given by,

$$(\mathrm{d}z \rho \mathcal{C})_{\tau+1/2}^{ijk} = (\mathrm{d}z \rho \tilde{\mathcal{C}})_{\tau+1/2}^{ijk} - \Delta \tau (J^{(s)}_{\tau+1/2}^{ijk-1} - J^{(s)}_{\tau+1/2}^{ijk}) \quad (6.95)$$

where  $J^{(s)} = \rho F^{(s)}$  is the SGS tracer flux and  $\tilde{\mathcal{C}}$  is the tracer concentration after explicit updates but before the implicit updates have been applied. The vertical fluxes generally take the form of a down gradient diffusion,

$$J^{(s)}_{\tau+1/2}^{ijk} = -\rho_0 \kappa^{ijk} \left( \frac{\tilde{\mathcal{C}}_{\tau+1/2}^{ijk} - \tilde{\mathcal{C}}_{\tau+1/2}^{ijk+1}}{\mathrm{dzwt}_{\tau+1/2}^{ijk}} \right), \quad (6.96)$$

where we recall that  $\mathrm{dzwt}$  is the vertical distance between tracer points (see Section 6.3.1).

Setting  $\rho = \rho_0$  is a choice justified by the fact that the error introduced by such an approximation is less than the uncertainty in the choice for the diffusivity,  $\kappa$ .

The entrainment and detrainment terms evolve the seawater mass and tracer mass from  $\tau - \frac{1}{2}$  to  $\tau + \frac{1}{2}$ . As discussed in Section 7.3.2, entrainment and detrainment are calculated at each sub-cycled step. Thus, if a blob transits a grid cell between  $\tau - 1/2$  and  $\tau + 1/2$ , it may contribute to the entrainment and detrainment terms even though that blob may not necessarily be in a particular grid cell at  $\tau - \frac{1}{2}$  or  $\tau + \frac{1}{2}$ . Thus when calculating the contributions to the E system tracer mass and seawater mass budgets, the sum is over all blobs present in a grid cell  $(i, j, k)$  between  $\tau - \frac{1}{2}$  and  $\tau + \frac{1}{2}$ , which is written as  $\mathcal{Q}_{\tau \mp 1/2}^{ijk}$ . The sum is also over the total number of sub-cycles that a blob was in the cell, which is written as  $(N^q)^{ijk}$ . The blob step size,  $h$ , for each sub-cycle can vary (see Section 7.3.1), thus the contribution to entrainment and detrainment needs to be time weighted. So, the entrainment and detrainment terms with the time labels revealed is

$$\sum_{q=1}^{\mathcal{Q}_{\text{new}}^{ijk}} \frac{(\rho_{\tau-1/2} \mathcal{C}_{\tau-1/2})_{\text{E}}^{ijk} \mathcal{E}^q A_{\text{interface}}^q}{dA^{ij}} = \sum_{q=1}^{\mathcal{Q}_{\tau \mp 1/2}^{ijk}} \sum_{n=1}^{(N^q)^{ijk}} \frac{h^n (\rho_{\tau-1/2} \mathcal{C}_{\tau-1/2})_{\text{E}}^{ijk} \mathcal{E}^n A_{\text{interface}}^n}{\Delta \tau dA^{ij}} \quad (6.97a)$$

$$\sum_{q=1}^{\mathcal{Q}_{\text{new}}^{ijk}} \frac{(\rho \mathcal{C})_{\text{L}}^q \mathcal{D}^q A_{\text{interface}}^q}{dA^{ij}} = \sum_{q=1}^{\mathcal{Q}_{\tau \mp 1/2}^{ijk}} \sum_{n=1}^{(N^q)^{ijk}} \frac{h^n (\rho \mathcal{C})_{\text{L}}^n \mathcal{D}^n A_{\text{interface}}^n}{\Delta \tau dA^{ij}}, \quad (6.97b)$$

The term for new blobs must be divided into two parts, one for blobs that are formed explicitly in time ( $\mathcal{Q}_{(\text{explicit})}$  at time level  $\tau - \frac{1}{2}$ ) and the other for blobs that are formed implicitly in time ( $\mathcal{Q}_{(\text{implicit})}$  at time level  $\tau + \frac{1}{2}$ ). Examples of blobs that are formed explicitly in time are those formed by the shallow-ocean/deep-ocean condition (i.e. dynamically active bottom blobs), while blobs that are formed implicitly in time are those formed by the vertical stability condition (i.e. dynamically active open ocean blobs). The term is thus written as

$$\sum_{q=1}^{\mathcal{Q}_{(\text{new})}^{ijk}} m_{\text{L}}^q \mathcal{C}_{\text{E}}^{ijk} = \left[ \sum_{q=1}^{\mathcal{Q}_{(\text{explicit})}^{ijk}} m_{\text{L}}^q \mathcal{C}_{\text{E}}^{ijk} \right]_{\tau-1/2} + \left[ \sum_{q=1}^{\mathcal{Q}_{(\text{implicit})}^{ijk}} m_{\text{L}}^q \mathcal{C}_{\text{E}}^{ijk} \right]_{\tau+1/2}. \quad (6.98)$$

As discussed in Sections 6.5.3 and 8.1, the test to ascertain whether it is necessary to destroy one or more blobs in a grid cell to maintain a positive E system thickness is conducted when updating the thickness to time  $\tau + \frac{1}{2}$ . As such, the appropriate time label for the destroyed blobs is to be  $\tau + \frac{1}{2}$ , and the term is written as

$$\sum_{q=1}^{\mathcal{Q}_{(\text{dstry})}^{ijk}} (m \mathcal{C})_{\text{L}}^q = \left[ \sum_{q=1}^{\mathcal{Q}_{(\text{dstry})}^{ijk}} (m \mathcal{C})_{\text{L}}^q \right]_{\tau+1/2}. \quad (6.99)$$

All necessary information about time labels has now been discussed. Synthesising all of this information, the discrete equation for the algorithm for updating the tracer content

of the E system is

$$\begin{aligned}
\frac{[(dz\rho\mathcal{C})_{\tau+1/2} - (dz\rho\mathcal{C})_{\tau-1/2}]_E^{ijk}}{\Delta\tau} &= -\nabla \cdot [(dz\rho)_\tau \mathbf{u}_\tau \mathcal{C}_{\tau-1/2} + (dz\rho)_{\tau-1/2} \mathbf{F}_{\tau-1/2}]_E^{ijk} \\
&\quad + [(w\rho)_\tau \mathcal{C}_{\tau-1/2} + \rho_0 F_{\tau+1/2}]_E^{ijk} \\
&\quad - [(w\rho)_\tau \mathcal{C}_{\tau-1/2} + \rho_0 F_{\tau+1/2}]_E^{ijk-1} \\
&\quad - \sum_{q=1}^{\mathcal{Q}_{\tau+1/2}^{ijk}} \sum_{n=1}^{(N^q)^{ijk}} \frac{h^n(\rho_{\tau-1/2} \mathcal{C}_{\tau-1/2})_E^{ijk} \mathcal{E}^n A_{\text{interface}}^n}{\Delta\tau dA^{ij}} \\
&\quad + \sum_{q=1}^{\mathcal{Q}_{\tau+1/2}^{ijk}} \sum_{n=1}^{(N^q)^{ijk}} \frac{h^n(\rho\mathcal{C})_L^n \mathcal{D}^n A_{\text{interface}}^n}{\Delta\tau dA^{ij}} \\
&\quad - \left[ \sum_{q=1}^{\mathcal{Q}_{(\text{explicit})}^{ijk}} \frac{m_L^q \mathcal{C}_E^{ijk}}{\Delta\tau dA^{ij}} \right]_{\tau-1/2} - \left[ \sum_{q=1}^{\mathcal{Q}_{(\text{implicit})}^{ijk}} \frac{m_L^q \mathcal{C}_E^{ijk}}{\Delta\tau dA^{ij}} \right]_{\tau+1/2} \\
&\quad + \left[ \sum_{q=1}^{\mathcal{Q}_{(\text{dstry})}^{ijk}} \frac{(m\mathcal{C})_L^q}{\Delta\tau dA^{ij}} \right]_{\tau+1/2}.
\end{aligned} \tag{6.100}$$

This equation is closed, that is, it obeys the compatibility condition and all terms are compatible without any order of operation problems.

## 6.6 Summary

Some discretisation and implementation issues that relate to the partitioning of an ocean climate model into an E system and an L system have been discussed. Details of the implementation in MOM, as well as some simplification to the combined system's momentum and energy budgets have also been detailed.

Discrete methods for calculating the combined system's density, tracer concentration and hydrostatic pressure have also been outlined. The methods for calculating prognostic variables related to the technology that facilitates generalised level coordinates was also detailed.

Strategies for ensuring tracer and mass compatibility, as well as closed time stepping schemes were also discussed. To this end, time labels and the order of certain important operations were detailed. A vertical CFL condition was derived for the E system, as well as a strategy to ensure that the total mass of the L system in a particular grid cell, does not exceed the total allowable mass for that grid cell.



# Chapter 7

## Space and Time Methods for Blobs

The embedded Lagrangian system can be viewed as a sub-model of the ocean model. The previous chapter is largely concerned with the implementation of admitting a Lagrangian sub-model. The present chapter is concerned with the methods and implementation of the embedded Lagrangian model itself.

### 7.1 Blobs and the Eulerian Grid

How the blobs use the information provided by the Eulerian model and grid is the main subject of this section. Methods for point location and interpolation methods are discussed.

#### 7.1.1 Horizontal Projection Considerations

To conduct finite difference calculus on an arbitrary orthogonal grid, it is not sufficient to merely know the location of the four grid points defined in Figure 6.3, it is also necessary to know the metric function that specifies the coordinate system being used. Distances along the  $\hat{x}$  and  $\hat{y}$  directions are approximated as (see Chapter 4.1.2.2 of Griffies et al., 2004, for further details),

$$\Delta x[a, b] = |\xi_1^a - \xi_1^b| (h_1^a + h_1^b) / 2 \quad (7.1a)$$

$$\Delta y[c, d] = |\xi_2^c - \xi_2^d| (h_2^c + h_2^d) / 2, \quad (7.1b)$$

where  $\boldsymbol{\xi} = (\xi_1, \xi_2)$  is some generalised lateral orthogonal coordinate system and  $\mathbf{h} = (h_1, h_2)$  are the stretching functions that describe the system. In MOM, the stretching functions are defined on the  $\mathcal{T}$  and  $\mathcal{U}$  grids.

For a longitude-latitude grid on a symmetric spherical domain, the zonal and merid-

ional displacements are defined as

$$\Delta x[a, b] = R \cos \phi_s (\lambda_s^b - \lambda_s^a) \quad (7.2a)$$

$$\Delta y[c, d] = R(\phi_s^d - \phi_s^c), \quad (7.2b)$$

where  $\phi_s$  is latitude,  $R$  is the radius of the sphere (Earth) and  $\lambda_s$  is the longitude.

In order to calculate distances accurately, the Lagrangian blobs are required to utilise two sets of coordinate variables. One set is the coordinate of the orthogonal coordinate system,  $\xi^q = (\xi_1^q, \xi_2^q)$ , and the other,  $\mathbf{x}^q = (x^q, y^q)$ , is a distance in metres relative to the initial position of the blob at time  $\tau$ , that is,  $x^q(\tau) = 0$ , and  $y^q(\tau) = 0$ .

As discussed in Section 7.3, the prognostic variables are  $\mathbf{x}^q$  rather than  $\xi$ . Thus, the position of a blob on the grid is diagnosed after a blob trajectory from  $a$  to  $b$  has been calculated

$$\xi_1^b = \xi_1^a + \frac{2(\Delta x[a, b])}{h_1^b + h_1^a} \quad (7.3a)$$

$$\xi_2^b = \xi_2^a + \frac{2(\Delta y[a, b])}{h_2^b + h_2^a}. \quad (7.3b)$$

It should be noted that in general  $h_1 = h_1(\xi_1, \xi_2)$  and  $h_2 = h_2(\xi_1, \xi_2)$ . In turn, there may be dependencies on  $\xi_1$  and  $\xi_2$  in the momentum equations (e.g. the Coriolis term). How such a system is evolved will be discussed in Sections 7.3.1 and 7.3.2.

### 7.1.2 Blob Grid Cell Search Algorithms in the Horizontal

Knowing when a blob has moved from one grid cell to another is key to calculations of the E system-L system interaction. A method is required to tell if a blob has crossed from one water column into a neighbouring column. Use of grids that are not regular in longitude-latitude space (such as the tripolar grid of Murray, 1996) complicate the logic required to tell what grid cell a blob is in.

One method that is grid independent is to test whether the vector,  $\overrightarrow{\xi_T^{ij} \xi^b}$ , which joins the tracer point  $\mathcal{T}^{ij}$  to the final position of the blob,  $\xi(b)$ , intersects any of the line segments  $\overrightarrow{\xi_U^{ij} \xi_U^{i-1j}}$ ,  $\overrightarrow{\xi_U^{ij} \xi_U^{ij-1}}$ ,  $\overrightarrow{\xi_U^{i-1j-1} \xi_U^{i-1j}}$ , or  $\overrightarrow{\xi_U^{i-1j-1} \xi_U^{ij-1}}$ . Here, the short hand  $\overrightarrow{\cdot \cdot}$  indicates a line segment joining the positions indicated. The subscript indicates the type of grid point the end-point of the vector is (i.e. tracer or velocity) superscript indicates which specific grid point we are considering. For example, the vector  $\overrightarrow{\xi_U^{ij} \xi_U^{i-1j}}$  joins the  $(i, j)$  velocity point with the  $(i-1, j)$  velocity point (i.e. it is the vector of the eastern face of a tracer grid cell).

The properties of the five vectors defined above, and the properties of the cross product shown in Figure 7.1 (Cormen et al., 2001) can be exploited to decide whether a blob's

trajectory has taken it outside of a grid cell or not. That is, whether the vector  $\overrightarrow{\xi_T^{ij}\xi^b}$  crosses a line formed by the line segments joining the four corners of the grid cell.

Firstly, the properties of the cross product are reviewed. If there are three points,  $\xi^0$ ,  $\xi^1$  and  $\xi^2$ , and we wish to know whether  $\xi^2$  is clockwise, anticlockwise or collinear relative to  $\xi^0$ , the cross product of  $\overrightarrow{\xi^0\xi^1}$  and  $\overrightarrow{\xi^0\xi^2}$  can be taken. The sign of that cross product yields information about the relative direction of the points, as illustrated in Figure 7.1.

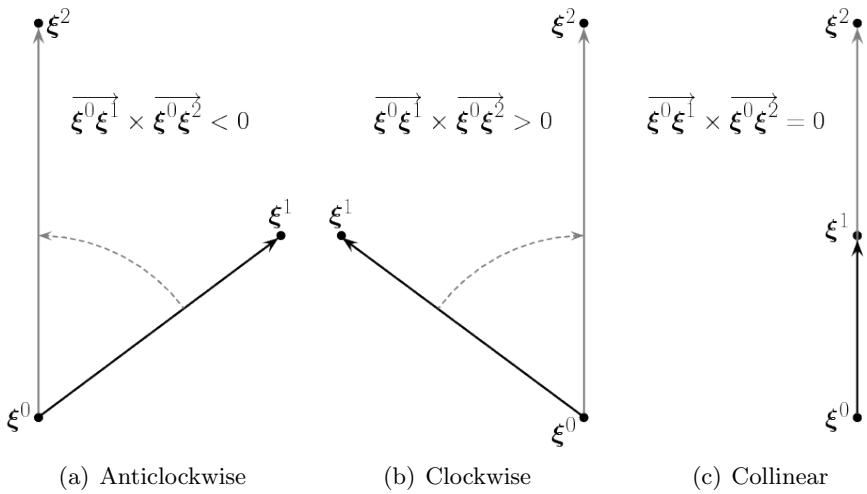


Figure 7.1: Properties of the cross product, which allows us to tell the angular orientation of one vector relative to another without the explicit calculation of the angle between them.

If two vectors  $\overrightarrow{\xi^0\xi^1}$  and  $\overrightarrow{\xi^2\xi^3}$  intersect, then the end points of one vector must be on either side of the first vector. The same must also be true for the converse, as can be seen in Figure 7.2.

The cross product can be calculated as the determinant of a  $2 \times 2$  matrix, and this can be used to determine whether two vectors intersect without the need to calculate angles, or use division (both of which are relatively expensive operations) . As illustrated in Figure 7.2, if we have,

$$\begin{aligned} \overrightarrow{\xi^0\xi^2} \times \overrightarrow{\xi^0\xi^1} &> 0 & \overrightarrow{\xi^0\xi^3} \times \overrightarrow{\xi^0\xi^1} &< 0 \\ \overrightarrow{\xi^2\xi^1} \times \overrightarrow{\xi^2\xi^3} &< 0 & \overrightarrow{\xi^2\xi^0} \times \overrightarrow{\xi^2\xi^3} &> 0, \end{aligned}$$

then the vectors intersect. One other possibility exists, and that is that  $\xi^3$  is collinear with  $\overrightarrow{\xi^0 \xi^1}$ .

The validity of the above method depends largely on the properties of the grid in question. It is assumed that lines formed by the grid are straight lines in latitude-longitude space. For spherical and Mercator grids, this is true by definition. For a grid such as the

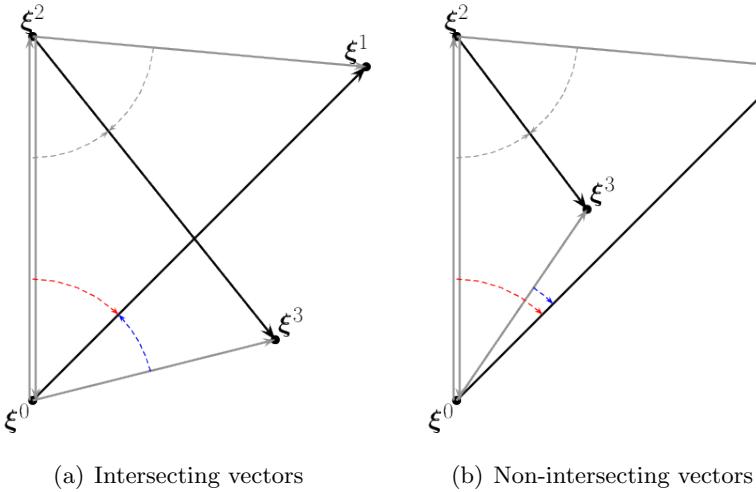


Figure 7.2: The conditions under which vectors intersect and do not intersect. All points are the same between the two sub-figures, except for  $\xi^3$ . In the left panel  $\xi^3$  is to the right of the vector joining  $\xi^0$  and  $\xi^1$ . Thus the vector joining  $\xi^2$  and  $\xi^3$  does intersect with the vector joining  $\xi^0$  and  $\xi^1$ . In the other panel,  $\xi^3$  is to the right and thus, the vectors do not intersect. We can tell if the vectors intersect or not from the properties of the cross product.

tripolar grid of Murray (1996) it is not true. If the grid spacing is refined enough, however, it is a reasonable approximation (and is the same approximation already in use by MOM when using the bipolar arctic grid). We proceed assuming that the grid is refined enough so as not to introduce large errors.

For the blobs, the intersection condition can be checked separately for each face. The eastern face's condition is,

$$\overrightarrow{\xi_T^{ij} \xi_U^{ij}} \times \overrightarrow{\xi_T^{ij} \xi^b} < 0 \quad (7.4a)$$

$$\overrightarrow{\xi_T^{ij} \xi_U^{ij-1}} \times \overrightarrow{\xi_T^{ij} \xi^b} > 0 \quad (7.4b)$$

$$\overrightarrow{\xi_U^{ij-1} \xi_T^{ij}} \times \overrightarrow{\xi_U^{ij-1} \xi_U^{ij}} < 0 \quad (7.4c)$$

$$\overrightarrow{\xi^b \xi_U^{ij}} \times \overrightarrow{\xi_U^{ij} \xi_U^{ij-1}} > 0, \quad (7.4d)$$

the northern face's condition is,

$$\overrightarrow{\xi_T^{ij} \xi_U^{i-1j}} \times \overrightarrow{\xi_T^{ij} \xi^b} > 0 \quad (7.5a)$$

$$\overrightarrow{\xi_T^{ij} \xi_U^{ij}} \times \overrightarrow{\xi_T^{ij} \xi^b} > 0 \quad (7.5b)$$

$$\overrightarrow{\xi_U^{ij} \xi_T^{ij}} \times \overrightarrow{\xi_U^{ij} \xi_U^{i-1j}} < 0 \quad (7.5c)$$

$$\overrightarrow{\xi^b \xi_U^{ij}} \times \overrightarrow{\xi_U^{ij} \xi_U^{i-1j}} < 0, \quad (7.5d)$$

the western face's condition is,

$$\overrightarrow{\xi_T^{ij} \xi_U^{i-1j-1}} \times \overrightarrow{\xi_T^{ij} \xi^b} < 0 \quad (7.6a)$$

$$\overrightarrow{\xi_T^{ij} \xi_U^{i-1j}} \times \overrightarrow{\xi_T^{ij} \xi^b} > 0 \quad (7.6b)$$

$$\overrightarrow{\xi_U^{i-1j} \xi_T^{ij}} \times \overrightarrow{\xi_U^{i-1j} \xi_U^{i-1j-1}} < 0 \quad (7.6c)$$

$$\overrightarrow{\xi^b \xi_U^{i-1j-1}} \times \overrightarrow{\xi_U^{i-1j-1} \xi_U^{i-1j}} > 0, \quad (7.6d)$$

and finally, the southern face's condition is,

$$\overrightarrow{\xi_T^{ij} \xi_U^{i-1j}} \times \overrightarrow{\xi_T^{ij} \xi^b} > 0 \quad (7.7a)$$

$$\overrightarrow{\xi_T^{ij} \xi_U^{ij}} \times \overrightarrow{\xi_T^{ij} \xi^b} < 0 \quad (7.7b)$$

$$\overrightarrow{\xi_U^{ij-1} \xi_T^{ij}} \times \overrightarrow{\xi_U^{ij-1} \xi_U^{i-1j-1}} > 0 \quad (7.7c)$$

$$\overrightarrow{\xi^b \xi_U^{ij}} \times \overrightarrow{\xi_U^{ij} \xi_U^{i-1j}} < 0. \quad (7.7d)$$

If all of the conditions for one direction is found to be true, all directions are checked again for the new grid cell (except for the line segment that has already been crossed). Retesting of the condition for the new grid cell is required if, for instance, a blob moves diagonally (e.g. from the cell  $\mathcal{T}^{ij}$  to cell  $\mathcal{T}^{i+1j+1}$ ).

It is also worth noting that in practice, only three conditions for each direction need to be tested, as each condition has one cross product that does not involve the position of a blob. The condition that does not involve a blob is always true as the horizontal Eulerian grid is fixed in time.

Implicit in the conditions stated above is that if a blob is precisely on a grid cell boundary (i.e. the blob position is collinear with the grid cell face), the blob is not considered to have changed water columns.

### 7.1.3 Cyclic Grids

Knowledge of the index of the grid cell that a blob resides in, as well as its position in the E coordinate system, is essential to the correct functioning of the L system. For cyclic (periodic) grids, extra care must be taken to ensure that a blob remains within the bounds of the grid cell indexing system, and, within the coordinate bounds of the model domain. This requires the resetting of variables if a blob goes off the domain.

To ensure this it must be first ascertained whether a blob has crossed a domain boundary. Then, what the blob's position in the model domain should be calculated (e.g. if a blob travels 3km to the east of the domain, it should be placed 3km from the western edge of the model domain). Figure 7.3 illustrates a blob crossing a domain boundary, and, where it should be placed after crossing the boundary.

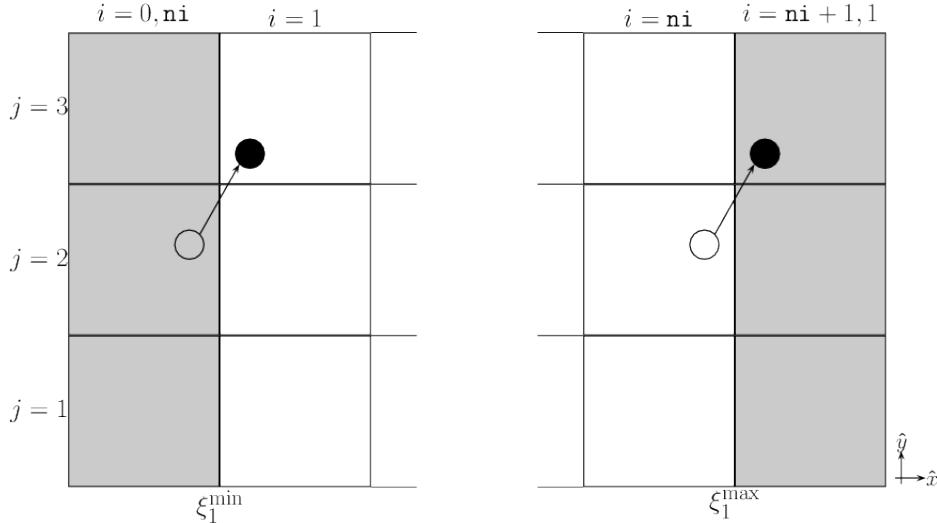


Figure 7.3: An illustration of how a blob's grid cell index and location must be reset when crossing a domain boundary on a cyclic grid. Depicted here is a cyclic grid in the zonal direction. Shaded cells indicate halo cells (i.e. duplicate grid cells that enable the model to wrap from its eastern most grid cell  $i = \text{ni}$  to its western most grid cell  $i = 1$ ). The black circles and line indicate a blob that has gone from the grid cell  $(\text{ni}, 2)$  to  $(\text{ni}+1, 3) = (1, 3)$ .

To reset the zonal direction,

$$i^{\text{reset}} = \begin{cases} i + \text{ni} & \text{if } i < 1 \\ i - \text{ni} & \text{if } i > \text{ni} \end{cases}, \quad (7.8a)$$

where  $\text{ni}$  is the total number of zonal grid cells. Similarly for the meridional direction,

$$j^{\text{reset}} = \begin{cases} j + \text{nj} & \text{if } j < 1 \\ j - \text{nj} & \text{if } j > \text{nj} \end{cases}. \quad (7.8b)$$

To reset the values for the zonal and meridional coordinate, the Fortran elemental function `modulo` is utilised. `modulo` is defined as (see Section 8.3 of Metcalf et al., 2004),

$$\text{modulo}(a, b) = a - \text{floor}(a/b) \times b, \quad (7.9)$$

where `floor` is also a Fortran elemental function, and returns the greatest integer less than or equal to its argument (Metcalf et al., 2004).

The zonal coordinate of the  $q^{\text{th}}$  blob is reset as

$$\xi_1^{\text{reset}} = \begin{cases} \xi_1^{\text{max}} - \text{modulo}(\xi_1^{\text{min}}, \xi_1^q) & \text{if } \xi_1^q < \xi_1^{\text{min}} \\ \xi_1^{\text{min}} + \text{modulo}(\xi_1^q, \xi_1^{\text{max}}) & \text{if } \xi_1^{\text{max}} < \xi_1^q \end{cases}, \quad (7.10a)$$

and, similarly for the meridional coordinate,

$$\xi_2^{\text{reset}} = \begin{cases} \xi_2^{\max} - \text{modulo}(\xi_2^{\min}, \xi_2^q) & \text{if } \xi_2^q < \xi_2^{\min} \\ \xi_2^{\min} + \text{modulo}(\xi_2^q, \xi_2^{\max}) & \text{if } \xi_2^{\max} < \xi_2^q \end{cases}, \quad (7.10b)$$

#### 7.1.4 The Tripolar Grid

In order to avoid problems with grid convergence at the north pole, many configurations of MOM utilise the composite tripolar grid of Murray (1996) which places one pole over Antarctica and two poles in the Arctic region. Thus, all three poles may be placed over land, obviating problems with singular points within the ocean domain. In configurations utilising the tripolar grid, the typical spherical grid is used generally from approximately 65°N southwards and the bipolar Arctic grid to the north (see Section 4.2 of Griffies et al., 2004).

The bipolar Arctic grid is a simply rotated sphere such that at the join (at latitude  $\phi_P$ ), the rotated grid's parallels align with the unrotated grid's meridians (see Figure 7 of Murray, 1996). The rotated grid's parallels are  $\lambda_c$  and the meridians are  $\phi_c$ . The rotated grid's parallels match the sphere's meridians at the join, that is,  $\xi_2(\lambda_c) = \xi_2(\phi_s)$  at  $\phi_P$ .

Utilisation of grids where the coordinate system does not coincide with latitude and longitude has implications for the Lagrangian system. For instance, calculating point locations to decide which grid cell a blob resides in. The location scheme discussed in Section 7.1.2 is robust for such grids.

As for the cyclic grid, a strategy for resetting the grid cell index that a blob resides in when it crosses the bipolar fold must be developed. The Arctic bipolar grid is illustrated in Figure 7.4.

When crossing the bipolar fold, the zonal and meridional grid cell indices are reset as

$$\left. \begin{array}{l} i^{\text{reset}} = (\mathbf{n}i + 1) - i \\ j^{\text{reset}} = \mathbf{n}j \end{array} \right\} \text{if } j > \mathbf{n}j. \quad (7.11)$$

While the grid cell faces are defined by the Arctic bipolar grid, the blobs still operate on a longitude-latitude grid. Zonally, that is a cyclic grid and is handled as described in Section 7.1.3. Meridionally, however, a blob's latitude and meridional velocity need to be reset if the blob passes through the geographic north pole

$$\left. \begin{array}{l} \xi_2^{\text{reset}} = 180 - \xi_2^q \\ v^{\text{reset}} = -v^q \end{array} \right\} \text{if } \xi_2^q > 90 \quad (7.12)$$

The E system's velocity within the Arctic bipolar grid does not align with latitude and longitude, while, the blob's velocity does. Thus, the E system's lateral velocity must be

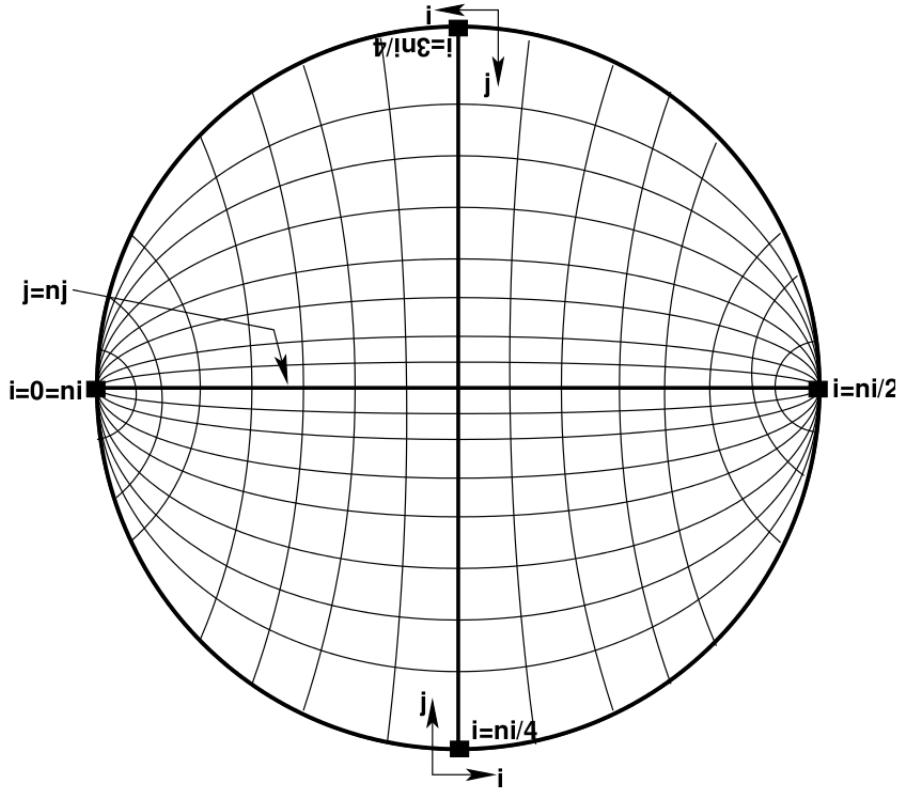


Figure 7.4: The Arctic bipolar grid with selected grid labels. The bipolar fold is the line that joins  $i = 0, ni$  and  $i = ni/2$ . From Figure 4.6 of Griffies et al. (2004).

rotated in order to calculate the drag terms in the blob momentum equations

$$u_E^{\text{zonal}} = v_E \sin \theta + u_E \cos \theta \quad (7.13)$$

$$v_E^{\text{merid}} = v_E \cos \theta + u_E \sin \theta, \quad (7.14)$$

where  $\theta$  is the clockwise angle between lines of longitude-latitude and the grid.

### 7.1.5 Blob Grid Cell Search Algorithms in the Vertical

It is not necessary to search for the vertical grid cell index of a dynamically active bottom blob since it is assumed that a dynamically active bottom blob always resides in the bottom grid cell. We are only required to find the vertical grid cell for free blobs as bottom blobs are always assumed to reside in the bottom grid cell.

A blob must also know which grid cell it resides in at each blob sub-cycle. When deciding on a blob's vertical grid cell at each sub-cycle a number of computationally cheap best guesses are attempted, which are based on the depth of coordinate surfaces, the depth of the blob and the vertical velocity of the blob. If those best guesses fail a brute force search of the entire water column is attempted. The coordinate surfaces are assumed to remain constant over the period of  $\Delta\tau$  between E system time steps.

At each Eulerian time step, information about the grid cell location of a blob must be given to the E system. As discussed in Sections 7.3.1 and 8.1, knowledge of which grid cell a blob resides in is required prior to knowing the depth of coordinate surfaces. Thus, a brute force search of the water column at time  $\tau + 1/2$  is conducted in the model's native vertical coordinate at the new Eulerian time step. The search is conducted, checking whether a blob resides between two coordinate surfaces, starting at either the top or the bottom grid cell (depending on the coordinate system).

If it is found that a blob no longer resides in the model domain (i.e. has penetrated the surface or bottom), it is either destroyed, or transferred to another dynamic regime as appropriate. This can happen during a sub-cycle, or, it can happen at an Eulerian time step.

Here, we detail the brute force approach taken for searching for a blob's vertical grid cell in the various vertical coordinate systems. For depth based coordinates, the search begins in the upper most grid cell and proceeds downward. In the case of  $z^*$  coordinates, the blob's position is calculated in  $z^*$  space,

$$z^{*q} = H^{ij}(z^q - \eta^{ij})/(H^{ij} + \eta^{ij}). \quad (7.15a)$$

The values for  $z_{\mathcal{W}}^{*k}$  is known by the model (where  $z_{\mathcal{W}}$  is the value of the vertical coordinate at a  $\mathcal{W}$  point), the algorithm thus searches downward until one of the conditions,

$$\begin{aligned} z_{\mathcal{W}}^{*k} \leq z^{*q} \leq 0 & \quad \text{for } k = 1 \\ z_{\mathcal{W}}^{*k} \leq z^{*q} < z_{\mathcal{W}}^{*k-1} & \quad \text{for } 1 < k < k_{\text{bot}}^{ij} \\ -(H + \eta)^{ij} \leq z^{*q} < z_{\mathcal{W}}^{*k-1} & \quad \text{for } k = k_{\text{bot}}^{ij}, \end{aligned} \quad (7.15b)$$

is satisfied.

For pressure based coordinates, the search begins in the bottom grid cell,  $k_{\text{bot}}$ , since the prognostic variable is bottom pressure. For the pressure coordinate, it is convenient to use the geopotential, rather than pressure space

$$\Phi + gH = g \int_{-H}^z dz \quad (7.16a)$$

$$= g \int_{p_b}^p z_{,p} dp \quad (7.16b)$$

$$= g \int_{p_b^0}^{p^*} z_{,p} dp^*, \quad (7.16c)$$

where  $p_b^0$  is a reference bottom pressure. The known variable for the blobs is the depth relative to  $z = 0$ . As such, the blob's geopotential is given using Equation (7.16a)

$$\Phi^q = gz^q. \quad (7.17)$$

Similarly to  $z_{\mathcal{W}}^{*k}$ , the values for  $p_{\mathcal{W}}^k$  coordinate surfaces are known, and as such, the

geopotential is calculated as

$$\Phi = -g \int_{p_b}^p dp - gH, \quad (7.18a)$$

where the identity for  $z_{,p}$  has been used (see Table 6.1). In discrete form Equation (7.18a) is given by

$$\Phi_{\mathcal{W}}^{ijk} = g \sum_{k=k_{\text{bot}}^{ij}}^K dp^{ijk} - gH^{ij}. \quad (7.18b)$$

For  $p^*$  coordinates the geopotential is calculated as

$$\Phi = -g \int_{p_b^0}^{p^*} \frac{p_b - p_a}{p_b^0} dp^* - gH, \quad (7.19a)$$

where the identity for  $z_{,p^*}$  from Table 6.1 has been used. In discrete form Equation (7.19a) is given by

$$\Phi_{\mathcal{W}}^{ijk} = g \sum_{k=k_{\text{bot}}^{ij}}^K \frac{(p_b - p_a)^{ij}}{p_b^0} dp^{*k} - gH^{ij}. \quad (7.19b)$$

The values for  $dp$  and  $dp^*$  are values that are carried by the model prior to the calculation of the grid cell thickness (and indeed, remain constant throughout the model run with the exception of the upper and bottom grid cells for pressure coordinates). The search conditions for both  $p$  and  $p^*$  are the same, but, the method for calculating the geopotential is different. The algorithm searches from the bottom upwards until one of the conditions,

$$\begin{aligned} -gH^{ij} &\leq \Phi^q < \Phi_{\mathcal{W}}^{ijk-1} && \text{for } k = k_{\text{bot}}^{ij} \\ \Phi_{\mathcal{W}}^{ijk} &\leq \Phi^q < \Phi_{\mathcal{W}}^{ijk-1} && \text{for } k_{\text{bot}}^{ij} < k < 1 \\ \Phi_{\mathcal{W}}^{ijk} &\leq \Phi^q \leq 0 && \text{for } k = 1, \end{aligned} \quad (7.20)$$

is satisfied.

### 7.1.6 Topography Considerations

For dynamically active bottom blobs to have the impetus to move down a slope, they must “feel” the slope. It can be seen in the bottom blob momentum Equations (5.18) that the impetus to move down the slope comes from a non-zero topographic gradient,  $\nabla H \neq 0$ . The question thus becomes, how to interpret the topography for the blobs. In principle, the topography used by the blobs may be arbitrary. This facilitates the possibility of being able to use a higher resolution topography than that used by the Eulerian model. This capability has not, however, been implemented in MOM, and we utilise the depth of the E system topography to find the topography for the blobs.

The remapping of topography has received some attention in recent times due to the advent of hybrid coordinate systems. The constraints on remapping to accurately represent the underlying data is an important aspect of choosing a remapping scheme (White and Adcroft, 2008; White et al., 2009). The present situation has different requirements

to the typical remapping problem faced by ocean modellers, in that the blob scheme is not concerned with conserving volume (in order to accurately calculate transport), but, it is mostly concerned with obtaining a continuous, and relatively smooth surface that joins the topography data points.

Smooth topography,  $H$ , and a smooth topographic gradient  $\nabla H$ , are important for integrating smooth trajectories for bottom blobs. However, the computational expense of obtaining smooth functions must also be considered. The strategy employed in the present implementation is to take the topography, defined on the  $\mathcal{T}$  grid, and calculate the gradient of the topography on the  $\mathcal{U}$  grid during initialisation, and then use the horizontal interpolation scheme described in Section 7.1.8 to interpolate  $H$  and  $\nabla H$  to individual blobs. This is a relatively cheap operation, as the coefficients required for the interpolation must be calculated for other variables anyway.

### 7.1.7 Grounding and Penetration of the Surface

Blobs that are grounded (i.e. a blob that has entered a column with no wet cells) are destroyed at the last calculated position prior to being grounded. Similarly, blobs that penetrate the free surface are destroyed at their last calculated position.

### 7.1.8 Interpolation to Blobs

Some E system variables have to be projected onto individual blobs in order to calculate the evolution of the blob prognostic variables. For smooth behaviour, the E system variables being used to calculate blob trajectories must also vary smoothly.

There are four types of interpolations that may required, two in the horizontal and two in the vertical. In the horizontal, some variables are interpolated from the  $\mathcal{U}$  grid and some from the  $\mathcal{T}$  grid. In the vertical, there are some variables that are interpolated from tracer points on the  $\mathcal{T}$  grid and there are some from the vertical velocity grid,  $\mathcal{W}$  to a blob.

In practice, many variables need to be interpolated in the horizontal and the vertical for free blobs, while for bottom blobs, interpolation only occurs in the horizontal. For free blobs, horizontal interpolation occurs along coordinate surfaces. The values are then interpolated in the vertical. In contrast, bottom blobs interpolate values along topographic surfaces (i.e. the bottom most grid cells).

The horizontal interpolation schemes are based on inverse distance weighting (Shepard, 1968). The points that participate in the interpolation for the  $\mathcal{T}$  and  $\mathcal{U}$  grids are illustrated in Figure 7.5.

To find the value of a variable,  $A_H$ , at the horizontal position of a blob,  $q$  the inverse

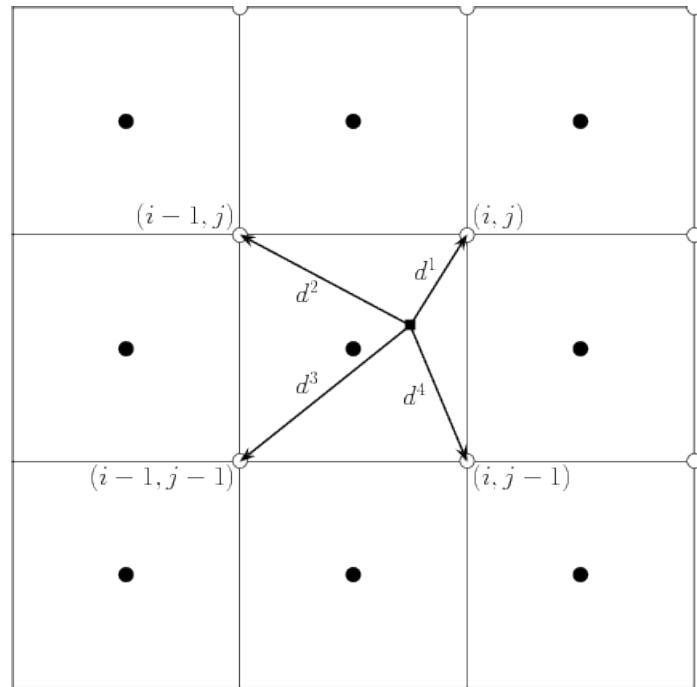
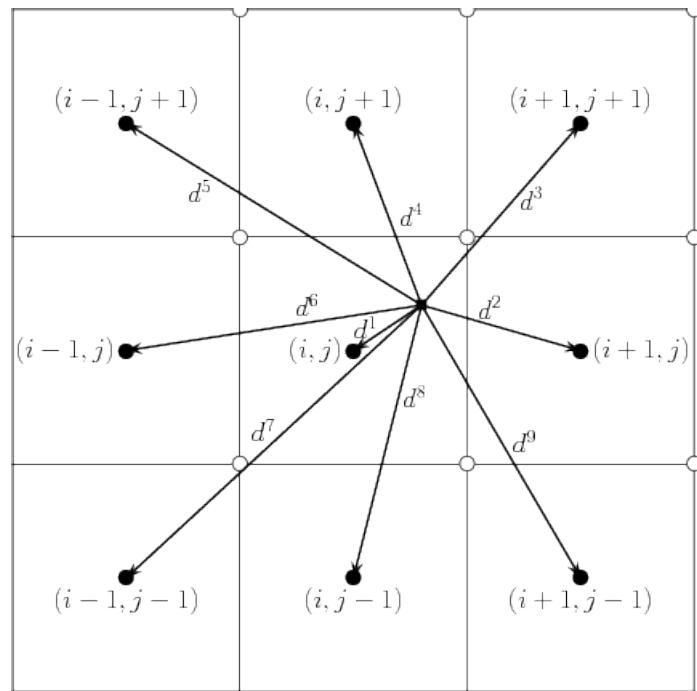
(a)  $\mathcal{U}$  grid interpolation(b)  $\mathcal{T}$  grid interpolation

Figure 7.5: Horizontal interpolation from the  $\mathcal{U}$  grid (unfilled circles) and the  $\mathcal{T}$  grid (solid circles) to a blob (solid square). The distances,  $d^l$ , are used for the distance weighting with  $l = 1, \dots, 4$  for the  $\mathcal{U}$  grid and  $l = 1, \dots, 9$  for the  $\mathcal{T}$  grid.

distance weighting formula is

$$A_H^q = \begin{cases} D^{-1} \sum_{l=1}^L (d^l)^{-2} A^l & \text{if all } d^l \neq 0 \\ A^l & \text{if any } d^l \equiv 0 \end{cases}, \quad (7.21)$$

where  $d^l$  indicates the distance to the  $l^{\text{th}}$  point (see Figure 7.5) and is calculated using the calculus described in Section 7.1.1

$$d^l = \frac{1}{2} \left\{ |\xi_1^q - \xi_1^l|^2 (h_1^q + h_1^l)^2 + |\xi_2^q - \xi_2^l|^2 (h_2^q + h_2^l)^2 \right\}^{\frac{1}{2}}. \quad (7.22)$$

Here,  $(\xi_1^q, \xi_2^q)$  and  $(h_1^q, h_2^q)$  indicate the coordinate and metric functions at the blob's location, and  $(\xi_1^q, \xi_2^q)$  and  $(h_1^q, h_2^q)$  indicates the coordinate and metric function at the  $l^{\text{th}}$  grid location.  $D$  is calculated as

$$D = \sum_{l=1}^L (d^l)^{-2}. \quad (7.23)$$

There must be special treatment given to solid wall boundaries (i.e. not cyclic and not tripolar), so that when a blob is in the vicinity of a boundary, all of the values that are participating in the interpolation are from the grid. For instance, for a blob in a grid cell neighbouring a western solid wall boundary, the  $\mathcal{U}$  interpolation will only use the values for  $l = 1, 4$  and the  $\mathcal{T}$  interpolation will only use the values for  $l = 1, 2, 3, 4, 8, 9$ .

Different variables have different requirements for land points. For instance the interpolation for density should not include solid land points, so, for the density interpolation, solid land points are ignored. On the other hand, the grid stretching function is still valid over land points, so, land points are not ignored for the stretching function.

One final note is with regards to how the stretching function,  $\mathbf{h}^q$  is interpolated. In MOM, the stretching function is originally defined on the  $\mathcal{T}$  grid, however, it is noted that from Equations (7.21) to (7.23) a value for  $\mathbf{h}^q$  is required to obtain the coefficients needed to perform the interpolation. This presents an order of operations problem. To overcome this problem, the value of  $\mathbf{h}^q$  from the previous value at time step  $n$  is used and updated the same as any other interpolated variable to time step  $n+1$ , that is to calculate,

$$\mathbf{h}_{n+1}^q = \begin{cases} D_{n+1}^{-1} \sum_{l=1}^L (d_{n+1}^l)^{-2} \mathbf{h}^l & \text{if all } d^l \neq 0 \\ \mathbf{h}^l & \text{if any } d^l \equiv 0 \end{cases} \quad (7.24a)$$

$$d_{n+1}^l = \frac{1}{2} \left\{ |\xi_{1n+1}^q - \xi_1^l|^2 (h_{1n}^q + h_1^l)^2 + |\xi_{2n+1}^q - \xi_2^l|^2 (h_{2n}^q + h_2^l)^2 \right\}^{\frac{1}{2}}, \quad (7.24b)$$

where  $\xi_{n+1}^q$  is calculated using equations (7.3),

$$\xi_{1n+1}^q = \xi_{1n}^q + \frac{\Delta x}{h_{1n}^q} \quad (7.24c)$$

$$\xi_{2n+1}^q = \xi_{2n}^q + \frac{\Delta y}{h_{2n}^q}. \quad (7.24d)$$

Here,  $\Delta x$  and  $\Delta y$  come directly from solving the momentum equations (discussed in subsequent sections).

In the vertical (which is only relevant to the free blobs) a linear interpolation is conducted considering only two points. Interpolation from the  $\mathcal{T}$  and  $\mathcal{W}$  grids are considered in turn.

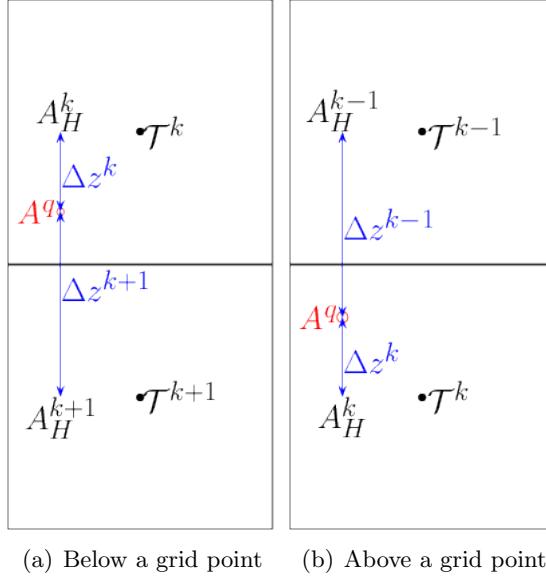


Figure 7.6: Dimensions for the vertical interpolation when a blob lies between two grid points.  $A^q$  is the value of the variable at the blob, while  $A_H$  is the value of the variable at either the  $k - 1$ ,  $k$  or  $k + 1$  level, interpolated to the horizontal position of the blob.  $\Delta z$  is the distance between the blob and the vertically adjacent grid points.

To decide which are the appropriate  $\mathcal{T}$  points to interpolate from, it first must be decided whether the blob is above or below the grid point of the cell that the blob resides in. If the blob is above the  $k$  grid point, the  $k - 1$  and  $k$   $\mathcal{T}$  points are used for the interpolation. If the blob is below the point, the  $k$  and  $k + 1$  points are used for the interpolation. This is illustrated in Figure 7.6. The equation to obtain the value of the  $\mathcal{T}$  grid variable,  $A$ , at the blob (after it has been interpolated in the horizontal to  $A_H^k$  and  $A_H^{k+dk}$ ) is given by,

$$A^q = \frac{A_H^k \Delta z^{k+dk} + A_H^{k+dk} \Delta z^k}{\text{dzwtT}^{k+dk}}, \quad (7.25a)$$

where,

$$dk = \begin{cases} +1 & \text{if } z^q \leq z_{\mathcal{T}}^k \\ -1 & \text{if } z^q > z_{\mathcal{T}}^k \end{cases}. \quad (7.25b)$$

Strictly speaking, since the horizontal interpolation has occurred along the coordinate surface and not along a geopotential surface, it would be more correct to also horizontally interpolate the values for  $\Delta z$  and  $\text{dzwtT}$  to the horizontal position of the blob for the  $k$  and  $k + dk$  surfaces. It is, however, assumed that the slope of the coordinate sur-

face is small and thus that the error introduced by not interpolating  $\Delta z$  and  $\text{dzwtT}$  is small.

There are two special circumstances where the situation depicted in Figure 7.6 does not occur, and that is if a blob resides between the surface and the top most grid point, or if a blob resides between the bottom and the lowest most wet grid point – both situations are depicted in Figure 7.7.

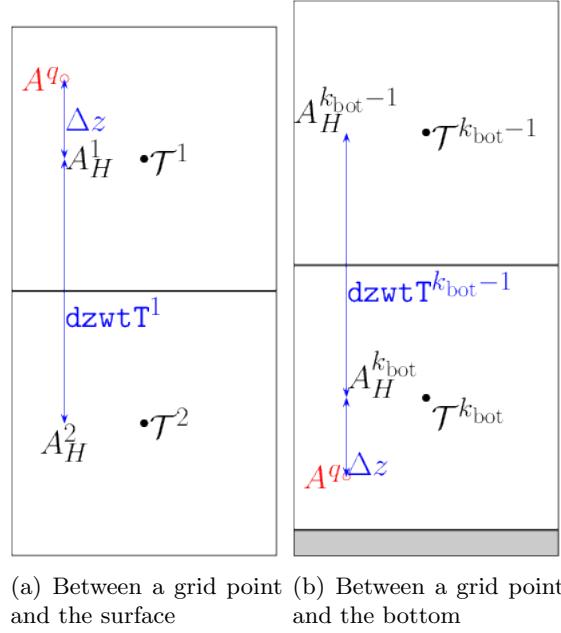


Figure 7.7: Dimensions for the vertical interpolation when a blob lies between a grid point and a boundary.

The equation for finding  $A^q$  at the surface and bottom then becomes

$$A^q = \begin{cases} A_H^{k_{bot}-1} + \frac{\text{dzwtT}^{k_{bot}-1} + \Delta z}{\text{dzwtT}^{k_{bot}-1}} (A_H^{k_{bot}} - A_H^{k_{bot}-1}) & \text{if } z^q < z_T^{k_{bot}} \\ A_H^2 + \frac{\text{dzwtT}^1 + \Delta z}{\text{dzwtT}^1} (A_H^1 - A_H^2) & \text{if } z^q > z_T^1 \end{cases}. \quad (7.26)$$

The  $\mathcal{W}$  grid has no special cases since the  $\mathcal{W}$  grid points lie at the top and bottom of every tracer cell, and, if a blob is within the ocean domain then it must lie between two  $\mathcal{W}$  points. The dimensions required for the interpolation on the  $\mathcal{W}$  grid are shown in Figure 7.8. The equation for finding the value at the blob is given by

$$A^q = \frac{A_H^k \Delta z^{k-1} + A_H^{k-1} \Delta z^k}{\text{dzwtT}^k}. \quad (7.27)$$

## 7.2 Blob List Sorting Algorithms

As discussed in Section 6.1.3 there are certain issues surrounding bitwise reproducibility across an arbitrary number of processors that must be taken into account in the La-

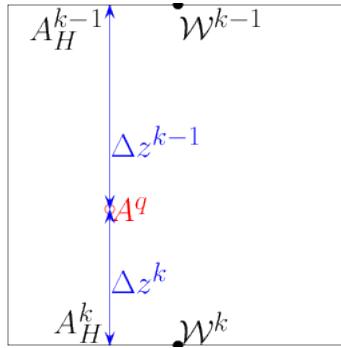


Figure 7.8: Dimensions for the vertical interpolation of  $\mathcal{W}$  grid variables.

grangian framework. These issues arise from the order in which calculations are carried out. Insisting on maintenance of bitwise agreement across an arbitrary number of processors requires that operations on linked lists are always done in the same relative sequence.

When parallelising the blobs the linked list in which the blobs are stored is decomposed across processors in analogy to the way that grids are decomposed across processors in the Eulerian system. Similar to grid operations in the Eulerian model, the order of operations on the linked lists must always occur in the same relative order to preserve bitwise reproducibility.

To achieve bitwise agreement when decomposing linked lists requires a method for sorting the list so that when cycling through the list, the blobs always appear in the same relative sequence. This sequencing is achieved via two sorting conditions. The first is a hash function based on the grid cell of origin for the blob, and the second is a counter, which allocates a unique number to each blob originating in a particular grid cell. The hash function is given by

$$\mathcal{H} = k + \mathbf{n}_k \cdot j + \mathbf{n}_k \cdot \mathbf{n}_j \cdot i, \quad (7.28)$$

where  $\mathcal{H}$  is the hash,  $\mathbf{n}_k$  is the total number of grid cells in the vertical and  $\mathbf{n}_j$  is the total number of grid cells in the nominal north-south direction. With this hash function, each grid cell can be identified uniquely with a single number.

The second condition for sorting is a counter which counts the number of blobs that are formed in a particular grid cell. Each blob stores its number and its hash. These two numbers allow for blobs to be uniquely identified from all other blobs in a given simulation.

This sorting method does not allow one to take output from one model setup with a particular grid and move it onto another model with a different grid. The technique is not, however, dependent on the grid decomposition and will remain robust across restarts with different computational domain decompositions.

### 7.3 Blob Trajectories

Unlike an Eulerian model, where the prognostic kinematic variable is velocity, the prognostic kinematic variable for a Lagrangian model is position. As such, the equations of motion generally take the form of coupled second order equations,

$$\ddot{\mathbf{x}} = \mathbf{G}(t, \mathbf{x}, \dot{\mathbf{x}}), \quad (7.29)$$

where we have used Newton's notation to indicate the material derivative of position with respect to time to help emphasise the importance of position. We write the position as a vector,  $\mathbf{x} = [x(t), y(t), z(t)]$  and the corresponding functions that describe the evolution of the position are also in vector form,  $\mathbf{G}$ . We note that the prognostic position distance variables are the physical distances,  $(x^q, y^q)$ , as opposed to the grid variables  $(\xi_1^q, \xi_2^q)$ . Also, the prognostic depth variable is the depth relative to the geoid,  $z = 0$ , not the depth relative to the sea surface,  $z = \eta$ . As a convention,  $(x^q, y^q)$  are set to zero at the beginning of each blob sub-cycle.

When deciding upon algorithms to solve these equations we must bear in mind that we wish to maximise accuracy while minimising cost. Furthermore, a prerequisite for an algorithm is that it must be stable. We must also take into consideration aspects that are unique to the particular situation we are considering, namely,

1. we do not have the freedom to choose an arbitrary step size. Ultimately, the step size must be chosen so that the Lagrangian system couples to the Eulerian system at each Eulerian model time step, and
2. in order to accurately represent the process of properties transferring between the Eulerian and Lagrangian systems, we must assume that there will be some instances where a blob may traverse more than one grid box in a given Eulerian time step. This has implications for the accurate representation of entrainment/detrainment, as well as for the momentum of a blob.

Before beginning a discussion on the considerations for algorithmic choice, we note that we may reduce the order of our Equation (7.29) by defining  $\boldsymbol{\chi}$  to be the exact solution of our differential equations, and writing it as a vector,

$$\boldsymbol{\chi}(t) = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}, \quad (7.30)$$

which then suggests that we have a system of equations of the form,

$$\dot{\boldsymbol{\chi}}(t) = \mathbf{G}(t, \boldsymbol{\chi}). \quad (7.31)$$

Such a system of equations are initial value problems, lending themselves to numerical integration. As such, we denote the approximate numerical solution as  $\mathbf{X} = \mathbf{X}(t)$ . The

form of the discrete equation may be written as,

$$\mathbf{X}_{n+1} = \mathbf{X}_n + h\mathcal{R}(t_n, \mathbf{X}_n, h; \mathbf{G}). \quad (7.32)$$

where  $n$  indicates the  $n^{\text{th}}$  time step,  $t_n$  is the time,  $h$  is the size of the time step ( $h = t_{n+1} - t_n$ ) and  $\mathcal{R}$  is the integrating function.

There are several types of error that result from the discretisation of continuous equations, as well as floating point errors associated with the use of computers. We will only concern ourselves here with errors associated with algorithmic selection, namely, the local truncation error and the global truncation error (which we will refer to as the truncation error and global error respectively). From Equation (7.32), we can see that it is desirable if our numerical integrating function,  $\mathcal{R}$ , is close to the value of the derivative of the exact solution,

$$\mathcal{R}(t, \boldsymbol{\chi}(t), h; \mathbf{G}) \approx \dot{\boldsymbol{\chi}}(t) = \mathbf{G}(t, \boldsymbol{\chi}(t)). \quad (7.33)$$

We now define the local truncation error as the difference between the exact solution at  $t_{n+1}$  and the approximate solution, given the exact solution at  $t_n$ ,

$$\mathbf{T}_n(\boldsymbol{\chi}) = \boldsymbol{\chi}(t_{n+1}) - [\boldsymbol{\chi}(t_n) + h\mathcal{R}(t_n, \boldsymbol{\chi}(t_n), h; \mathbf{G})]. \quad (7.34)$$

The global error, on the other hand, is defined as,

$$\mathbf{e}_n(\boldsymbol{\chi}) = \boldsymbol{\chi}(t_{n+1}) - [\mathbf{X}(t_n) + h\mathcal{R}(t_n, \mathbf{X}(t_n), d; \mathbf{G})], \quad (7.35)$$

which gives the cumulative effect of the local truncation error.

A common method for numerical integration is the Runge-Kutta (RK) method. There are many flavours of RK methods and a corresponding variety of accuracy and computational requirements. Ultimately, we wish to gain the greatest accuracy for the smallest computational cost. Firstly, we note that the general form for the generating function of an explicit RK method is (for instance Atkinson, 1989; Shampine, 1994),

$$\mathcal{R}(t, \mathbf{X}, h; \mathbf{G}) = \sum_{s=0}^p \gamma_s \mathcal{V}_s(\mathbf{X}) \quad (7.36a)$$

where  $\gamma_s$  is an arbitrary parameter. Here,

$$\mathcal{V}_0 = \mathbf{G}(t_n, \mathbf{X}_n) \quad (7.36b)$$

$$\mathcal{V}_s = \mathbf{G} \left( t_n + \delta_s h, \mathbf{X}_n + h \sum_{r=0}^{s-1} \beta_{sr} \mathcal{V}_r \right) \quad s = 1, \dots, p \quad (7.36c)$$

where  $\delta_s$  is a free parameter and  $\beta_s$  is a free parameter. The coefficients for an explicit

RK scheme are often represented in a Butcher tableau (see Section 304 of Butcher, 1987),

$$\begin{array}{c|ccccc}
 0 & & & & & \\
 \delta_1 & \beta_{10} & & & & \\
 \delta_2 & \beta_{20} & \beta_{21} & & & \\
 \vdots & \vdots & \vdots & \ddots & & \\
 \delta_p & \beta_{p0} & \beta_{p1} & \cdots & \beta_{pp-1} & \\
 \hline
 & \gamma_0 & \gamma_1 & \cdots & \gamma_{p-1} & \gamma_p
 \end{array} \tag{7.37}$$

The Butcher tableau may be generalised for implicit RK schemes as well, however, we shall only concern ourselves with the explicit representation as we are only considering explicit schemes.

When considering what method is best to use, one metric that may be used is the efficiency of the algorithm. A natural measure of efficiency in the RK family of algorithms is the stepsize divided by the number of function evaluations,

$$E = \frac{h}{p}, \tag{7.38}$$

where  $E$  is the efficiency. To begin with, it is desirable to have as high an order scheme for the fewest number of function evaluations,  $p$  while having a large stepsize,  $h$ . It can be shown (for instance, p.427 of Atkinson, 1989) that if a RK method has a truncation error  $\mathbf{T}_n(\chi) = O(h^{b+1})$ , then the rate of convergence of the method is  $O(h^b)$  and we refer to the scheme as an order  $b$  scheme.

### 7.3.1 Eulerian Grid Interaction and Step Size Considerations

If a blob traverses several grid cells in a single E system timestep, as is depicted in Figure 7.9, we wish the blob's effect on the E system (e.g. via detrainment) to be felt at the intermediate grid cells and not just at the end point cells. Furthermore, the size of the Eulerian time step is not necessarily going to be small enough to guarantee that the blobs will be well behaved. In other words the relative error,  $\mathbf{T}_n/\chi_n$ , can be greatly affected by the step size,  $h$ . We are thus motivated to find methods that allow us to sub-cycle blobs relative to the Eulerian system.

One choice would be to keep a fixed  $h$  such that

$$h \in \left\{ \Delta\tau, \frac{\Delta\tau}{2}, \frac{\Delta\tau}{3}, \frac{\Delta\tau}{4}, \dots \right\}. \tag{7.39}$$

Such a strategy is not desirable as it does not ensure that we are taking step sizes that are small enough, nor does it allow us to take larger step sizes if small step sizes are unnecessary.

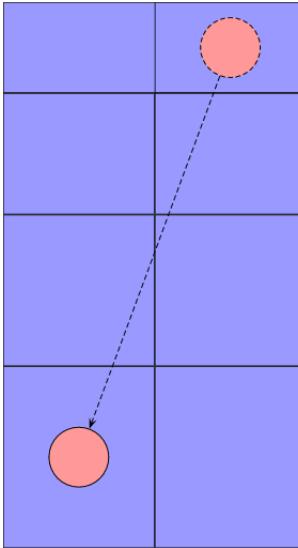


Figure 7.9: The dashed outlined circle is the blob at time  $\tau$  while the solid circle is the blob at time  $\tau + \Delta\tau$ . Since blobs may have relatively large vertical velocities, we must take into account that a blob may traverse several grid cells within a single Eulerian model time step due to the relatively thin grid cells commonly used in modern ocean climate models.

There are numerical integrators available that have a variable step size that adapts  $h$  to minimise the truncation error and maintain numerical stability. In the system that we are considering, we do not have the luxury of being able to arbitrarily vary  $h$  as the Eulerian system and Lagrangian system must exchange properties at the end of every Eulerian time step. The necessity of having the Eulerian and Lagrangian time steps coincide periodically implies that if we vary the Lagrangian time step, we must have

$$\sum_{n=1}^N h(n, \tau) = \Delta\tau \quad (7.40)$$

where  $\Delta\tau$  is the Eulerian time step,  $N = N(\tau)$  is the number of Lagrangian time steps between  $\tau$  and  $\tau + \Delta\tau$ . Note, that the Eulerian model's time step in MOM is fixed (and hence, no subscript).

### 7.3.2 A Series of Explicit Runge-Kutta Schemes with Error Control

Selection of a RK scheme entails selecting the coefficients,  $\gamma_s$ ,  $\delta_s$  and  $\beta_{sr}$  in equations (7.36) to maximise the efficiency,  $E$ , as well as to maximise the ratio  $b/p$ . Generally, a family of such values are derived and the family of solutions has particular properties. A great many number of possibilities confronts the practitioner, as is evidenced in Butcher (1987).

Suppose we evaluate our function using two different methods, one which is order  $b$  to obtain  $\hat{\mathbf{X}}_n$  and the other is order  $b + 1$  to obtain  $\mathbf{X}_n$ . An estimate of the leading local

truncation error,  $\mathcal{T}$ , may be given by,

$$\mathcal{T} = |\mathbf{X}_n - \hat{\mathbf{X}}_n| \quad (7.41a)$$

$$= \epsilon h_n^{b+1} + O(h^{b+2}), \quad (7.41b)$$

where  $\epsilon$  is a vector of proportionality parameters. Let us now define  $\zeta$  to be the relative local truncation error,

$$\zeta = \frac{\mathcal{T}}{|\mathbf{X}_n|}, \quad (7.42)$$

where we note that the division is element wise. If we now say that  $\zeta^*$  is the *desired maximum* local relative truncation error, we can see that, to leading order, the vector of time steps,  $\mathbf{h}^*$ , required to obtain the desired maximum truncation error,  $\mathcal{T}^*$ , is given by,

$$\mathcal{T}^* = \zeta^* |\mathbf{X}_n| \quad (7.43a)$$

$$= \epsilon (\mathbf{h}^*)^{b+1}. \quad (7.43b)$$

We may now use equations (7.41) and (7.43), to estimate the maximum allowable step size for the desired relative error. Equating the worst case scenario of each truncation error estimate,  $\mathcal{T}_{\max} = \max(\mathcal{T})$ , with the desired truncation error, we may gain an estimate for the timestep required,  $h^*$ , to obtain the required relative error,

$$h^* = h_n \left( \frac{\zeta^* |X_n|}{\mathcal{T}_{\max}} \right)^{1/(b+1)}. \quad (7.44)$$

where  $X_n$  corresponds to the variable represented by  $\mathcal{T}_{\max}$ . We may now use this estimate for the approximate maximum step size to accept or reject the solutions  $\mathbf{X}_n$ . If  $\zeta^* |X_n| > \mathcal{T}_{\max}$ , then we reject the solution and choose another value for  $h_n$ . Rejected steps are expensive, and there is still uncertainty in Equation (7.44). It is common to have an adjustable parameter,  $\varrho$ , specified *a priori* to avoid having a large number of rejected steps,

$$h^* \approx h_n \left( \frac{\varrho \zeta^* X_n}{\mathcal{T}_{\max}} \right)^{1/(b+1)}, \quad (7.45)$$

where  $0 < \varrho \leq 1$ . It is of course optimal to have  $\varrho$  as close to 1 as possible (as this allows a larger step size) while still minimising the number of rejected steps (see Section 7.3 of Shampine, 1994).

If the step is rejected (that is,  $h_n > h^*$ ), we wish go back and recalculate the values  $\mathbf{X}_n$  with the new step size,  $h_n = h^*$ . We note that there is a namelist variable for the minimum step size  $h_{\min}$ , such that in the event that the step size  $h^*$  is less than the minimum step, the new step is set to the minimum size,

$$h_n = \begin{cases} h^* & \text{if } h^* > h_{\min} \\ h_{\min} & \text{if } h^* \leq h_{\min} \end{cases}. \quad (7.46)$$

If the step is accepted, then we may make further use of the quantity  $h^*$  by conducting what is known as local extrapolation. Local extrapolation means that if  $h_n$  is accepted, we may then use  $h^*$  as our estimate for the next time step,  $h_{n+1}$  to find  $\hat{\mathbf{X}}_{n+1}$  and  $\mathbf{X}_{n+1}$ .

$$h_{n+1} = \begin{cases} \min(\Delta\tau, h^*) & \text{if } n = 0 \\ \min(\Delta\tau - \sum_{m=1}^n h_m - h^*, h^*) & \text{if } n > 0, \end{cases} \quad (7.47)$$

where here  $n$  is the number of steps since the last time that the discrete Eulerian model time,  $\tau$ , and the discrete blob time,  $t$ , coincided. If  $n = 0$ , then it is either the first step since the blob's creation, or the last step was the  $N^{\text{th}}$  step during the previous Eulerian model time step.

If it is the first step for a new blob, we set the step size to a step size specified in the namelist,

$$h_1 = h_{\text{first}} \quad \text{if } t = 0. \quad (7.48)$$

Notwithstanding the restrictions set out in Equation (7.47), there are some additional restrictions that we place on the local extrapolation in order to maintain stability and minimise the number of rejected steps. Namely, if the step  $n$  was ever rejected, we do not increase the step size, but maintain the same step size  $h_n$ . If the step was not rejected, we do not allow the next step to be any larger than twice the  $n^{\text{th}}$  step size,

$$h_{n+1} = \begin{cases} h_n & \text{if the } n^{\text{th}} \text{ step was rejected} \\ h^* & \text{if } h^* < 2h_n \text{ and the } n^{\text{th}} \text{ step was not rejected} \\ 2h_n & \text{if } h^* \geq 2h_n \text{ and the } n^{\text{th}} \text{ step was not rejected} \end{cases} \quad (7.49)$$

One open question that remains is how to efficiently compute  $\hat{\mathbf{X}}$  and  $\mathbf{X}$ . Intelligent selection of the parameters  $\gamma$ ,  $\delta$  and  $\beta$  – see Equation (7.36) – will give results of different order for the same function evaluations. Such formulae are known as embedded pairs. There are a great many number of embedded RK schemes (Section 6.10 of Atkinson, 1989). We shall follow the convention in the literature of appending the order of a method to the abbreviation RK, for example, a fourth order Runge-Kutta method shall be referred to as RK4. For an embedded RK method, we shall append the lower order embedded scheme in brackets, for example, an RK5 scheme with an embedded fourth order scheme shall be written as RK5(4).

Three schemes that have found much utility are the RK3(2) scheme of Bogacki and Shampine (1989) and the RK5(4) schemes of Dormand and Prince (1980) and Cash and Karp (1990). The former two are implemented in the MATLAB ODE solvers (Shampine and Reichelt, 1997), while the latter is the preferred method of Press et al. (1992). The Bogacki and Shampine (1989) and Cash and Karp (1990) schemes have been implemented in MOM. We present the Butcher tableaux – see Equation (7.37) – for the two implemented schemes. Note that we present the higher order  $\gamma$  coefficients followed by the lower order

coefficients. The Bogacki and Shampine (1989) is given by

$$\begin{array}{c|cccc}
 & 0 & & & \\
 \frac{1}{2} & & \frac{1}{2} & & \\
 \frac{3}{4} & 0 & \frac{3}{4} & & \\
 1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\
 \hline
 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 \\
 & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{8}
 \end{array} \tag{7.50a}$$

The Cash and Karp (1990) scheme is given by,

$$\begin{array}{c|ccccc}
 & 0 & & & & \\
 \frac{1}{5} & & \frac{1}{5} & & & \\
 \frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & \\
 \frac{3}{5} & \frac{3}{10} & -\frac{9}{10} & \frac{6}{5} & & \\
 1 & -\frac{11}{54} & \frac{5}{2} & -\frac{70}{27} & \frac{35}{27} & \\
 \hline
 \frac{7}{8} & \frac{1631}{55296} & \frac{175}{512} & \frac{575}{13824} & \frac{44275}{110592} & \frac{253}{4096} \\
 & \frac{37}{378} & 0 & \frac{250}{621} & \frac{125}{594} & 0 & \frac{512}{1771} \\
 & \frac{2825}{27648} & 0 & \frac{18575}{48384} & \frac{13525}{55296} & \frac{277}{14336} & \frac{1}{4}
 \end{array} \tag{7.50b}$$

The main advantage of using an embedded scheme for error control, is that the user gets two schemes for slightly more than the price of one. That is, the most expensive part of the computation (the evaluation of  $\mathcal{V}$ ) is the same for each method, and it is  $p$  extra multiplications in order to obtain the lower order method.

In the next two sections, we discuss the specific implementation for the two dynamically active regimes discussed in Sections 5.2 and 5.3.

### 7.3.3 The Open Ocean Implementation

We begin by restating the equations of motion for a free blob from Equations (5.6). As previously, Newton's notation is used for the material derivative of position,

$$\ddot{x} = -\alpha \dot{x} + f \dot{y} - f^* \dot{z} - \frac{\partial_x p}{\rho_L} + \alpha u_E \quad (7.51a)$$

$$\ddot{y} = -f \dot{x} - \alpha \dot{y} - \frac{\partial_y p}{\rho_L} + \alpha v_E \quad (7.51b)$$

$$\ddot{z} = f^* \dot{x} - \alpha \dot{z} - g \frac{\rho_L - \rho_E}{\rho_L} + \alpha w_E. \quad (7.51c)$$

We wish to represent these second order, coupled equations in matrix form as first order equations to simplify the task of solving them numerically. Writing the equations as,

$$\dot{\chi} = \mathbf{L}\chi + \mathbf{D}, \quad (7.52)$$

where  $\mathbf{L}$  is the linear operator,  $\mathbf{D}$  denotes the variables taken from the Eulerian grid, and  $\chi$  is the true solution. Written more explicitly in matrix form, we have,

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & 0 & f & 0 & -f^* \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -f & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & f^* & 0 & 0 & 0 & -\alpha \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{\partial_x p}{\rho_L} + \alpha u_E \\ 0 \\ -\frac{\partial_y p}{\rho_L} + \alpha v_E \\ 0 \\ -g \frac{\rho_L - \rho_E}{\rho_L} + \alpha w_E \end{bmatrix} \quad (7.53)$$

In the notation introduced in the preamble of Section 7.3,

$$\mathbf{G}(t, \chi) = \mathbf{L}\chi + \mathbf{D}. \quad (7.54)$$

The only constant parameters in time and space in  $\mathbf{G}$  is the coefficient of drag,  $\alpha$ , and gravity  $g$ . All other variables in  $\mathbf{L}$  and  $\mathbf{D}$  are potentially a function of position and/or time, and as such, may change from one sub-cycle to the next. We further note that  $\mathbf{v}_E$ ,  $\nabla p$  and  $\rho_E$  are values that we interpolate to a blob. We discuss how these values are interpolated in Section 7.1.8.

The remaining variables are  $f$ ,  $f^*$  and  $\rho_L$ . Both  $f$  and  $f^*$  are continuous functions of latitude only and may be easily evaluated.  $\rho_L$  on the other hand is a function of the pressure and the concentration of heat and salt of the blob.

### 7.3.4 The Price-Baringer Dynamic Streamtube Implementation

Restating the momentum Equations (5.18) for the Price-Baringer streamtube implementation,

$$\ddot{x} = -\dot{x}\frac{\mathcal{E} + C_d|\dot{\mathbf{x}}|}{h_L} + \dot{y}f + g\frac{\rho_L - \rho_E}{\rho_L\sqrt{H_{,x}^2 + H_{,y}^2 + 1}}\partial_x H \quad (7.55a)$$

$$\ddot{y} = -\dot{x}f - \dot{y}\frac{\mathcal{E} + C_d|\dot{\mathbf{x}}|}{h_L} + g\frac{\rho_L - \rho_E}{\rho_L\sqrt{H_{,x}^2 + H_{,y}^2 + 1}}\partial_y H \quad (7.55b)$$

$$\ddot{z} = -\dot{z}\frac{\mathcal{E} + C_d|\dot{\mathbf{x}}|}{h_L} + g\frac{\rho_L - \rho_E}{\rho_L}\left(\frac{1}{\sqrt{H_{,x}^2 + H_{,y}^2 + 1}} - 1\right). \quad (7.55c)$$

where we have from Equations (5.17),

$$\begin{aligned} \tau_{\text{bot}} &= \rho_L C_d |\dot{\mathbf{x}}| \dot{\mathbf{x}} \\ \mathcal{E} &= \begin{cases} |\dot{\mathbf{x}} - \mathbf{u}_E|^{\frac{0.08-0.1\text{Ri}}{1+5\text{Ri}}} & \text{if } \text{Ri} \leq 0.8 \\ 0 & \text{if } \text{Ri} > 0.8 \end{cases} \\ \text{Ri} &= g \frac{(\rho_L - \rho_E)h_L}{\rho_L |\dot{\mathbf{x}} - \mathbf{u}_E|^2}. \end{aligned}$$

These equations are in a form suitable for solving by the RK schemes discussed in Section 7.3.2. In matrix notation, equations (7.55) are written as,

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\mathcal{E} + C_d|\dot{\mathbf{x}}|}{h_L} & 0 & f & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -f & 0 & -\frac{\mathcal{E} + C_d|\dot{\mathbf{x}}|}{h_L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{\mathcal{E} + C_d|\dot{\mathbf{x}}|}{h_L} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ g\frac{\rho_L - \rho_E}{\rho_L\sqrt{H_{,x}^2 + H_{,y}^2 + 1}}\partial_x H \\ 0 \\ g\frac{\rho_L - \rho_E}{\rho_L\sqrt{H_{,x}^2 + H_{,y}^2 + 1}}\partial_y H \\ 0 \\ g\frac{\rho_L - \rho_E}{\rho_L}\left(\frac{1}{\sqrt{H_{,x}^2 + H_{,y}^2 + 1}} - 1\right) \end{bmatrix} \quad (7.56)$$

### 7.3.5 Reproducibility with an Adaptive Step Size

The introduction of sub-cycling complicates the requirement of strict bitwise reproduction (see Section 7.3.1). To restate what we are trying to achieve when it comes to bitwise reproduction: we need to process all calculations in the same relative order, regardless of the domain decomposition. This means that the interaction between the blob and the E system must be processed in the same relative order for each sub-cycle regardless of how the domain is decomposed.

There are a number of strategies that one could employ to ensure that the interaction at the end of each blob sub-cycle is processed in the same relative order. The most practical strategies involve storing information about a blob at each subcycle and processing

that information when all blobs have completed a full Eulerian time step.

One such strategy would be to create a linked list associated with each blob, with each node storing information about the blob (e.g. grid cell position, tracer tendency due to entrainment and detrainment, etc.) at each position at the end of each sub-cycle. An alternative would be to create an array to store such information and process the array sequentially.

The latter strategy of storing an array for each blob is chosen since the maximum number of sub-cycles is fixed (recall that the minimum blob step size is a namelist option). It is computationally less expensive to cycle through an array than through a linked list as there should never be a reason to resize the array.

When it comes to the transfer of a blob to a different compute domain, the blobs are transferred from one domain to another via buffers (as described in Section 6.1.3). The blobs are not transferred from one compute domain to another until all blob trajectories have been cycled through, and the blob tendencies on the Eulerian system are not calculated until after all blobs have been transferred. Herein lies a crucial assumption, we do not assume that a blob travels beyond the halo of the compute domain. This should not be a particularly impractical assumption, as the applications typical for the present method would have grid cells with a relatively large lateral extent.

When a blob is transferred from one domain to another at the end of the Eulerian time step, the blob's history is copied and sent to the new domain. Once the buffers are unpacked and the blob inserted in its appropriate place in the list, we process all the blob's histories in order. Note, that blobs that have been transferred from one domain to another will be processed twice, once on the beginning and ending domain. This does not provide a problem (aside from a slight inefficiency) as the history that does not pertain to the respective compute domain will only affect the halos of the relevant Eulerian fields (which will be overwritten at the next call to update the halos), and as such, will not adversely affect the reproducibility of the result.

So, to summarise,

1. a blob is time stepped, with fields that affect the Eulerian system at each sub-cycle being stored in an array associated with each blob,
2. if a blob is found to have gone off a compute domain, the blob and its history for the present Eulerian time step is transferred to a buffer and sent to the new domain (the blob on the original domain is given a zero mass and tracer value, and will be deleted after the history is processed),
3. incoming blobs and their histories are unpacked and inserted into the linked list,
4. the blobs and their histories are processed in the correct order,

5. blobs of zero mass (either detrained to zero mass, killed or having gone to another compute domain) are erased from memory.

The strategy described above can be very memory intensive if there are many blobs, and, the minimum blob step is a small fraction of the E system's time step. So, there is a namelist option that can be selected to turn off bitwise reproducibility. If `bitwise_reproduction` is false, then we do not save the history of each blob, but instead process them sequentially per compute domain. If a blob crosses from one compute domain to another, it is transferred to the new domain before the end of the E system time step, and its integration is resumed on that new domain. In other words, the order that the exchange of properties between the E system and the L system is processed depends on the domain decomposition and thus, bitwise reproducibility regardless of domain decomposition is not possible.

### 7.3.6 Limiting Velocity and Position

There are a number of instances in the present implementation which forces us to construct limits on the position and velocity of blobs.

When checking for vertical instability to create free blobs, we use the total system's square of the buoyancy frequency, Equation (6.37), rather than the E system's since it is the combined system that gives the stability of the water column. We can imagine a situation, however, where the combined system's square of the buoyancy frequency is less than a reference value,  $N_T^2 < N_0^2$ , and the E system in isolation is stable,  $N_E^2 > 0$ . Under these circumstance, we can see that the pseudo-non-hydrostatic term will give the blob an initially positive (upward) velocity, which defeats the purpose of creating a blob. Another situation where a positive velocity might occur is if there is a large positive vertical velocity that the pseudo-non-hydrostatic term is unable to overcome. So, if a blob has an initial vertical velocity which is positive,  $\dot{z}(t = 0) > 0$ , we do not form the blob. In this way, we ensure that all blobs formed have an initial velocity that is downward.

We are forced to limit the initial position and velocity of a free blob if the initial velocity is sufficiently large such that it would be placed lower than the target grid box. To illustrate, we restate equations (5.11),

$$\begin{aligned}\mathbf{x}^a(\tau + \Delta\tau) &= \left( x_{\mathcal{T}}^{ij} + \Delta\tau u^a, y_{\mathcal{T}}^{ij} + \Delta\tau v^a, z_{\mathcal{W}}^{ijk} + \Delta\tau w_{\mathcal{E}}^{ijk} - (\Delta\tau)^2 g \frac{\rho_L^a - \rho_E^{ijk+1}}{2\rho_L^a} \right) \\ \mathbf{v}^a(\tau + \Delta\tau) &= \left( u_{\mathcal{E}}^{ijk}, v_{\mathcal{E}}^{ijk}, w_{\mathcal{E}}^{ijk} - \Delta\tau g \frac{\rho_L^a - \rho_E^{ijk+1}}{\rho_L^a} \right),\end{aligned}$$

which is the initial position and velocity of a free blob that has been formed during the implicit part of the time step. We recall that as the blobs are formed implicitly in time (and as such, cell thickness has already been calculated), grid cells must not change mass

during this process. We are then forced to ask what happens if

$$\Delta\tau w_{\text{E}}^{ijk} - (\Delta\tau)^2 g \frac{\rho_{\text{L}}^a - \rho_{\text{E}}^{ijk+1}}{2\rho_{\text{L}}^a} > dz^{ijk+1},$$

that is, that the initial position of a blob places it below  $z_{\mathcal{W}}^{ijk+1}$ , causing a net decrease in mass of the cell  $(i, j, k+1)$ , and a net increase in mass of the cell that the blob is targeting. We are thus forced to limit the initial position of the blob to ensure that its initial position is in the cell  $(i, j, k+1)$ , just above the bottom cell boundary. To remain consistent with the vertical point location scheme, Section 6.5.1, at full Eulerian time steps we also use the native vertical coordinate to test and limit the blob's initial velocity and position

$$\mathbf{x}(\tau + \Delta\tau) = \left( x_{\mathcal{T}}^{ij} + \Delta\tau u_{\text{E}}^{ijk}, y_{\mathcal{T}}^{ij} + \Delta\tau v_{\text{E}}^{ijk}, z_{,s}^{ijk} [s_{\mathcal{W}}^{ijk} - ds^{ijk+1} + \epsilon] \right) \quad (7.57\text{a})$$

$$\mathbf{v}(\tau + \Delta\tau) = \left( u_{\text{E}}^{ijk}, v_{\text{E}}^{ijk}, \frac{z_{,s}^{ijk} [-ds^{ijk+1} + \epsilon]}{\Delta\tau} \right), \quad (7.57\text{b})$$

for depth based coordinates, and

$$\mathbf{x}(\tau + \Delta\tau) = \left( x_{\mathcal{T}}^{ij} + \Delta\tau u_{\text{E}}^{ijk}, y_{\mathcal{T}}^{ij} + \Delta\tau v_{\text{E}}^{ijk}, z_{,s}^{ijk} [s_{\mathcal{W}}^{ijk} + ds^{ijk+1} - \epsilon] \right) \quad (7.57\text{c})$$

$$\mathbf{v}(\tau + \Delta\tau) = \left( u_{\text{E}}^{ijk}, v_{\text{E}}^{ijk}, \frac{z_{,s}^{ijk} [ds^{ijk+1} - \epsilon]}{\Delta\tau} \right), \quad (7.57\text{d})$$

for pressure based coordinates. Here,  $\epsilon$  is a small parameter (e.g.  $10^{-3}\text{m}$ ).

### 7.3.7 Namelist Variables

Scattered throughout the preceding notes are a number of variables that can affect various aspects of the Lagrangian model. The aim of the present section is to summarise those parameters that the user has direct control over via the namelist.

Firstly, the main controlling namelist variable is `use_blobs` which is found in the ocean model namelist. Other namelist variables in `ocean_model_nml` which can affect performance is the `layout` variable. The layout refers to how the compute domain is subdivided, and thus, can affect load balancing. For instance, if we take the box sector test case, which is a flat bottomed box in the northern hemisphere (see Chapter 31 of Griffies, 2009) and run with blobs in a two processor configuration, we will get different performance results whether the processors are arranged in a north-south sense or an east-west sense. The case where the processors are next to each other in an east-west sense will obtain the better performance because the load will be better balanced. The majority of blobs are formed in the northern part of the domain, and thus, there is a greater computational load on the northern part of the domain.

In the `ocean_blob` module, there are a number of namelist variables that are asso-

ciated with debugging and reproducibility, which, when turned off, will improve performance. `debug_this_module` and `really_debug` are variables which when set to true, will write certain helpful debugging information to standard out. `really_debug` in particular writes a lot of information and produces very large standard out files very quickly. `bitwise_reproduction` ensures that the code will produce exactly the same answer regardless of the domain decomposition. It should be noted that in order to obtain bitwise reproduction, the machine should be capable of doing so and that certain compiler flags are also necessary. `do_bitwise_exact_sum` raises a flag to ensure that diagnostic global sums will be the same, regardless of the number of processors used. Setting all four of these variables to false will improve the speed and memory footprint of the model.

A parameter which is set in the `ocean_blob` namelist and used by passively active free and bottom blobs is the threshold at which the mass of a blob is considered “small”, and therefore destroyed. The parameter has units of kilograms. It is set by `blob_small_mass`, and any properties that are left in the blob, once it has been determined that it is small enough to be destroyed, are returned to the Eulerian system. A consequence is that the system will remain conservative even if a relatively large value for `blob_small_mass` is chosen.

Caution is required when choosing a small value of `blob_small_mass` where there is likely to be very low tracer concentrations. Blobs that have decreased in mass by many orders of magnitude, and are carrying very low tracer concentrations may start to suffer from catastrophic cancellation (e.g. Chapter 2.4 of Oliveira and Stewart, 2006) of the tracer concentration when the mass of the blob is approaching zero. Empirically, a value of  $10^3\text{kg}$  has been found to avoid problems with catastrophic cancellation, however, this value likely depends on the specifics of a particular model configuration.

Another parameter that is set in `ocean_blob` and indirectly affects the number of blobs is `max_prop_thickness`. It is a non-dimensional variable that sets the maximum proportion of the upper and lower portion of a grid cell that the Lagrangian system may occupy. `max_prop_thickness` must have a value of between 0 and 1. The default value is 0.7.

In both the free and bottom dynamically active modules, there are a number of namelist variables that are similar, or even identical, but, applied separately to each scheme. One is `rel_error`, which is the relative error,  $\zeta^*$ , as discussed in Section 7.3.2. Setting the relative error to be very small will significantly increase computational cost for very little gain in fidelity. On the other hand, a large value may compromise the accuracy of the simulation and worse, may lead to unstable behaviour. The relative error may be considered as a percentage error. A value in the region of 0.1 to 0.2 has been found useful thus far.

Two quantities that are related to the relative error is the minimum step,  $h_{\min}$ , and the safety factor,  $\varrho$ , both of which are discussed in Section 7.3.2. The minimum step is

set with the namelist variable `minstep`, while the safety factor is set with the variable `safety_factor`. With both variables, the smaller the value, the more computational time is required. `safety_factor` should be somewhere between zero and one, with a value in the region of 0.8 to 0.95 being common for RK solvers. The value for `minstep` is less straightforward to estimate. A value of 1 for the free dynamically active blobs and a value of 9 for the bottom dynamically active blobs has been found to be useful. Although, this is likely dependent on the exact model configuration.

The size of the first step size for a blob is also set via the respective namelists for the dynamically active free and bottom blobs via the variable `first_step`. Setting a large first step means that the first step is likely to be rejected. Setting a small first step means that it may take some time for the step size to increase to a size that will meet the relative error requirements (recall that we do not allow the step size to increase by more than a factor of two). So, the ideal size is some sort of intermediate value, where the step size is large enough to increase significantly within a few time steps, but, is still small enough such that the step is unlikely to be rejected. A default value of 50 for each scheme is chosen.

Another parameter that is common to both bottom and free dynamically active blobs, but is controlled independently, is the detrainment parameter discussed in Section 5.2,  $\Gamma$ , that has units  $\text{kg m}^2 \text{s}^{-1}$ . The namelist variable is `det_param`. A larger value means that blobs will have a shorter life time (and will therefore mean there are less blobs).

There is an error checking namelist variable that is common to both schemes. `large_speed` is the speed that the user considers to be unrealistically large, and thus, indicates some sort of instability in the blob scheme. If the velocity of a blob exceeds `large_speed` in any direction, a warning is raised. The default value is  $10\text{m s}^{-1}$ .

The user can choose the RK scheme for each of the dynamically active modules independently. `update_method` can take the value of either `BS_RK3(2)` for the Bogacki and Shampine (1989) method or `CK_RK5(4)` for the Cash and Karp (1990) method.

The detrainment parameter,  $\mathcal{D}$ , and the maximum detrainment value (see Sections 5.2.2 and 5.3) are set independently via the variables `det_param` and `max_detrainment`.

There are also a number of namelist parameters that are specific to each of the schemes. Starting with the free dynamically active scheme, we have:

- `rayleigh_drag`, which is the Rayleigh drag coefficient,  $\alpha$ , as described in Section 7.3.3. In this implementation, it is a constant value, however, infrastructure exists to allow its value to vary spatially.
- `size_fact` is the non-dimensional scaling parameter,  $\Lambda$ , discussed in Section 5.2, which scales the initial mass of a free dynamically active blob. It should have a value between zero and one.

- `bv_freq_threshold` is the stability threshold at which a blob will be formed and has units of  $s^{-2}$ . That is, if the square of the buoyancy frequency is less than `bv_freq_threshold`, then a blob will be formed. In general, the value should be less than or equal to zero. Limiting the creation of blobs to only areas that are more than marginally unstable limits the number of blobs that are created, and also only forms blobs in areas with strong convection, while other convection schemes can be left to remove small instabilities (see below).

With specific reference to the last point above, we include `diff_cbt_limit` as part of the discussion on namelist variables affecting the free dynamic blobs. `diff_cbt_limit` is the maximum vertical diffusivity of regions that are statically unstable and is set in `cean_vert_const` namelist. As described in Section 9.3 of Griffies et al. (2004), setting `diff_cbt_limit` equal to `kappa_h` eliminates the removal of instabilities by enhanced vertical diffusion. Using `diff_cbt_limit` in combination with `bv_freq_threshold` gives the practitioner a useful means to control how much instability is removed through enhanced vertical diffusion (a local process), and how unstable the water column needs to be before non-local processes come into play (the formation and evolution of dynamically active blobs).

For the dynamically active bottom blobs there are also a number of namelist options that affect their performance and behaviour,

- `blob_overflow_mu`, `blob_overflow_delta` and `blob_overflow_umax` are all related to the formation of a new blob and set the values for frictional dissipation,  $\mu$ , the fraction of a grid cell participating,  $\delta$  and the maximum velocity of the overflow  $\mathbf{u}^{slope}$  respectively (see Sections 4.4 and 5.3.1).
- `drag` is the coefficient of drag,  $C_d$ , (see Section 5.3).
- `elastic` is the elasticity,  $\varepsilon$ , of the interaction of a free blob with the bottom (see Section 5.4). The default value is 1.
- `min_do_levels` is the minimum number of grid cells deeper that a deep ocean column must be than (compared to the adjacent shallow ocean column) for it to be considered possible to have an overflow event taking place. The default value is 1.
- `rho_threshold` is the minimum density difference between the shallow ocean grid cell and the deep ocean grid cell before an overflow event can occur.
- `accept_free_blobs` is a flag that if false, destroys free blobs that interact with topography, rather than converting them to bottom blobs.
- `no_rotation` sets the Coriolis parameter,  $f$ , to zero.
- `critical_richardson` is the critical Richardson number used to calculate entrainment (see Section 5.3).

## 7.4 Summary

We began this chapter by examining how to track the trajectory of a blob on an arbitrary orthogonal grid, with special reference to a sphere. Using a technique from computational geometry that involves cross products, we then examined how to conduct point locations to figure out whether a blob has changed horizontal grid cells. We then expanded this discussion to include cyclic grids and the Murray (1996) tripolar grid.

In the vertical, we use a brute force approach to find which grid cell a blob resides in at full Eulerian time steps. Due to the order of operations, we need to conduct the search at Eulerian time steps in the native coordinate system of the Eulerian model. At blob sub-cycles, we can do a more simple and efficient search based on the depth of coordinate surfaces.

The blobs require information from the Eulerian grid in order to calculate their trajectories. We horizontally interpolate the salient variables to a blob using an inverse distance weighting method. The methodology for velocity grid and tracer grid variables is the same, however, the specifics are slightly different and the interpolation coefficients for the two grids needs to be calculated separately. Vertical interpolation is only relevant for dynamically active free blobs. In the vertical, we use a linear interpolation from the two nearest grid points.

In order to maintain bitwise reproduction for arbitrary domain decomposition, we have two numbers that in combination are able to uniquely identify a blob. One is a hash function, which is based on the grid cell of origin and the other is a counter that is incremented on a per grid cell basis every time a new blob is formed.

Blob trajectories are found by numerically integrating the momentum equation of a blob using a Runge-Kutta scheme. We use a Runge-Kutta-Fehlberg method that allows us to control the step size based on an estimate of the local truncation error. Use of an adaptive step size allows us to make the time step as long as possible without compromising the integrity of the solution for a small additional cost per time step.

We also discuss reasons and methods for limiting the initial velocity and position of dynamically active blobs. Finally, we summarise the various namelist options for the two dynamically active blob schemes, and how those namelist variables can affect various aspects of the performance and fidelity of a simulation using the blobs.

# Chapter 8

## Miscellanea

### 8.1 Overall Algorithmic Structure and Flow

This section complements the discussion on code organisation in Section 2.2 of Griffies et al. (2004), as well as the discussion on the order of operations in Section 6.1.2. The order of operations for the model is listed below, where we have ignored operations that do not involve the embedded Lagrangian system. We have also ignored initialisation and ending procedures, which the embedded Lagrangian scheme also impacts upon.

Time step the ocean (`update_ocean_model`).

- + Explicit blob updates (`ocean_blob_update`).
  - ++ Do the blob overflow scheme – see Section 4.4 (`blob_overflow_like`).
  - ++ Increment the dynamically active blobs (`blob_dynamic_free_update`) and (`blob_dynamic_bottom_update`).

We calculate the vertically integrated blob divergence. We can do so because the horizontal position of blobs relative to the grid cell walls is known at time  $\tau + 1/2$ . The cell thicknesses at  $\tau + \frac{1}{2}$  are unknown and we are thus unable to complete the calculation of the blob divergence of individual cells. We also add entrained and detrained properties as a tendency term.
  - ++ Blobs that interact with topography or separate from the bottom are transferred from one dynamic regime to another.
  - ++ New dynamically active bottom blobs are formed.
- + Calculate the sea surface/bottom pressure tendency, including the contribution from blobs which requires use of the column integrated blob divergence (`eta_and_pbot_tend_blob`).
- + Update the sea surface height or bottom pressure (`eta_and_pbot_update`).
- + Find the number of blobs in each grid cell at the new time,  $Q_{\tau+1/2}^{ijk}$ , which allows us to calculate the blob divergence. Knowledge of the sea surface height (bottom pressure) allows us to find the blob's grid cell at time  $\tau + \frac{1}{2}$  in the native coordinate for depth- (pressure-)based models even though we still do not know what the updated cell thickness/depth is (`ocean_blob_cell_update`).
- + Calculate the density weighted thickness tendency of grid cells (`rho_dzt_tendency`).
- + Diagnose the vertical velocity, including the contribution from blob convergence (`ocean_advection_vertical`).
- + Calculate the total thickness of tracer cells (`update_tcell_thick_blob`).

- + Calculate the Lagrangian system contribution to total thickness. If the L system thickness exceeds the total thickness, then enough blobs are “evaporated” to ensure that the L system thickness is less than the total system thickness (see Section 6.5.3). We are only able to carry out this thickness check at this point because until the call to `update_tcell_thick_blob`, we did not know what the physical thickness of the cell is (`update_L_thickness`).
- + Diagnose the Eulerian system contribution to total thickness (`update_E_thickness`).
- + Update the system tracer concentration (`update_ocean_tracer`).
  - ++ The tracers are updated explicitly in time, which includes advective tendencies and blob tendencies (entrainment, detrainment, new blobs created explicitly in time and destroyed blobs).
  - ++ Blob implicit physics are applied (`ocean_blob_implicit`).
    - +++ The NCON-like scheme is applied (`blob_static_free`).
    - +++ New blobs are formed from vertical instabilities (`blob_dynamic_free_implicit`).
    - +++ The L system thickness is adjusted to take into account the new blobs (`blob_thickness`).
    - +++ The E system thickness is diagnosed to reflect the new L system thickness (`update_E_thickness`).
  - ++ Update the combined tracer concentration for a grid cell at time  $\tau + \frac{1}{2}$ .
- + Diagnose the sea surface height or the bottom pressure, as well as the geodepth (the depth relative to  $z = 0$ ) of tracer grid points and tracer grid cell bottoms (`eta_and_pbot_diagnose`)
- + Diagnose the geodepth of new blobs (that were formed during the implicit part of the tracer update), and the depth (relative to the sea surface height) of already existing blobs (`ocean_blob_diagnose_depth`).
- + Update the ocean density (`update_ocean_density`).
  - ++ Update the pressure at tracer cell points – including the contribution from the L system (`pressure_in_dbars`).
  - ++ Update the combined density for a grid cell at time  $\tau + \frac{1}{2}$ .
- + Calculate the explicit acceleration for the E system (`ocean_explicit_accel_a`).
  - ++ Calculate the contribution to acceleration of pressure – including the contribution from the L system (`pressure_force`).
- + Update the velocity cell thickness for the T, L and E systems (`update_uceil_thickess`).
- + Add the E system source term for the exchange of properties between the E and L system (`energy_analysis`)
- + Conduct diagnostics – see Section 6.1.4 (`ocean_diagnostics`)

We note that in order to calculate the blob divergence, there has been a necessary change in the order of operations over the 2009 public release of MOM. Specifically, the embedded Lagrangian scheme requires that we update the sea surface height (bottom pressure) directly after calculating the sea surface height (bottom pressure) tendency. The reason for this requirement is so that the blobs position may be diagnosed in the native vertical coordinate, and that we may complete the calculation of the blob divergence with knowledge of  $Q_{\tau+1/2}^{ijk}$  which is used to calculate the vertical velocity (in conjunction with the tendency of the density weighted thickness tendency and the E system divergence). The alteration of the order of operations has trivial effects to other aspects of the code.

## 8.2 Known Limitations

There are some known bugs and known limitations of the present implementation in MOM. The origins of the bugs remain unknown at this time.

### 8.2.1 GOTM

One of those bugs is an interaction with the General Ocean Turbulence Model (Umlauf et al., 2005). Some experiments using the blobs have had the GOTM module in MOM to cause a runtime error.

### 8.2.2 Pressure Based coordinates

A much more subtle bug has to do with the interaction between the blobs and the pressure-based vertical coordinates. When using pressure or  $p^*$  coordinates, the simulations are not reproducible across restarts. The source of the problem originates in the thickness module, but, is only manifest when the blobs module is in use.

### 8.2.3 Momentum Transfer Between the E and L systems

As discussed in Section 6.2.4, the transfer of momentum between the E and L systems is not treated completely in the implementation in MOM. It is unclear whether a complete implementation of the momentum transfer would have a non-trivial impact on the solution or not.

### 8.2.4 The Evolution of Non-Conservative Tracers

Since the E and L systems are distinct models with rather different infrastructure, the biogeochemical models used for a purely Eulerian model are unable to operate on the blobs. Thus, there needs to be new methods developed to evolve non-conservative tracers (such as age tracers and biogeochemical tracers) for the Lagrangian model.

### 8.2.5 The Grid Cell Mass Constraint with Dynamically Active Bottom Blobs

Bottom blobs are only said to exist in the bottom cell (in fact as implemented, it is the bottom half of the bottom cell). This is a potential problem for large blobs entering thin partial cells, as this makes it relatively easy for the mass constraint of the grid cell to be violated (see Section 6.5.3) thus causing the blob to be destroyed. However, whether it is necessarily desirable to be able to have very large blobs is not clear, as is discussed in Section 12.4.

### 8.2.6 The Height and Width of Dynamically Active Bottom Blobs

In the DOME test case (Chapter 12), it was found that there is a link between the mass of a blob and the drag term – see Equation (12.13) in Section 12.4.5. The link arises due to the formulation for the height of a blob, in which a blob is naively considered to be a sphere. Thus, the height and the mass are intimately related. The height is then used in the calculation of the bottom drag, which then means that the bottom drag term and the blob mass are related. In reality the height of a gravity driven current is much less than its width. Furthermore the cross-sectional shape of a viscous gravity driven current with bottom friction will change with time. As such, an alternative formulation for evolution of the width and height of a blob should be sought.

### 8.2.7 Code Optimisation

Knuth (1974) famously wrote “premature optimization is the root of all evil.” The blob code has been written with this philosophy in mind. While all reasonable attempts were made to not write inefficient code, there are undoubtedly many opportunities to further optimise the code to make it more efficient and it is likely that attempts to do so will significantly improve the speed of the code.

## **Part II**

# **Proof of Concept**



# Chapter 9

## Introduction

Up to this point the theory and implementation of the embedded Lagrangian model has been discussed and examined in quite some detail. The rest of this document is devoted to examining the impact of the embedding of a Lagrangian model in selected test cases that elucidate the physical properties of the systems that the blobs are supposed to model, and to examine the impact of the blob schemes on those systems, as well as the sensitivity of the scheme to certain parameter choices.

The rest of this chapter is devoted to examining previous process studies of the physical systems that the current implementation of the blobs aims to model. Subsequent chapters then examine a number of different model configurations and assess the fidelity of the model solutions and examine the computational burden required to attain those solutions. In doing so, traditional parameterisations are also examined and compared, as well as the parameter space of the blobs, and the sensitivity of the solution and computational burden to those parameters.

### 9.1 Previous Studies

The two main physical processes that the present implementation of the Lagrangian blobs aims to model are that of overflows and open ocean deep convection. In this section, a review of previous process studies of those processes is presented.

#### 9.1.1 Deep Convection

Jones and Marshall (1993) use a fine scale flat bottomed, doubly periodic domain that is initially weakly stratified and has a circular cooling disc of  $800\text{W m}^{-2}$  at the surface (see Chapter 11) to examine the mechanisms of open ocean deep convection. They are able to do so because their model was able to resolve convective plumes for an entire convective patch using a non-hydrostatic model. They conducted experiments varying the natural Rossby number,

$$\text{Ro} = \frac{B_0^{1/2}}{f^{3/2}H}, \quad (9.1)$$

by two orders of magnitude. Here,  $B_0$  is the surface buoyancy forcing ( $\text{m}^2\text{s}^{-3}$ ),  $f$  is the Coriolis parameter and  $H$  is the depth of the ocean. They found that the scaling arguments of previous laboratory experiments were in good agreement with their numerical simulations and corroborated existing scaling laws.

Send and Marshall (1995) used the model of Jones and Marshall (1993) to examine the integral effects of a population of plumes. They found that the net effect of a convective event is to mix properties through the water column. Their paper largely motivated Klinger et al. (1996) to advocate for an enhanced diffusion coefficient to parameterise convection in hydrostatic models in order to capture the effect of open ocean deep convection.

Campin et al. (2011) also used a configuration based on Jones and Marshall (1993) to test their super-parameterisation. The super-parameterisation is a 2d-nonhydrostatic slice embedded in a hydrostatic model. They ran three cases, one that is hydrostatic, one that is hydrostatic with the super-parameterisation and one that is fully non-hydrostatic. They find that the super-parameterisation improves the representation of the convective patch, with the greatest improvement being that the variance of the tracer distribution of the hydrostatic run with the super-parameterisation having a tracer variance much more similar to the nonhydrostatic run than the hydrostatic run without the super-parameterisation.

### 9.1.2 Overflows

There have been a number of idealised studies to assess the fidelity of overflows and gravity currents in ocean climate models. These have largely focussed on z-coordinate models, as they are the most widespread class of vertical coordinates in use for global climate simulations. Z-coordinate models also fare relatively poorly in terms of their representation of downslope flows.

One of the first of this class of experiments was that of Winton et al. (1998) which compared the results of a z-coordinate model with that of an isopycnal model in a relaxation experiment using a “bowl” topography (see Chapter 13). Deficiencies in the representation of overflows in level coordinate models found in this and other studies has motivated much research into the accurate representation of overflows in ocean climate models and motivated various collaborations such as the Dynamics of Overflow, Mixing and Entrainment (DOME) and the Gravity Current Climate Process Team (Legg et al., 2009).

Ezer and Mellor (2004) conducted experiments using the DOME configuration (see Section 12) and other configurations to examine the properties of a hybrid  $\sigma$ -z model, with particular reference to overflows. They showed that because of the fundamentally different representation of plumes in the different coordinate systems, changing parameters (such as the lateral diffusion) can have very different effects on the model’s representation

of a plume.

Legg et al. (2006) conducted a series of experiments at varying resolution in the DOME configuration with the MITgcm and the Hallberg Isopycnal Model. They examined the effects of changing resolution, slope, stratification, rotation and buoyancy anomalies. Their focus was predominantly on entrainment and diapycnal mixing in different vertical coordinate systems (isopycnal and geopotential) at varying resolutions. They found that in level models, coarse resolution models tend to have a lot of spurious mixing, however, in the range between coarse resolution and very fine resolution, the models may not have enough mixing.

Tseng and Dietrich (2006) also used the DOME configuration and found that the results of a the purely z-coordinate DieCAST model (Dietrich, 1997) are robust once the Rossby radius of deformation and the thickness of the plume become resolved. They found their results to be more robust than the results reported by Ezer and Mellor (2004) and they conjectured that it may be due to better numerics.

Özgökmen and Fischer (2008) used a very fine resolution model to look at the role of bottom roughness on gravity currents. Their estimates for the skin drag coefficient are consistent with the value of  $3 \times 10^{-3}$  used by Price and Baringer (1994) and other observational estimates (for example, Girton and Sanford, 2003). Their results indicated, however, that form drag can be up to an order of magnitude larger than the bottom drag (or, skin drag) for rough topography in their model configuration.

Using a fine resolution isopycnal model with idealised topography, Ilicak et al. (2011b) performed a number of experiments to investigate the effect of canyons and ridges on overflow properties. They found that corrugations steer the flow downslope and that ridges are more effective at facilitating downslope transport than canyons. They found a relation between downslope transport and the corrugation Burger number. They then suggested a mixing parameterisation based on the corrugation Burger number.

## 9.2 Preliminaries

In the idealised tests considered in this thesis density is a linear function of temperature

$$\rho = \rho_0 - \alpha T, \quad (9.2)$$

where  $\rho$  is the density,  $\rho_0 = 1035 \text{ kg m}^{-3}$  is a reference density,  $\alpha = 0.255 \text{ kg m}^{-3} \text{ C}^{-1}$  is the thermal coefficient of expansion and  $T$  is temperature. Buoyancy is defined as (for instance, section 2.4 of Vallis, 2006),

$$b = -g \frac{\delta \rho}{\rho_0} \quad (9.3)$$

where  $\delta\rho$  is the density deviation relative to a reference density,  $\rho_0$  and  $g = 9.80\text{m s}^{-2}$  is the gravitational acceleration. Combining Equations (9.2) and (9.3), we can see that,

$$b = \frac{g\alpha\Theta}{\rho_0}. \quad (9.4)$$

We also introduce the buoyancy frequency,  $N$ ,

$$N^2 = \frac{\partial b}{\partial z}. \quad (9.5)$$

The initial conditions for all stratified experiments (i.e. the deep convection test case and the DOME test case) are that,

$$b(x, y, z) = b_0 - N^2 z, \quad (9.6)$$

where  $b_0$  is the initial buoyancy at the surface.

The value for the heat capacity in all experiments is taken from Jackett et al. (2006) of  $C_p^0 = 3992.1\text{J C}^{-1}\text{kg}^{-1}$ . There is no surface stress in any of the experiments.

The quasi-barotropic streamfunction can be calculated as

$$\psi(x, y, t) = - \int_{y_0}^y U^\rho(x, y', t) dy', \quad (9.7a)$$

however, alternative formulations do exist

$$\psi^*(x, y, t) = \int_{x_0}^x V^\rho(x', y, t) dx', \quad (9.7b)$$

where  $\mathbf{U}^\rho = (U^\rho, V^\rho)$  is the depth integrated density weighted horizontal velocity and is defined in Equation (6.54). Since there is no rigid lid approximation employed and the depth integrated density weighted velocity can be divergent,  $\partial_x \psi$  does not, in general, yield the zonal transport and neither does  $\partial_y \psi^*$  necessarily yield the meridional transport. Thus, use of the barotropic quasi-streamfunction must be done with care. In the context of blobs, the barotropic quasi-streamfunction is only able to give information about the E system transport, not about the combined E and L system transport.

# Chapter 10

## Dynamically Passive Blobs

The objective of this chapter is to demonstrate that the NCon-like scheme of Section 4.1 and the blob overflow scheme of Section 4.4 successfully replicate their purely Eulerian analogues. This is achieved by comparing the horizontally averaged temperature for the deep convection test case described in Chapter 11 and the DOME test case described in Chapter 12.

### 10.1 The NCon Scheme

For the deep convection test case of Jones and Marshall (1993), described in Chapter 11, two experiments were run with the only difference between them being that one used the traditional NCon scheme for convective adjustment and the other used the blob NCon scheme for convection adjustment. Both experiments were run for four days. The magnitude of the difference between the two experiments is very small, with the largest difference being approximately  $6.4 \times 10^{-14}\text{C}$ . If we were using computers with infinite precision, the difference between the two schemes should be zero, however, as we are using finite precision computers the difference is non-zero, but is trivially small.

The run with the NCon scheme had a run length per core of 699.1s, while the blob NCon scheme had an average run length per core of 922.7s. This represents approximately a 30% increase in the cost of the experiment. Of the 223.6s difference between the two experiments, only 27.4s of that difference comes directly from the difference between the NCon and the blob NCon schemes. The rest of the difference arises due to the other additional infrastructure associated with the blobs model.

### 10.2 The Overflow Scheme

Two experiments are conducted using the DOME test case (described in Chapter 12) to compare the Campin and Goosse (1999) overflow scheme and the blob overflow scheme, described in Section 4.4. The experiments were run for one year and were identical except for the overflow parameterisation used. The difference between the two experiments is

very small, with the largest difference between the two experiments being  $3.0 \times 10^{-10}\text{C}$ . Such a small difference is what would expect as the schemes are equivalent.

The average time taken per core for the run is 2466.7, while the time was 3237.0s for the blob overflow scheme. This is approximately a 30% increase in the average time taken to compute each timestep. Interestingly, the actual blob scheme is cheaper than the original overflow scheme, taking an average of 5.2s and 23.1s respectively. However, it is all of the associated infrastructure that makes the blobs that much more expensive.

# Chapter 11

## The Deep Convection Test Case

The deep convection case is based on the configurations of Jones and Marshall (1993) and Campin et al. (2011). It is a doubly periodic domain, with a flat bottom and initially uniform stratification. A circular negative buoyancy flux is applied to the surface, creating static instability and causing deep convection. The objective of this experimental configuration is to examine whether the blobs are able to improve the representation of the physics of a deep convection mixed patch. The physics of open ocean deep convection is discussed in Section 2.1. It is hypothesised that use of the blobs will improve the profile and variability of a passive tracer in the convecting patch.

### 11.1 Configuration Details

The experiments are conducted on an  $f$ -plane, with a value of  $f_0 = 10^{-4}\text{s}^{-2}$ , which is the value at approximately 43°N. The domain is 40km×40km and 2000m deep. The radius of the circular cooling disc is 10km, and a heat loss of  $800\text{W m}^{-2}$  is applied to the disc. The timestep is 5s in all experiments<sup>1</sup>. The domain is initially stratified with a buoyancy frequency of  $N = 3 \times 10^{-4}\text{s}^{-1}$ , and has an initial surface temperature of 20°C. A passive dye tracer is used in all experiments. Initially, the dye tracer is set to zero uniformly except in the surface cells where it is set to 1.0. The dye tracer is restored to a value of 1.0 at the surface with a timescale of 300s.

Several model configurations are run, including three cases using the MITgcm (Marshall et al., 1997a,b, 1998). The MITgcm experiments are run with a rigid lid with geopotential coordinates, while all of the MOM cases are run with the  $z^*$  vertical coordinate with a free surface. There are two different resolutions that are examined. The coarse resolution case is run in both the MITgcm and MOM and is a  $20(x) \times 20(y) \times 20(z)$  hydrostatic configuration with a uniform grid spacing of 2km in the horizontal and 100m in the vertical. The fine resolution case is only run in the MITgcm and is a  $400(x) \times 400(y) \times 20(z)$  configuration with a grid spacing of 100m in the horizontal and the vertical.

---

<sup>1</sup>Campin et al. (2011) have a 60s timestep, and, it was found that most experiments were stable with a 60s timestep, however, some blob experiments (particularly experiments with large numbers of blobs) were catastrophically unstable.

Name	Model	Resolution	Convective Parameterisation
NH	MITgcm	Fine	N/A (non-hydrostatic)
HYD_FR	MITgcm	Fine	Enhanced diffusion
HYD	MITgcm	Coarse	Enhanced diffusion
noblob	MOM	Coarse	Enhanced diffusion
ctrl	MOM	Coarse	Free Blobs

Table 11.1: Experimental details for the deep convection test case for all experiments not using dynamically active blobs and one that does (see Table 11.2 for a description of all experiments using blobs). The fine resolution configuration has a resolution of  $100\text{m} \times 100\text{m} \times 100\text{m}$ , while the coarse resolution case has a resolution of  $2\text{km} \times 2\text{km} \times 100\text{m}$ . All experiments use a constant viscosity.

The details of the various experiments are summarised in Tables 11.1 and 11.2. No experiments use lateral or vertical diffusion. A constant viscosity, with a value of  $1.0\text{m}^2\text{s}^{-1}$ , is used in the vertical for all experiments. In the horizontal, a value of  $1.0\text{m}^2\text{s}^{-1}$  is used for the fine resolution cases and  $20.0\text{m}^2\text{s}^{-1}$  is used in the coarse resolution cases. For the hydrostatic runs, an enhanced vertical diffusivity of  $50.0\text{m}^2\text{s}^{-1}$  is used for the cases where convective adjustment is used. The MOM cases use the multi-dimensional piecewise parabolic method of Colella and Woodward (1984) with the flux limiter described by Lin (2004) for both horizontal and vertical tracer advection. For tracer advection the MITgcm runs use a non-linear third order direct space time with a Sweby flux limiter.

### 11.1.1 The Hydrostatic Approximation and Model Effects

The NH experiment is a fine resolution setup in the MITgcm that employs non-hydrostatic dynamics, while the HYD\_FR experiment is also a fine resolution MITgcm experiment that employs the hydrostatic approximation. The purpose of the NH experiment is to provide a “ground truth” to which the other solutions can be compared. The HYD\_FR experiment is designed to examine the effect of introducing the hydrostatic approximation on the solution, since it is identical to NH except for its employment of the hydrostatic approximation. A coarse resolution version, HYD, is also run, which has two purposes. The first purpose is to examine the effect of resolution on the solution (by comparison with NH and HYD\_FR) and to provide an experiment by which to compare the two different models (being the MITgcm and MOM) on the solution.

The HYD experiment is directly comparable to the noblob experiment, which is a coarse resolution experiment in MOM which employs convective adjustment and does not use the Lagrangian blob scheme. The two experiments are as similar as possible and thus, differences between the solutions of the HYD and noblob experiments are due to differences in model formulation and between parameterisations (such as tracer advection) where there is no directly comparable schemes in both models. The ctrl experiment uses the Lagrangian blob scheme and is used to examine the effects of the introduction of the

pseudo-non-hydrostatic dynamics admitted by the Lagrangian blob scheme.

### 11.1.2 Lagrangian Blob Parameters

For all experiments using the Lagrangian blob scheme, a blob's mass is considered “small” when it is less than 1000kg, and all remaining properties are returned to the E system when a blob's mass is less than 1000kg. In other words, a blob is considered small when its volume is less than approximately 1m<sup>3</sup>. The maximum proportion of a grid cell that the L system may occupy is 80% (see Section 6.5.3). The Runge-Kutta scheme used is that of Bogacki and Shampine (1989) with a local relative truncation error of  $\zeta^* = 0.05$ , a safety factor of  $\varrho = 0.8$  and a minimum step size of  $h_{\min} = 0.05$ s (see Section 7.3.2). The maximum allowable detrainment velocity is  $\mathcal{D}_{\max} = 10^{15}$ m s<sup>-1</sup>, which is a large enough upper bound that the detrainment velocity is effectively not limited.

The `ctrl` experiment is the baseline experiment from which the other blob experiments are compared. The other blob experiments, which are summarised in Table 11.2, vary a single parameter at a time to explore the parameter space of the blobs and to investigate how those parameters affect the properties of the convecting patch.

#### Interfacial Drag Coefficient, $\alpha$

The first parameter to be varied is the interfacial Rayleigh drag coefficient,  $\alpha$ , which has units of s<sup>-1</sup>. The `ctrl` value for  $\alpha$  is chosen to be  $5 \times 10^{-3}$ s<sup>-1</sup>. The `c5e-2` and `c5e-5` experiments vary that parameter to be larger and smaller to investigate the sensitivity of the solution to that Rayleigh drag parameter.

Name	$\alpha$	$\Gamma$	$\Lambda$	$N_0^2$
<code>ctrl</code>	$5 \times 10^{-3}$	$1 \times 10^{-6}$	0.10	$-5 \times 10^{-7}$
<code>c5e-2</code>	$5 \times 10^{-2}$	$1 \times 10^{-6}$	0.10	$-5 \times 10^{-7}$
<code>c5e-5</code>	$5 \times 10^{-5}$	$1 \times 10^{-6}$	0.10	$-5 \times 10^{-7}$
<code>G1e-4</code>	$5 \times 10^{-3}$	$1 \times 10^{-4}$	0.10	$-5 \times 10^{-7}$
<code>G1e-5</code>	$5 \times 10^{-3}$	$1 \times 10^{-5}$	0.10	$-5 \times 10^{-7}$
<code>G1e-7</code>	$5 \times 10^{-3}$	$1 \times 10^{-7}$	0.10	$-5 \times 10^{-7}$
<code>G1e-8</code>	$5 \times 10^{-3}$	$1 \times 10^{-8}$	0.10	$-5 \times 10^{-7}$
<code>L25</code>	$5 \times 10^{-3}$	$1 \times 10^{-6}$	0.25	$-5 \times 10^{-7}$
<code>L05</code>	$5 \times 10^{-3}$	$1 \times 10^{-6}$	0.05	$-5 \times 10^{-7}$
<code>N5e-5</code>	$5 \times 10^{-3}$	$1 \times 10^{-6}$	0.10	$-5 \times 10^{-5}$
<code>N5e-6</code>	$5 \times 10^{-3}$	$1 \times 10^{-6}$	0.10	$-5 \times 10^{-6}$
<code>N5e-9</code>	$5 \times 10^{-3}$	$1 \times 10^{-6}$	0.10	$-5 \times 10^{-9}$

Table 11.2: Experimental details for the coarse resolution deep convection test cases that use dynamically active blobs.  $\alpha$  is the coefficient of drag,  $\Gamma$  is the detrainment parameter,  $\Lambda$  is the scaling parameter for the initial blob size and  $N_0^2$  is the threshold at which a new blob is formed.

### Detrainment Parameter, $\Gamma$

The detrainment parameter is a parameter for which there is no observational evidence or previous study to base an initial value on. As such, a large parameter space is spanned to examine the effect of the parameter on the solution. The largest value chosen is  $1 \times 10^{-4} \text{ kg m}^2 \text{s}^{-1}$  in the `G1e-4`, while the smallest value chosen is  $1 \times 10^{-8} \text{ kg m}^2 \text{s}^{-1}$  in the `G1e-8` experiment.

### Blob Scaling Parameter, $\Lambda$

Similarly, the non-dimensional scaling parameter for the initial size of a blob,  $\Lambda$ , has no observational or scaling estimates to base an estimate on. Since all of the grid cells in these experiments have the same initial mass, a value of  $\Lambda = 1.0$  would mean that blobs are half of the mass of a grid cell – see the discussion surrounding Equation (5.8a). An experiment was run in which  $\Lambda = 1.0$ , however, it was found that the vast majority of blobs were destroyed almost immediately. Thus, a much smaller value is desirable for practical simulations. The `L25` experiment has a value of  $\Lambda = 0.25$ . The value in the `ctrl` experiment is  $\Lambda = 0.10$  and for the small value, it was chosen that the `L05` experiment should have the value of  $\Lambda = 0.05$ . An experiment was also run with a value of  $\Lambda = 0.01$ , however, this experiment proved impractically expensive due to the very large numbers of blobs being produced.

### Instability Threshold, $N_0^2$

The instability threshold,  $N_0^2$ , at which a blob is formed is investigated. A wide parameter range is explored to investigate the sensitivity of the solution to this parameter. The experiments `N5e-5`, `N5e-6`, `ctrl` and `N5e-9` investigate the effect of this parameter over four orders of magnitude. In fact, other experiments were devised that explore this parameter over a wider range, however, parameters outside of this range were deemed to be impractical. As it is, the `N5e-5` experiment threshold is too large to produce any blobs at all, while the `N5e-9` experiment produces a very large number of blobs which severely affects the performance of the model (and thus, very significantly increases the runtime).

## 11.2 Results

### 11.2.1 Hydrostatic Approximation

To examine the effect of the hydrostatic approximation, two fine resolution experiments are run using the MITgcm. The `NH` experiment uses the non-hydrostatic capability of the MITgcm, while `HYD_FR` is virtually identical to the `NH` experiment but makes the hydrostatic approximation.

Figure 11.1 shows the evolution of the dye tracer under the cooling disc for the `NH` experiment, while Figure 11.2 shows the dye tracer evolution under the cooling disc for the `HYD_FR` experiment. Following the procedure used by Campin et al. (2011), both of

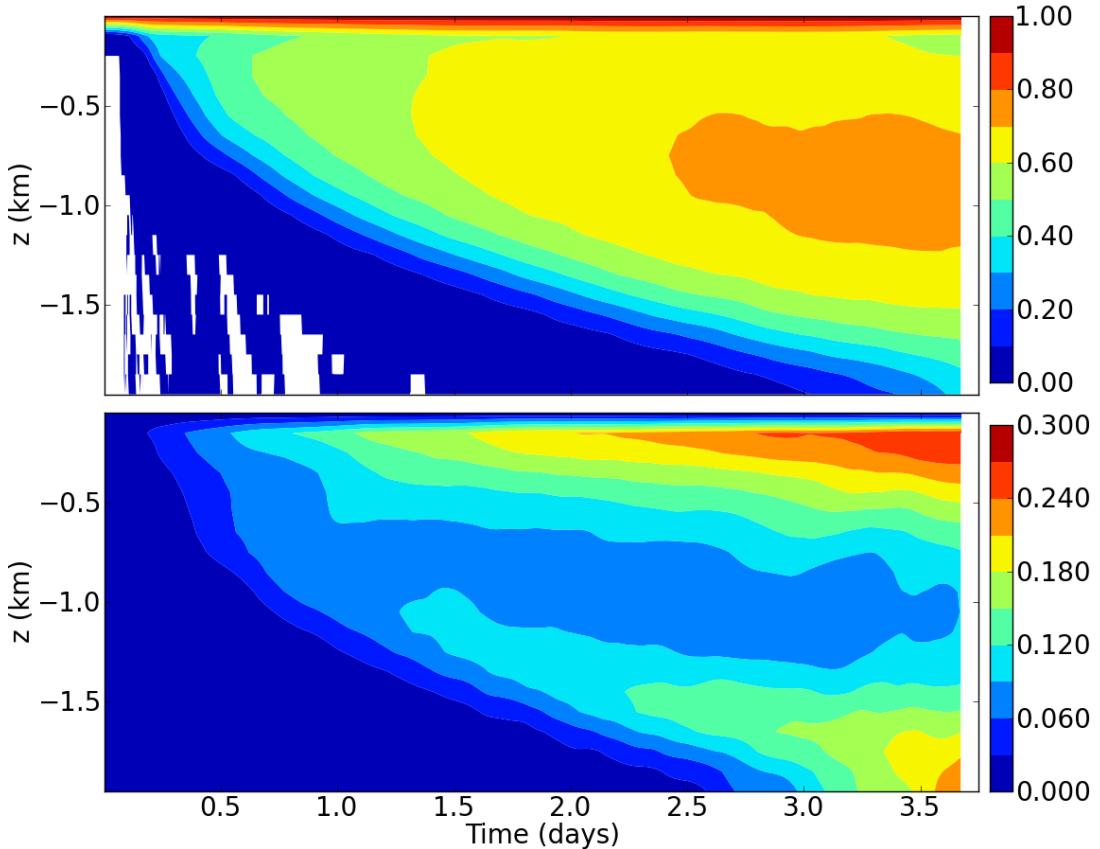


Figure 11.1: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer under the cooling disc in the NH experiment. The NH experiment is a fine resolution, non-hydrostatic experiment and is taken as a “ground truth” for the deep convection test case. The results are averaged onto the coarse resolution grid so as to be comparable to the coarse resolution experiments.

these fine resolution experiments are horizontally averaged onto the coarse grid in order to be comparable to the coarse resolution experiments. There is quite a difference between these two experiments, both with the tracer concentration and the standard deviation of the tracer concentration.

The NH experiment develops two dye tracer maxima, one at the surface, and the other is at a depth of about 1000m. By contrast, the dye tracer in the HYD\\_FR experiment is a maximum at the surface and the thickness of that maximum increases with time. By the end of the simulation, the entire water column under the cooling disc has a mean value of close to 1.0. The standard deviation of the NH experiment is a maximum above and below the tracer maximum at 1000m, while the HYD\\_FR experiment has a maximum standard deviation that roughly follows the tracer front as it descends. The magnitude of the standard deviation of the tracer is also much smaller in the HYD\\_FR experiment. The difference between these two experiments highlights the effect of using a convective adjustment parameterisation in the case where non-hydrostatic dynamics are important.

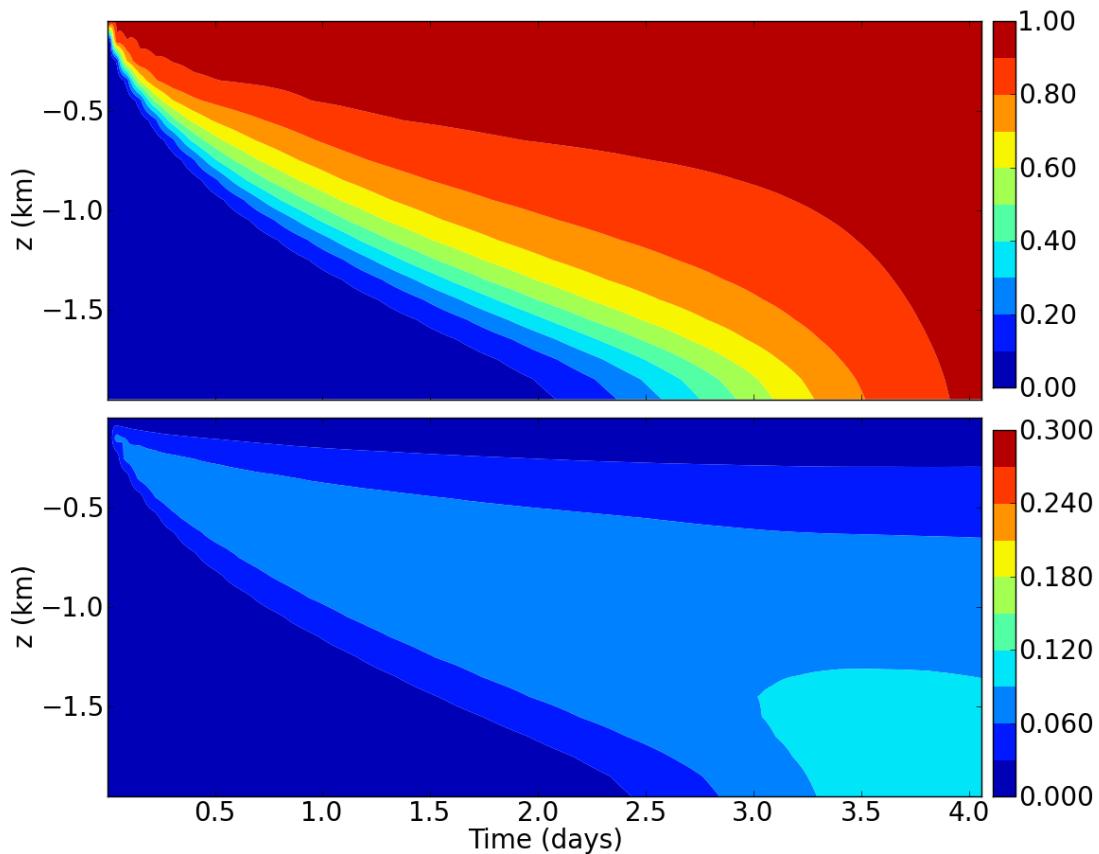


Figure 11.2: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer under the cooling disc in the HYD\\_FR experiment. The results are averaged onto the coarse resolution grid so as to be comparable to the coarse resolution experiments. As can be seen when comparing with Figure 11.2 the use of the hydrostatic approximation significantly alters the distribution and variability of the passive tracer under the cooling disc.

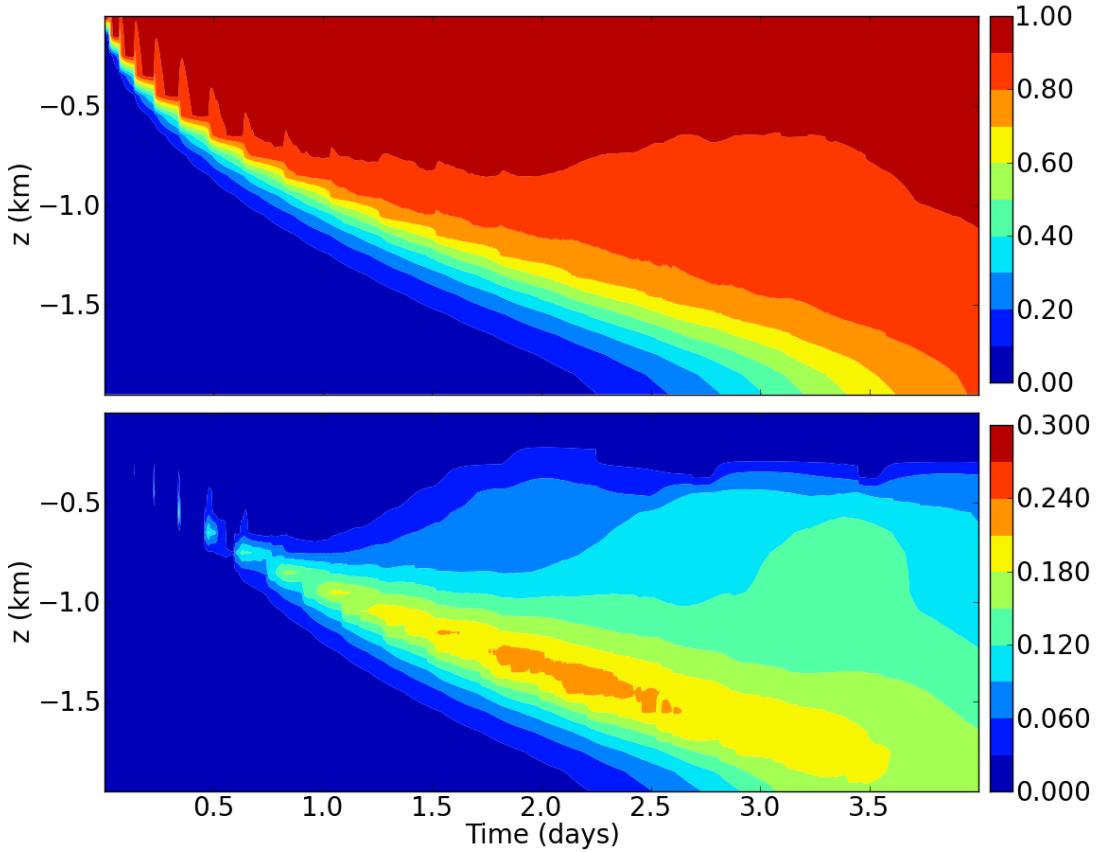


Figure 11.3: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer under the cooling disc in the HYD experiment. When comparing with the results of the non-hydrostatic NH experiment shown in Figure 11.1 and the fine resolution hydrostatic HYD\_FR experiment, shown in Figure 11.2, it can be concluded that the effect of introducing the hydrostatic approximation is much greater than the change in resolution between these experiments.

To examine the sensitivity of the result to resolution, the HYD experiment is shown in Figure 11.3. The pattern of tracer distribution and standard deviation is similar to that of the HYD\_FR experiment, however, the standard deviation is appreciably larger in the HYD experiment from the end of the first day onwards. Toward the beginning of the HYD experiment, the tracer depth deepens in jumps. These jumps are likely associated with the convective adjustment scheme. It can be concluded from these experiments, however, that the introduction of the hydrostatic approximation has a much greater effect on the solution than the coarsening of the resolution.

The `noblob` experiment, shown in Figure 11.4, is the most comparable experiment of the experiments conducted in the MOM to the HYD experiment. The step-like nature of the way that the tracer migrates down the water column is more pronounced in the `noblob` experiment than the HYD experiment. Although another major difference is that the tracer standard deviation is extremely small.

The effects of the hydrostatic approximation are to change the way that the tracer

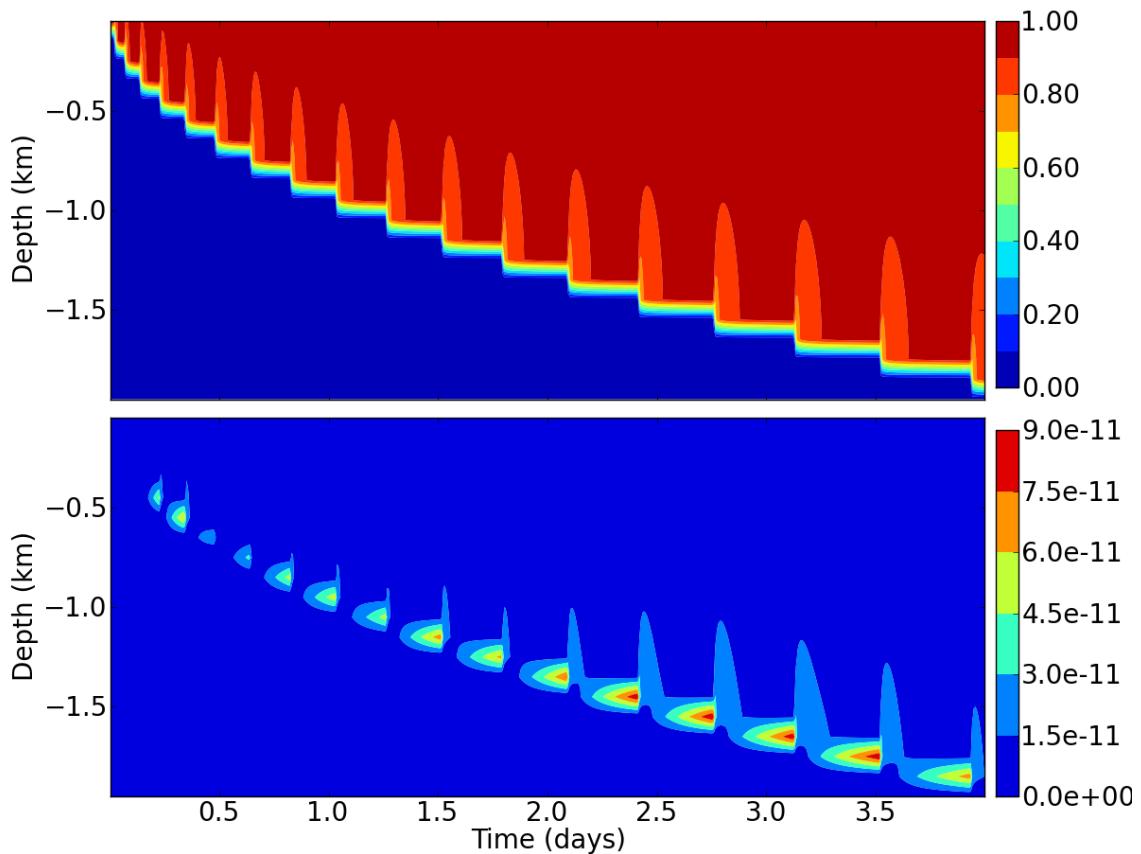


Figure 11.4: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer in the `noblob` experiment, which is in MOM and uses convective adjustment to remove vertical instability in the water column. Note the different colour scale on the standard deviation to other experiments. The `noblob` experiment is largely similar to the `HYD` experiment, except the step-like nature of the increase in depth is much more accentuated, and, there is much less horizontal variability in the passive tracer.

under the cooling patch is distributed, both vertically (as is clear from the vertical tracer profile) but also horizontally (as is clear from the horizontal standard deviation). This change is due to the differences in the way that tracer is transported between the non-hydrostatic and the hydrostatic models.

The introduction of the pseudo-non-hydrostatic dynamics via the embedded Lagrangian framework is illustrated in the `ctrl` experiment via the evolution of tracer concentration and horizontal tracer standard deviation in Figure 11.5. Firstly, it can be seen that the introduction of the pseudo-non-hydrostatic embedded Lagrangian model makes a very large difference to both the concentration and the standard deviation when compared to the `noblob` experiment, shown in Figure 11.4. Qualitatively, the passive tracer concentration evolution of the `ctrl` experiment and the `NH` experiment is more similar than the evolution of the `NH` experiment and the `HYD_FR` experiment. This indicates that the introduction of the pseudo-non-hydrostatic dynamics admitted by the Lagrangian blob scheme is potentially a large improvement in the representation of the underlying physical processes. It shall be seen in subsequent sections in this chapter, however, that the solution is very dependent on the parameter values chosen. There is also a numerical instability that develops in the `ctrl` experiment and a number of other experiments using the Lagrangian blob scheme. The onset of the instability results in rapid mixing and homogenisation of the water column, starting just before day 3.5 in the `ctrl` experiment. It could be that the root cause of the numerical instability is the simplification of the non-hydrostatic dynamics. The causes and consequences of this instability in specific experiments is discussed in more detail the following sections.

### 11.2.2 Mean Depth of Passive Tracer

The mean depth of the passive tracer is a measure of the penetration of the surface waters into the water column. The mean depth is taken over the entire domain, not just under the cooling disc. A grid cell is considered to be tagged with passive tracer if its concentration is greater than or equal to 0.01. In the cases where the Lagrangian blob scheme is used, it is the combined E and L system concentration that is used. Since the domain has a flat bottom with a depth of 2000m, a mean depth of 1000m indicates that there are as many tagged grid cells in the the bottom half of the water column as the top half. The mean depth is shown in Figure 11.6. The tagged water makes it all the way to a mean depth of 1000m in most experiments, however, it is noteworthy that the `NH` experiment does not make it to a mean depth of 1000m. Also, the increase in depth of the `HYD` experiment is much more step like than the `NH` experiment. This is likely due to the nature of convective adjustment being a threshold condition.

It can be seen when comparing the `NH` experiment with the `HYD` experiment that the introduction of the hydrostatic approximation with convection adjustment tends to deepen the mean depth of tagged water. The `HYD` and `HYD_FR` experiments are very similar and thus the `HYD_FR` experiment is not shown. The `noblob` experiment tends to have its

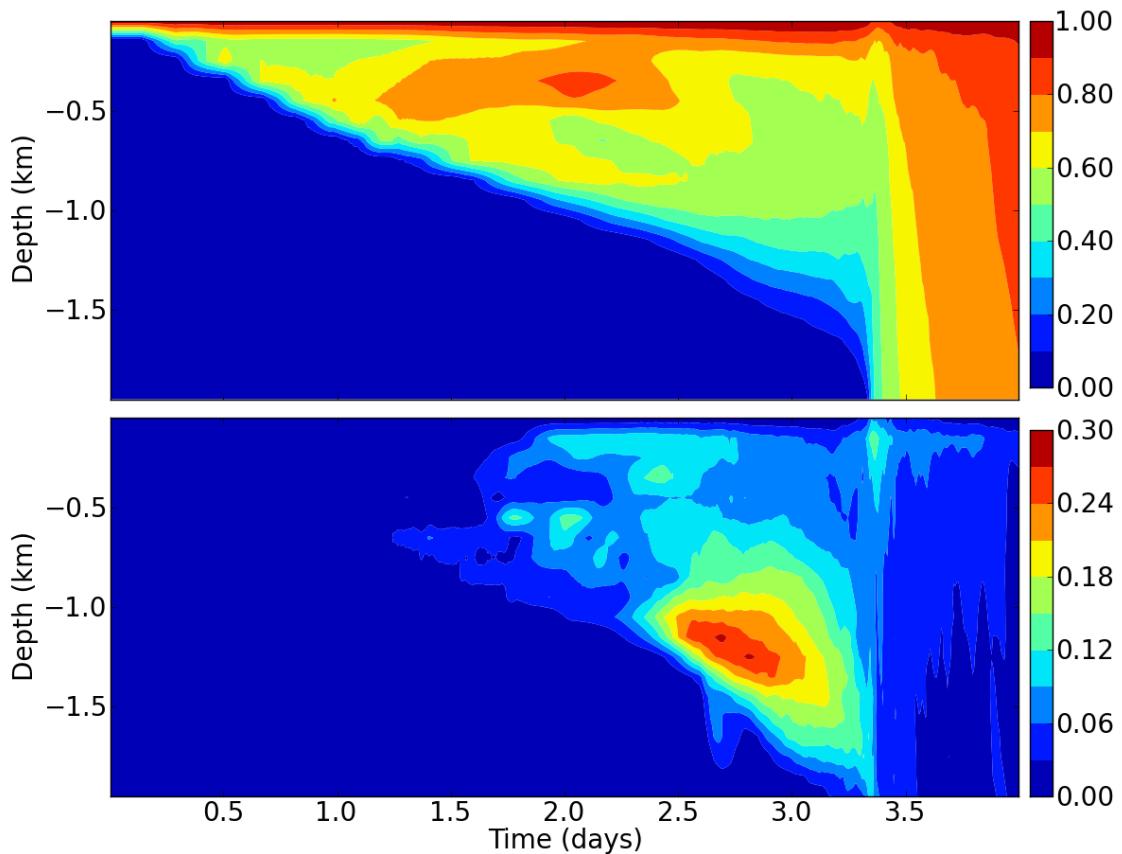


Figure 11.5: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer in the `ctrl` experiment. The introduction of the pseudo-non-hydrostatic dynamics admitted by the Lagrangian blob scheme clearly affects the solution when compared to the `noblob` experiment in Figure 11.4. In many ways the solution is a large step towards the solution exhibited by the non-hydrostatic `NH` experiment (Figure 11.1). The solution is, however, quite sensitive to the parameter value choices of the blobs.

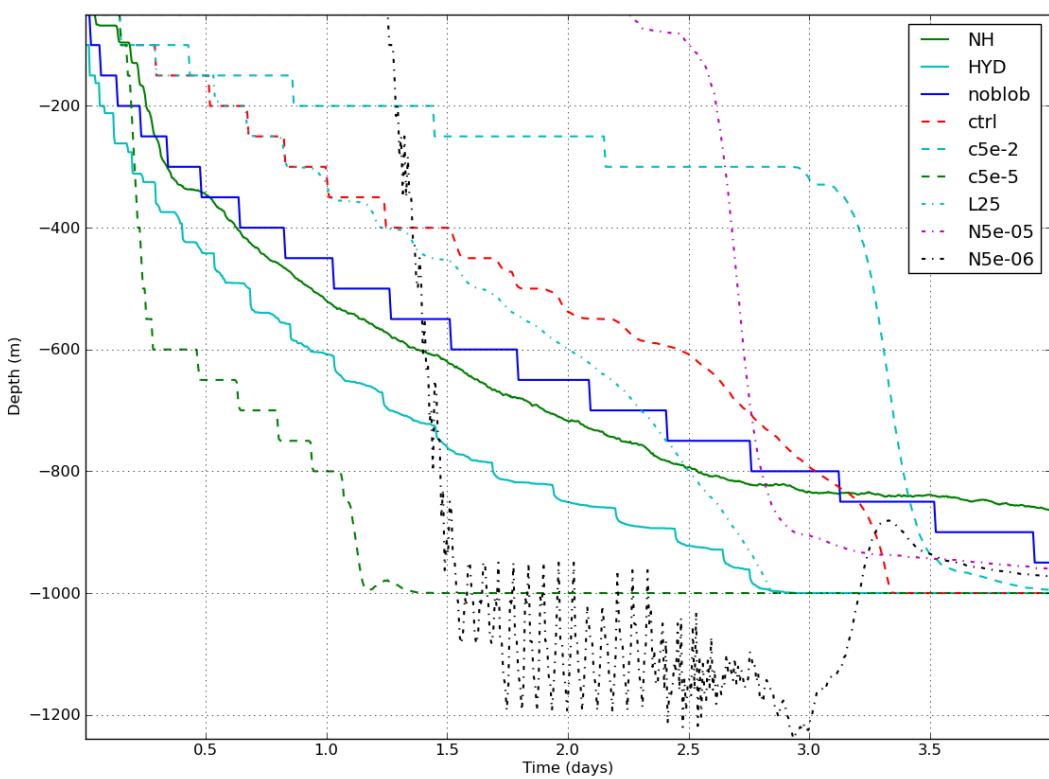


Figure 11.6: The mean depth of grid cells that have been tagged with passive tracer (i.e. that have concentration greater than or equal to 0.01). The effect of the hydrostatic approximation, pseudo-non-hydrostatic dynamics and the various blob parameters are discussed in the text.

tracer more shallow than the HYD experiment (the **noblob** experiment is the equivalent experiment in the MOM to the HYD experiment in the MITgcm). The introduction of pseudo-non-hydrostatic dynamics changes the nature of the curves significantly. In the case of the **ctrl** experiment, it shallows the mean depth of tagged water for most of the experiment, when compared to the **noblob** experiment.

By comparing the **c5e-5**, the **ctrl** and the **c5e-2** experiments (which are identical, save for their interfacial drag coefficients which are  $5 \times 10^{-5}$ ,  $5 \times 10^{-3}$  and  $5 \times 10^{-2}\text{s}^{-1}$  respectively), it can be seen that the influence of the Rayleigh drag coefficient,  $\alpha$ , is to alter how rapidly the mean depth of the passive tracer increases, with the larger drag coefficient having the shallower mean passive tracer depth. The reason for the difference is that blobs that have a lower interfacial friction coefficient travel faster than those that have a higher friction coefficient, thereby penetrating deeper into the water column. It can be seen in Figure 11.7 that in the **c5e-5** experiment highest concentration of the passive tracer is found at the surface and at the bottom. This occurs after the development of numerical instability in the **c5e-5** experiment. It is hypothesised that the numerical instability exhibited by the **c5e-5** is different to other experiments that exhibit numerical instability in that the source of the instability is with the Lagrangian blob scheme. It is thought that the large velocities and deep penetration of the blobs causes there to be a rapid change in properties that the numerics of the blob scheme are not able to handle.

The threshold square of the buoyancy frequency at which a blob is formed,  $N_0^2$ , also has a large influence on the mean depth of the passive tracer. The experiment that has a small threshold, **N5e-9**, behaves vastly different to the **ctrl** experiment, which in turn behaves vastly different to the experiment with a large threshold, **N5e-5**. The **N5e-9** experiment, which has a threshold of  $N_0^2 = 5 \times 10^{-9}\text{s}^{-2}$  has behaviour that is extremely similar to the **noblob** experiment (and is thus not shown). This is because there are a very large number of blobs produced with such a small threshold. The large number of blobs then means that a significant proportion of mass resides in the L system, which in turn causes the grid cell mass constraint to be violated frequently (see Section 6.5.3). This is discussed in more detail in Section 11.2.6. In the context of the present discussion, the consequence is that the scheme is effectively running as the blob-NCon scheme (see Section 4.1).

On the other hand, having a relatively large threshold, as is the case in the **N5e-5** experiment means that the water column must be very unstable in order to form blobs. In fact, in the case of the **N5e-5**, no blobs are formed in the experiment at all. This explains why it takes over two days in Figure 11.7 for the **N5e-5** experiment to begin to deepen the mean depth of the passive tracer. The deepening, when it eventually occurs is very rapid and is likely a result of some kind of numerical instability, as there is no convective adjustment scheme active in that experiment. The **N5e-6** experiment also takes quite a while (more than one day) for the mean tracer depth to begin deepening. As can be seen in Figure 11.14, it is the time of the rapid deepening of the mean depth of the passive

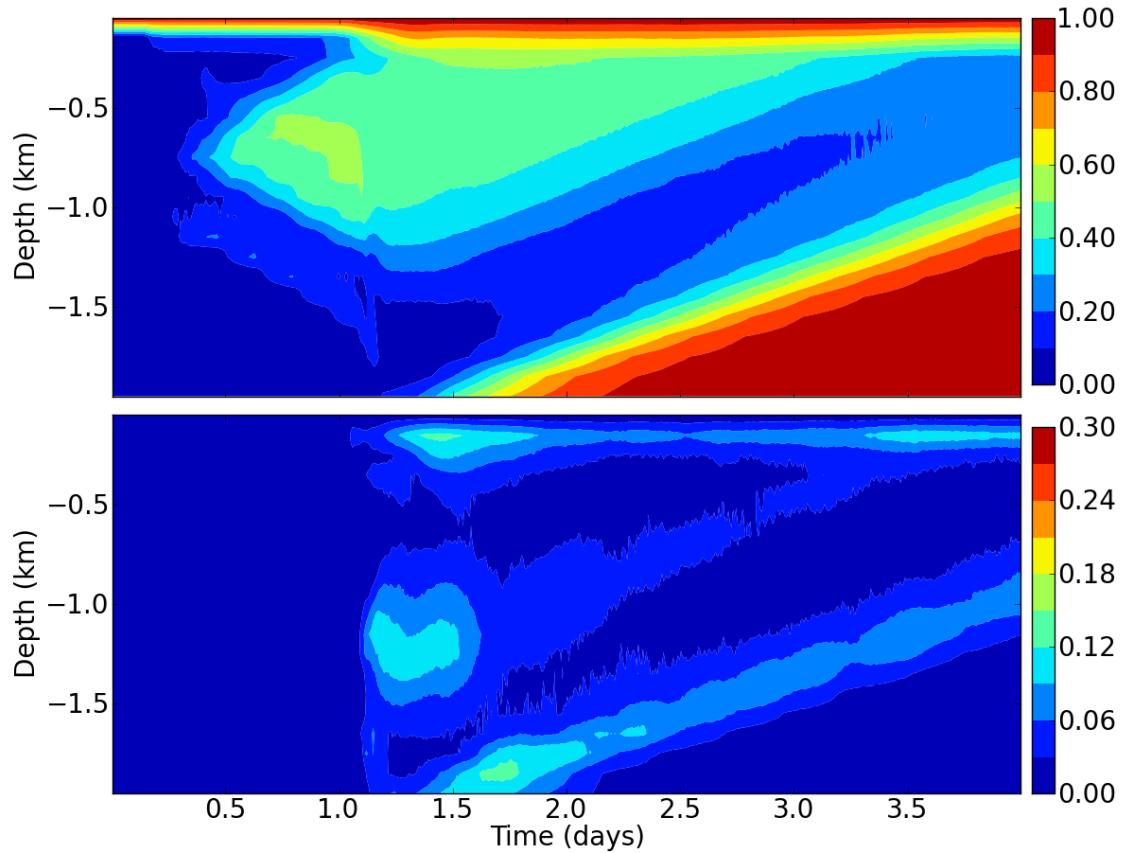


Figure 11.7: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer in the  $c5e-5$  experiment, which has a small value for the Rayleigh drag coefficient. The small drag coefficient means that blobs sink rapidly and interact with the bottom, thereby transferring their properties back to the E system. This means that the value of the passive tracer is large near the bottom, producing an unphysical result.

tracer that blobs are formed. This indicates that the deepening of the tracer is largely caused by the Lagrangian scheme. The mean depth does, however, fluctuate rapidly for a period. This indicates that there is some sort of numerical instability in the experiment.

From this, we can conclude that in this test case a threshold of  $N_0^2 = 5 \times 10^{-6}\text{s}^{-2}$  as in the **N5e-6** experiment is too large, while a value of  $N_0^2 = 5 \times 10^{-9}\text{s}^{-2}$  as in the **N5e-9** is too small. The optimal value likely lies somewhere in the vicinity of  $N_0^2 = 5 \times 10^{-7}\text{s}^{-2}$ , as in the **ctrl** experiment.

### 11.2.3 Depth of Blobs

There are three main factors that determine the mean depth of the blobs in an experiment:

- where the blobs are formed,
- how far the blobs travel vertically, and
- where the blobs are destroyed;

all three factors are in turn controlled by the model's stratification. In the deep convection experiment, blobs should initially form near the surface (the source of static instability) and then sink. How fast they sink depends on the value of the drag parameter,  $\alpha$ , and the value of the reduced gravity term. How far they travel depends on how fast they are travelling, and, when they are destroyed. As can be seen in Figure 11.8 there is a variety of behaviour amongst the experiments.

The behaviour of the experiments which test the effect of the Rayleigh drag coefficient,  $\alpha$ , confirm what was discussed in Section 11.2.2. That is, the low coefficient of drag experiment, **c5e-2**, has blobs that are quite deep because the blobs sink rapidly in the water column. The **ctrl** experiment slowly increases, however, toward the end of the experiment, the average depth of blobs does deepen rapidly so as to exceed the mean depth of the **c5e-2** experiment. The reason for this is discussed later in this section. The experiment with the large coefficient of drag **c5e-2** has relatively shallow blobs for much of the simulation. Similar to the **ctrl** experiment, there is a rapid deepening of the mean blob depth toward the end of the experiment. The reason for this also shall be discussed later in this section.

Somewhat surprisingly the value of the detrainment parameter,  $\Gamma$ , has a large impact on the mean depth of the blobs for larger values, that is, there is a large difference in behaviour between the **G1e-4**, the **G1e-5** and the **ctrl** experiments. The reason for this has to do with the third factor listed at the beginning of this section, i.e. where the blobs are destroyed. The **G1e-4** experiment, which has  $\Gamma = 1 \times 10^{-4}\text{kg m}^{-2}\text{s}^{-1}$ , maintains a fairly shallow mean blob depth for most of the experiment, except for a large trough in the second half of the second day. In fact, most experiments exhibit some sort of trough, or, rapid decrease and then stabilisation of the mean blob depth, with the notable exception of the **G1e-5** experiment, which has  $\Gamma = 1 \times 10^{-5}\text{kg m}^{-2}\text{s}^{-1}$ . This begs the question,

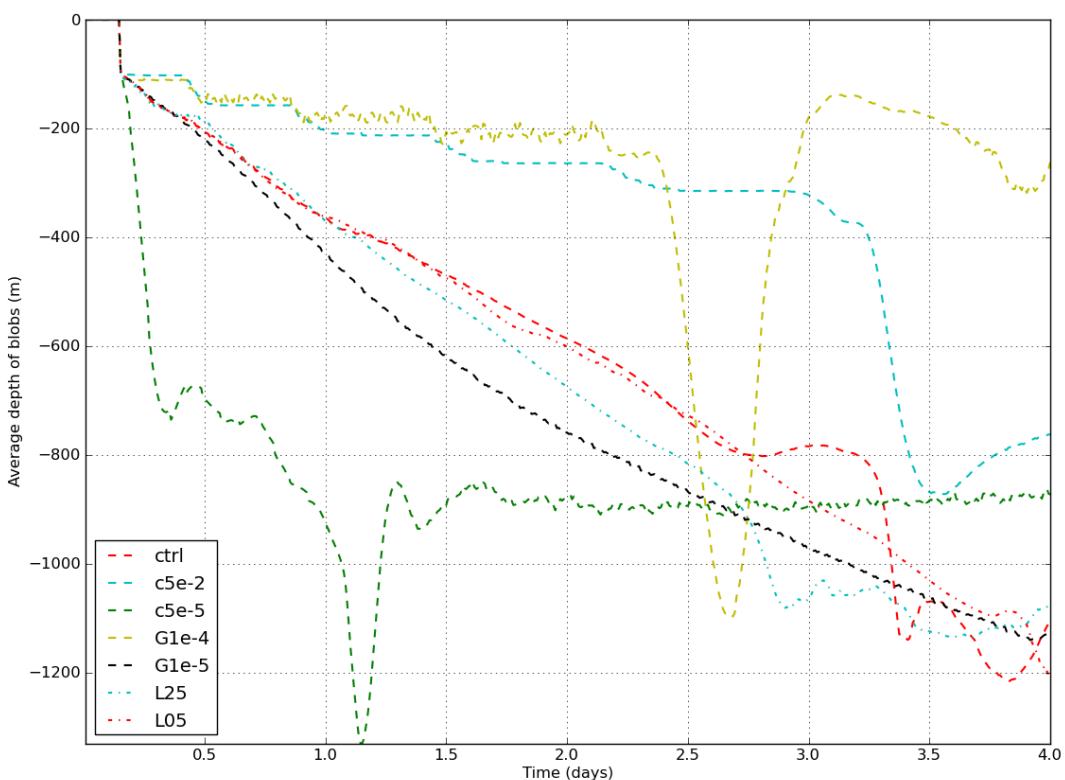


Figure 11.8: The mean depth of blobs in various experiments. The values of the Rayleigh drag coefficient,  $\alpha$  and the detrainment parameter,  $\Gamma$  have a large influence on the mean depth of blobs. The initial blob size scaling parameter,  $\Lambda$  appears to have relatively little influence, however, it does have an influence on the numerical stability of the simulation. See details in text.

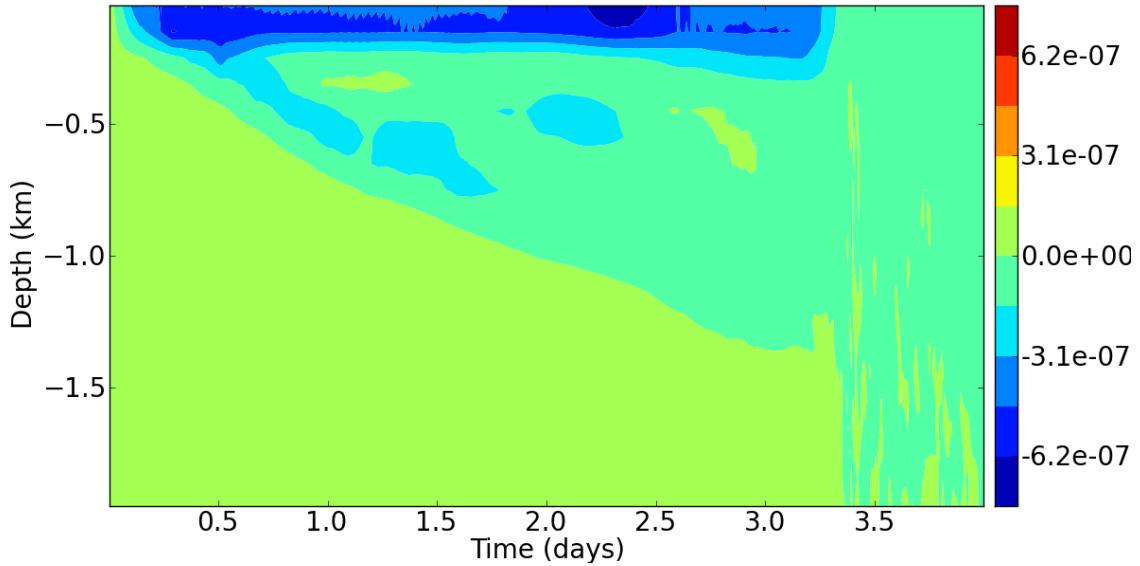


Figure 11.9: The value of the square of the buoyancy frequency ( $\text{s}^{-2}$ ) for the `ctrl` experiment for the region under the cooling disc. The experiment experiences an increase in static instability approximately 2.5 days into the experiment, which is just prior to the experiment rapidly mixing properties through the water column.

what causes this rapid change, and, why does it not occur in the `G1e-5` experiment?

As can be seen in Figure 11.13, the `G1e-4` experiment has relatively small blobs, due to the large detrainment. This means that most blobs fully detraining before they can penetrate to depth, thereby limiting their ability to restabilise the water column. The rapid deepening of the blobs is due to the numerical instability in the E system developing and is the same reason why the `N5e-5` rapidly mixes its properties when the static instability of the system becomes too great for the numerics to handle (see Section 11.2.2). The mechanism that causes the numerical instability in the `ctrl` experiment is different to that of the `G1e-4` experiment, but, the numerical instability nevertheless develops and causes a rapid change in properties of the simulation (see Figure 11.5). The reason for the numerical instability developing in the `ctrl` experiment is that a relatively large number of blobs are destroyed in the `ctrl` experiment by the grid cell mass constraint being violated. The blobs are being destroyed near the site of creation, thereby inhibiting the ability of the L system to restabilise the water column. Thus, the static instability of the water column increases with time, as can be seen in Figure 11.9, which shows the time and depth evolution of the square of the buoyancy frequency,  $N^2$ , of the `ctrl` experiment. By contrast, the equivalent plot for the `G1e-5` experiment is shown in Figure 11.10. The reason why the `G1e-5` is able to stop the static instability from increasing is that less blobs are being destroyed because they are smaller (due to the larger detrainment parameter). This can be seen clearly in the mean blob mass Figure 11.13 which shows that the `G1e-5` experiment has a mean blob mass approximately the same as the `L05` experiment (which is an experiment that produces blobs of a small initial mass) for much of the duration of the experiment. Having less blobs that are destroyed means that more blobs are able to

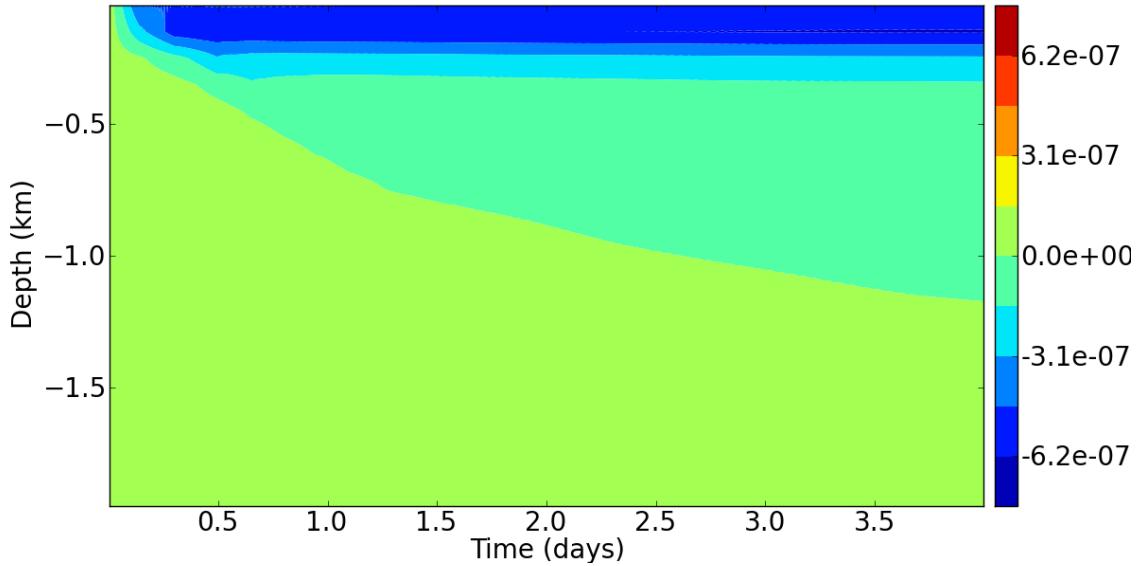


Figure 11.10: The value of the square of the buoyancy frequency ( $s^{-2}$ ) for the `G1e-5` experiment for the region under the cooling disc. In contrast to the `ctrl` experiment, shown in Figure 11.9, the stability of the water column remains relatively steady throughout the experiment once the initial reaction to the surface cooling is established.

sink, which explains the deeper average blob depth.

The inability of the blobs to restabilise the water column in the `ctrl` experiment is accentuated in the `c5e-2` experiment, because the blobs are travelling slower. Since there are even more blobs near the creation site, the grid cell mass constraint is violated even more frequently.

A consequence of having a more numerically stable simulation is that the distribution of the tracer in the horizontal becomes much more uniform, as is shown in Figure 11.11. Thus, one concludes that part of the reason for the reason why the passive tracer distribution in the `ctrl` experiment (Figure 11.5) is so similar to that in the `NH` experiment (Figure 11.1) is at least partly due to the numerical instability allowed to develop in the E system of the `ctrl` experiment. As shall be discussed in Section 11.2.6, however, the onset of that numerical instability appears to at least be partly controlled by the blob scaling parameter,  $\Lambda$ .

The effect of the blob scaling parameter,  $\Lambda$ , on mean blob depth appears to be relatively minor, since the curves for the `L05`, the `ctrl` and the `L25` experiments (which represent small, medium and large blobs respectively) are all reasonably similar in Figure 11.10. However, as is discussed in detail in Section 11.2.6,  $\Lambda$  affects the timing of the onset of the development of numerical instability. A larger initial blob size brings forward the onset of numerical instability, which is manifested in the mean blob depth by a rapid increase in blob depth. The mechanism for the alteration in the timing of the onset of numerical instability is discussed in Section 11.2.6.

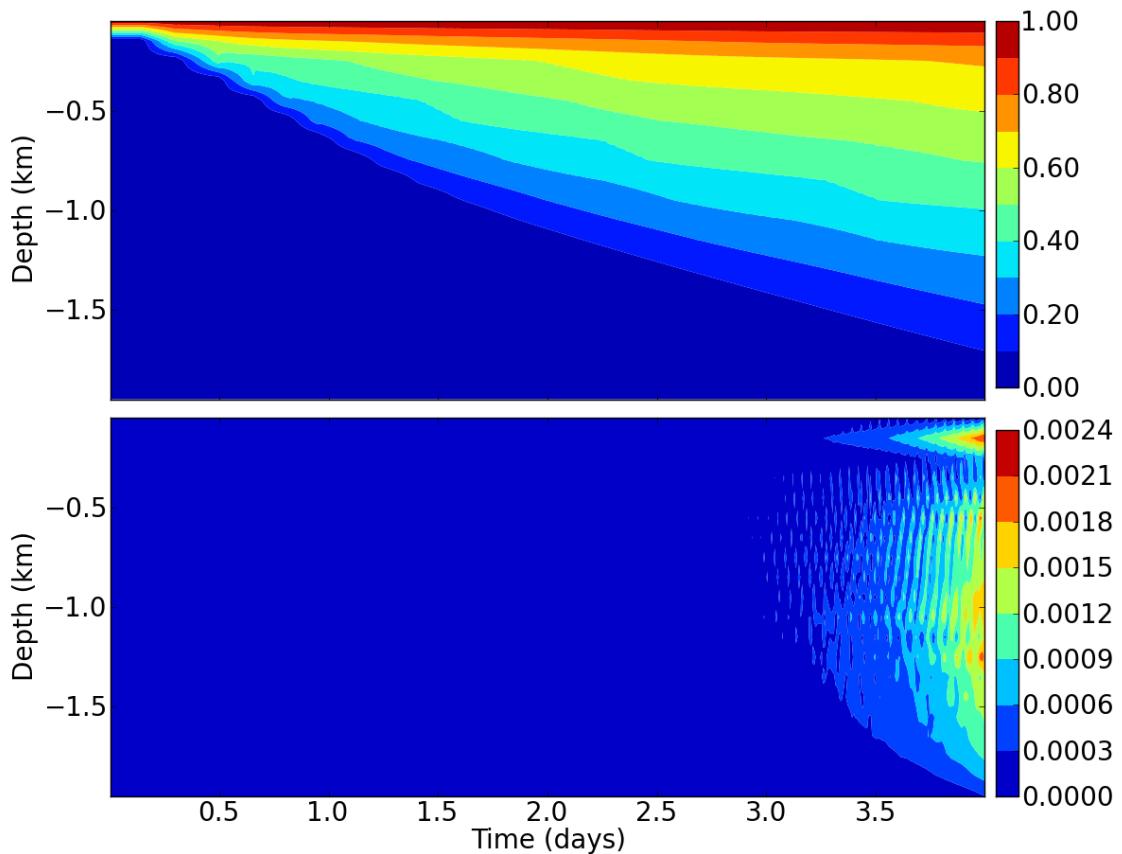


Figure 11.11: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer under the cooling disc in the `G1e-5` experiment. Note the different scale for the bottom panel from most other experiments. It can be seen from the top panel that no numerical instability develops, but, it can be seen in the bottom panel that the passive tracer is much more horizontally homogeneous than, for instance, the `ctrl1` experiment (Figure 11.5). The reasons are discussed in text.

### 11.2.4 Blob Speed

Two factors dominate the mean speed of blobs, being, the interfacial drag (controlled in part by the Rayleigh drag coefficient,  $\alpha$ ) and the value of the pseudo-non-hydrostatic term, which is dependent on the density difference between the blob and the surrounding E system. Figure 11.12 shows the evolution of the mean speed of blobs in a selection of experiments. Related to the mean depth of blobs, discussed in Section 11.2.3, there are sharp changes in blob speed when a numerical instability develops. This is most obvious in the `ctrl` experiment where the mean blob speed peaks at almost  $6\text{ m s}^{-1}$  (not shown), which is clearly unphysical. Several other experiments also exhibit a large increase in blob speed, which all coincide with the development of numerical instability. All experiments, prior to developing a numerical instability, have mean speeds that are  $O(0.01)\text{ m s}^{-1}$  or smaller, with the exception of the `c5e-5` experiment which has a mean blob speed of approximately  $0.1\text{ m s}^{-1}$  prior to the onset of instability. The `c5e-2` experiment, on the other hand has a large value for  $\alpha$  and it has a very small mean velocity, with the curve not being discernible from the bottom axis for most of the duration of the experiment. Thus, it can be seen that  $\alpha$  has a direct impact on the mean speed of blobs.

In terms of the horizontal speed relative to the vertical speed, the `ctrl` experiment's blob speed (prior to the onset of numerical instability) is dominated by the vertical speed (not shown). The ratio of horizontal speed to vertical speed is typically of order  $0.01--0.1$ .

### 11.2.5 Blob Mass and Density

For most experiments, the average mass of a blob is quite consistent throughout the experiment, as can be seen in Figure 11.13. In contrast to the bottom blobs, the initial size of free blobs (when formed by vertical instability) are not dependent on the magnitude of the static instability (see Section 5.2.1). This means that there is less variation in the initial size of blobs. Furthermore, the deep convection test case has uniform initial grid mass throughout the domain. Therefore, all blobs for a given initial blobs size parameter,  $\Lambda$ , have approximately the same initial mass. This explains why all of the experiments shown have the same mean blob mass when the experiments first start producing blobs, except for the `L05` experiment (which produces smaller blobs) and the `L25` experiment (which produces larger blobs).

After the first blobs are formed, the mean blob mass of the experiments changes. The way that a blob changes mass is via detrainment. It can be seen that blobs with a small detrainment parameter, like `G1e-7`, have relatively massive blobs while experiments with a large detrainment parameter, like `G1e-4`, have relatively small blobs.

There is very little variation in blob density (not shown) since the free blobs do not entrain, and, therefore their density remains fixed throughout their lifetime (since a linear equation of state for seawater has been used). The only variation that occurs is the small changes in density that occur in the regions of formation, which, are quite minor given

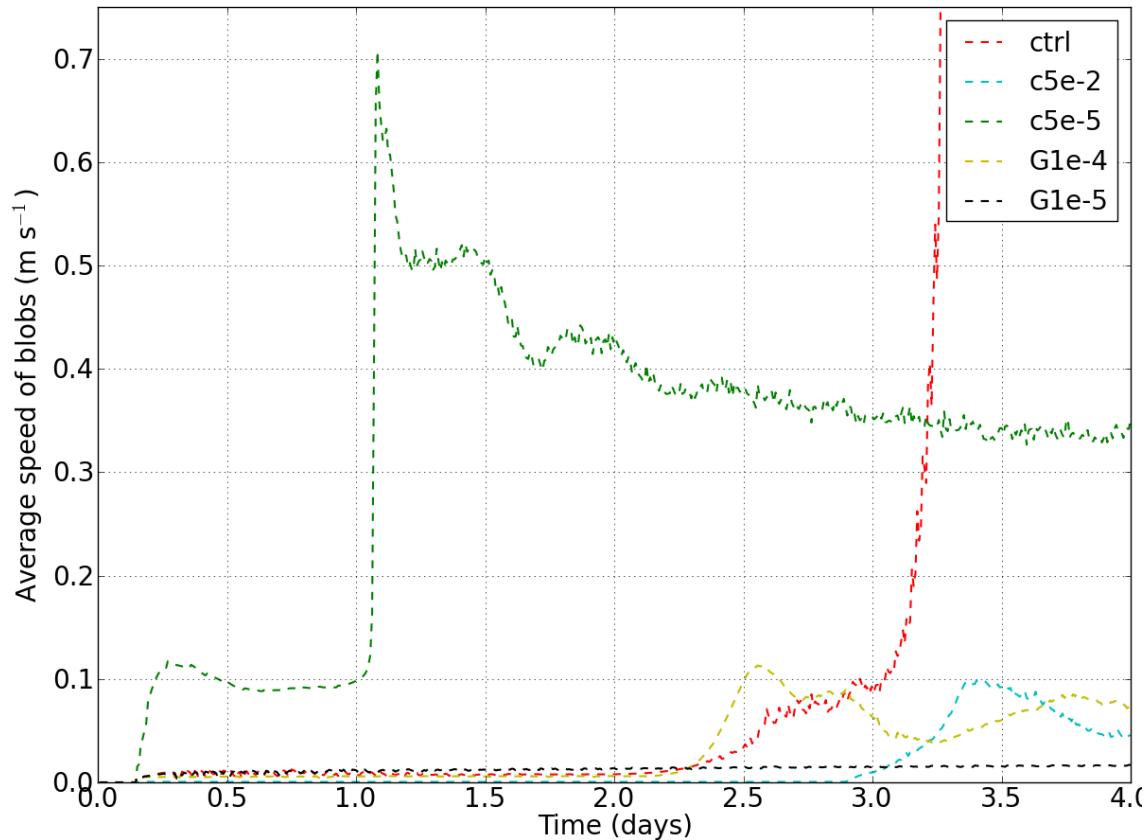


Figure 11.12: The mean speed of blobs. Most experiments have blob velocities that are  $O(0.01)\text{m s}^{-1}$  or smaller, unless they develop a numerical instability. In some instances where an experiment develops a numerical instability, such as the `ctrl` experiment, the mean velocity can exceed  $6\text{m s}^{-1}$ . Instabilities aside, the value for  $\alpha$  has a large effect on the mean speed of the blobs, with small values making for fast blobs and large values making for slow blobs.

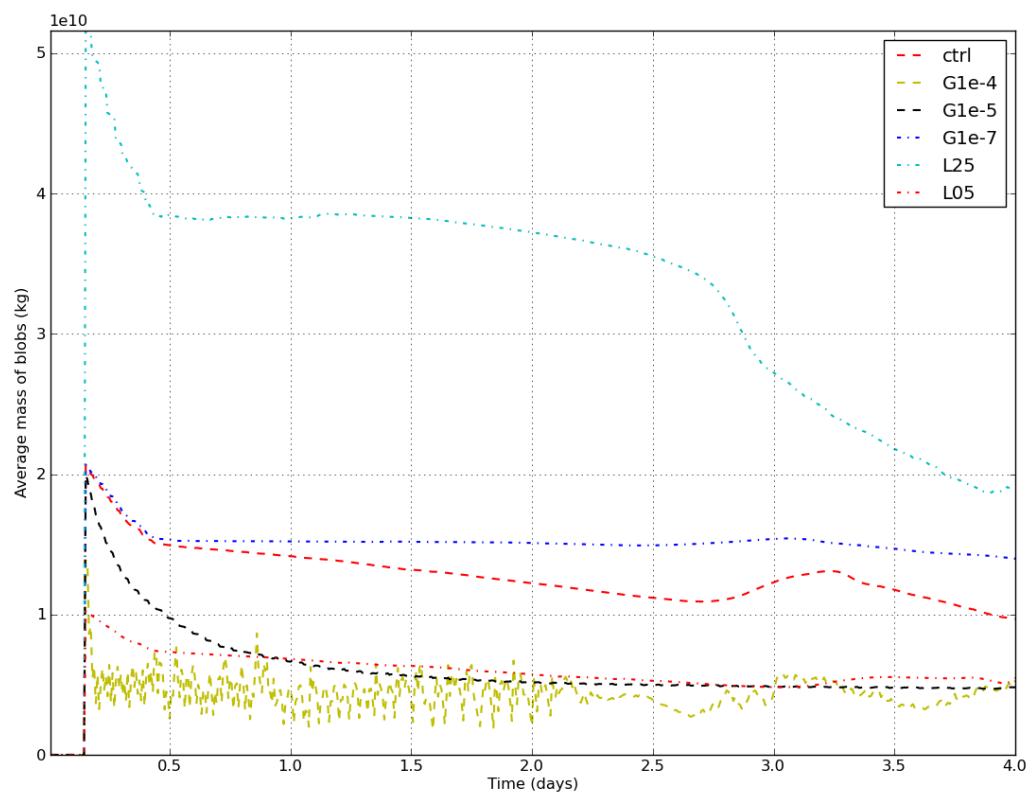


Figure 11.13: The mean mass of blobs. The scaling parameter for the initial size of a blob,  $\Lambda$  is a large influence on the initial mass of blobs, while the detrainment parameter,  $\Gamma$ , is a large influence on the longer term mass of a blob.

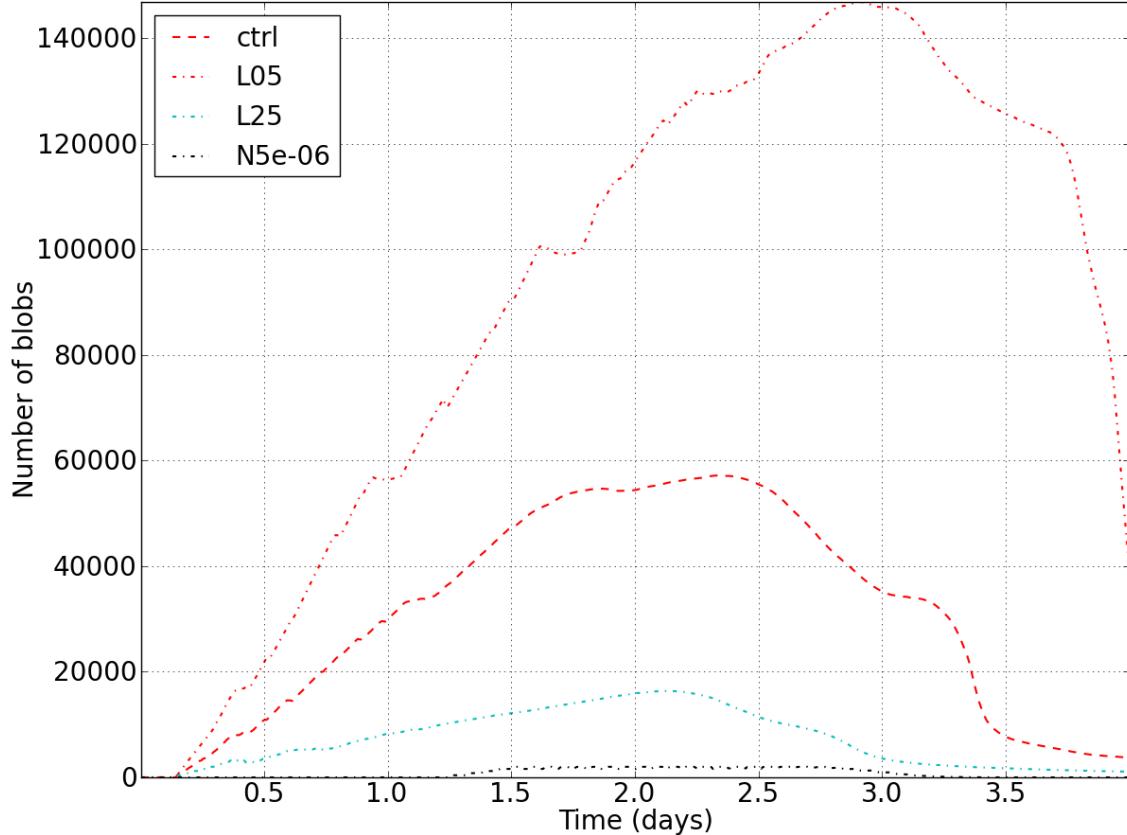


Figure 11.14: The evolution of the number of blobs in various simulations. For comparison, the number of grid cells in the Eulerian system is 8000. Thus, in most experiments there are significantly more Lagrangian blobs than there are Eulerian grid cells. There are various factors, which are discussed in the text, as to the effect of various parameters on the number of blobs.

the weakly stratified nature of this test case.

### 11.2.6 Number of Blobs

The number of blobs in each experiment at any given time is shown in Figure 11.14. As a means of comparison, the total number of grid cells for the coarse resolution experiments is 8000. The number of blobs in most experiments greatly exceeds the number of grid cells, in some instances by more than an order of magnitude.

The time evolution of the average age of blobs in various experiments is shown in Figure 11.15. The age of blobs depends on the how long it is from creation to destruction. One way that a blob can be destroyed is to fully detrain its properties to the E system. This is largely controlled by the detrainment parameter,  $\Gamma$ , as can be seen in Equation (5.13). To test the sensitivity of the simulation to  $\Gamma$ , the experiments G1e-4, G1e-5, ctrl, G1e-7 and G1e-8 were run (in order of decreasing  $\Gamma$ ). The results of G1e-7 and G1e-8 are very similar and thus G1e-8 is not shown. The results of G1e-7 and ctrl are quite similar, indicating that the largest sensitivity to the parameter is for values greater than  $1 \times 10^{-6} \text{ kg m}^{-2} \text{s}^{-1}$ .

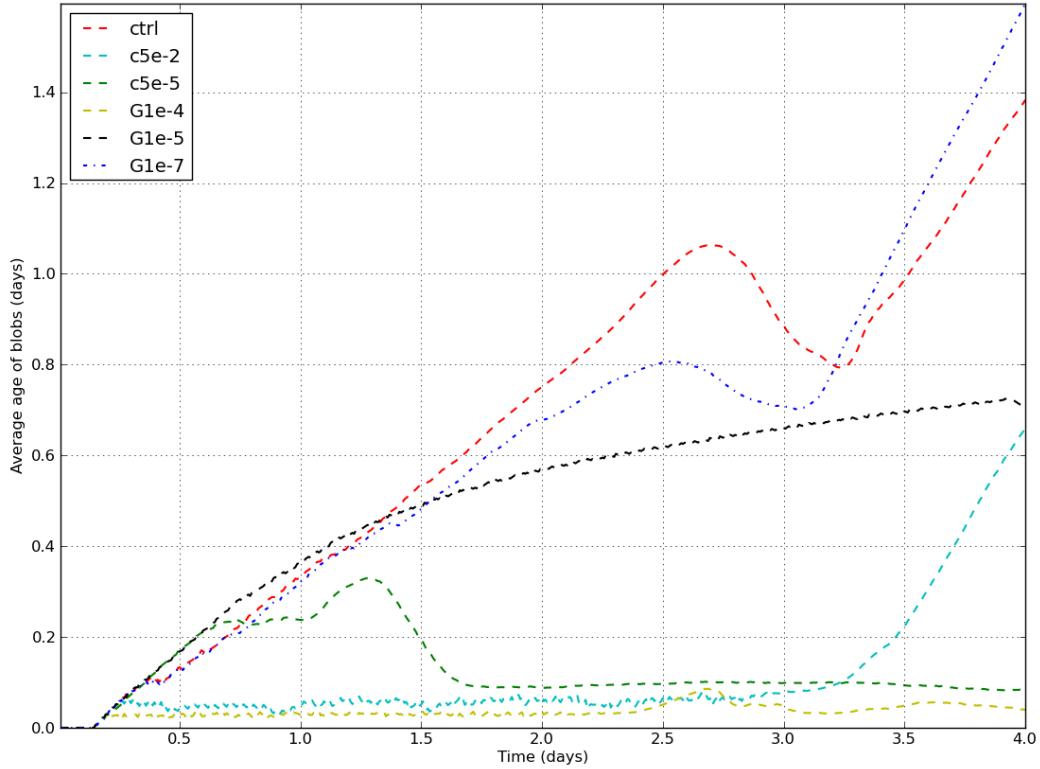


Figure 11.15: The evolution of the mean age of blobs (days) in various simulations. The larger the value of the detrainment parameter,  $\Gamma$ , the shorter amount of time that the blobs tend to exist for.

The **G1e-4** experiment has the youngest blobs throughout the simulation, indicating that the blobs in that experiment detrain rapidly.

The number of blobs in the **ctrl** and **G1e-5** experiments are similar until about 2.5 days into the experiment. This is about the same time as when the numerical instability, discussed in Section 11.2.3, in the **ctrl** experiment begins to develop. The number of blobs in the **ctrl** experiment drops rapidly, due to many blobs being destroyed by the grid cell mass constraint being violated. This eventually leads to numerical instability in the E system to develop. The average age of blobs in the **ctrl** experiment increases and the number of blobs decreases. This indicates that it is largely younger blobs being destroyed, which adds weight to the assertion that it is blobs near the surface (i.e. close to their creation site, and, therefore young) that are being destroyed.

The **c5e-2** experiment, which has a high Rayleigh drag coefficient, has a relatively small number of blobs throughout the experiment. The reason for this is that many blobs are destroyed (not shown) due to the grid cell mass constraint being violated frequently. Their relatively low speed (see Section 11.2.4) means that they tend to bunch up near the region of creation, violating the grid cell mass constraint. The destruction of a large

number of blobs near their formation site means that the Lagrangian scheme is not able to effectively transport material downwards. This contributes to the slow increase in depth of the tracer, as shown in Figure 11.6. The **c5e-5** experiment also has a relatively low number of blobs, but, for different reasons. The decrease in the number of blobs in the **c5e-5** experiment that occurs after about one day coincides with an increase in the number of blobs per timestep that are interacting with the bottom (and thus, being destroyed). This also coincides with the development of numerical instability (see Section 11.2.2) and a dramatic increase in the amount of tracer near the bottom (see Figure 11.7). The rate at which blobs are destroyed for this reason is reasonably constant throughout the rest of the experiment (not shown) and thus, helps to keep the number of blobs in the experiment relatively low.

The **ctrl** experiment on the other hand has less blobs that are destroyed by interacting with the bottom than the **c5e-5** and has less blobs that are destroyed by the grid cell mass constraint than the **c5e-2** experiment. Thus, the **ctrl** experiment mostly has a larger number of blobs than either experiment.

The choice of the blob scaling parameter,  $\Lambda$ , appears to have a large impact on the number of blobs given the large difference in the number of blobs in each experiment in the **L25**, **ctrl** and **L05** experiments. The **L25** experiment has a relatively low number of blobs throughout. This is because it forms relatively large blobs (ones that are 2.5 times larger than that of the **ctrl** experiment) and as a result, the grid cell mass constraint is frequently violated, causing blobs to be destroyed. The small blobs formed by the **L05** experiment means that the grid cell mass constraint is not violated very frequently, allowing many blobs to be in existence. In addition, the small mass of the blobs means that more blobs are required to be formed in order to move the same properties as a larger blob, thereby requiring more blobs to remove the static instability. As is seen from Table 11.3, the large number of blobs in the **L05** experiment increases the computational time required by the experiment. One thing that is interesting to note is that the experiment with the larger blobs, **L25**, has a rapid homogenisation of the water column similar to that exhibited by the **ctrl** experiment in Figure 11.5, except it occurs before the beginning of the third day (not shown). The **L05** experiment, on the other hand, does not exhibit this homogenisation, except at the end of the run, as can be seen in Figure 11.16. This indicates that the smaller sized, greater number of blobs leads to a more stable solution.

It was mentioned previously (Section 11.2.2) that the **N5e-9** behaves more similar to a convective adjustment scheme because it creates a very large number of blobs which then causes the grid cell mass constraint to be violated frequently, and, the blobs properties to be returned to the E system very soon after blob creation. By the end of four days the **N5e-9** experiment is producing and destroying approximately 4000 blobs per timestep. This is in stark contrast to all other experiments that produce and destroy  $O(1)$  to  $O(10)$  blobs per timestep (not shown). This also helps to explain the significant compute time

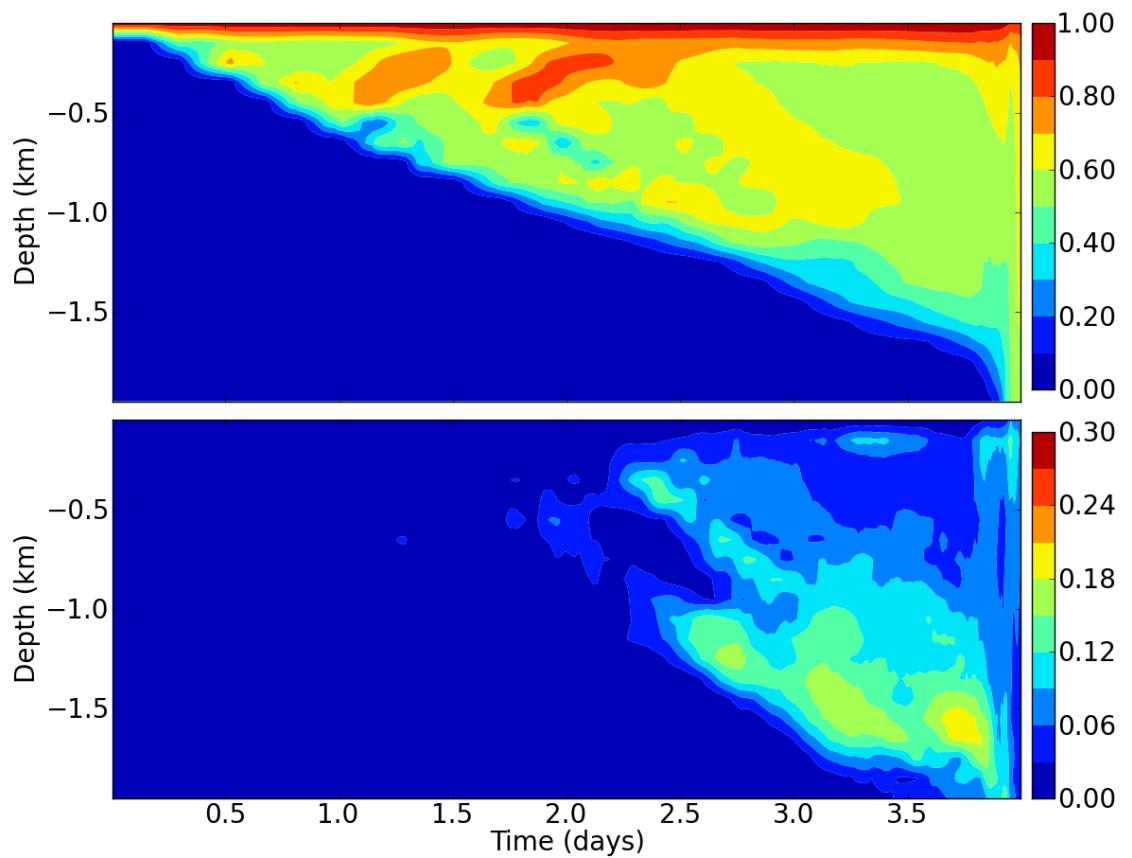


Figure 11.16: The mean (top panel) and standard deviation (bottom panel) of the passive dye tracer in the L05 experiment, which has a small blob scaling parameter, producing blobs that have a small initial mass. Having smaller, but more numerous blobs appears to improve the stability of the simulation.

Experiment	Model	CPU-seconds	Comments
NH	MITgcm	$2.27 \times 10^6$	Non-hydrostatic, fine resolution
HYD_FR	MITgcm	$1.36 \times 10^5$	Hydrostatic, fine resolution
HYD	MITgcm	$8.23 \times 10^3$	Hydrostatic, coarse resolution
noblob	MOM	$7.76 \times 10^3$	Convective adjustment
ctrl	MOM	$1.98 \times 10^4$	Uses blobs
c5e-2	MOM	$1.32 \times 10^4$	Uses blobs
c5e-5	MOM	$1.12 \times 10^4$	Uses blobs
G1e-4	MOM	$8.72 \times 10^3$	Uses blobs
G1e-5	MOM	$2.45 \times 10^4$	Uses blobs
G1e-7	MOM	$1.81 \times 10^4$	Uses blobs
L25	MOM	$1.11 \times 10^4$	Uses blobs
L05	MOM	$4.11 \times 10^4$	Uses blobs
N5e-5	MOM	$7.70 \times 10^3$	Uses blobs
N5e-6	MOM	$7.29 \times 10^3$	Uses blobs
N5e-9	MOM	$7.82 \times 10^4$	Uses blobs

Table 11.3: The compute time required for each 4-day experiment. Given the relatively large number of blobs in most experiments, compared to the number of Eulerian grid cells (which is equal to 8000 in the coarse resolution configuration), means that the additional compute time required to run the Lagrangian blobs is by and large proportional to the number of blobs.

required by the experiment (Table 11.3). In contrast to the N5e-9 experiment, the N5e-5 experiment does not produce any blobs at all (not shown), and, it takes over one day before the N5e-6 experiment produces an appreciable number of blobs. The relatively small number of blobs produced by the N5e-6 experiment is because of the high threshold before a blob can be formed. This leads to instabilities, which have been documented in earlier sections.

### 11.2.7 Compute Time

The compute time for each experiment is shown in Table 11.3. All experiments were run for four model days. All experiments were run on four cores, except for the NH and HYD\_FR experiments, which were run on 16 cores and the HYD experiment, which was run on two cores. The code was run on the Sun Constellation cluster, Vayu machine at the Australian National Computational Infrastructure National Facility. The Sun Constellation cluster has 1492 nodes in Sun X6275 blades. The code was compiled using the Intel Fortran Compiler version 12.0.4.191. Because of the two different models in use, the way that the compute time is calculated and presented in Table 11.3 is different to that in the DOME (Chapter 12) or Bowl (Chapter 13) test cases. To remain consistent in calculating the timing between the two models, the elapsed time for each experiment was taken and multiplied by the number of cores to get the total compute time in seconds.

As can be seen, the fine resolution cases are significantly more expensive than the coarse resolution cases. On top of that, the NH experiment, which uses non-hydrostatic dynamics takes almost 20 times longer than the HYD\_FR experiment, which uses the hydro-

static approximation. This indicates, unsurprisingly, that the hydrostatic approximation increases the speed of simulations significantly since the equations being solved are more simple. The `HYD` experiment takes approximately two orders of magnitude less time than the `HYD_FR` experiment, however, this is to be expected since the `HYD` experiment has 100 times fewer grid points than the `HYD_FR` experiment. The `HYD` experiment, which is run in the MITgcm, and the `noblob` experiment, which is run in the MOM, have approximately the same computing requirements, which is to be expected as they have the same resolution and employ many of the same approximations and parameterisations.

One thing that is important to note is that the blob experiments were run with an E system timestep of 5s. It was found that timesteps larger than this can lead to catastrophic numerical instability in some experiments. On the other hand, experiments like the `NH` experiment and `HYD_FR` experiment are able to run with a timestep as large as 60s (which was the value used by Campin et al., 2011). The coarse resolution, hydrostatic experiments can run with timesteps even larger. So, there is scope for the experiments that do not use the Lagrangian blob scheme to take less computational resources, while there is little scope beyond efficiencies to be gained from code alterations (see Section 8.2.7) to decrease the computational requirements of the Lagrangian blob experiments.

The use of the Lagrangian blob scheme increases the amount of compute time required significantly. For instance, the `ctrl` experiment takes more than 2.5 times the CPU time to run the simulation than the `noblob` experiment, which uses traditional convective adjustment. This represents a significant relative increase in cost, and, is unsurprising given that the number of blobs in most experiments significantly outnumber the grid cells, as is shown in Figure 11.14.

Of the blob experiments, it is unsurprising that the most expensive are the `L05` and `N5e-9` experiments since these are the experiments that produce the most number of blobs (see Section 11.2.6). That exception aside, the time taken for blob experiments, relative to one another, correlates well to the number of blobs in Figure 11.14. This is because the overhead of the infrastructure required to admit the embedded Lagrangian model is relatively small in comparison to the requirements to evolve the properties of all of the individual blobs. This is in contrast to the findings in later chapters where this is not always the case, due to the much lower number of blobs relative to the number of grid cells in the domain.

## 11.3 Summary

The admission of the pseudo-non-hydrostatic dynamics of the embedded Lagrangian scheme greatly affects the nature of the solution of the deep convection test case in MOM. When looking at the evolution of a passive tracer directly under the cooling disc, which is restored at the surface, the solution appears to be a great step toward emulating the non-hydrostatic results in a run using the MITgcm. One caveat is that the parameter choices made in the

Lagrangian blob scheme greatly affect the solution, as outlined below.

### 11.3.1 The Lagrangian Blobs

These results indicate that, with the right parameter choices, the Lagrangian blob scheme is a cheaper, potentially viable alternative to the use of a fully non-hydrostatic model in process studies in which non-hydrostatic processes, such as open ocean deep convection, are important. The utility of the Lagrangian blob scheme for more coarse resolution experiments, however, is unclear from the results presented. Send and Marshall (1995) indicate that the effect of convective adjustment captures the integral effects of open ocean deep convection. Hughes et al. (2009) however caution that the energetically incomplete representation of deep convection through the traditional convective adjustment process may have implications for the meridional overturning circulation. A worthwhile extension of the present work would be to investigate the effect of the embedded Lagrangian scheme on the overturning circulation to assess what utility the scheme may have in global scale ocean modelling.

Numerical instability, resulting in rapid mixing of the entire water column is a feature that occurs in a number of experiments using the Lagrangian blobs, including the `ctrl` experiment (see Figure 11.5). The `G1e-5` experiment, which does not exhibit any numerical instability, has a much less variable horizontal distribution of passive tracer under the cooling disc (see bottom panel of Figure 11.11). One of the characteristics of the non-hydrostatic `NH` experiment (which is being used as a “ground truth” in this test case) is the horizontal variability of the passive tracer under the cooling disc. It would appear that in the experiments that use the Lagrangian blobs, the numerical instability and rapid mixing of the water column is preceded by an increase in the horizontal variability of the passive tracer. It is clear that the `G1e-5` experiment is an improvement over the `noblob` experiment (Figure 11.4), which uses convective adjustment to stabilise the water column. It is less clear, however, whether experiments that exhibit this numerical instability (and therefore, increased horizontal variability in the passive tracer) are a further improvement or are a degradation of the simulation. Further investigation as to the representation of non-hydrostatic dynamics through the pseudo-non-hydrostatic term is warranted to assess whether this approximation is a contributing factor to the numerical instability that develops.

### Interfacial Friction

The main effect of the Rayleigh drag coefficient,  $\alpha$ , is to control how fast the blobs move through the water column. A large drag coefficient means that blobs move slowly. Under conditions where there is strong forcing, as in the deep convection test case, many blobs are formed. But, when the blobs are travelling slowly, they tend to bunch up near the site of formation. This bunching up causes the grid cell mass constraint to be frequently violated (see Section 6.5.3) and inhibits the ability of the Lagrangian scheme to stabilise

the water column. On the other hand, having a value for  $\alpha$  which is small means the blobs travel through the water column rapidly and do not interact or detrain very much with the E system before interacting with the bottom. In this test case, the blobs are destroyed once they interact with the bottom. Thus, in the case where the value of the friction coefficient is small, blobs mostly deposit their properties near the bottom, which causes properties to be transported rapidly from the surface to the bottom, with very little mixing in between (see Figure 11.7). Thus, an intermediate value for  $\alpha$  must be chosen.

### **Detrainment Rate**

The direct effect of the detrainment parameter,  $\Gamma$ , is to alter the rapidity at which a blob detrains. This affects the length of time that a blob is in existence for. In the deep convection test case, a large detrainment rate gives the blobs a short lifetime. Having a detrainment rate that is too large can mean that blobs are fully detrained before they have had the chance to transport properties into the deep ocean. This inhibits the Lagrangian scheme from restabilising the water column and eventually leads to the growth of the static instability and the onset of numerical instability.

On the other hand, having a very small detrainment parameter causes the blobs to not detract and they largely maintain their mass. Under strong forcing, a large number of blobs can be created and thus a large proportion of the mass (particularly near the surface) can be in the Lagrangian system. This causes the grid cell mass constraint to be violated (see Section 6.5.3) frequently. If this happens enough, it can be difficult for the blobs to get far from their region of creation, since they are destroyed before being able to transport their properties deep enough to maintain the static stability of the water column. Again, if this occurs and the static instability grows large enough, it can lead to numerical instabilities developing in the Eulerian model.

### **Initial Size**

The blob scaling parameter,  $\Lambda$ , sets the initial size of a blob. This affects the mass of the blobs throughout the simulation. It also affects the number of blobs that the simulation creates. Larger blobs remove the static instability quicker, however, having a large initial mass also increases the propensity for the simulation to develop numerical instability (this is due to the grid cell mass constraint being violated more easily and, therefore, frequently). On the other hand, creating smaller blobs reduces the propensity for the grid cell mass constraint to be violated (and makes the simulation less susceptible to numerical instability), but, the creation of more blobs is required in order to achieve the same end in terms of restabilising the water column. The creation of more blobs requires more computational resources.



## Chapter 12

# The DOME Test Case

The dynamics of overflow mixing and entrainment (DOME) test case considered here uses a similar experimental configuration to previous DOME studies (for example, Legg et al., 2006). The DOME test case was originally conceived to examine plume dynamics on a slope with a realistic gradient. A coarse resolution ( $1/2^\circ$ ) configuration is used to primarily explore the parameter space of the bottom blob implementation, however, the effects of friction schemes and other overflow parameterisations are also examined. The resolution of the setup is chosen as it is representative of the current generation of global scale ocean climate models.

It is known that coarse resolution level coordinate models have difficulty transporting material down slope (Winton et al., 1998; Legg et al., 2006). It is also known that level models tends to be too diffusive, and, this affects the along slope velocity of the plume. To begin with, a suite of experiments is conducted to examine how various parameterisations in a purely Eulerian model affect the properties of the plume. Then, the dynamic blob scheme is introduced and various parameters of the scheme are varied to examine how they influence the plume properties. The parameters that are tested are

- the initial blob size,
- the coefficient of drag,  $C_d$
- the detrainment parameter,  $\Gamma$ , and
- the formation condition threshold,  $\Delta\rho$ .

### 12.1 Configuration Details

The domain for the DOME configuration has a maximum depth of 3600m. The domain extends approximately 2000km zonally and approximately 750km meridionally. To the north of the domain there is an embayment of depth  $h_e = 600\text{m}$  approximately  $2/3$  the way along from the western wall. From the embayment, there is a uniform slope of  $0.01\text{m/m}$  until the maximum depth. The topography of the domain is shown in Figure 12.1. The domain has solid wall boundaries with sponge boundary conditions for tracers on the western and

eastern walls. The sponge extends out from the solid walls by three grid cells, with a restoring timescale of 10, 25 and 50 days (with the strongest restoring being closest to the wall and the weakest being furthest from the wall). A constant quadratic bottom drag for the E system is used, with a dimensionless bottom drag coefficient of  $1 \times 10^{-3}$  (unless otherwise stated).

The model uses a constant value for the Coriolis parameter of  $f_0 = 10^{-4}\text{s}^{-1}$ , which is the value at approximately 43°N. The grid cells have a horizontal aspect ratio of 1 : 1. The timestep for all experiments is 3600s (although, as discussed in Section 7.3.2, the timestep for individual blobs is variable). The domain is initially uniformly stratified with a buoyancy frequency of  $N = 2.3 \times 10^{-3}\text{s}^{-1}$ , and the initial temperature at the surface is 20°C.

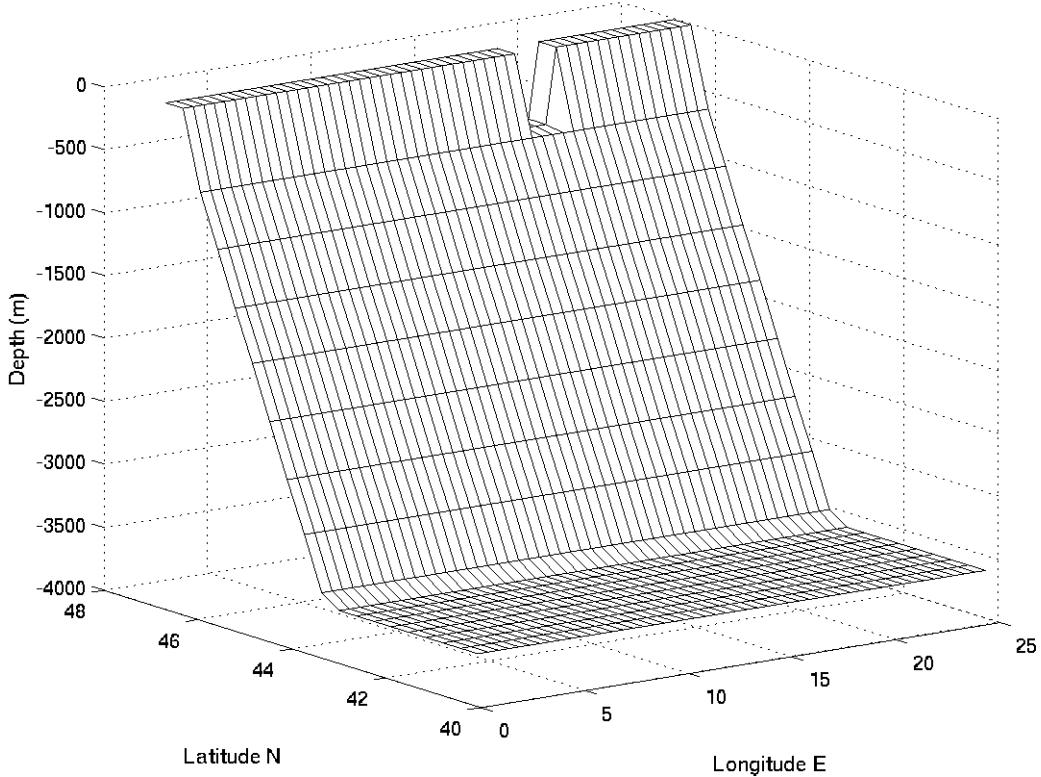


Figure 12.1: A wire-frame of the topography of the DOME test case. The model uses no slip side wall boundary conditions (the side walls are not depicted here).

In a departure from previous DOME studies, there is no mass flux at the northern end of the embayment, but, water is set to a constant temperature. Thus the plume is purely density driven, and, there is some recirculation required from the bulk ocean into the embayment to balance the cold water leaving. The reason why there is no mass flux prescribed is because we were unable to provide a mass flux through the northern

boundary on the Arakawa (1966) B grid of MOM without inducing significant instabilities.

In this experiment, density is a linear function of temperature only. A passive dye tracer is used to tag the inflowing water. The dye is initialised to zero throughout the domain, but is set to one in the injected waters. Temperature and the dye are damped to initial conditions at the eastern and western boundaries by the sponge boundary conditions.

The inflow at the northern boundary of the embayment is defined, respectively, by its buoyancy anomaly and height,

$$b_I(x_w, z) = b_0 - \Delta b_0 \quad (12.1a)$$

$$h_I(x_w) = h_0 \exp(-x_w/L_\rho) \quad (12.1b)$$

where  $x_w$  is the distance (in metres) from the western edge of the embayment. Also,

$$L_\rho = \frac{\sqrt{h_0 \Delta b_0}}{f} \quad (12.2)$$

is the deformation radius,  $h_0$  is the height scale of the inflow,  $b_0$  is the initial buoyancy at the surface and  $\Delta b_0$  is the buoyancy anomaly of the inflow relative to the initial conditions (a value that is constant and chosen *a priori*).

Legg et al. (2006) have a Froude number condition in their formulation for the inflow buoyancy and velocity to ensure that there is no spurious mixing by Kelvin waves. This condition is not included here since the sponge boundary conditions will damp out any induced Kelvin wave (Chapter 33 of Griffies, 2009) and should achieve the same end as the Froude number condition.

It is noted that  $h_I$  is a continuous function that will not necessarily coincide with grid cells. A condition is thus required for the exclusion or inclusion of a grid cell for which  $h_I$  passes through. A value of  $h_0 = 380\text{m}$  is chosen and the condition is that a grid cell is included in the inflow if  $h_I$  passes anywhere through the grid cell.

No experiments use explicit lateral prescribed diffusion, with stability being ensured through numerical diffusion via the tracer advection scheme. The tracer advection scheme used is in both the vertical and horizontal is the multi-dimensional piecewise parabolic method of Colella and Woodward (1984) with the flux limiter described by Lin (2004). Partial bottom cells are used with the thinnest partial cell allowable being 20% of the thickness of a full cell. A constant vertical diffusivity of  $1.0 \times 10^{-5}\text{m}^2\text{s}^{-1}$  is used. An enhanced vertical diffusivity of up to  $10.0\text{m}^2\text{s}^{-1}$  is used in gravitationally unstable regions, unless otherwise stated.

The configuration is a  $50(x) \times 19(y) \times 50(z)$  grid, which is a resolution of approximately

40km (which equates to roughly  $1/2^\circ$  at  $43^\circ\text{N}$ ). The grid cells increase linearly from 20m in thickness at the surface to 124m in thickness at the bottom. A constant viscosity of  $500.0\text{m}^2\text{s}^{-1}$  is used in the horizontal unless otherwise stated while a constant viscosity of  $0.1\text{m}^2\text{s}^{-1}$  is used for all experiments in the vertical.

## 12.2 Analysis Techniques

Since the DOME test case is concerned with plume dynamics, the behaviour of simulations after the plume front has interacted with the western wall is largely ignored as the wall then begins to influence the behaviour of the plume. There are other, more appropriate, experimental designs for the examination of plume dynamics interacting with topography (e.g. Ilicak et al., 2011b).

Following the convention of Legg et al. (2006), we define the plume to be the region where  $\tau > 0.01$ , and the bulk ocean to be where  $\tau \leq 0.01$ , where  $\tau = \tau(t, x, y, z)$  is the calculated value for the passive dye tracer. This definition facilitates calculation of the total mass of the plume

$$M_P(t) = \iiint_P \rho \, dx \, dy \, dz. \quad (12.3)$$

The mean plume path can be defined in the following way,

$$Y(t, x) = \frac{\iint y \tau(t, x, y, z) \, dy \, dz}{\iint \tau(t, x, y, z) \, dy \, dz}. \quad (12.4)$$

Note, that for the mean path to exist, it is required that  $\tau > 0.01$  somewhere along a surface of constant  $x$  (for a given  $t$ ), otherwise the path is said to not exist for those values of  $x$  and  $t$ .

Similarly, a mean plume depth can be defined

$$Z^{(x)}(t, x) = \frac{\iint z \tau(t, x, y, z) \, dy \, dz}{\iint \tau(t, x, y, z) \, dy \, dz} \quad (12.5)$$

where it is again recognised that if there is no value of  $\tau > 0.01$  for a surface of constant  $(t, x)$ , then the path is said to not exist.

Another useful quantity is the Nof (1983) speed, defined as,

$$U_N = \frac{\theta g'}{f}. \quad (12.6)$$

The Nof speed is the along slope speed of an advancing plume in the absence of friction. In the experiments presented  $\theta = 0.01\text{m/m}$  is the slope and  $f = 10^{-4}\text{s}^{-1}$  is the Coriolis parameter; both of which are fixed. Thus, the Nof speed only depends on the reduced

gravity,  $g'$ .

The reduced gravity is defined as

$$g' = g \frac{\rho_B - \rho_P}{\rho_B} \quad (12.7)$$

where  $\rho_P$  is a measure of the density of the plume and  $\rho_B$  is a measure of the density of the bulk ocean. A measure of the density of the plume at a given level can be found by taking the mass of grid cells where the passive tracer concentration is greater than 0.01, and dividing it by the volume of the grid cells

$$\rho_P^k = \frac{\sum_{i,j} dV^{ijk} \rho^{ijk}}{\sum_{i,j} dV^{ijk}} \quad \forall \quad \tau^{ijk} > 0.01. \quad (12.8)$$

Similarly, the density of the bulk ocean can be found

$$\rho_B^k = \frac{\sum_{i,j} dV^{ijk} \rho^{ijk}}{\sum_{i,j} dV^{ijk}} \quad \forall \quad \tau^{ijk} \leq 0.01. \quad (12.9)$$

The total measure of the reduced gravity can be found by thickness weighting the right hand side of Equation (12.7) and summing over all levels

$$g' = g \left( \sum_k dz^k \frac{\rho_B^k - \rho_P^k}{\rho_B^k} \right) \left( \sum_k dz^k \right)^{-1}, \quad (12.10)$$

where  $dz^k$  is the average depth of the  $k^{\text{th}}$  level. We note that unless the plume occupies every single grid cell in the domain,  $\rho_B$  is never 0, and thus  $g'$  is always finite.

## 12.3 Experimental Details

All of the experiments conducted using the DOME test case are described in this section. The baseline experiment is the `noblob` experiment and is, in the traditional sense of the term, the “control” experiment. However here the “`ctrl`” experiment label is assigned to the control experiment incorporating blobs. The experiment takes the mid-range of sensible parameter choices from which the values are varied in other blob experiments. The parameter values in the `ctrl` experiment are not chosen for being the optimal or best parameter values.

A number of experiments are conducted, and, all of them are described in Tables 12.1 and 12.2. All of the experiments listed in Table 12.1 are designed to test the representation of the plume with different parameterisations that affect downslope flow (either directly or indirectly). Table 12.2 lists experiments that use the Lagrangian blob scheme. These experiments are designed to examine how the properties of the plume change with different parameter values.

Name	Viscosity	Comment
<code>noblob</code>	Constant $500\text{m}^2\text{s}^{-1}$	
<code>hdrag</code>	Constant $500\text{m}^2\text{s}^{-1}$	High bottom drag ( $1 \times 10^{-2}$ )
<code>gm</code>	Constant $500\text{m}^2\text{s}^{-1}$	Gent and McWilliams (1990) neutral physics
<code>ksmag2</code>	Lap Smag <code>ksmag</code> = 2.0	MICOM velocity scale=0.1
<code>ksmag3</code>	Lap Smag <code>ksmag</code> = 3.0	MICOM velocity scale=0.1
<code>nomicom</code>	Lap Smag <code>ksmag</code> = 2.0	No background friction
<code>BD97</code>	Constant $500\text{m}^2\text{s}^{-1}$	Beckmann and Dösscher (1997) overflow param.
<code>CG99</code>	Constant $500\text{m}^2\text{s}^{-1}$	Campin and Goosse (1999) overflow param.
<code>NoReturn</code>	Constant $500\text{m}^2\text{s}^{-1}$	No return overflow
<code>ctrl</code>	Constant $500\text{m}^2\text{s}^{-1}$	Dynamically active bottom blobs
<code>free</code>	Constant $500\text{m}^2\text{s}^{-1}$	Dynamically active free and bottom blobs

Table 12.1: Experimental details for the DOME test case for all experiments not using dynamically active blobs and some that do (see Table 12.2 for a description of all experiments using blobs). Lap Smag means Laplacian Smagorinsky friction. The experiments that use the Laplacian Smagorinsky friction scheme also use the isotropic MICOM velocity scale for the background friction, except for the `nomicom` experiment.

### 12.3.1 Friction

The `noblob` experiment uses a constant horizontal viscosity with a value of  $500\text{m}^2\text{s}^{-1}$ . It has recently been suggested that friction schemes play an important role in diapycnal mixing in overflows (Ilicak et al., 2011a). Here, the effect of friction is investigated by introducing the Smagorinsky Laplacian viscosity (Smagorinsky, 1963; Griffies and Hallberg, 2000). The Smagorinsky Laplacian viscosity is defined according to

$$A_{SM} = (C_s \Delta s)^2 |E| \quad (12.11)$$

where  $C_s$  is the non-dimensional Smagorinsky coefficient,  $\Delta s$  is the horizontal grid spacing and  $E$  is the total deformation rate with units of inverse time. In addition to the Smagorinsky viscosity, a background viscosity may also be defined where

$$A_{min} = U_A \Delta s \quad (12.12)$$

where  $U_A$  is a constant velocity scale. In MOM, a background velocity scale (originating from studies conducted by MICOM) suggests a velocity scale of  $0.1\text{m s}^{-1}$ .

There are three experiments that use the Smagorinsky viscosity. The `ksmag2` experiment uses a Laplacian Smagorinsky viscosity with a dimensionless parameter of 2.0 and a MICOM velocity scale of  $0.1\text{m s}^{-1}$ . The `ksmag3` experiment is identical to the `ksmag2` experiment, except the Smagorinsky dimensionless parameter has a value of 3.0. The `nomicom` experiment is identical to the `ksmag2` experiment, except the background velocity scale is set to zero.

### 12.3.2 Lateral Diffusion and Bottom Drag

There is also an experiment that uses the neutral physics scheme of Gent and McWilliams (1990); Gent et al. (1995), with the main purpose of this experiment to compare the computational cost of the blobs to an existing, widely used parameterisation. The **gm** experiment is identical to the **noblob** experiment, except it has a constant coefficient of  $\kappa_{GM} = 600\text{m}^2\text{s}^{-1}$  and a constant isoneutral diffusion coefficient of  $\kappa_{iso} = 600\text{m}^2\text{s}^{-1}$  with linear tapering near the surface.

The **noblob** experiment specifies a non-dimensional coefficient of bottom drag of  $1 \times 10^{-3}$ . The **hdrag** experiment is identical to the **noblob** experiment except it specifies a larger non-dimensional coefficient of bottom drag of  $1 \times 10^{-2}$ .

### 12.3.3 Traditional Overflow Schemes

A number of experiments are run in which the DOME test case is used to evaluate the fidelity of some existing overflow parameterisations. A summary of traditional bottom boundary layer schemes is given in Section 2.4.

The **noblob** experiment uses no overflow parameterisations. The **BD97** experiment is identical to the **noblob** experiment, except it uses the sigma scheme of Beckmann and Döscher (1997). It is worthwhile noting that Döscher and Beckmann (2000) found that the advective component of the scheme to not be critical in coarse models and as such, only the diffusive aspect of the Beckmann and Döscher (1997) scheme is implemented in the MOM code used in this study (Griffies, 2009). The **CG99** experiment is identical to **noblob**, except it uses the overflow parameterisation of Campin and Goosse (1999). On the other hand, the **NoReturn** experiment uses the no return overflow scheme, which is based on the scheme of Campin and Goosse (1999), however, the prescribed return flow is not utilised in this parameterisation (described in Section 4.5).

### 12.3.4 Lagrangian Blob Schemes

The **ctrl** experiment is the baseline experiment from which other experiments vary a single parameter at a time to explore the parameter space of the blobs and investigate how those changes affect the properties of the plume. The first parameter to be varied from the **ctrl** experiment is the coefficient of drag. Observational estimates of skin drag have consistently shown to be approximately  $3 \times 10^{-3}$  (see discussion in Section 9.1.2). Form drag over rough topography, however, can also be a large contributing factor to the overall bottom drag. Özgökmen and Fischer (2008) show in numerical experiments that form drag can be approximately an order of magnitude higher than skin drag over rough topography. With this in mind, the coefficient of drag is increased by an order of magnitude to  $C_d = 3.0 \times 10^{-2}$  in the **c3e-2** experiment and is also decreased by an order of magnitude to  $C_d = 3.0 \times 10^{-4}$  in the **c3e-4** experiment to cover a parameter range that could be reasonably expected.

Name	$C_d$	$\Gamma$	$\delta$	$\Delta\rho$	Comments
ctrl	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1	0.0	
c3e-2	$3.0 \times 10^{-2}$	$1.0 \times 10^{-6}$	1	0.0	
c3e-4	$3.0 \times 10^{-4}$	$1.0 \times 10^{-6}$	1	0.0	
G1e-4	$3.0 \times 10^{-3}$	$1.0 \times 10^{-4}$	1	0.0	
G1e-5	$3.0 \times 10^{-3}$	$1.0 \times 10^{-5}$	1	0.0	
G1e-7	$3.0 \times 10^{-3}$	$1.0 \times 10^{-7}$	1	0.0	
G1e-8	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.0	
r100	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.100	
r010	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.010	
r001	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.001	
d20	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	20.0	0.0	
d23	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	2/3	0.0	
d13	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1/3	0.0	
free	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1	0.0	Allows blob separation
bblob	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	N/A	0.0	Enforces large blob creation
adiab	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1	0.0	Has adiabatic blobs
noconv	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1	0.0	No convective adjustment

Table 12.2: Experimental details for the DOME experiments that use dynamically active blobs.  $C_d$  is the coefficient of drag,  $\Gamma$  is the detrainment parameter,  $\delta$  is the fraction of a grid cell participating in an overflow event and  $\Delta\rho$  is the density difference threshold above which a blob is formed.

For all of the experiments using the embedded Lagrangian scheme, a blob’s mass is considered “small” when it is less than 1000kg, and all remaining properties are returned to the E system (see Section 5.3). In other words, a blob is considered small when its volume is less than approximately  $1\text{m}^3$ , or, its mass is less than about 0.000003% of the total mass of the smallest grid cell in the domain. The maximum proportion of a grid cell that the L system may occupy is 70% (see Section 6.5.3). The Runge-Kutta scheme used is that of Cash and Karp (1990) with a local relative truncation error of  $\zeta^* = 0.01$ , a safety factor of  $\varrho = 0.8$  and a minimum step size of  $h_{\min} = 18\text{s}$  (see Section 7.3.2). The maximum allowable detrainment is  $\mathcal{D}_{\max} = 10^{15}\text{m s}^{-1}$  (see Section 5.3) and the frictional parameter used to calculate the initial mass of bottom blobs is  $\mu = 10^{-4}\text{s}^{-1}$  (see Section 5.3.1).

### Detrainment Parameter, $\Gamma$

A parameter for which there is no observational evidence or previous study to base an initial value on is the detrainment parameter,  $\Gamma$ . This is largely because it is a parameter that controls a fairly heuristic means by which to destroy (i.e. control the number of) blobs. As such, a number of experiments are performed over a large range of values from  $1 \times 10^{-8}\text{kg m}^{-2}\text{s}^{-1}$  to  $1 \times 10^{-4}\text{kg m}^{-2}\text{s}^{-1}$  in the G1e-4, G1e-5, ctrl, G1e-7 and G1e-8 experiments. The **adiab** experiment forces the blobs in the experiment to be adiabatic, that is, there is no entrainment or detrainment permitted.

### Threshold Density Difference, $\Delta\rho$

The threshold density difference at which a blob is formed,  $\Delta\rho$ , is also varied to explore the parameter space. The threshold density difference is essentially a means by which the number of blobs that are in existence can be controlled. In contrast to the detrainment parameter, it controls how many blobs are formed, rather than impacting on how long they live. Again, there is no guidance from the literature as to what this value should be, so a large parameter range is spanned, from  $0.0\text{kg m}^{-3}$  to  $0.1\text{kg m}^{-3}$ .

### Fraction of a Cell Overflowing, $\delta$

The fraction of a grid cell participating in an overflow event,  $\delta$ , is also varied. Campin and Goosse (1999) use a value of  $1/3$  for their original formulation. For the bottom blob formation, there are certain constraints in the code to ensure that a grid cell is never entirely “drained” of fluid. There is no reason why an entire grid cell cannot participate in the overflow event. As such, the a value of  $1$  is chosen for the `ctrl` experiment and the smaller value of  $2/3$  is chosen for the `d23` experiment and a value of  $1/3$  is chosen for the `d13` experiment. Since there are checks and balances in the code to ensure that a grid cell is never entirely drained of fluid, the value of  $\delta$  can be increased to be greater than  $1$ . Experiment `d20` uses a value of  $20.0$  to see how larger blobs affect the solution. There is also an experiment, called `bblob`, where the creation of large blobs is enforced by making all blobs  $1/5^{\text{th}}$  the mass of the grid cell of origin.

## Other Experiments

There are two additional experiments listed in Table 12.2: the `free` and `noconv` experiments. The `free` experiment has the free blob module turned on, which facilitates the separation of bottom blobs to free blobs. This experiment aims to examine the importance of blobs separating from the bottom boundary. The `noconv` experiment has no convective adjustment scheme while the bottom blobs are in use. This experiment aims to investigate whether convective adjustment still plays a significant role in plume dynamics with the blobs turned on.

For the `free` experiment, which uses the dynamically active free blobs, the Runge-Kutta scheme of Bogacki and Shampine (1989) is used with a local relative truncation error of  $\zeta^* = 0.05$ , a safety factor of  $\varrho = 0.8$  and a minimum step size of  $h_{\min} = 18\text{s}$  (see Section 7.3.2). The buoyancy frequency threshold is very large, so, blobs that are separating from the bottom are the only source of dynamically active free blobs. The drag coefficient is  $10^{-7}\text{s}^{-1}$ . The detrainment factor,  $\Gamma$  is the same as for the bottom blobs, as is the maximum detrainment.

## 12.4 Results

The DOME test case is mostly concerned with the properties of the plume on the slope. The presence of the solid walls can alter the behaviour of the plume on the slope. Thus, for long times after the plume has interacted with the western boundary, the useful information that can be gleaned from the experiment is somewhat limited. Therefore, most of the analysis looking at the physical properties of the plume focusses on the period prior to the plume interacting with the western wall.

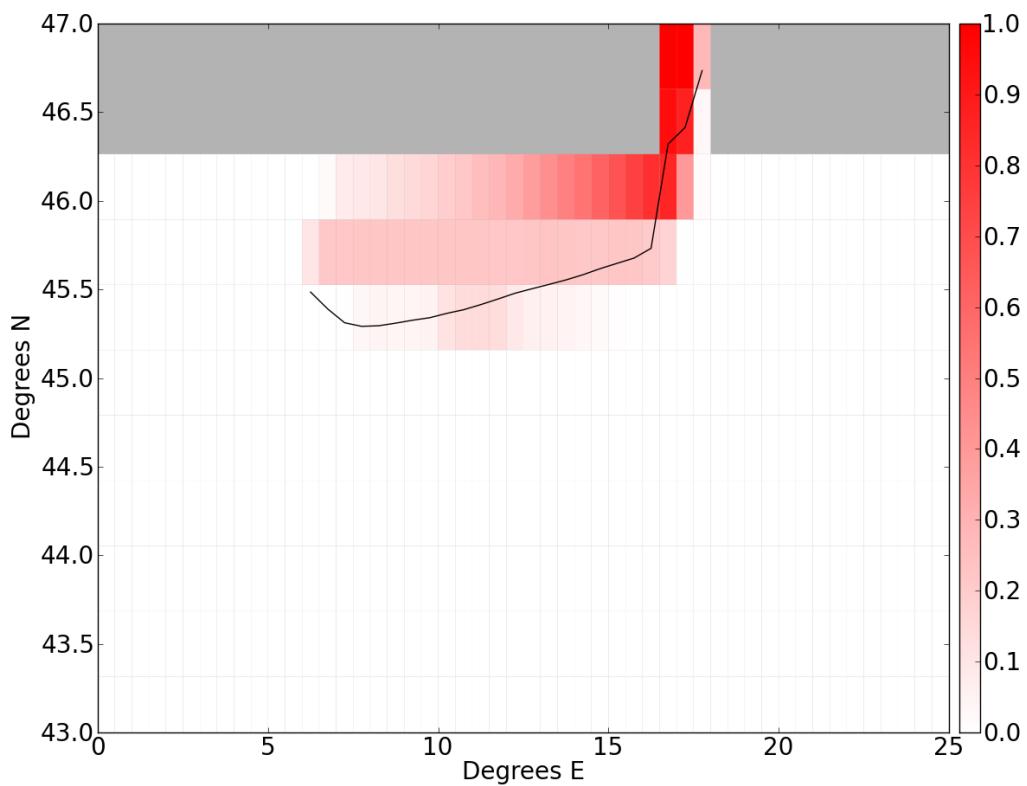
Previously, Legg et al. (2006) found in the MITgcm that coarse resolution (50 km) experiments are sluggish and have shallower plume penetration than finer resolution experiments. As resolution is increased, they find more phenomena are resolved, for instance at 10 km resolution, the plume penetrates deeper and has a faster along slope velocity, however, eddies are not resolved.

### 12.4.1 Preliminaries

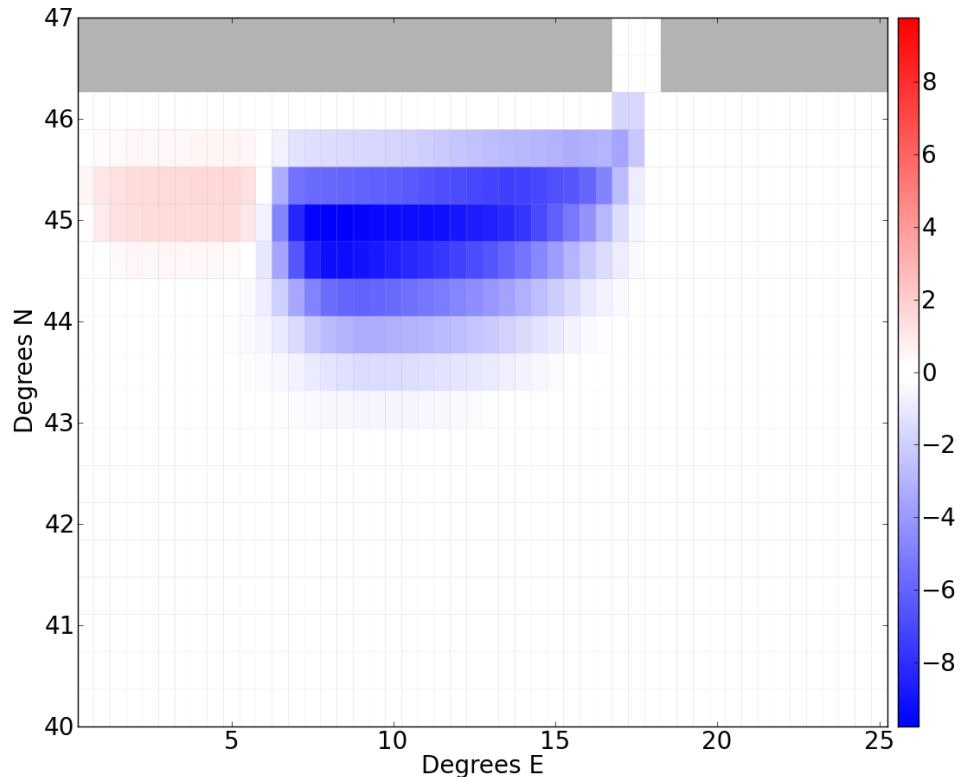
The objective of this section is to obtain a qualitative feel for the nature of the solutions to the experiments listed in Tables 12.1 and 12.2. This is done by examining plots from a selection of experiments. From a subset of representative experiments, a pseudo-colour plot of the bottom dye tracer concentration, with the centre of mass of the plume overlaid – see Equation( 12.4) – as well as a pseudo-colour plot of the barotropic quasi-streamfunction, defined in Equations (9.7). For experiments that use dynamically active blobs, the longitude-latitude position of the blobs are also overlaid on the bottom dye concentration plot. As shall be seen in Section 12.4.4, there are vastly different plume speeds for different experiments. The plots for the experiments are shown when the plume front is at approximately 6°E (thus, showing the experiments at different times).

The first experiment chosen, shown in Figure 12.2, is the **noblob** experiment (which has no overflow parameterisation) at day 180. The bottom dye concentration in Figure 12.2(a) shows the plume largely hugging the northern boundary. The barotropic quasi-streamfunction in Figure 12.2(b) indicates a cyclonic circulation centred on the southern part of the plume. As shall be seen in Section 12.4.4, the progress of the plume front is relatively slow. The **gm** experiment is qualitatively similar to the **noblob** experiment, and is thus not shown.

The **ksmag2** experiment, which uses the Laplacian Smagorinsky viscosity, is shown in Figure 12.3 at day 270. As can be seen in Figure 12.3(a), the footprint of the dye tracer on the bottom is not coincident with the centre of mass of the plume. This indicates that there is a significant portion of the plume that is not near the bottom and is being transported offshore. The cyclonic circulation associated with the plume, shown in Figure 12.3(b) supports this hypothesis. The bottom of the plume, as shown by the dye concentration, has more of a downslope component to it than that displayed by the **noblob**



(a) The dye concentration of the bottom cells and the centre of mass of the plume.



(b) The barotropic quasi-streamfunction.

Figure 12.2: The `noblob` experiment at day 180. Note the different latitudinal scales. A cyclonic cell, centred over the southern portion of the plume, is present in the quasi-streamfunction.

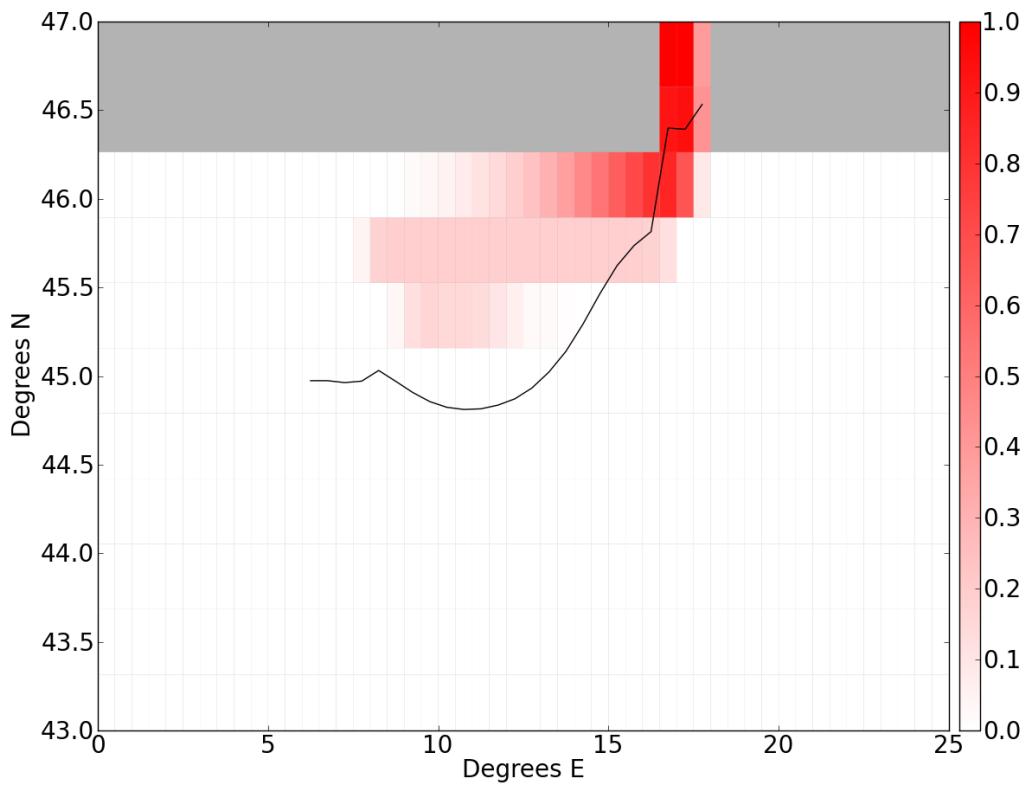
experiment. While the use of the Smagorinsky friction scheme increases the downslope transport, it takes approximately an additional 90 days more than the `noblob` experiment for the plume front to reach 6°E – indicating a much slower plume speed. The `ksmag3` and `nomicom` experiments are qualitatively similar to the `ksmag2` experiment, and are thus not shown.

Figure 12.4 shows the `BD97` experiment, which uses the Beckmann and Döscher (1997) overflow parameterisation, at day 60. The diffusive nature of the scheme is immediately obvious in the bottom dye concentration, shown in Figure 12.4(a), with the dye spreading isotropically to just south of 45°N. The spreading of the dye tracer to the east of the embayment is spurious and is due to the fact that the scheme is not affected by rotation. The speed of the plume is much faster than any of the other experiments that have been shown thus far. As shall be seen in Section 12.4.4, this increased plume speed is due to the enhanced diffusion prescribed by the scheme. The barotropic quasi-streamfunction in Figure 12.4(b) is much more smaller in extent, but similar in magnitude than that in `noblob` experiment. The quasi-streamfunction at later times (not shown) is much more similar to that of the `noblob` experiment. This indicates that the Beckmann and Döscher (1997) scheme does not have a large influence over the large scale flow, but, mostly just influences the tracer concentration of the bottom most cells.

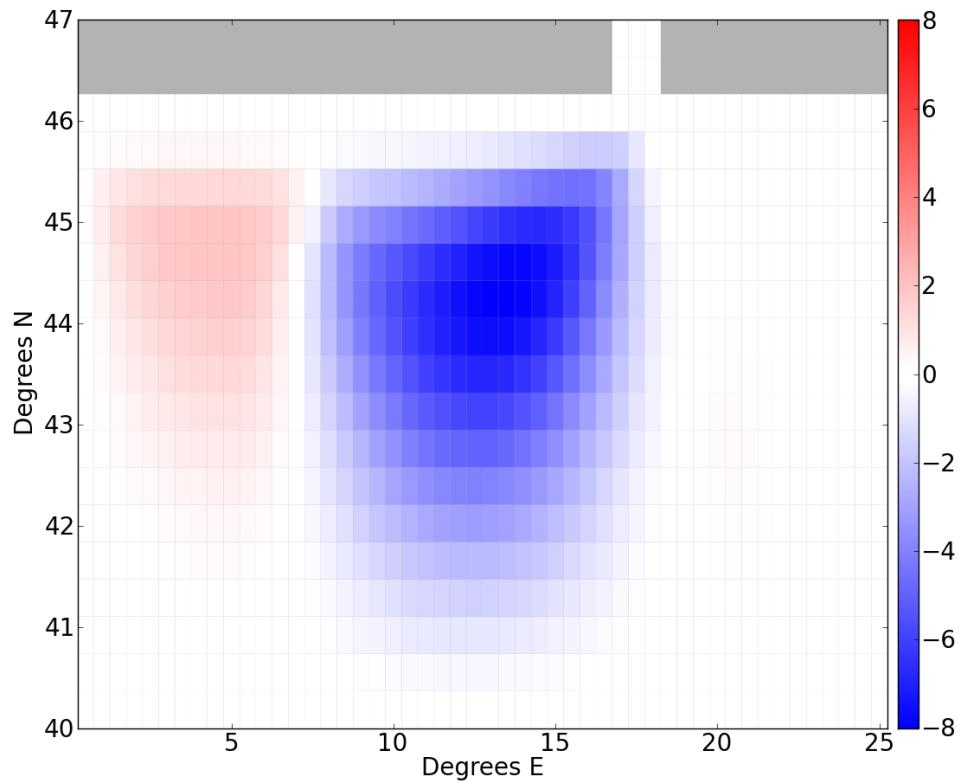
The `CG99` experiment, which uses the overflow parameterisation of Campin and Goosse (1999), is shown in Figure 12.5. From the bottom dye concentration, Figure 12.5(a), the parameterisation is effective in getting the plume further downslope when compared to the `noblob` experiment (Figure 12.2(a)), evidenced by the bottom tracer distribution and the latitude of the plume centre of mass. The plume speed, however, is much slower, with the plume front passing 6°E at approxiamtely day 230, as compared to day 180 for the `noblob` experiment. Similar to the `noblob` experiment (Figure 12.2(b)), the `CG99` experiment shows a strong cyclonic cell centred on the southern end of the plume in Figure 12.5(b). The circulation is, however, appreciably stronger (note the different colour scales) than the other experiments that have been examined thus far.

The tracer distribution of the `NoReturn` experiment, which uses the modified Campin and Goosse (1999) scheme (described in Section 4.5) and is shown in Figure 12.6(a), is similar to the `CG99` experiment, except that material is moved even further downslope near the embayment. The westward progress of the plume is slower than the `CG99` experiment, with the along slope plume progress being faster upslope (45°N–46°N) than further downslope (44°N–45°N). Figure 12.6(b) shows that the latitudinal extent of the cyclonic cell is greater than that of the `CG99` experiment (Figure 12.5(b)), but, the magnitude of the cirulcation is less.

When compared to the `noblob` experiment, it can be seen that the existing downslope flow parameterisations that have been investigated improve some aspects of the represen-

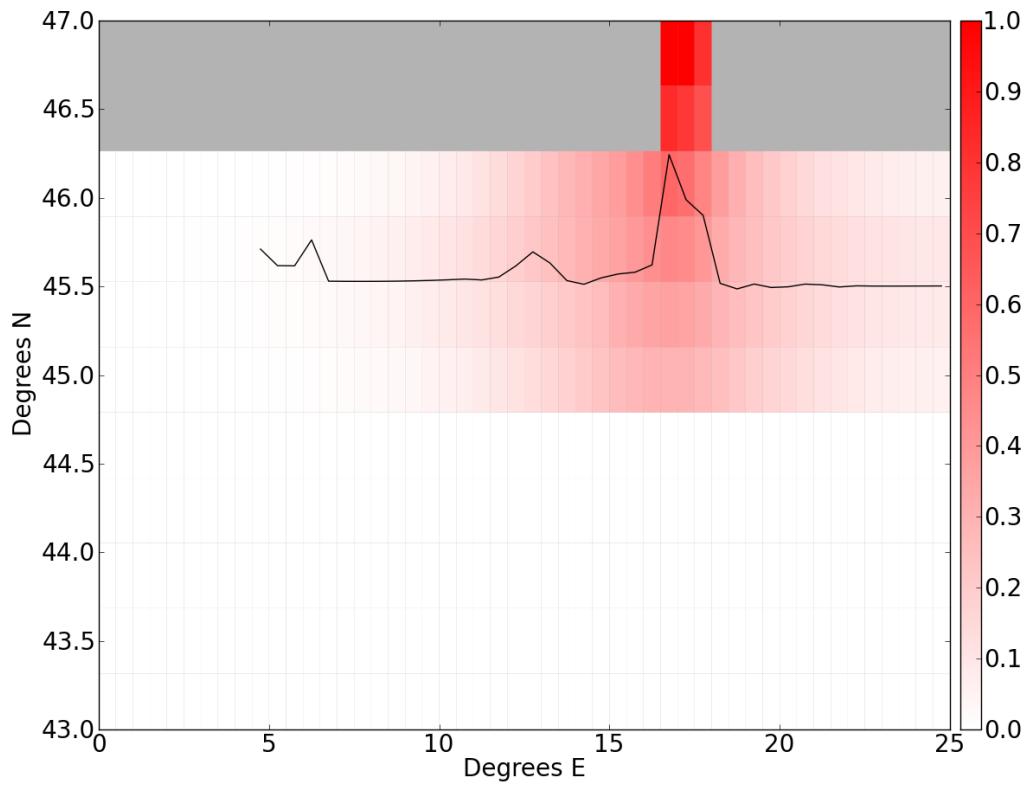


(a) The dye concentration of the bottom cells and the centre of mass of the plume.

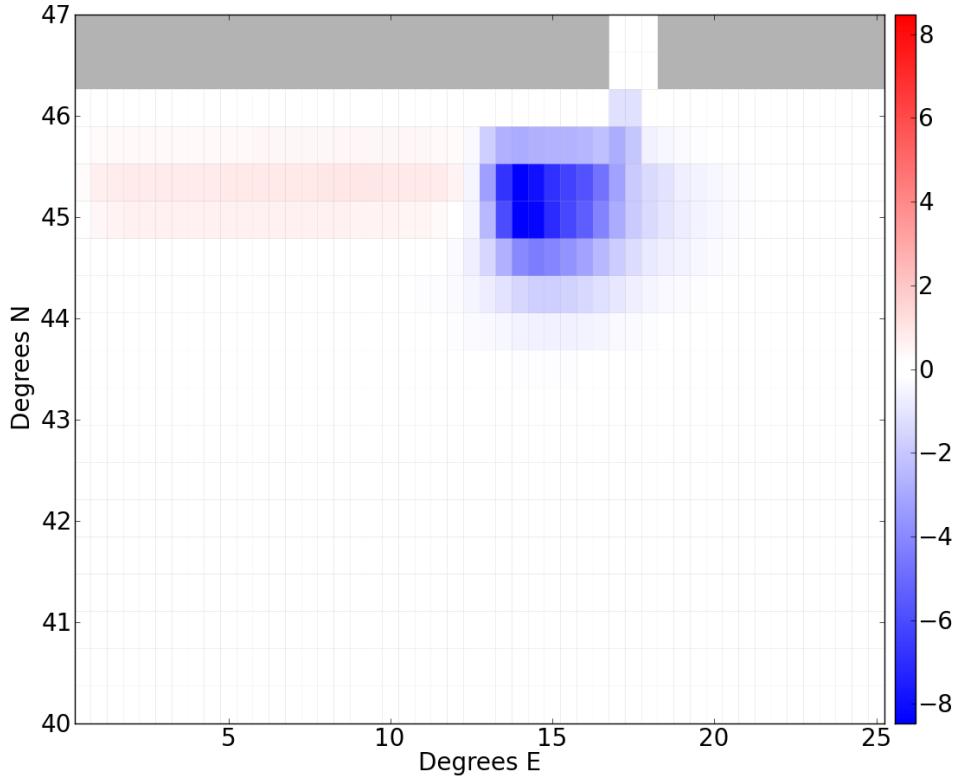


(b) The barotropic quasi-streamfunction.

Figure 12.3: The `ksmag2` experiment at day 270. Note the different latitudinal scales. A cyclonic cell, centred over the southern portion of the plume, is present in the quasi-streamfunction.

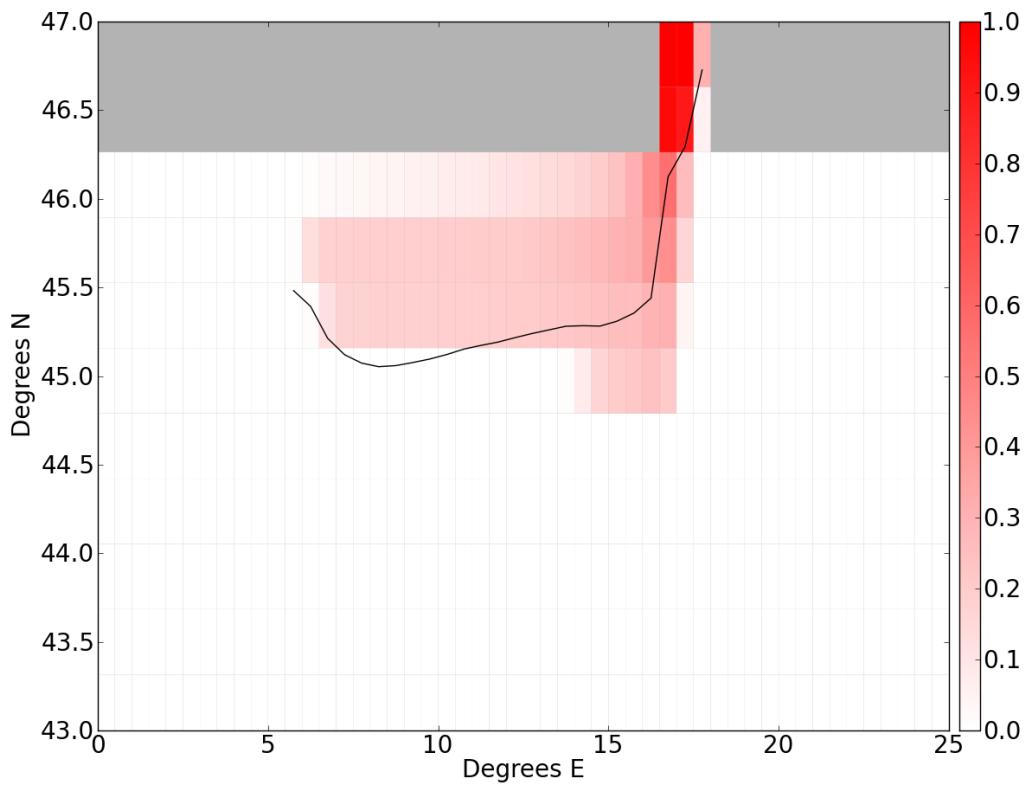


(a) The dye concentration of the bottom cells and the centre of mass of the plume.

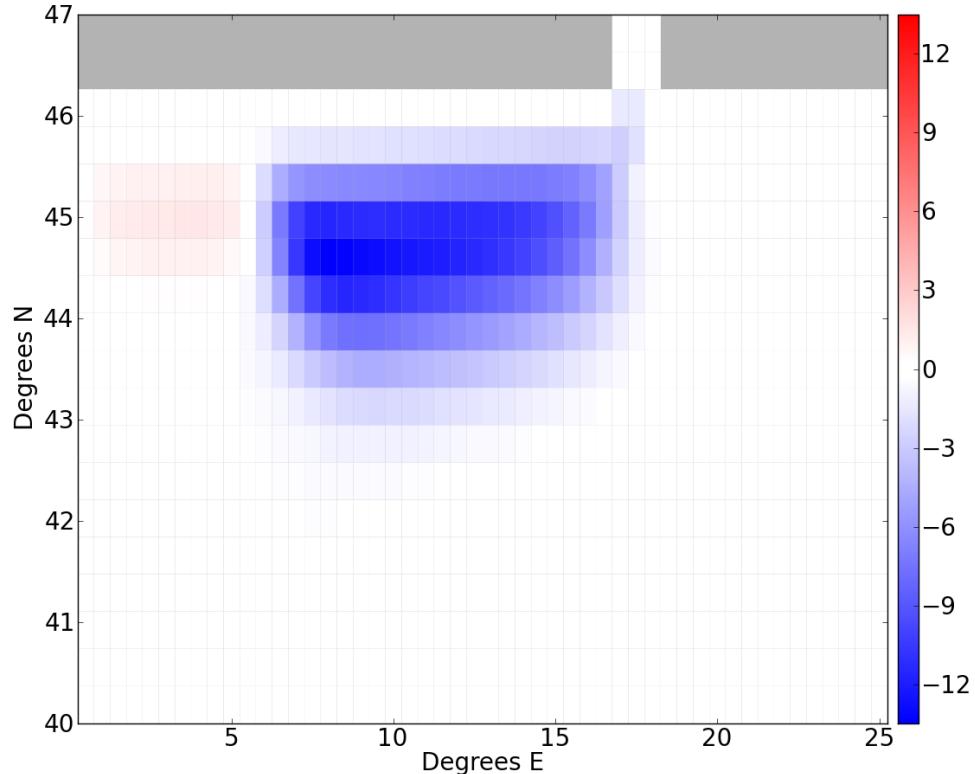


(b) The barotropic quasi-streamfunction.

Figure 12.4: The BD97 experiment at day 60. Note the different latitudinal scales. The spreading of the dye tracer to the east of the embayment is spurious and is due to the diffusive nature of the overflow parameterisation used. The large scale flow is not significantly affected by the parameterisation.

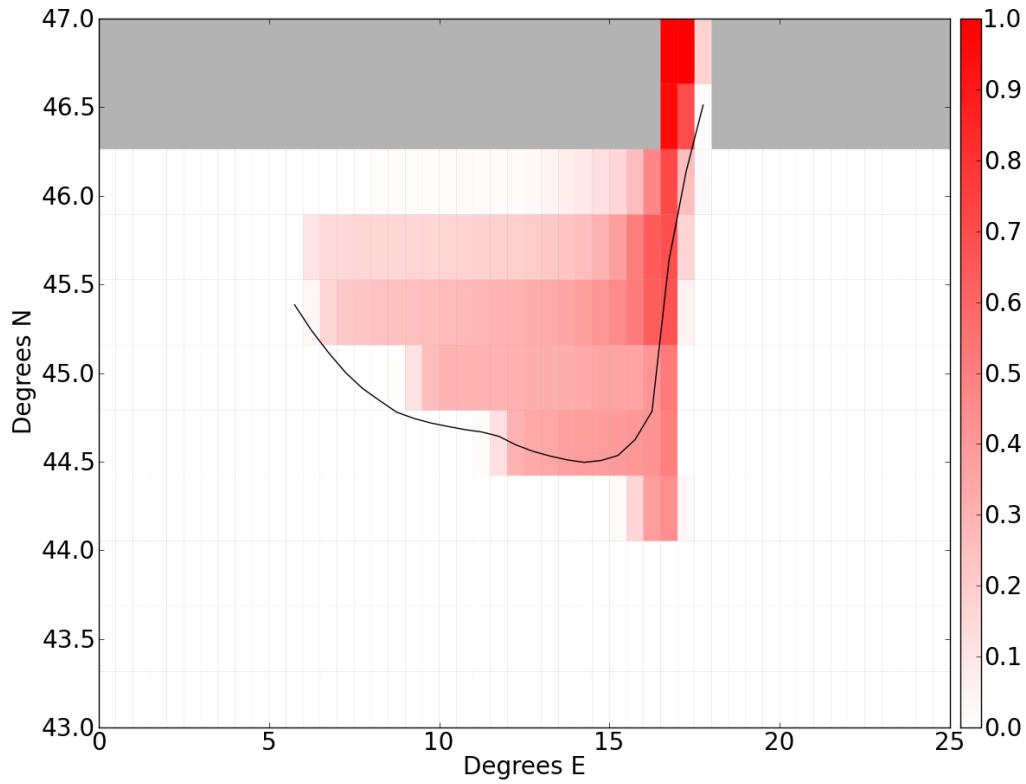


(a) The dye concentration of the bottom cells and the centre of mass of the plume.

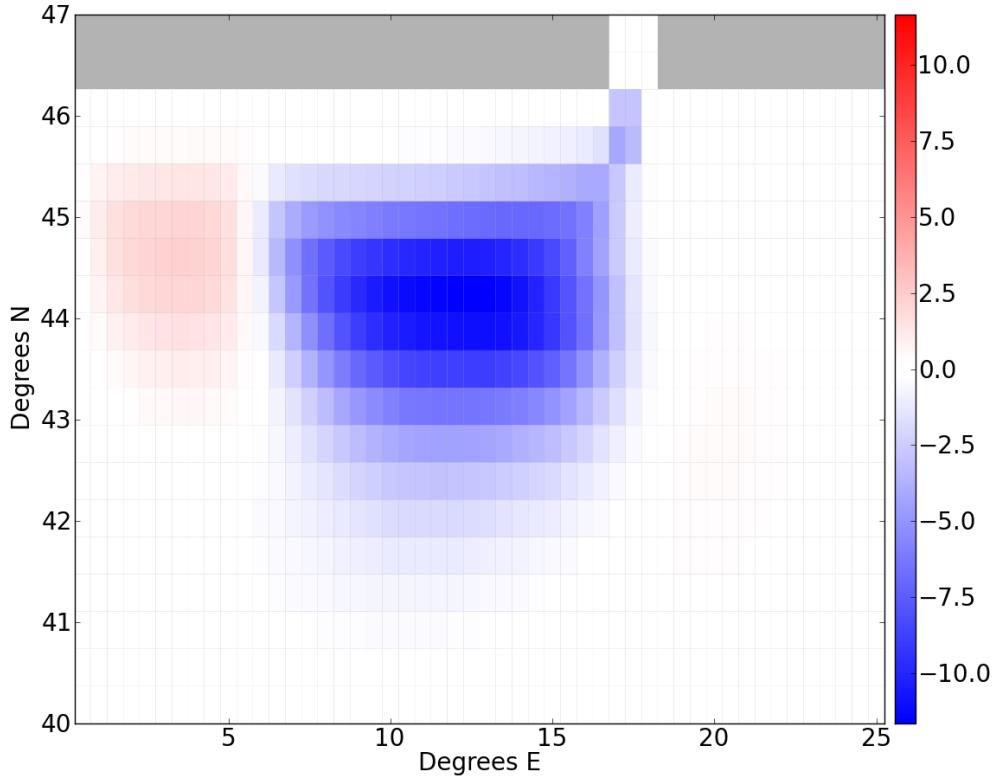


(b) The barotropic quasi-streamfunction.

Figure 12.5: The CG99 experiment at day 230. Note the different latitudinal scales. The parameterisation is effective at getting tracer downslope. It also induces a strong cyclonic cell, centred over the southern portion of the plume.



(a) The dye concentration of the bottom cells and the centre of mass of the plume.



(b) The barotropic quasi-streamfunction.

Figure 12.6: The `NoReturn` experiment at day 340. Note the different latitudinal scales. The parameterisation moves material downslope near the embayment, but, slows the westward propagation of the plume. This promotes a more latitudinally extensive circulation that is less intense, when compared to the CG99 experiment in Figure 12.5(b).

tation of a plume on a slope in level coordinate coarse resolution ocean climate models, however, these improvements come at a cost by degrading other aspects of the solution. Firstly, the embedded sigma later of Beckmann and Döscher (1997) diffuses the plume's properties isotropically in bottom grid cells. Doing so moves tracer further down the slope, and, increases the along slope velocity of the plume, however, the isotropic nature of the tracer diffusion means that the plume spreads spuriously to the east of the embayment. The increased plume velocity is mostly due to the diffusion of the scheme and there is little corresponding effect on the large scale circulation.

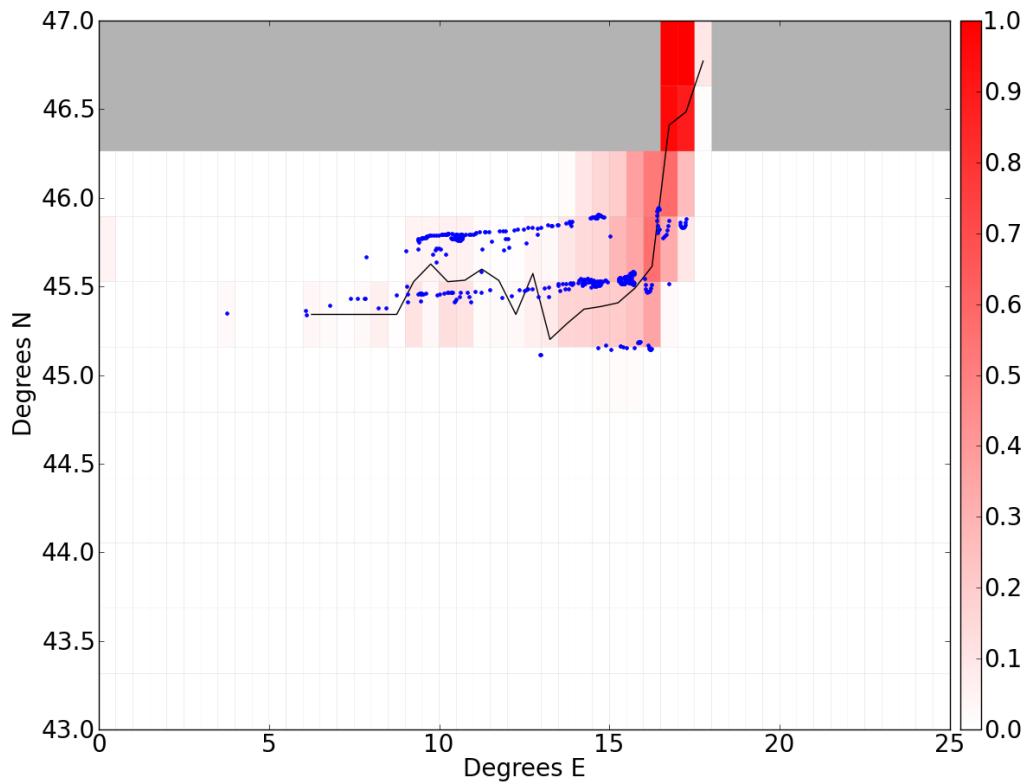
The overflow parameterisation of Campin and Goosse (1999) and the no return version of their parameterisation presented in Section 4.5 are successful in getting tracer further down the slope. This strengthens the circulation, but, it comes at a cost of a much slower along slope propagation of the plume. The ramifications of the improvements and degradations of the solution by these parameterisations are investigated further in the analysis in subsequent sections.

The bottom cell dye tracer concentration, centre of mass of the plume and blob positions at day 32 are shown in Figure 12.7(a), while the E system's barotropic quasi-streamfunction is shown in Figure 12.7(b) (see discussion in Section 9.2 regarding why the quasi-streamfunction is only for the E system). Compared to all other experiments shown thus far, there is a very large difference in both the bottom tracer concentration and the quasi-streamfunction. Firstly, away from the immediate area south of the plume, virtually all of the along slope tracer transport has been done by blobs. The blobs obviously travel at a relatively high speed and thus, transport tracer much more rapidly than what the E system is able to.

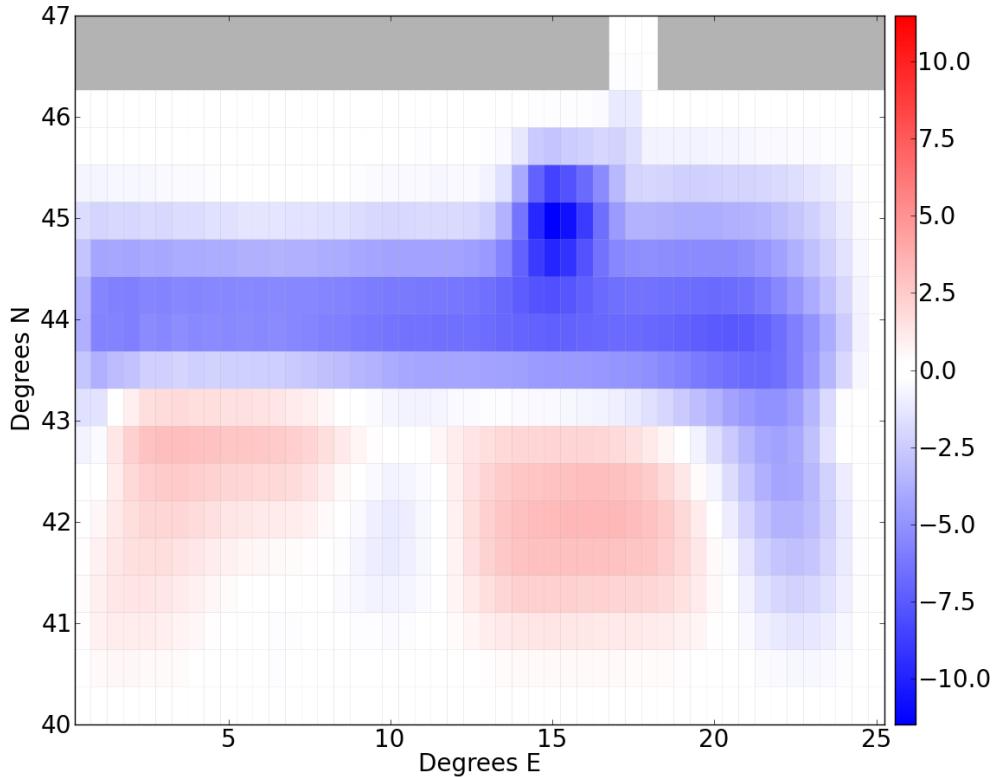
In Figure 12.7(b) the quasi-streamfunction shows a relatively strong cyclonic cell focussed over the south of the centre of the plume, and extends virtually the full longitudinal extent of the domain. There are two weaker anti-cyclonic cells that are in the southern part of the domain. This circulation pattern must be a direct response of the E system to the movement of mass affected by the L system's blobs.

Between the experiments listed in Table 12.2 there is a reasonable amount of variation in tracer and blob distribution, as well as the quasi-streamfunction. For brevity, only a selection of the experiments listed in Table 12.2 that offer insight into the way that blob parameters affect the solution are presented here.

The effect of having a higher bottom friction coefficient for the blobs is investigated in the `c3e-2` experiment, with the results at day 46 shown in Figure 12.8. The increased bottom drag causes the plume's centre of mass to be further south than in the `ctrl` experiment (Figure 12.8(a)). The reason for the greater downslope component to the plume path is because friction breaks the geostrophic balance that causes the plume to follow



(a) The combined E and L system dye concentration of the bottom cells, the centre of mass of the plume and the location of blobs.



(b) The E system barotropic quasi-streamfunction.

Figure 12.7: The `ctrl` experiment at day 32. Note the different latitudinal scales. The blobs propagate the plume front along the slope much faster and instigate a barotropic flow that is very different to all previously presented experiments.

topographic contours. Thus, increasing the frictional force increases the downslope component of the flow. Another consequence of having a larger bottom friction is that the plume velocity is a little bit slower than the `ctrl` experiment, however, is still much faster than the `noblob` experiment.

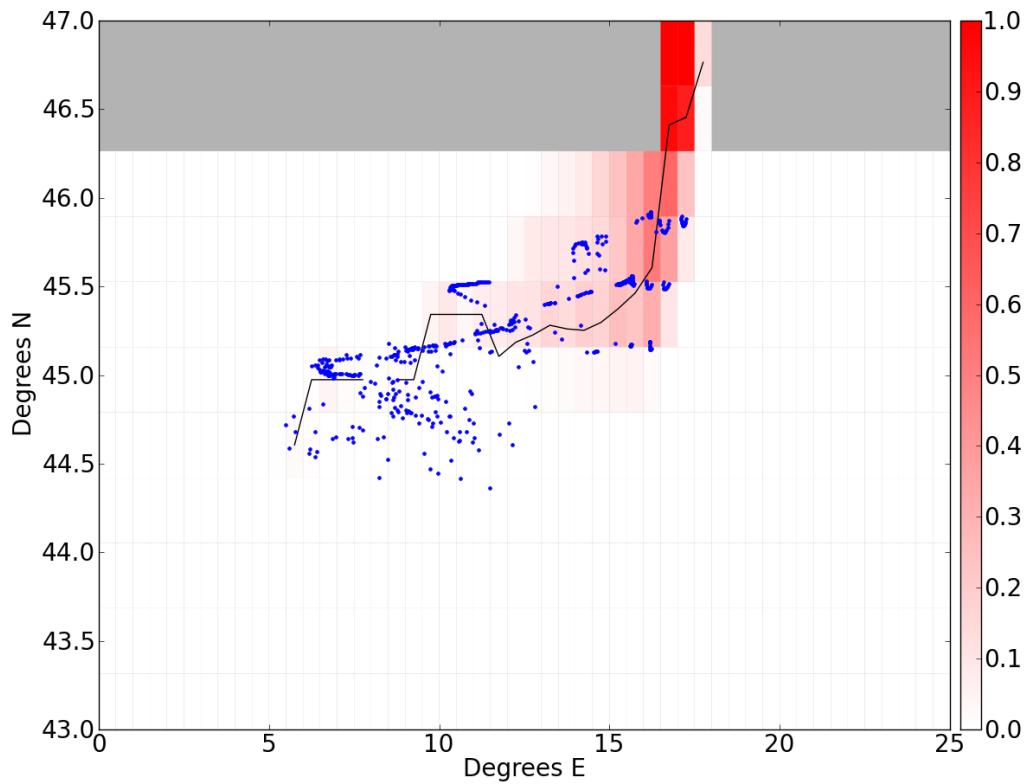
One thing that is noteworthy is that the centre of mass of the plume (the black line in Figure 12.8(a)) is not continuous. This indicates that somewhere between the western front of the plume and the plume’s eastern extreme, there is some longitude of grid cells where the combined E and L system dye tracer concentration is less than 0.01. It can be seen that there are blobs in the region where the plume is not defined. This does not mean that the blobs dye tracer concentration is less than 0.01, but, it means that the mass weighted average of the blob(s) in a grid cell, and the grid cell tracer concentration is less than 0.01.

The E system’s quasi-streamfunction for the `c3e-2` experiment, shown in Figure 12.8(b), is largely the same as the quasi-streamfunction of the `ctrl` experiment (Figure 12.7(b)), however, the cyclonic cell in the southern part of the domain centred on  $\sim 10E$  is appreciably larger and stronger in the `c3e-2` experiment. This change in circulation is likely due to the slightly different pathway that blobs take along and down the slope between the two experiments.

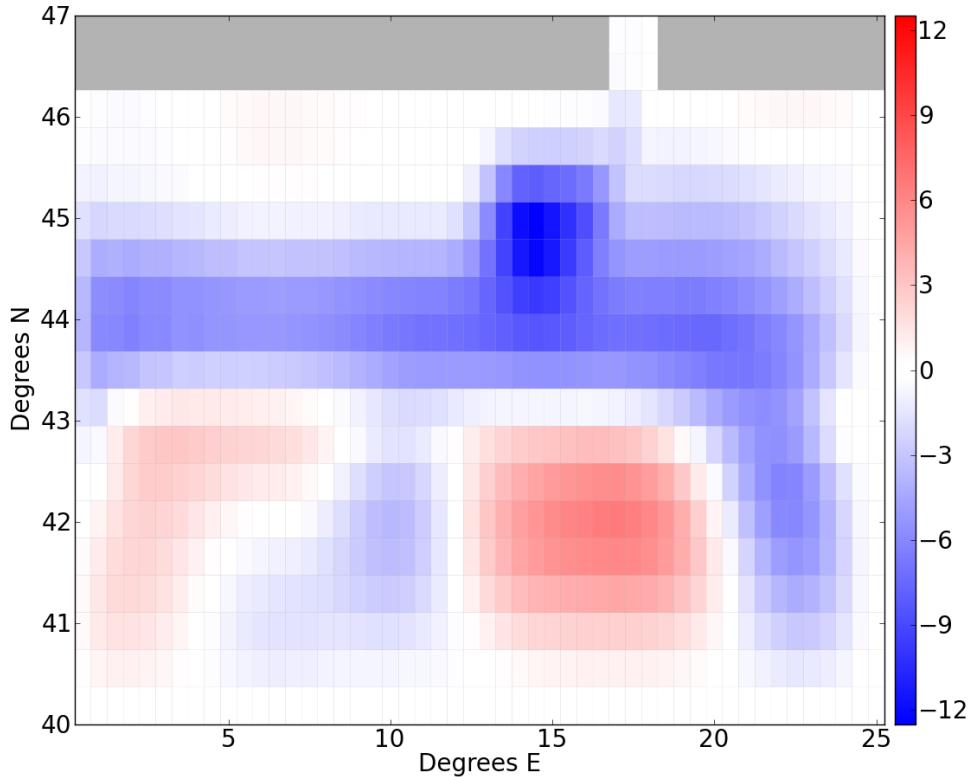
The final experiment to be examined in this section, shown in Figure 12.9, is the `bblob` experiment, which enforces the creation of massive blobs (one fifth the size of the grid cell of origin). As can be seen, the tracer distribution, number of blobs and quasi-streamfunction are all vastly different to any of the other experiments presented previously. Firstly, there are much fewer blobs, with none away from the near vicinity of the embayment. As shall be seen in Section 12.4.7, this is due to the grid cell mass constraint being violated by the large blobs (see Section 6.5.3), thus, causing them to be destroyed. This is an illustration of the limitation of having bottom blobs only considered to be residing in the bottom most grid cell (see Section 8.2).

The lack of blobs to transport the properties away from the embayment significantly slows down the plume when compared with other blobs experiments. Further, the large scale circulation of the `bblob` experiment exhibits a quadrupole of very, very strong cells in the southern half of the domain that dominates the E system’s barotropic quasi-streamfunction (Figure 12.9(b)). The `d20` experiment, which also forms massive blobs, has a quasi-streamfunction that is much more akin to that of the `bblob` experiment than that of the `ctrl` experiment.

The experiments that use the Lagrangian blobs have a vastly different character to their solutions than all the other experiments. They are mostly qualitatively similar, with the largest differences in tracer distribution occurring when the coefficient of bottom fric-

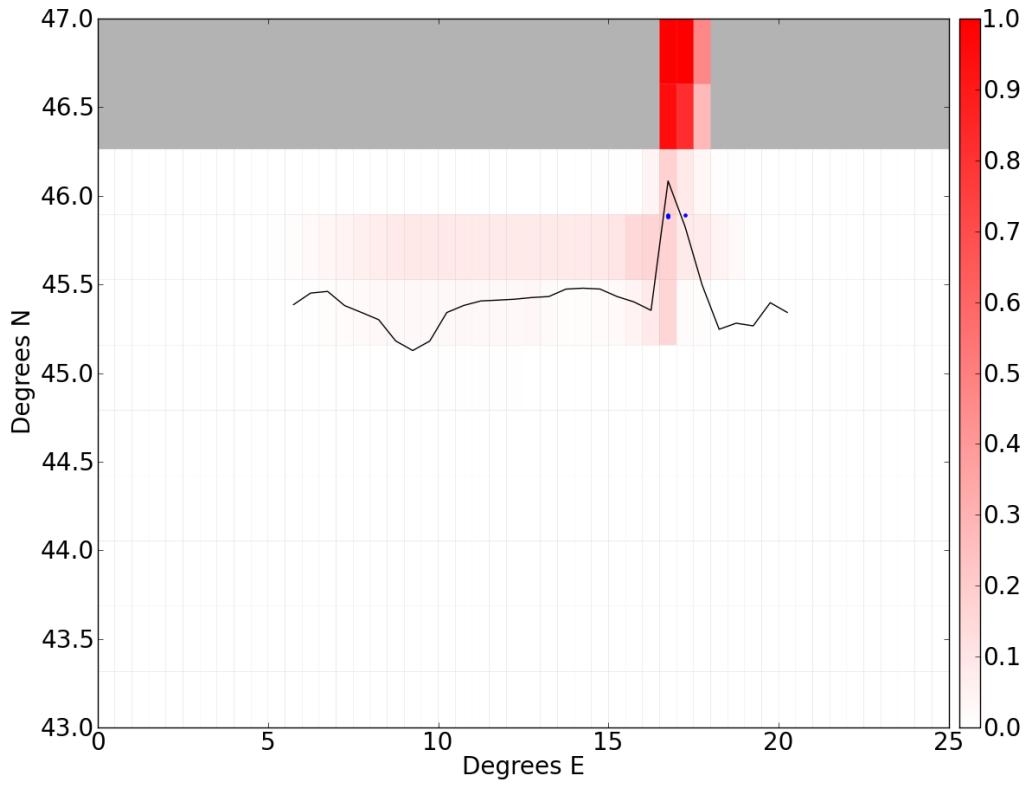


(a) The combined E and L system dye concentration of the bottom cells, the centre of mass of the plume and the location of blobs.

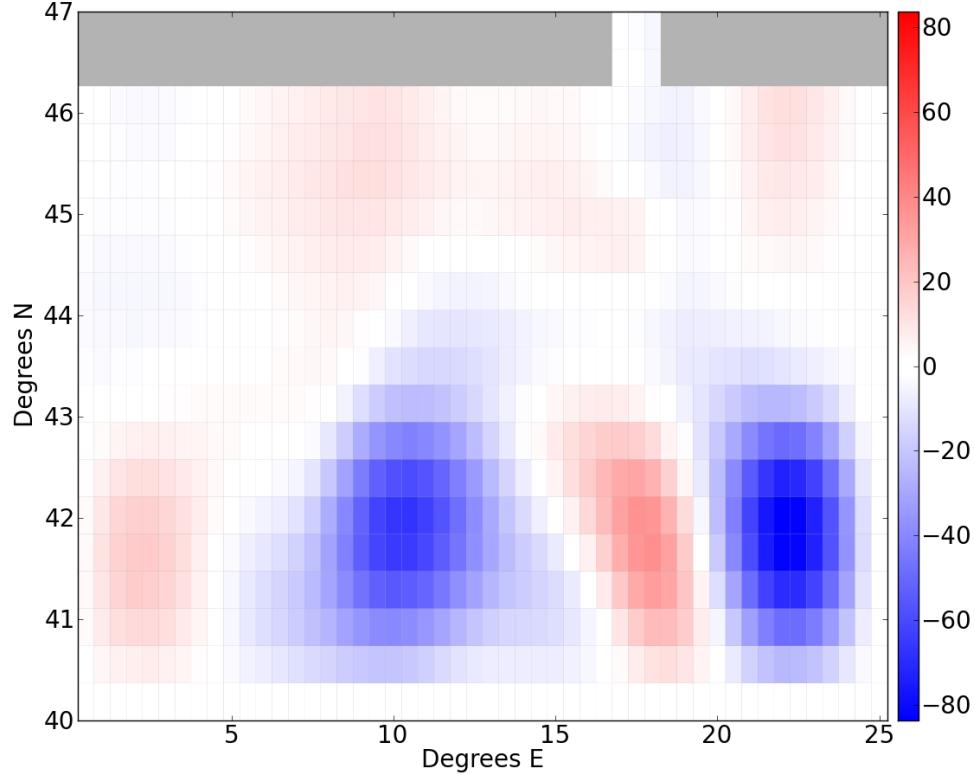


(b) The E system barotropic quasi-streamfunction.

Figure 12.8: The `c3e-2` experiment at day 46. Note the different latitudinal scales. The larger coefficient of bottom friction for the blobs increases the downslope component of the blob pathways, with ramifications for the plume pathway, speed and E system circulation.



(a) The combined E and L system dye concentration of the bottom cells, the centre of mass of the plume and the location of blobs.



(b) The E system barotropic quasi-streamfunction.

Figure 12.9: The **blob** experiment at day 126. Note the different latitudinal scales. There **blob** experiment enforces the creation of large blobs, which causes a strong E system circulation. The enforcement of large blobs also means that most blobs are destroyed shortly after being created in order to satisfy the grid cell mass constraint.

tion is increased by an order of magnitude, or, the formation of massive blobs is enforced. Dramatic changes in the pattern and magnitude of the E system's barotropic flow occur when the creation of large blobs is enforced.

In the following sections, a more quantitative analysis of the experiments is undertaken, examining various metrics to determine the effect of the Lagrangian blob scheme (and other parameterisations) on the properties of the plume.

### 12.4.2 Plume Mass and Density

A time-series of the plume mass outside of the embayment is shown for all experiments in Figure 12.10. Relative to the `noblob` experiment, it can be seen that virtually all of the overflow parameterisations increase the mass of the plume. Changing the horizontal friction scheme to the Laplacian Smagorinsky scheme, or the increasing coefficient of bottom drag decreases the plume mass. The `gm` experiment, which uses an neutral physics scheme, also has a larger plume mass. For the rest of this chapter, only a representative subset of the time-series of experiments will be plotted.

The mass of the plume is directly related to the dimensions of the plume, see Equation (12.3). For the plume mass to increase, it must be either thicker and/or have an greater longitudinal and/or have a greater latitudinal extent. It can be seen from the previous section that most of the blob experiments, and the `BD97` experiment have a much greater longitudinal extent than the `noblob` experiment (which has no overflow parameterisation). The other experiments with overflow parameterisations (`CG99` and `NoReturn`) have a lesser longitudinal extent than the `noblob` experiment, however, they have a greater latitudinal extent and thickness (due to the parameterisation transporting properties down the slope) that more than compensates for the lesser longitudinal extent. The increase in plume mass in the `BD97` experiment is due to the spreading of tracers via the enhanced along slope diffusion prescribed by the scheme (see Figure 12.4(a)). This leads to the plume spreading without any reference to rotation, and thus, a portion of the mass of the plume is spurious.

The experiments that change the friction scheme (`ksmag2`, `ksmag3` and `nomicom`) and the bottom drag (`hdrag`) all have lesser plume masses, presumably due to the lesser longitudinal extent of their plumes. The larger plume mass of the `gm` experiment, which uses an neutral physics scheme, is likely due to the neutral physics scheme spreading the plume along isopycnals, rather than along or down the slope.

The evolution of the average density of the plume from a selection of experiments is shown in Figure 12.11. As a reference, the coldest of the inflowing waters is  $10.81^{\circ}\text{C}$ , which corresponds to a density of  $1032.24\text{kg m}^{-3}$ . A non-trivial amount of mixing occurs in and near the embayment with the amount of mixing depending in part on the various schemes used by the model (not shown). The `noblob` experiment's plume density

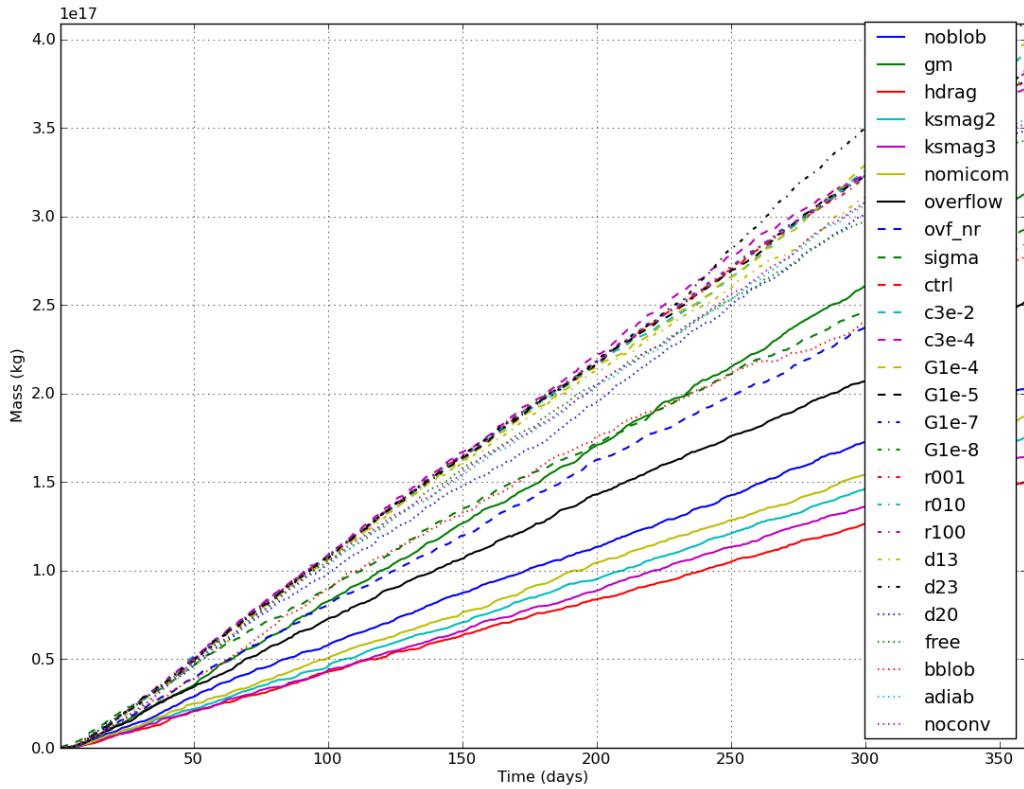


Figure 12.10: The mass of the plume outside of the embayment. The mass of the plume is related to the dimensions of the plume, with the various parameterisations acting to change the dimensions of the plume, relative to the `noblob` experiment. The blob schemes mostly lengthen the plume, while other overflow parameterisations tend to widen and thicken it.

has a very similar trajectory to the other experiments where the bottom friction (`hdrag`) and lateral friction scheme are varied (`ksmag2`, `ksmag3` and `nomicom`). The `gm` experiment follows a parallel trajectory, with the density at an appreciably smaller magnitude. The less dense plume in the `gm` experiment is likely due to the diffusion of the plume's properties along isopycnals (recall that explicit diffusion is absent from other experiments).

The CG99 and BD97 experiments have plumes that are more dense than the `noblob` experiment and the evolution of density largely parallels the `noblob` experiment. Similar to the blob experiments, the initial spike is most likely due to the parameterisation initially being able to somewhat preserve the density signal of the inflowing water before being diluted as the plume spreads.

The `ctrl` experiment is shown as it is qualitatively representative of the experiments using Lagrangian blobs, with the exception of the `bblob` experiment. The `ctrl` experiment eventually has a lower plume density than the `noblob` experiment, although, this is a long time after the plume front of the `ctrl` experiment interacts with the western

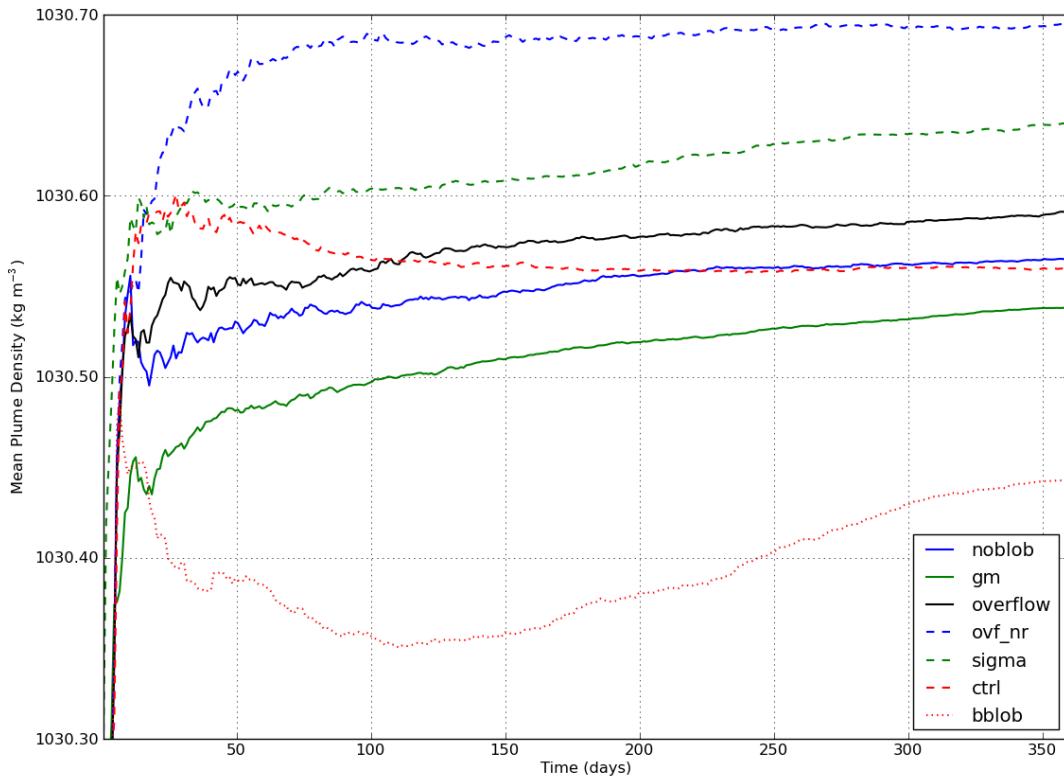


Figure 12.11: Average density of the plume (of the combined E and L system) outside of the embayment. Mostly, overflow parameterisations increase the mean plume density, while the neutral physics scheme decreases the plume’s density.

wall, so, the importance or relevance of this result is uncertain. The `bblob` experiment has a large decrease in the density of the plume over the first 100 days, before the density recovers somewhat over the rest of the simulation.

The effect of the choice of model setup, whether it be overflow parameterisation, neutral physics scheme or friction scheme, affects the basic properties of a plume on a uniform slope. Here, the properties investigated were the average mass and density of the plume. In most instances, using an neutral physics scheme, or, a more complex friction parameterisation tended to decrease the mass of the plume, and use of an overflow parameterisations increased the mass of the plume. The mechanism for the increase in mass of the plume in experiments using one of the traditional overflow parameterisations can have other spurious effects (some of which are investigated in subsequent sections of this chapter).

### 12.4.3 Blob Mass and Density

The average mass of blobs from a selection of experiments using the embedded Lagrangian model is shown in Figure 12.12. One feature common to all experiments is an initial spike in the blob mass, before most experiments plateau to a reasonably constant value through-

out the duration of the experiment. The average density of blobs is shown in Figure 12.13. The spike that is seen initially in the mass of blobs is also present in the density, although, it is much less pronounced.

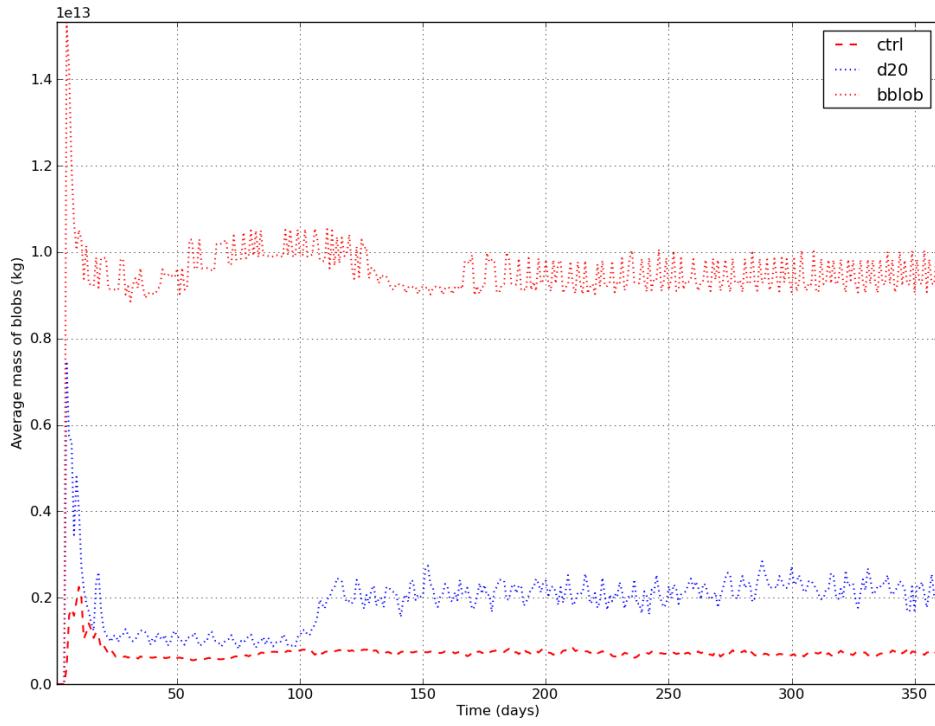
As is to be expected, blobs in the **bblob** (which enforces the production of blobs that are one fifth the mass of the grid cell of origin) and **d20** (which has  $\delta = 20$ ) experiments are significantly more massive than blobs in other experiments. These two experiments are shown in Figure 12.12(a) along with the **ctrl** experiment for comparison. Interestingly, these two experiments have average blob densities that are significantly less than the other experiments. In the main region just south of the embayment (where the majority of blobs are formed) the E system density is lower in these two experiments than the other experiments (not shown). It is hypothesised that the movement of large blobs changes the E system circulation such that the dense water leaving the embayment is rapidly diluted. This hypothesis is supported by the very strong E system circulation shown for the **bblob** experiment in Figure 12.9(b). The **d20** experiment also has a strong E system circulation with a similar pattern to the **bblob** experiment.

Figure 12.12(b) shows a selection of blob experiments where very massive blobs are not formed. Most experiments not shown roughly follow the trajectory of the **ctrl** blob mass curve. The **r100** experiment, which has a blob creation threshold of  $\Delta\rho = 0.1\text{kg m}^{-3}$ , has consistently more massive blobs than the other experiments shown in Figure 12.12(b). This is mostly likely the result of a relatively large initial mass of blobs formed in this experiment, which results from the initial mass being proportional to the onshelf/offshelf density difference – see Equation (5.22a).

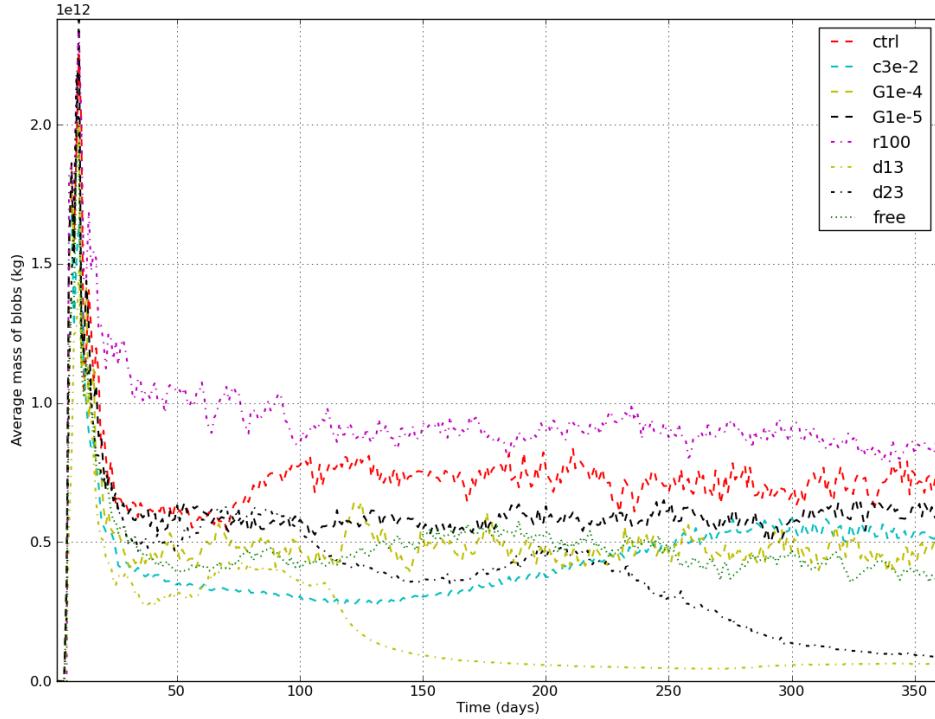
The experiments **d13** and **d23** (which have a value for  $\delta = 1/3$  and  $\delta = 2/3$  respectively) have lower blob masses than most other experiments. Since the initial mass of a blob is directly proportional to the fraction of a grid cell participating in the overflow event, Equation (5.22a), it is not surprising that these experiments produce lower mass blobs compared to the **ctrl** experiment, which has  $\delta = 1.0$ . The **d13** experiment has an appreciably higher average blob density than the other experiments for most of the simulation, however, the **d23** experiment has an average blob density that largely tracks that of the **ctrl** experiment.

The **c3e-2** experiment also has a lower average blob mass than the **ctrl** experiment (Figure 12.12(b)). It is unclear as to what the connection between having a higher coefficient of bottom friction and the blob mass is. The experiment also has a lower average blob density than the **ctrl** experiment, as shown in Figure 12.13. Similar to average blob mass, it is unclear what the connection is between a higher value for the dimensionless bottom friction and the average blob density.

The **G1e-5** and **G1e-4** experiments, in which the detrainment parameter ( $\Gamma$ ) is in-



(a) The **d20** and **bblob** experiments form massive blobs when compared to all other experiments. The **ctrl** experiment is shown as well for comparison.



(b) A sample of experiments that do not form massive blobs.

Figure 12.12: The average mass of blobs in various experiments. All of the blob parameters being examined ( $C_d, \delta, \Delta\rho$ ) affect, to some degree, the average blob mass.

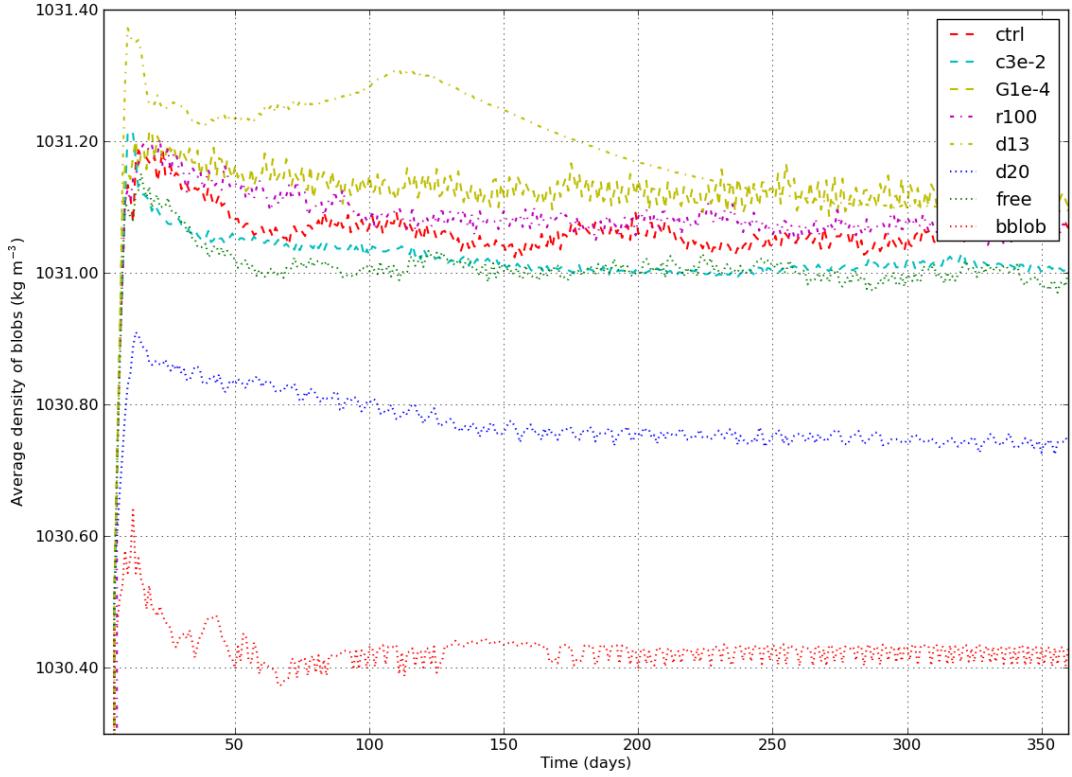


Figure 12.13: The average density of blobs for a subset of experiments. The experiments that form the most massive blobs also form the least dense blobs.

creased relative to the `ctrl` experiment, also have lower average blob masses than the `ctrl` experiment. This is likely due to the faster rate of detrainment that arises from having a larger detrainment parameter – see Equation (5.17c). The `G1e-5` does not have an appreciably different blob density to the `ctrl` experiment, but, the `G1e-4` experiment does have an appreciably higher average blob density. This could be due to the shorter life time (Figure 12.21) of the blobs in the `G1e-4` experiment (arising from their relatively rapid detrainment) which does not give them as long to entrain material and dilute the blob’s properties. The experiments `G1e-7` and `G1e-8` are very similar to the `ctrl` experiment, indicating that changes to the detrainment parameter at small values have less of an effect than changes at larger values.

In the `free` experiment blobs that satisfy the separation criterion, Equation (5.27), become free blobs rather than being destroyed, as is the case in the `ctrl` experiment. This means that some blobs will live longer and detrain for longer, bringing down the average mass of the blobs in existence. Thereby explaining why blobs in the `free` experiment have a smaller average mass and lower average density.

### 12.4.4 Plume Speed

It can be seen from the Nof Speed, Equation (12.6), that in the absence of friction the greater the magnitude of the reduced gravity, the faster the plume will travel. Since the bulk ocean in the DOME experiment is stratified, the density of the plume is not necessarily the best measure of the reduced gravity, as the density of the bulk ocean is a function of depth.

The more negative the value of the reduced gravity, the greater the westward Nof speed will be, since the slope and the Coriolis parameter are fixed in the DOME configuration. The initial reduced gravity of the inflowing water is  $1.89 \times 10^{-2} \text{ m s}^{-2}$ . On the slope, this would equate to a Nof speed of  $-1.89 \text{ m s}^{-1}$ . If the plume were to travel at this speed, it would take roughly 8 days to travel the approximately 1300km from the embayment to the western wall. This is, of course, an upper bound on the plume speed, as it does not take into account the sinking of the plume into a stratified environment (which decreases the magnitude of the reduced gravity) or other effects such as bottom friction and dilution of the plume via mixing.

The average reduced gravity of the plume is calculated according to Equation (12.7) and is shown in Figure 12.14. All of the blob experiments, with the exception of `blob` and `d20` (experiments that form large blobs), have similar trajectories for the evolution of their plume's reduced gravity. The experiments that vary the horizontal friction scheme (`ksmag3` and `micom`) and that vary the bottom drag (`hdrag`) have trajectories that are similar to that of `ksmag2`.

Interestingly, even though the `NoReturn` experiment has one of the highest average plume densities (Figure 12.11), it has one of the lowest values for its reduced gravity. It can be seen in Figure 12.6(a) that the parameterisation is effective at getting material down the slope. However, moving the material further down the slope makes the reduced gravity smaller, due to the higher density of the background environment. A consequence of the smaller reduced gravity is that the model is slower to transport the plume properties along the slope, which can be seen by the slow progress of the plume front in Figure 12.15. The `CG99` experiment is a less extreme version of the `NoReturn` experiment, in which the mean density of the plume is a little higher than that of the `noblob` experiment, but, the magnitude of the reduced gravity is also slightly smaller too. This too is reflected in the plume front position, with `CG99` having a plume front speed between `noblob` and `NoReturn`.

The `BD97` experiment generally has a comparable reduced gravity to the experiments without any overflow parameterisation, however, it can be seen from the plume front position in Figure 12.15 that it has a significantly faster plume front speed. This is most likely due to the sigma diffusion acting along the slope (in addition to down the slope), thus, enhancing the along slope transport.

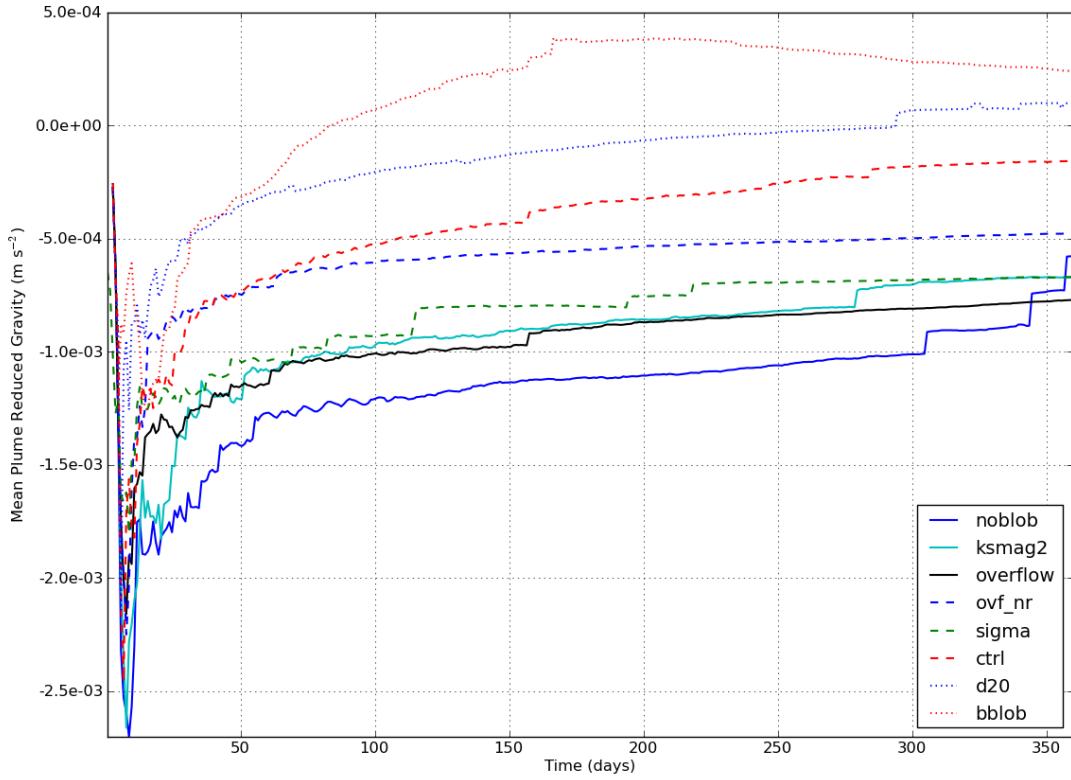


Figure 12.14: The average reduced gravity of the plume, as calculated by Equation (12.7). The experiments using the Lagrangian blob scheme generally have a lower reduced gravity, however, their plumes still travel faster.

The **bblob** experiment actually has a positive mean plume reduced gravity for the majority of the experiment. This explains why part of the plume extends towards the east of the embayment (see Figure 12.9(a)). Despite the positive reduced gravity, the plume front still advances westward faster than the **noblob** experiment. Due to the lack of blobs to transport properties in the **bblob** experiment (see Section 12.4.1) there must be other explanations to explain why the plume continues to travel westward. It could be due to the intense circulation in the experiment (see Figure 12.9(b)), or, it could be that some parts of the plume have a negative reduced gravity, which allows at least part of the plume to propagate westward.

The **d20** experiment, which forms large blobs, has a low reduced gravity that becomes positive toward the end of the simulation. When looking at the position of the plume front, it can be seen that the front still progresses faster than the **noblob** experiment, aided in part by the blobs, and, likely also aided in part by the strong circulation in that experiment.

One feature of the plume front position of the **d20** experiment, that is even more pronounced in the **G1e-4** experiment, is that there is a lot of noise in the position of the plume front. The reason for the noisy plume front position is that these plumes are dis-

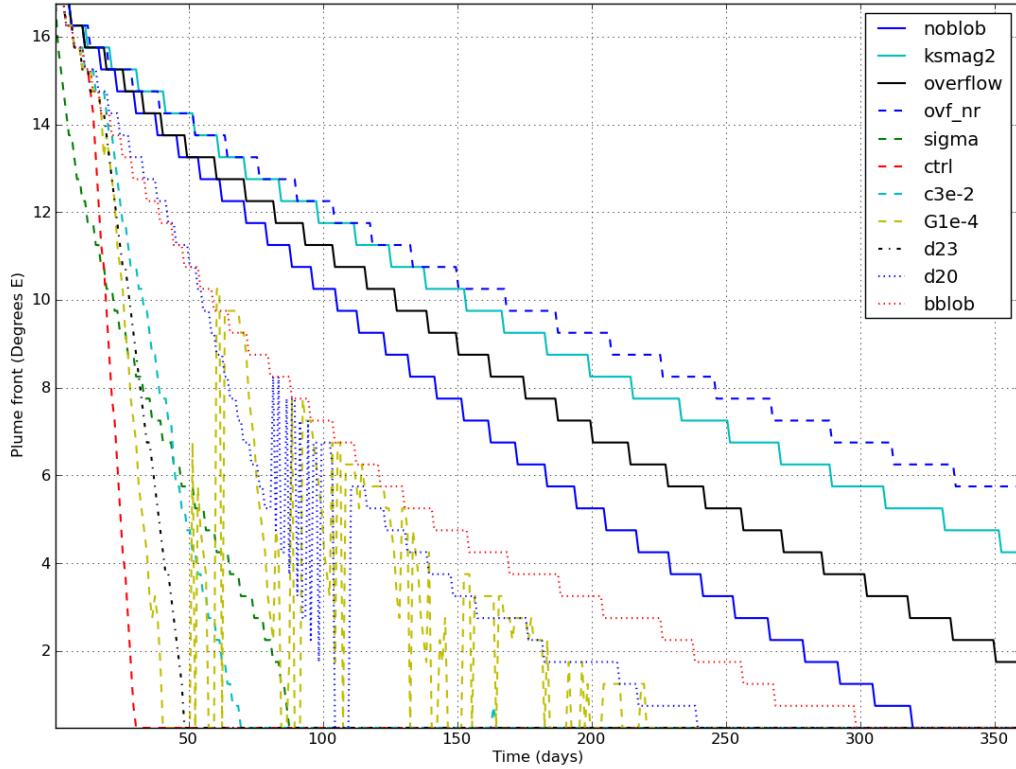


Figure 12.15: The position of the plume front for a representative subset of experiments. The plume front is defined as the western most grid cell where the plume is defined. The step like nature of some of the experiments is due to the resolution of the model used (half a degree). The noise in the plume front position in some of the experiments is not spurious, it is due to the definition of the plume, see Figure 12.16.

continuous, as discussed in relation to Figure 12.8(a). A fairly extreme example of a discontinuous plume is shown in Figure 12.16 (the plume centre of mass is the black line and longitudes where there is no black line, the plume is undefined). The reason for the discontinuity is because the blobs are transporting tracer along the slope, however, there are regions where their mass weighted average tracer concentration (with the E system) is not sufficient to be greater than 0.01. Once a blob interacts with the western wall, it is destroyed (as it is considered to have become grounded) and its properties are returned to the E system. The 10 day restoring timescale in the western most grid cells then fairly rapidly removes the dye tracer signal. Thus, for there to not be any noise in the plume front time-series, there must be blobs of sufficient size and number all the way along the slope, or, the E system must have sufficient concentration have a continuously defined plume.

The `ksmag2` experiment is representative of the `ksmag3`, `nomicom` and `hdrag` experiments. It can be seen that the Smagorinsky friction scheme acts to decrease the reduced gravity (relative to the constant friction used in the `noblob` experiment). Since `hdrag`

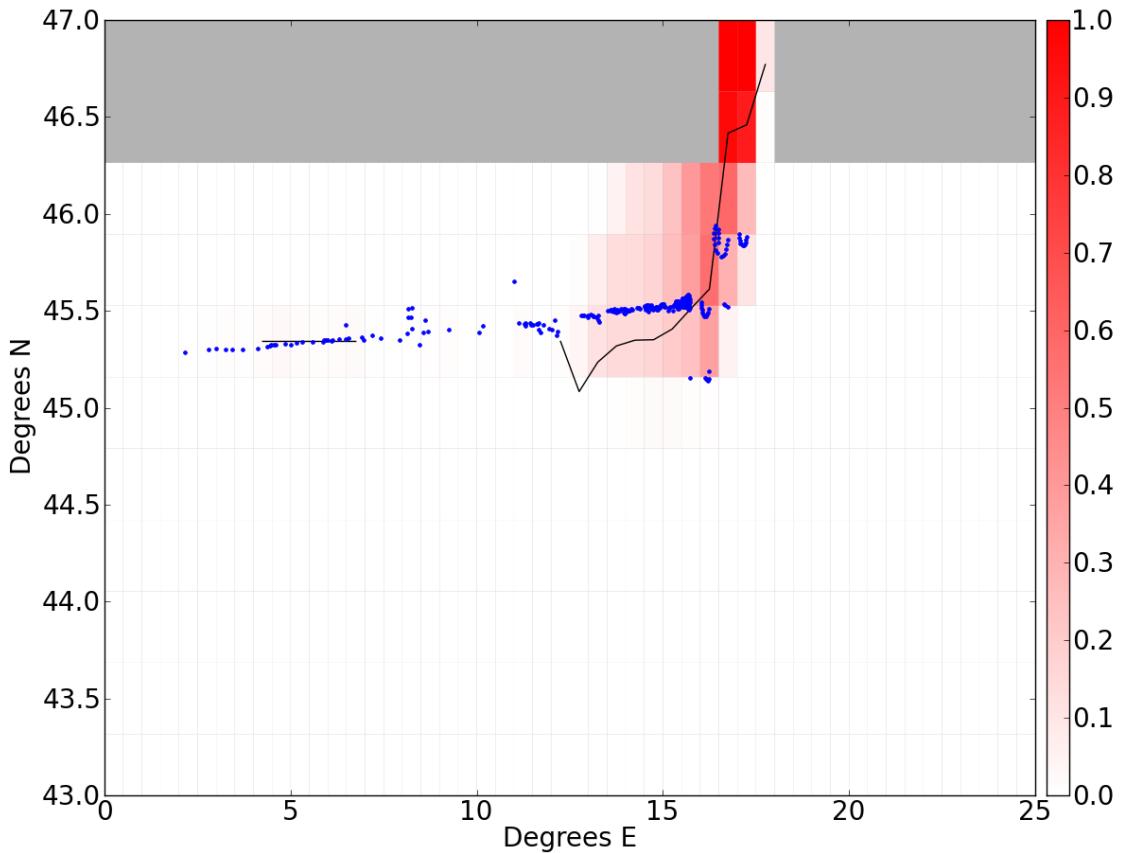


Figure 12.16: An example of a discontinuous plume, which is the `G1e-4` experiment at day 36. The discontinuous plume arises because there are regions where the combined E and L system dye tracer concentration is less than 0.01. The discontinuous plume can cause noise in the time-series for the plume front, shown in Figure 12.15.

tracks the `ksmag2` experiment, it can also be said that increasing the Eulerian bottom drag reduces the magnitude of the reduced gravity. Looking at the `ksmag2` experiment, it can be seen in the plume front position in Figure 12.15 that the reduction in the reduced gravity contributes to a reduction in the plume speed.

In terms of the speed of the plume front in the blob experiments, most of them have a plume front speed that is fairly similar to that of the `ctrl1` experiment. The blob experiment that has an appreciably slower plume front speed is the `c3e-2` experiment, which has a coefficient of friction that is 10 times larger than the `ctrl1` experiment. The slower progress of the plume in that experiment is likely due to slower blob speeds, which result from a combination of increased friction and lower blob reduced gravity (see Figure 12.17).

#### 12.4.5 Blob Speed

There is not a significant enough difference between the reduced gravity of the blob experiments shown in Figure 12.14 and that of the experiments that do not use any parameterisation to explain the very large difference in plume front speed shown in Figure 12.15.

The reason for the large difference in plume front speed is due to the blobs themselves. The blobs travel much faster than the E system, largely because of a reduced gravity that is much greater in magnitude (as shown in Figure 12.17), allowing them to have a much greater Nof speed.

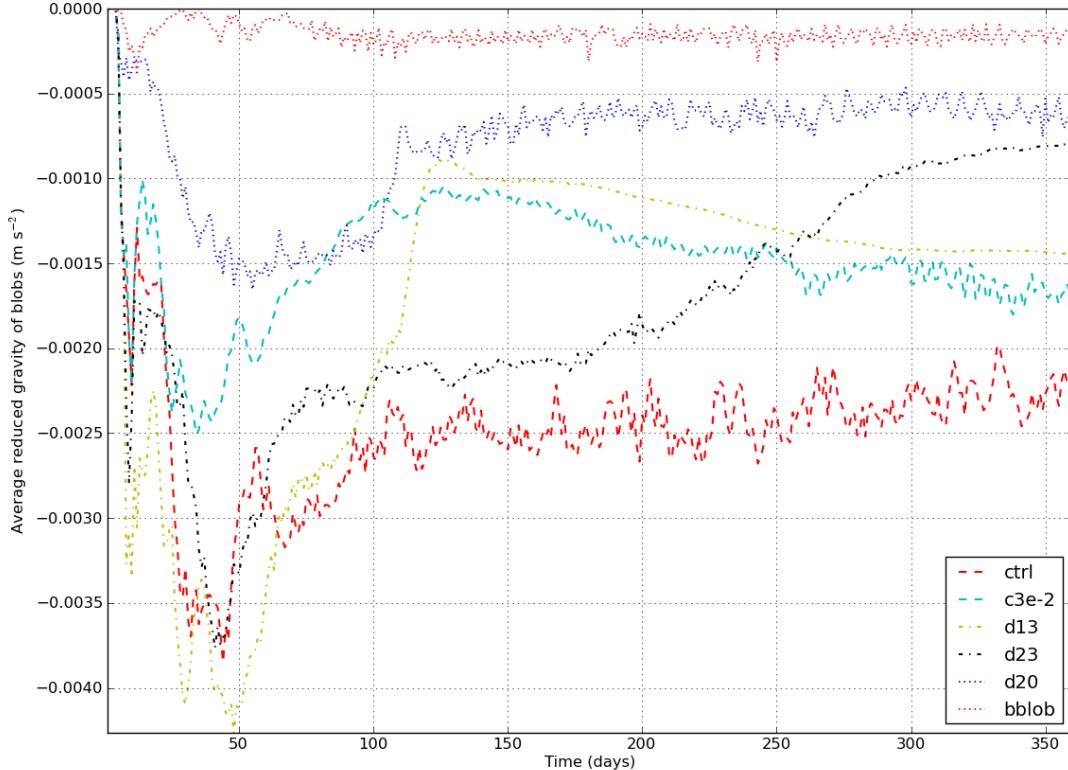


Figure 12.17: The time evolution of the average reduced gravity of the blobs for a representative subset of experiments. The reduced gravity of the blobs is not calculated using the background density, but rather, the density of the E system at the point where the blob is (i.e. it is interpolated from the surrounding E system grid points – see Section 7.1.8 for details on the interpolation scheme).

The reduced gravity of the blobs is calculated using the E system's density interpolated to the blob (rather than the background density, as is the case for the plume reduced gravity). This is done because it is the blob relative to its surroundings that dictates the Nof speed of the blob. Since the blob is typically inside the plume (which is generally more dense than the background stratification), the reduced gravity relative to the background stratification would be even larger. Most experiments have a similar trajectory to the `ctrl` experiment, with a few exceptions, shown in Figure 12.17.

For both the average blob reduced gravity and the average blob speed, most experiments track the time-series of the `ctrl` experiment quite closely. Those experiments that do not are plotted along with the `ctrl` experiment. The small average reduced gravity

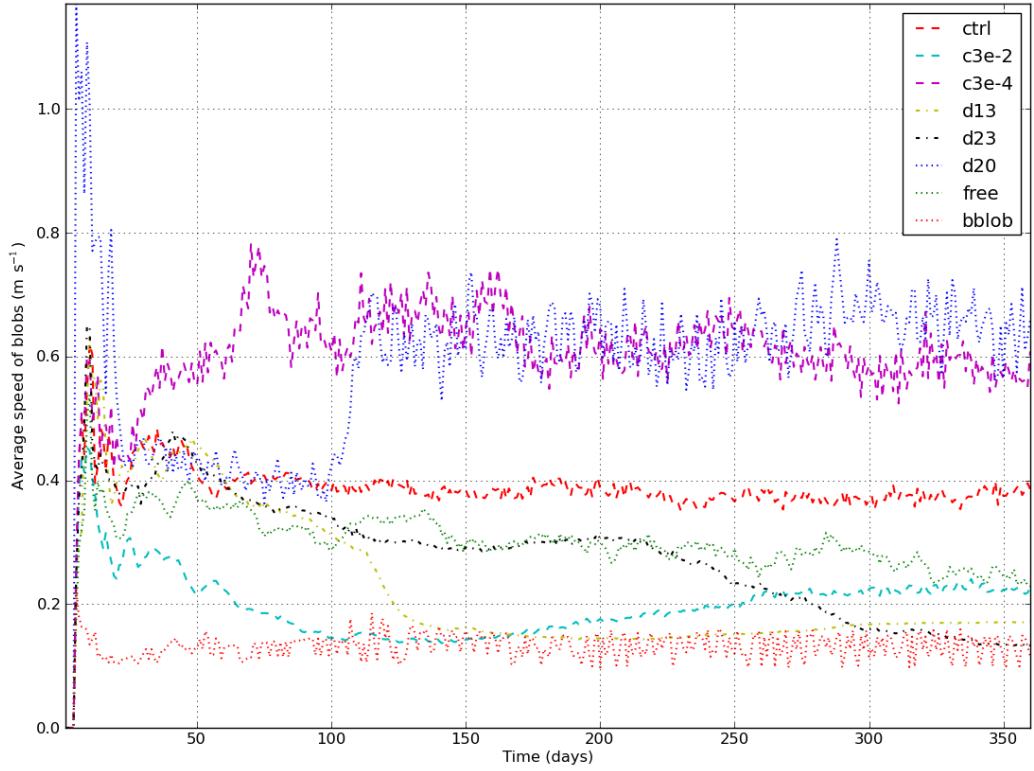


Figure 12.18: The average speed of blobs for a representative subset of experiments. The bottom friction greatly affects the average speed of blobs, as is evidenced when comparing the `c3e-2`, `ctrl` and `c3e-4` experiments. In the evaluation of blob speed, it was noted that there is an undesirable relationship between blob mass and blob speed, in which the drag term of the momentum equation is proportional to  $m^{-1/3}$ .

and the low blob speed in the `bblob` experiment is likely due to two factors. One is the Nof speed, which says that a low reduced gravity will produce small speeds, while the second is that most of the blobs in the `bblob` experiment are destroyed shortly after creation (see Figure 12.9(a)), before they have had a chance to get up to full speed.

The `d20` experiment has a small reduced gravity, but, has a relatively large speed, both initially and then a dramatic increase at day  $\sim 100$ . Interestingly, the increase in speed is concurrent with a dramatic increase in mass (Figure 12.12(a)). This indicates that there is a relationship between the blob speed and blob mass. This relationship can be seen by examining the drag term of the bottom blob momentum Equations (5.18)

$$\dot{\mathbf{x}} \propto -\dot{\mathbf{x}} C_d |\dot{\mathbf{x}}| / h_L \quad (12.13)$$

where the height of the blob is calculated using Equation (5.19b) ,

$$h_L = \left( \frac{6 m}{\pi \rho_L} \right)^{\frac{1}{3}}. \quad (12.14)$$

Thus, we can see that the drag term scales with  $m^{-\frac{1}{3}}$ . Therefore, a larger mass decreases the drag term, while a smaller mass increases the drag term. This explains why the **d20** experiment has blobs with a high speed despite having a small reduced gravity, and why there is an increase in the average speed of blobs concurrent with an increase in the average mass of blobs. This relationship appears to be a shortcoming of the formulation, and, a more physically based formulation for finding the blob height would be a desirable improvement on the present formulation.

The **c3e-2** experiment, which has a large coefficient of blob bottom drag, has an average blob speed that is approximately half that of the **ctrl** experiment for most of the experiments. The **c3e-2** experiment has a smaller reduced gravity than the **ctrl** experiment. While the smaller reduced gravity is due, in part, to a lower blob density (Figure 12.13) it would mostly be due to the deeper penetration of the blobs in the **c3e-2** experiment (Figure 12.8(a)), arising from a greater downslope position. The other contributing factor to the slow speed is the increased bottom drag itself.

The **d13** and **d23** experiments, which decrease the initial size of blobs compared to the **ctrl** experiment, have their changes in average blob speed mirroring the change in the average blob reduced gravity. The changes in the reduced gravity to the **d13** and **d23** experiments do not mirror changes in the average blob density (Figure 12.13) and thus, must be caused by some other mechanism. There is also an obvious relationship between the average blob mass (Figure 12.12) and the speed, which is linked to the relationship discussed in Equation (12.13).

While there has been an undesirable relationship uncovered between the blob mass and the blob speed during the analysis of these experiments, it is still possible to make robust conclusions about how different parameters affect the blob speed (such as the coefficient of bottom drag for blobs), and, how there is a clear relationship (via the Nof speed, as one would expect) between the reduced gravity of the blobs and the speed of the blobs.

#### 12.4.6 Plume Depth

Figure 12.19 shows the mean plume depth of a subset of experiments. Most of the blob experiments has similar mean plume depths, with the **ctrl** experiment being fairly representative of those experiments. The anomalous experiments are the **free** and **bblob** experiment.

By and large, it can be seen that overflow parameterisations deepen the plume when compared to experiments without any parameterisation. The modified Campin and Goosse (1999) parameterisation appears to be the most effective at deepening the mean plume depth, however, as noted in Section 12.4.4 that parameterisation has shortcomings in that it inhibits the along slope transport of the plume due to a small reduced gravity. Similarly the **BD97** experiment, which features the parameterisation of Beckmann and Döscher

(1997) deepens the plume, but, as noted in Section 12.4.1 the isotropic diffusion that facilitates the deepening of the plume creates some spurious behaviour. The original scheme of Campin and Goosse (1999) has a limited effect on the mean plume depth when compared with the modified version of their scheme.

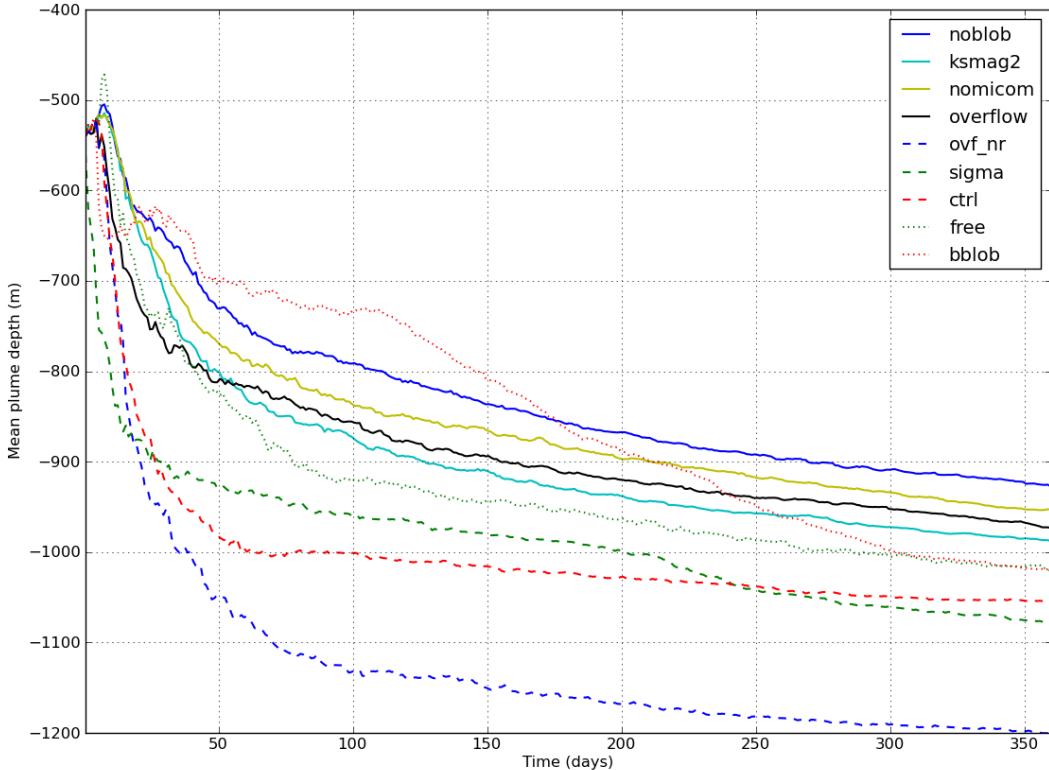


Figure 12.19: The mean plume depth (m) for a representative subset of experiments. Note that the maximum depth of the domain is 3600m. By and large, overflow parameterisations assist in deepening the mean plume depth – a known shortcoming of level coordinate models.

The **ctrl** experiment (and all other blob experiments, sans the **free** and **bblob** experiments) appears to have been effective in increasing the depth of the plume, without slowing down the plume speed. In fact, it was shown in Section 12.4.4 that the blobs greatly assist in the along slope transport of material.

The two experiments that exhibited behaviour that was markedly different to that of the **ctrl** experiment is the **free** experiment, in which blobs that satisfy the separation condition are made into free blobs, and the **bblob** experiment, where a large initial mass of blobs is enforced. The **free** experiment’s mean plume depth largely tracks that of the **ctrl** experiment, except is consistently shallower. This could be because the free blobs are rising from the bottom and are detraining (thereby transferring their properties back to the E system) at a shallower depth than if they were destroyed once they satisfy the

separation condition.

#### 12.4.7 Blob Age and the Number of Blobs

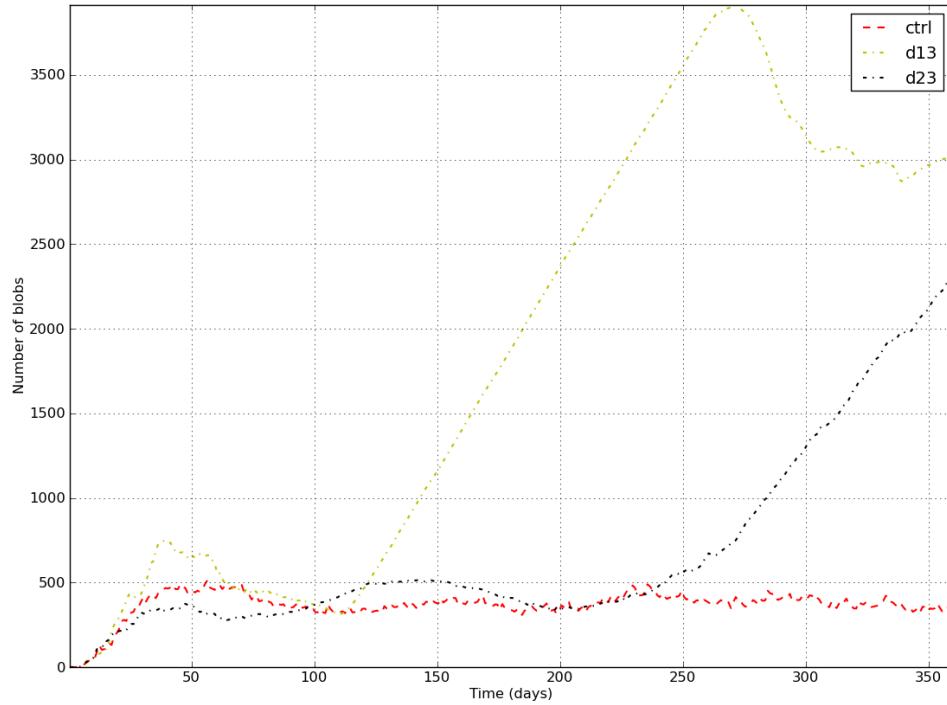
Here, the number of blobs and the age of blobs in each simulation is investigated and inferences about the effect of different parameter choices on those variables is made. The number of blobs is shown in Figure 12.20. The reason for an increase or decrease in blob numbers must be because there is an imbalance between blobs that are being created and blobs that are being destroyed (whether that be by fully detraining, grounding, etc.).

In the DOME test case, blobs are formed when the on shelf/off shelf density difference threshold is exceeded. For most experiments, this threshold is zero (i.e. if the off shelf water at the bottom of the deep ocean column is less dense than the on shelf water, a blob is formed – see Section 5.3.1). Blobs can be destroyed in multiple ways. One way is to fully detraining, that is, a blob's properties have been returned to the E system via detrainment, over the lifetime of the blob. Blobs may also ground (that is, they enter a water column with no wet grid cells). Grounding is a common occurrence in the DOME test case because of the solid wall boundary conditions at the western boundary. Blobs may also separate from the bottom if the ambient fluid that they are in becomes more dense than the blob. Blobs that separate are destroyed in all experiments in the DOME test case with the exception of the `free` experiment. For free blobs, a blob may also be destroyed if it penetrates the free surface (this is not possible for bottom blobs). Finally, a blob may also be destroyed if the grid cell mass constraint, described in Section 6.5.3, is violated. That is, if the mass of material in the L system exceeds some proportion of the total mass allowed in the grid cell, a blob is destroyed and the properties of the blob is returned to the E system.

As can be seen, there are large differences in the number of blobs between the experiments, with the `d13` and `d23` experiments creating large numbers of blobs, while other experiments tending to create less. The reason for the increase is a particularly large number of blobs are created and go in a continuous stream from the area of creation to the western wall, where they are destroyed (not shown). Reflecting the large number of blobs in the `d13` and `d23` experiments is an increase in blob age, shown in Figure 12.21.

Most other experiments have a similar number of blobs and similar evolution of the age of blobs to the `ctrl` experiment, however, there are some other exceptions which are shown in Figures 12.20 and 12.21. The number of blobs in the `ctrl` experiment increases steadily in the `ctrl` experiment for approximately the first 50 days. Reflecting this, the age of the blobs increases over this period too. Both the number of blobs and the age of the blobs then remain relatively constant for the rest of the experiment.

There is a significant decrease in the number of blobs in the `d20` experiment at about day 100. This decrease is concurrent with a large decrease in the average age of blobs in the `d20` experiment as well. This indicates that there are a sizeable number of “older”



(a) The relatively large number of blobs in the **d12** and **d23** experiment is due to an imbalance between the number of blobs created and destroyed.



(b) There are various factors that affect the number of blobs. These can include transient behaviour of the simulation, the value of the detrainment parameter and the threshold density difference at which blobs are formed.

Figure 12.20: The number of blobs in existence at any given time.

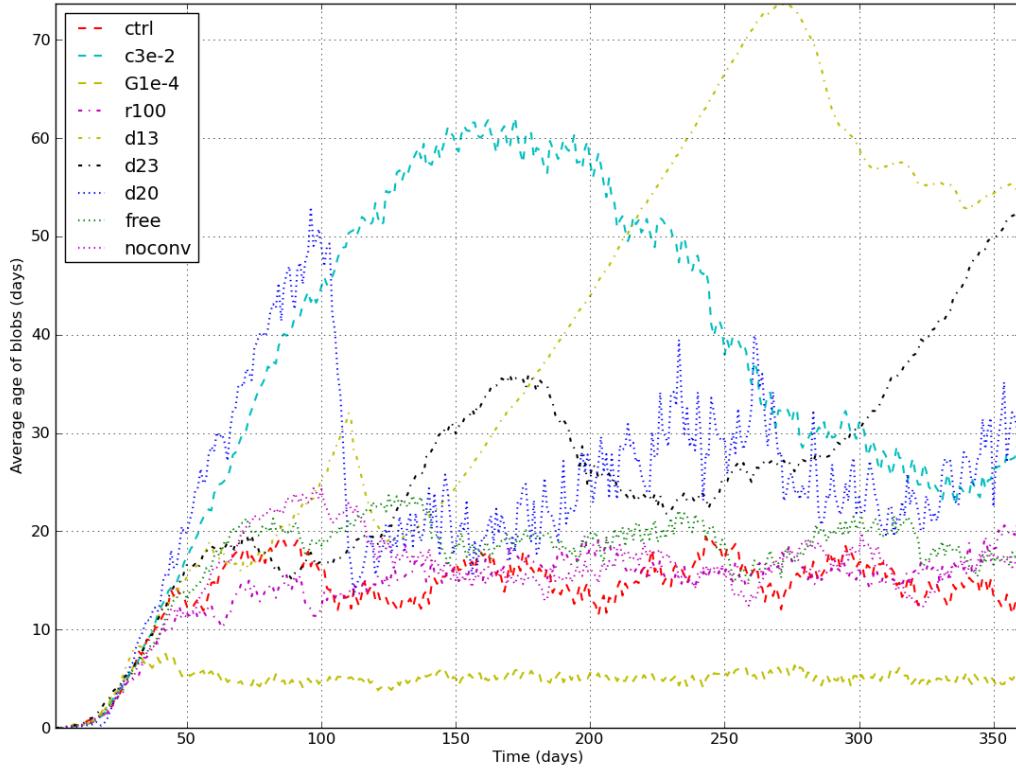


Figure 12.21: The average age of blobs (i.e. how many days it has been since their creation). The average age of blobs is related to the number of blobs in existence (Figures 12.20), with many of the same factors influencing the age as what influences the number.

blobs which are destroyed. Indeed, there is a sizeable group of blobs that interact with the western wall at approximately day 100 in the experiment (not shown). The group is formed not long after the experiment is started, and travels from near the embayment all the way to the western boundary, where they all ground at approximately the same time.

The `noconv` experiment, which has no convective adjustment scheme, initially produces a large number of blobs, however, the number drops again by about day 130 to be comparable to the `ctrl` experiment. There is not a large concurrent decrease in the age of blobs when the decrease in the number of blobs occurs. This indicates that it is blobs that are close to the average age that are being destroyed.

The `G1e-4` experiment and the `r100` also has appreciably fewer blobs than most other experiments. There are, however, different reasons for their similar number of blobs. As is shown in Figure 12.22, the `G1e-4` experiment has an appreciably higher number of blobs that fully detrain (and are thus, destroyed) than all other experiments. This explains the relatively low number of blobs shown in Figure 12.20 for the the `G1e-4` experiment. The relatively large number of blobs that detrain is due to the large detrainment parameter,  $\Gamma$ ,



Figure 12.22: The average number of blobs that fully detrain per timestep. The values are not necessarily integer values because the average is taken over a model day. The detrainment parameter is the largest influence, however, the ability for blobs to separate from the bottom also boosts the number of blobs that fully detrain.

in the **G1e-4** experiment. The **r100** experiment, on the other hand, forms far fewer new blobs than most other experiments, as is evidenced in Figure 12.23. The **r100** experiment creates fewer new blobs because of the relatively large density threshold required to create a new blob.

The **free** experiment, in which blobs that satisfy the blob separation condition become free blobs, also has appreciably more blobs than the **ctrl** experiment (Figure 12.20). It also has marginally older blobs than the **ctrl** experiment, as is shown by the average blob age in Figure 12.21. The **free** experiment having older blobs than the **ctrl** experiment is likely due to the fact that blobs that satisfy the separation condition in the **ctrl** experiment are destroyed, while they are not destroyed in the **free** experiment. Thus, blobs that would be destroyed in the **ctrl** experiment continue to exist as free blobs in the **free** experiment. As shown in Figure 12.23 the **free** experiment also has more blobs than the **ctrl** experiment that fully detrain, which again is a result blobs not being destroyed when the separation condition is satisfied, and instead, continuing to exist until they are fully detrained.

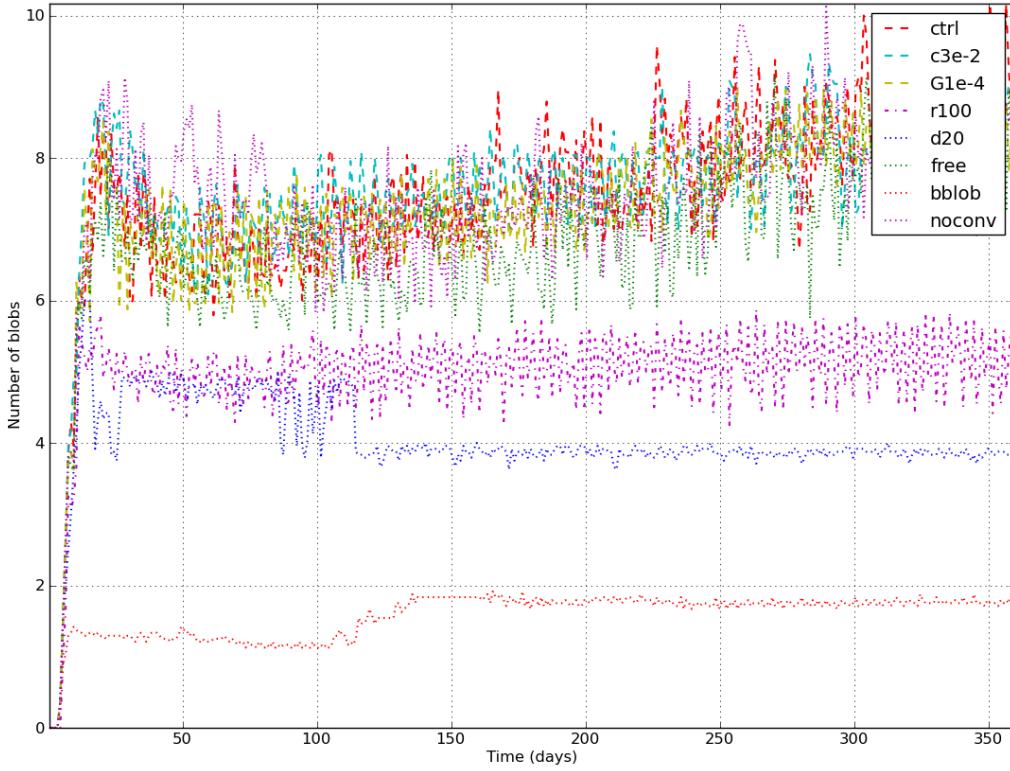


Figure 12.23: The average number of new blobs created per timestep. The values are not necessarily integer values because the average is taken over a model day. Experiments that form very massive blobs also tend to form fewer new blobs. Also, the **r100** experiemnt, which has a large density difference threshold for the formation of blobs, creates fewer new blobs.

The **c3e-2** experiment, which has a high bottom coefficient of drag, has a relatively large number of blobs (Figure 12.20) that are also relatively old (Figure 12.21) when compared to the **ctrl** experiment. This could be because the blobs are travelling slower (see Figure 12.17) which means that they take longer to get to a stage where a destruction condition is satisfied (e.g. grounding at the western boundary).

#### 12.4.8 Compute Time

The average compute time per core for each experiment is shown in Table 12.3. All experiments were run for 360 model days on 8 cores. The domain decomposition is the same across all experiments with there being four compute domains in the longitudinal direction (consisting of 13, 12, 12 and 13 grid points) and two compute domains in the latitudinal direction (consisting of 10 and 9 grid points). Thus, the smallest compute domain consists of 5400 grid points, while the largest compute domain consists of 6500 grid points (when taking into account that there are 50 vertical levels). The code was run on the Sun Constellation cluster, Vayu machine at the Australian National Computational Infrastructure National Facility. The Sun Constellation cluster has 1492 nodes in Sun X6275 blades. The

code was compiled using the Intel Fortran Compiler version 12.0.4.191.

Examining the total time taken (less the time for the diagnostic module) for each of the experiments in Table 12.3 it can be seen that, with the exception the BD97 experiment, the total time taken for each experiment is  $990s \pm 11\%$ . All of the blob experiments, as well as the gm and BD97 experiments, have a greatly reduced computational burden for the tracer routine. This reduction arises due to a much smaller requirement in the tracer advection routine. It is unclear as to why there is such a dramatic decrease in compute time required by the tracer advection scheme when using either the Lagrangian blobs, the Gent and McWilliams (1990) or Beckmann and Döscher (1997) parameterisations.

There is a reduction in the amount of time spent in the velocity and barotropic modules when using the Lagrangian blob scheme. The noblob experiment spent an additional 52s in the velocity module and an additional 127s in the barotropic module, when compared to the ctrl experiment. It is unclear as to the reason why there is such a reduction when using the Lagrangian blobs. Preliminary tests were run on another super computer and there was not a corresponding decrease in compute time in these modules. This leads to speculation that the changes could be compiler and/or hardware dependent.

A source of increased computational cost of the blob scheme is the thickness module, with most experiments using the Lagrangian blob scheme spending 200-300s in the thickness routines, while the time taken in the noblob experiment was 165s. The additional time in the thickness module for the experiments using the Lagrangian blob scheme is unsurprising as there is a number of additional variables and calculations required in order to handle the partitioning of mass between the E and L systems.

Overall, the Lagrangian blob scheme takes approximately 31% of the compute time of the ctrl experiment, spending 332s in that module. By comparison the gm experiment, which uses the Gent and McWilliams (1990) neutral physics parameterisation, spent nearly 250s in the neutral physics module which was approximately 24% of the compute time of the experiment. The CG99 and NoReturn experiments overflow parameterisations are very cheap to run, however, the scheme of Beckmann and Döscher (1997) took 236s and did not have a corresponding decrease in computational cost in other areas of simulation. Thus, in the DOME test case, all of the experiments that use the Lagrangian blob scheme are significantly cheaper, by at least 312s, than the BD97 experiment.

As shall be seen in Section 13.3.5, the time taken by the blob modules in the bowl test case is significantly greater than that in the DOME test case. This is because relatively few blobs are formed in the DOME test case, since the blobs are predominantly formed in a limited area just to the south of the embayment. The limited number of blobs helps to keep the computational requirements of the Lagrangian blob modules in the DOME test case to a relatively low level. As is illustrated in the bowl test case, the number of blobs

Experiment	Total	Tracer	Tracer Adv.	Velocity	Thickness	Barotropic	Overflows	Neutral Physics	Blobs
noblob	987.8	285.8	254.0	114.7	144.6	272.1	0.1	0.0	0.0
gmn	1039.9	56.8	31.9	145.7	154.9	247.3	0.1	249.9	0.0
hdrag	969.8	300.2	268.9	103.1	157.5	246.8	0.1	0.0	0.0
ksmag2	940.0	221.9	194.7	163.9	141.8	234.2	0.1	0.0	0.0
ksmag3	1008.7	293.9	262.7	158.4	155.1	224.8	0.1	0.0	0.0
nomicom	889.9	245.5	216.9	137.2	143.8	203.4	0.1	0.0	0.0
overflow	996.5	278.8	252.6	115.1	158.2	253.2	4.5	0.0	0.0
ovf_nr	905.9	245.9	217.1	101.4	153.7	226.9	5.8	0.0	0.0
sigma	1394.8	134.9	32.5	323.8	172.4	294.3	236.2	0.0	0.0
ctrl	1071.4	78.3	35.6	62.5	386.1	144.8	0.1	0.0	332.0
c3e-2	994.3	76.3	33.9	58.3	241.5	145.2	0.1	0.0	370.1
c3e-4	887.3	75.8	34.3	63.8	191.7	145.0	0.1	0.0	279.6
G1e-4	1039.7	78.9	33.6	88.5	186.6	144.2	0.1	0.0	343.1
G1e-5	1082.7	76.5	33.5	75.9	244.2	145.0	0.1	0.0	398.0
G1e-7	997.0	75.5	33.4	96.8	210.7	145.2	0.1	0.0	301.1
G1e-8	911.8	76.0	33.6	62.0	219.3	144.6	0.1	0.0	299.1
r001	1012.2	76.4	35.9	57.9	251.9	144.5	0.1	0.0	370.1
r010	1025.4	73.5	33.6	76.9	253.8	144.8	0.1	0.0	340.4
r100	1046.3	73.4	33.4	67.4	273.3	144.7	0.1	0.0	371.1
d13	1060.8	74.2	33.4	63.7	296.7	144.9	0.1	0.0	380.5
d23	1018.3	80.5	34.8	57.9	239.9	145.1	0.1	0.0	387.3
d20	972.1	80.5	34.8	63.4	248.6	145.3	0.1	0.0	324.0
free	1036.7	80.4	34.4	63.1	292.5	145.0	0.1	0.0	355.2
bblob	992.0	81.0	33.8	79.3	213.4	145.1	0.1	0.0	317.9
adiab	974.5	81.0	34.8	61.3	268.0	144.8	0.1	0.0	333.2
noconv	1031.1	79.5	35.3	88.1	238.5	144.8	0.1	0.0	341.2

Table 12.3: The average amount of time taken (in seconds) per core for all experiments up to day 360. All experiments were run with 8 cores. The first column is the total time taken, less the time taken in the diagnostic module. Each column after that is the average time spent in routines associated with the column heading. Tracer Adv. stands for tracer advection, and, is a component of the tracer module. The list of modules is not comprehensive and the sum of the columns does not add up to the total.

in a simulation has a strong influence on the amount of computational time taken by the scheme.

## 12.5 Summary

There are fundamental differences between the experiments that use the Lagrangian blob schemes and those that do not. The Lagrangian blob scheme affects all aspects of the DOME simulations, from the properties of the plume itself through to the circulation of the E system.

### 12.5.1 Friction Scheme

The grid dependent Laplacian friction scheme of Smagorinsky (1963) was compared to a constant friction scheme to ascertain the effect of the friction scheme of the properties of the plume in the absence of overflow parameterisations. It was found that the Smagorinsky friction scheme is effective in getting the plume further down the slope, but, in most other ways degrades the properties of the plume. In particular, the Smagorinsky scheme appreciably reduces the speed of the plume front.

### 12.5.2 Traditional Overflow Schemes

Some traditional overflow schemes are discussed in Section 2.4, of which several were tested. It can be seen that the scheme of Beckmann and Döscher (1997) improves some plume properties. The scheme makes the reduced gravity smaller (Figure 12.14) which would otherwise slow the plume down, however, the isotropic diffusive nature of the scheme means that the diffusion process that is a part of the scheme assists in moving the material along the slope (i.e. it is not an advective process, as is the case in the `noblob` experiment). The diffusive nature of the scheme also means that there are some spurious results, most notably being the isotropic way that the plume spreads down and along the slope without regard to rotation (Figure 12.4(a)).

The overflow scheme of Campin and Goosse (1999) and the “no return” version of their scheme (described in Section 4.5) tends to get material down slope (Figures 12.5(a) and 12.6(a)), thereby deepening the mean plume depth (Figure 12.19), however in doing so, it gives the plume a smaller reduced gravity (Figure 12.15), which slows the along slope velocity of the plume. Unlike the scheme of Beckmann and Döscher (1997), there is no mechanism to increase the along slope velocity. As a result of the small reduced gravity, the along slope progress of the plume is slower, as shown in Figure 12.15.

### 12.5.3 The Lagrangian Blobs

In terms of the experiments using the Lagrangian blob scheme, some general observations that can be made are that the blobs tend to increase the mass of the plume (Figure 12.10), increase the along slope speed of the plume front (the slope of the curves in Figures 12.15) and deepen the mean depth of the plume (Figure 12.19). The increased plume front speed is a result of the reduced gravity of the blobs, rather than the reduced gravity of the combined E and L system properties of the plume (see Figures 12.14 and 12.17).

The experiments using the Lagrangian blob scheme, by and large, have approximately the same computational requirements as experiments that do not use the Lagrangian blobs (to within 11%). The Lagrangian blob scheme uses approximately 30% of the compute time (not including E system diagnostics). In comparison, the Gent and McWilliams (1990) neutral physics scheme uses approximately 25% of the computate time (Table 12.3). It is noted, however, that the DOME configuration results in a relatively low computational load for the blobs, due to the relatively small number of blobs produced, which in turn is a result of the limited area, just to the south of the embayment, where blobs are generally formed. The bowl experiment, described in Chapter 13, is a configuration that produces a large number of blobs, and, the increases in computational load is more significant.

### Bottom Friction

Figure 12.15 shows how an increased dimensionless bottom friction decreases the plume front speed slightly (Figure 12.15). This arises due to a slightly slower blob speed, as shown in Figure 12.18. Conversely, decreasing the coefficient of friction slightly increases the average blob speed. The most striking difference, however, is illustrated when comparing the centre of mass of the plumes in the `ctr1` experiment, which has a coefficient of bottom drag of  $C_d = 3 \times 10^{-3}$  (Figure 12.2(a)) with that of the `c3e-2` experiment, which has a coefficient of bottom drag of  $C_d = 3 \times 10^{-2}$ . This indicates that an increase in the coefficient of friction creates a more downslope trajectory for the plume. Such a result is to be expected, since increased friction helps to break the geostrophic balance that makes the plume want to follow isobaths (Griffiths, 1986). The variation of the coefficient of friction does not have an appreciable effect on the required computational cost of the blob schemes within the parameter range examined.

One spurious result arising from the formulation of the momentum Equations (5.18) for the bottom blob, is a relationship in which the bottom drag term is proportional to the inverse of the cube root of the mass of the blob. The relationship was noted when examining the similarity in shape between some of the curves in the evolution of the average blob mass in Figure 12.10, and the average blob speed in Figure 12.18.

### The Detrainment Rate

The detrainment rate,  $\Gamma$ , is examined over a wide parameter space in the `G1e-4`, `G1e-5`, `ctrl`, `G1e-7`, `G1e-8` experiments and the `adiab` experiment. There appears to be relatively little effect, except for the `G1e-4` experiment. However, it can be seen that the `G1e-4` experiment has a larger number of blobs that fully detrain (Figure 12.22), which results in noticeably fewer blobs (Figure 12.20) and a significantly lower average age than most other experiments (Figure 12.21).

### Density Threshold for Blob Formation

The density difference threshold was investigated in experiments `r100` ( $\Delta\rho = 0.1$ ), `r010` ( $\Delta\rho = 0.01$ ) and `r001` ( $\Delta\rho = 0.001$ ). There appears to be very little appreciable difference between the `r010`, `r001` and `ctrl` experiments. The `r100` experiment on the other hand is different in several ways. Firstly, the average mass and density of the blobs in the `r100` experiment are appreciably greater than that of the `ctrl` experiment (Figures 12.12 and 12.13). This is a direct result of the Equation (5.22a) for the initial mass of a blob. The `r100` experiment also generally had a lower number of blobs than the `ctrl` experiment (Figure 12.20), due to a lower number of blobs being formed (Figure 12.23). The lower number of blobs formed is a result of the blob formation criterion not being satisfied as frequently, due to the larger density difference threshold.

### Initial Blob Size

The initial size of blobs is examined over a very wide parameter range in these experiments, from the `bblob` experiment that enforces blobs to be one fifth of the mass of the grid cell of origin, to the `d13` experiment that uses the velocity formulation and parameter value of Campin and Goosse (1999) to calculate the initial blob mass. The first thing that can be taken from these experiments is that there are pitfalls to having blobs that are too massive. Blobs that are too massive expose the limitation of the present implementation that only allows bottom blobs to occupy the bottom most grid cell. Allowing blobs to only occupy the bottom most grid cell means that the mass constraints for the combined system discussed in Section 6.5.3 are frequently violated and cause many blobs to be destroyed. There are other issues, however, such as a smaller blob reduced gravity (Figure 12.17) and a very strong induced circulation (Figure 12.9(b)) that provide arguments against having very massive blobs. On the other hand, having blobs that are small seems to increase the number of blobs in the simulation (Figure 12.20).

### Convective Adjustment

The effect of turning off convective adjustment while using the Lagrangian blobs scheme was investigated in the `noconv` experiment. Turning off convective adjustment seemingly

had a negligible effect when compared to the `ctrl` experiment, with there being no appreciably large differences between the results of the `noconv` experiment and the `ctrl` experiment in any of the diagnostics conducted, except that for a period, there were appreciably more blobs in existence than in the `ctrl` experiment (Figure 12.20).

### **Bottom Blob Separation**

Having the ability to have blobs separate from the bottom was investigated in the `free` experiment. The ability of the model to have blobs separating from the bottom boundary. The main effects of introducing the ability to separate blobs from the bottom boundary were to make the mean plume depth shallower, relative to the `ctrl` experiment (Figure 12.19). The other thing that happened was that there were more blobs in the `free` experiment, presumably because in the `ctrl` experiment, blobs are destroyed when they satisfy the separation condition. Thus, blobs that would have been destroyed in the `ctrl` experiment were able to continue as free blobs (thereby increasing the number of blobs in existence). There were no compelling results from the `free` experiment to argue for the necessity of blobs to have the ability to separate from the bottom.

# Chapter 13

## The Bowl Test Case

The bowl test case is based on the bathymetry constructed by Winton et al. (1998) to test overflow schemes, with some alterations to make much of the setup analogous to the DOME test case described in Section 12. As in the DOME test case, the blob parameters that are being tested here are

- the initial blob size,
- the coefficient of drag,  $C_d$
- the detrainment parameter,  $\Gamma$ , and
- the formation condition threshold,  $\Delta\rho$ .

The bowl test case is similar to the DOME test case (see Section 12) in that it was designed to test the fidelity of overflows in ocean climate models, however, it has complementary properties to the DOME test case. Unlike the DOME test case, which reaches a steady state, the bowl test case is an adjustment test, and as such there is no forcing of any kind. The main utility of the DOME test case was to examine the properties of a plume on a uniform slope. The limited source of instability in the DOME test case means that there is only a limited region from which blobs may be formed – thus, limiting their number.

The nature of the bathymetry of the bowl test case means that blobs can be formed over a large area. The unstratified nature of the initial conditions, and the large density contrast between the shelf and the bowl means that the dynamics of the experiment are quite a bit different to that of the DOME test case.

### 13.1 Configuration Details

The model domain spans from  $0^\circ$  to  $20^\circ$  (approximately 1700km) zonally and from  $35^\circ$  to  $50^\circ$  meridionally (approximately 1300km). It has a 500m deep flat shelf in the north of

the domain from  $45^{\circ}\text{N}$ . To the south of  $45^{\circ}\text{N}$  the topography obeys the following equation

$$H = 500 + 3000 \left\{ \exp\left(\frac{-(y-35)^2}{4}\right) \exp\left(\frac{-(y-45)^2}{4}\right) \exp\left(\frac{-x^2}{16}\right) \exp\left(\frac{-(x-20)^2}{16}\right) \right\}, \quad (13.1)$$

as is shown in Figure 13.1. The domain has solid wall boundaries with a no slip condition.

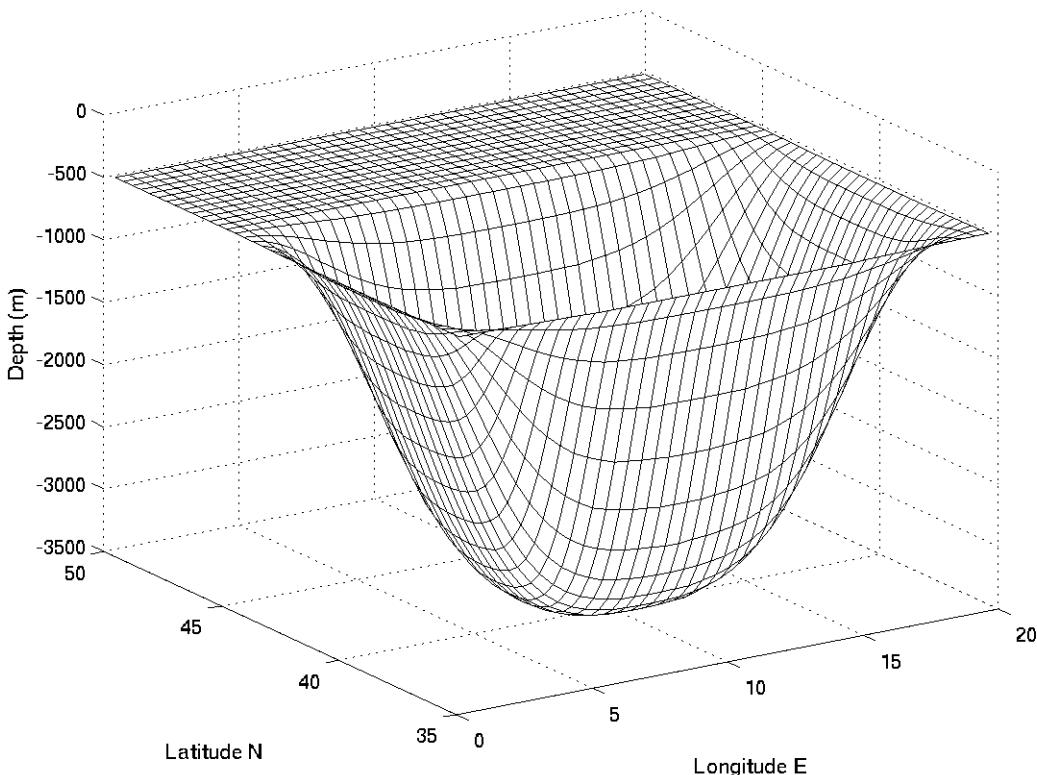


Figure 13.1: A wire frame of the bowl test case, based on the experimental design of Winton et al. (1998). The model uses no slip side wall boundary conditions (the side walls are not depicted here).

Similarly to the DOME configuration, the model uses a constant value for the Coriolis parameter of  $f_0 = 10^{-4}\text{s}^{-1}$ , which is the value at approximately  $43^{\circ}\text{N}$ . The grid has a cells have a horizontal aspect ratio of 1 : 1. The bowl test case is a  $40(x) \times 41(y) \times 50(z)$  grid, which is a resolution of approximately 40km (which equates to approximately  $\frac{1}{2}^{\circ}$  at  $43^{\circ}\text{N}$ ). The grid cells increase linearly from 20m in thickness at the surface to 120m in thickness at the bottom. The timestep of all experiments is 3600s (although, as discussed in Section 7.3.2, the timestep for individual blobs is variable). A constant quadratic bottom drag for the E system is used, with a dimensionless bottom drag coefficient of  $1 \times 10^{-3}$ .

Water on the shelf is set to  $15^{\circ}\text{C}$  while water to the south of  $45^{\circ}\text{N}$  is uniformly  $20^{\circ}\text{C}$ . There is also a passive tracer that is used to tag the shelf water. The dye tracer is initially

set to 1 on the shelf and is initially set to 0 south of 45°N. Density is a linear function of temperature only. The experiment has no forcing or restoring, thus, it is purely a relaxation experiment.

The details of the parameterisations and parameter values for the bowl experiment are virtually identical with the DOME configuration. The details are repeated here in their entirety, for completeness.

No experiments use explicit lateral prescribed diffusion, with stability being ensured through numerical diffusion via the tracer advection scheme. The tracer advection scheme used in both the vertical and horizontal is the multi-dimensional piecewise parabolic method of Colella and Woodward (1984, MDPPM) with the flux limiter described by Lin (2004) is used in both the horizontal and vertical. A constant horizontal viscosity of  $500.0\text{m}^2\text{s}^{-1}$  is used, unless otherwise stated. In the vertical, a constant viscosity of  $0.1\text{m}^2\text{s}^{-1}$  is used. Partial bottom cells are used with the thinnest partial cell allowable being 20% of the thickness of a full cell. A constant vertical diffusivity of  $1.0 \times 10^{-5}\text{m}^2\text{s}^{-1}$  is used. An enhanced vertical diffusion of up to  $10.0\text{m}^2\text{s}^{-1}$  is used in vertically unstable regions, unless otherwise stated.

## 13.2 Experimental Details

All of the experiments conducted using the bowl test case are described in this section. The baseline experiment is the `noblob` experiment and is, in the traditional sense of the term, the “control” experiment. The `ctrl` experiment on the other hand is the control experiment for the blobs. It takes the mid-range of sensible parameter choices from which the values are varied in other blob experiments. The parameter values in the `ctrl` experiment are not chosen for being the optimal or best parameter values.

The experiments used in the bowl test case are largely the same (with a couple of omissions) as those used in the DOME test case described in Chapter 12. All of the experiments conducted are listed in Tables 13.1 and 13.2.

### 13.2.1 Friction

The `noblob` experiment uses a constant horizontal viscosity with a value of  $500\text{m}^2\text{s}^{-1}$ . It has recently been suggested that friction schemes play an important role in diapycnal mixing in overflows (Ilicak et al., 2011a). Here, the effect of friction is investigated by introducing the Smagorinsky Laplacian viscosity (Smagorinsky, 1963; Griffies and Hallberg, 2000). The Smagorinsky Laplacian viscosity is defined according to

$$A_{SM} = (C_s \Delta s)^2 |E| \quad (13.2)$$

where  $C_s$  is the non-dimensional Smagorinsky coefficient,  $\Delta s$  is the horizontal grid spacing and  $E$  is the total deformation rate with units of inverse time. In addition to the

Name	Viscosity	Comments
<code>noblob</code>	Constant $500\text{m}^2\text{s}^{-1}$	None
<code>gm</code>	Constant $500\text{m}^2\text{s}^{-1}$	Gent and McWilliams (1990) neutral physics
<code>ksmag2</code>	Lap Smag* <code>ksmag</code> = 2.0	MICOM velocity scale=0.1
<code>nomicom</code>	Lap Smag† <code>ksmag</code> = 2.0	No background friction
<code>BD97</code>	Constant $500\text{m}^2\text{s}^{-1}$	Beckmann and Döscher (1997) overflow param.
<code>CG99</code>	Constant $500\text{m}^2\text{s}^{-1}$	Campin and Goosse (1999) overflow param.
<code>NoReturn</code>	Constant $500\text{m}^2\text{s}^{-1}$	No return overflow
<code>ctrl</code>	Constant $500\text{m}^2\text{s}^{-1}$	Dynamically active bottom blobs

Table 13.1: Experimental details for the bowl test case for all experiments not using dynamically active blobs and some that do (see Table 13.2 for a description of all experiments using blobs). Lap Smag means Laplacian Smagorinsky friction. The experiment `ksmag2`, which uses the Laplacian Smagorinsky friction scheme, also uses the isotropic MICOM velocity scale for the background friction, while the `nomicom` experiment does not.

Smagorinsky viscosity, a background viscosity may also be defined where

$$A_{min} = U_A \Delta s \quad (13.3)$$

where  $U_A$  is a constant velocity scale. In MOM, a background velocity scale (originating from studies conducted by MICOM) suggests a velocity scale of  $0.1\text{m s}^{-1}$ .

There are two experiments that use a non-constant friction scheme. The `ksmag2` experiment uses a Laplacian Smagorinsky viscosity with a dimensionless parameter of 2.0 and a MICOM velocity scale of  $0.1\text{m s}^{-1}$ , while the `nomicom` experiment is identical, except the MICOM velocity scale is set to zero (thus, there is no background friction).

### 13.2.2 Lateral Diffusion

There is also an experiment that uses the neutrally physics scheme of Gent and McWilliams (1990); Gent et al. (1995), with the main purpose of this experiment to compare the computational cost of the blobs to an existing, widely used parameterisation. The experiment has a constant coefficient of  $\kappa_{GM} = 600\text{m}^2\text{s}^{-1}$  and a constant isoneutral diffusion coefficient of  $\kappa_{iso} = 600\text{m}^2\text{s}^{-1}$  with linear tapering near the surface.

### 13.2.3 Traditional Overflow Schemes

A number of experiments are run in which the bowl test case is used to evaluate the fidelity of some existing overflow parameterisations. A summary of traditional bottom boundary layer schemes is given in Section 2.4.

The `noblob` experiment uses no overflow parameterisations. The `BD97` experiment is identical to the `noblob` experiment, except it uses the embedded sigma scheme of Beckmann and Döscher (1997), while the `CG99` experiment uses the overflow parameterisation

of Campin and Goosse (1999). The **NoReturn** experiment uses the no return overflow scheme, which is based on the scheme of Campin and Goosse (1999), however, the prescribed return flow is not utilised in this parameterisation (described in Section 4.5).

### 13.2.4 Lagrangian Blob Schemes

The **ctrl** experiment uses the dynamically active bottom blob scheme, described in Section 5.3 and is the baseline experiment from which various blob parameters are varied.

The various parameters that control the dynamically active bottom blobs, and the experiments used to examine their sensitivity are shown in Figure 13.2. The **c3e-2** and **c3e-4** experiments vary the non-dimensional coefficient of bottom drag by an order of magnitude larger and smaller from the **ctrl** experiment. The value for the **ctrl** experiment,  $C_d = 3.0 \times 10^{-3}$  is based on observations of skin drag, however, Özgökmen and Fischer (2008) show in numerical experiments that form drag can be approximately an order of magnitude higher than skin drag over rough topography. Thus, if one considers form drag as well, a value of  $C_d = 3.0 \times 10^{-2}$  is not unrealistically large.

Name	$C_d$	$\Gamma$	$\delta$	$\Delta\rho$
<b>ctrl</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1	0.0
<b>c3e-2</b>	$3.0 \times 10^{-2}$	$1.0 \times 10^{-6}$	1	0.0
<b>c3e-4</b>	$3.0 \times 10^{-4}$	$1.0 \times 10^{-6}$	1	0.0
<b>G1e-4</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-4}$	1	0.0
<b>G1e-5</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-5}$	1	0.0
<b>G1e-7</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-7}$	1	0.0
<b>G1e-8</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.0
<b>r100</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.100
<b>r010</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.010
<b>r001</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-8}$	1	0.001
<b>d23</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	2/3	0.0
<b>d13</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1/3	0.0
<b>noconv</b>	$3.0 \times 10^{-3}$	$1.0 \times 10^{-6}$	1	0.0

Table 13.2: Experimental details for the bowl experiments that use dynamically active blobs.  $C_d$  is the coefficient of drag,  $\Gamma$  is the detrainment parameter,  $\delta$  is the fraction of a grid cell participating in an overflow event and  $\Delta\rho$  is the density difference threshold above which a blob is formed.

For all of the experiments using the embedded Lagrangian scheme, a blob's mass is considered “small” when it is less than 1000kg, and all remaining properties are returned to the E system(see Section 5.3). The maximum proportion of a grid cell that the L system may occupy is 70% (see Section 6.5.3). The Runge-Kutta scheme used is that of Cash and Karp (1990) with a local relative truncation error of  $\zeta^* = 0.01$ , a safety factor of  $\varrho = 0.8$  and a minimum step size of  $h_{\min} = 18\text{s}$  (see Section 7.3.2). The maximum allowable detrainment is  $\mathcal{D}_{\max} = 10^{15}\text{m s}^{-1}$  (see Section 5.3) and the frictional parameter

used to calculate the initial mass of bottom blobs is  $\mu = 10^{-4}\text{s}^{-1}$  (see Section 5.3.1).

### Detrainment Parameter

The detrainment parameter,  $\Gamma$ , controls how quickly a blob detains material to the ambient fluid and has units of  $\text{kg m}^2 \text{s}^{-1}$ . Since it is a heuristic variable its values are varied over a wide parameter space from  $\Gamma = 1.0 \times 10^{-4}\text{kg m}^2\text{s}^{-1}$  to  $1.0 \times 10^{-8}\text{kg m}^2\text{s}^{-1}$  in the G1e-4, G1e-5, ctrl, G1e-7 and G1e-8 experiments.

### Threshold Density Difference

The density threshold at which a blob is formed is,  $\Delta\rho$ , is varied from  $0.0\text{kg m}^{-3}$  to  $0.1\text{kg m}^{-3}$  in the ctrl, r001, r010 and r100 experiments. The threshold density difference is essentially a means by which the number of blobs that are in existence can be controlled. In contrast to the detrainment parameter, it controls how many blobs are formed, rather than impacting on how long they live. Again, there is no guidance from the literature as to what this value should be, so a large parameter range is spanned, from  $0.0\text{kg m}^{-3}$  to  $0.1\text{kg m}^{-3}$ .

### Fraction of a Cell Overflowing

The fraction of a grid cell participating in an overflow event,  $\delta$ , is also varied over a wide parameter space. The parameter is based on Campin and Goosse (1999) who use a value of  $1/3$  for their original formulation. For the bottom blob formation, there are certain constraints in the code to ensure that a grid cell is never entirely “drained” of fluid. There is no reason why an entire grid cell cannot participate in the overflow event. As such, the a value of 1 is chosen for the ctrl experiment and the smaller values of  $2/3$  and  $1/3$  are chosen for the d23 and d13 experiments, respectively. Since there are checks and balances in the code to ensure that a blob is never entirely drained of fluid, the value of  $\delta$  can be increased to be greater than 1.

## 13.3 Results

The purpose of the bowl experiment is to examine the properties of the solutions with different parameterisations and to examine the computational cost of each experiment. The bowl configuration is different to the DOME configuration of Chapter 12 in several major ways. The first is that it is not a uniform slope, and thus, the dynamics of the plume are somewhat different. The second major difference relates to the blob scheme directly in that the region where blobs form is over a broad area, and thus, there is a propensity to create a large number of blobs. The experiment is also a relaxation experiment, with no external forcing. Finally, the bulk ocean is unstratified.

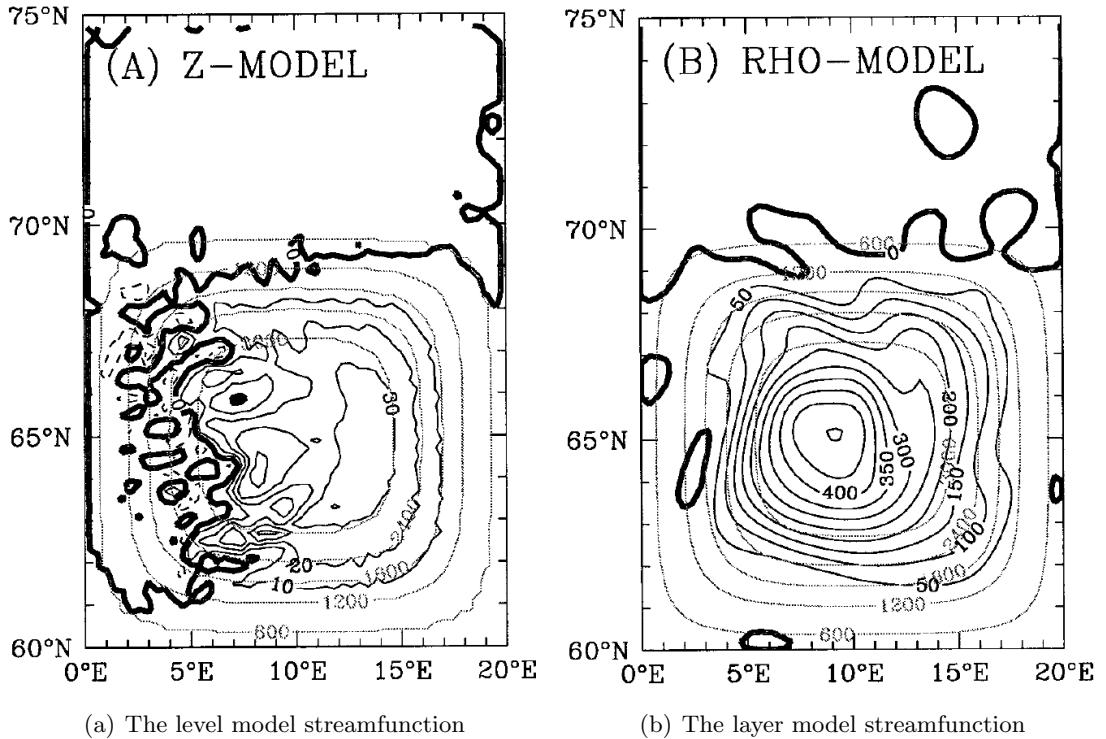


Figure 13.2: The barotropic streamfunction (Sv) of the experiments of Winton et al. (1998) at their day 360. The layered model (b) shows a strong anticyclonic flow centred on the bottom of the bowl, while the level model has much weaker and less coherent flow. From Figure 4 of Winton et al. (1998).

### 13.3.1 Preliminaries

The barotropic streamfunction for the level model and layered model of Winton et al. (1998) are shown in Figure 13.2. Their models have a resolution of approximately  $0.2^\circ$  in latitude and  $0.4^\circ$  in longitude (at  $60^\circ\text{N}$ ) on an  $f$ -plane with  $f = 1.26 \times 10^{-4}$ . For the level model, they use a biharmonic viscosity of  $1 \times 10^{10} \text{m}^4\text{s}^{-1}$ . In this section the bottom salinity and barotropic quasi-streamfunction are plotted for different experiments. The quasi-streamfunction, defined in Equations (9.7), for the various experiments is compared with the results of Winton et al. (1998).

Winton et al. (1998) say that the structure of the flow in the layered model is due to the compression of light fluid by dense fluid filling the bottom and is related to the hypsometric effect (Rhines and MacCready, 1989). There is a large difference between the two solutions, with the level model having flow that is an order of magnitude weaker, and, having much less vigorous sinking. Notwithstanding the significant differences between the present bowl configuration and that of Winton et al. (1998), the present section aims to examine the effect of parameterisations on the flow and tracer distribution in level models, with reference to the results of Winton et al. (1998).

The **noblob** experiment, which has no downslope flow parameterisation, is shown in

Figures 13.3. This is the most comparable experiment to the level model setup of Winton et al. (1998, see Figure 13.2(a)). The solutions are qualitatively similar, with the main difference between the `noblob` experiment’s quasi-streamfunction in Figure 13.3(b) and the level model solution of Winton et al. (1998) in Figure 13.2(a) is that there is a much stronger anti-cyclonic flow in the `noblob` experiment. This is possibly due to the different friction coefficients. There is similarity, however, in so much as there is some cyclonic flow in the western part of the domain, which in the `noblob` experiment, extends around the corner to the south of the domain too. When comparing the quasi-streamfunction to the bottom temperature in Figure 13.3(a), it can be seen that the areas of cyclonic and weak circulation correspond to regions of cold water (i.e. water that has originated on the shelf).

To examine the effect of friction on the `ksmag2` experiment, which uses a Laplacian Smagorinsky friction scheme with the MICOM isotropic velocity scale, the bottom temperature and quasi-streamfunction is shown in Figure 13.4. When comparing the bottom cell temperature, it can be seen that the different friction scheme assists in getting the density signal from the shelf down the slope. This result is in quite a contrast with the DOME test case which indicates that use of the Smagorinsky friction scheme only marginally increases the plume depth (Figures 12.3(a) and 12.19).

There are several things to observe in the quasi-streamfunction, shown in Figure 13.4(b). Firstly, the magnitude of the flow is smaller than that of the `noblob` experiment (Figure 13.3(b)). Similar to the `noblob` experiment, there is some weaker and less coherent flow near the western boundary, however, the largest difference is the relatively strong cyclonic cell that appears near the middle of the bowl. If one takes the solution of the layered model of Winton et al. (1998) as being closer to the “truth,” then, the price paid for improving the downslope transport of tracer is a degradation of the solution for the barotropic flow.

To examine the effect of the MICOM isotropic velocity, another experiment is run with Laplacian Smagorinsky viscosity, however, without the MICOM velocity scale. The experiment is called the `nomicom` experiment. The tracer distribution of the `nomicom` experiment, shown in Figure 13.5(a), is qualitatively similar to that of the `ksmag2` experiment (Figure 13.4(a)), however, there is a very large difference in the quasi-streamfunction, both in the magnitude and the pattern. Absent from the `nomicom` experiment is the strong cyclonic cell apparent in the `ksmag2` experiment, thus, making the pattern of the streamfunction more similar to the layered model solution of Winton et al. (1998), shown in Figure 13.2(b).

The BD97 experiment, which uses the overflow parameterisation of Beckmann and Döscher (1997), is shown at day 360 in Figures 13.6. The diffusive nature of the Beckmann and Döscher (1997) scheme is apparent in the tracer distribution of the bottom cells. Initially, the tracer diffuses down the slope from the north, however, as the tracer is

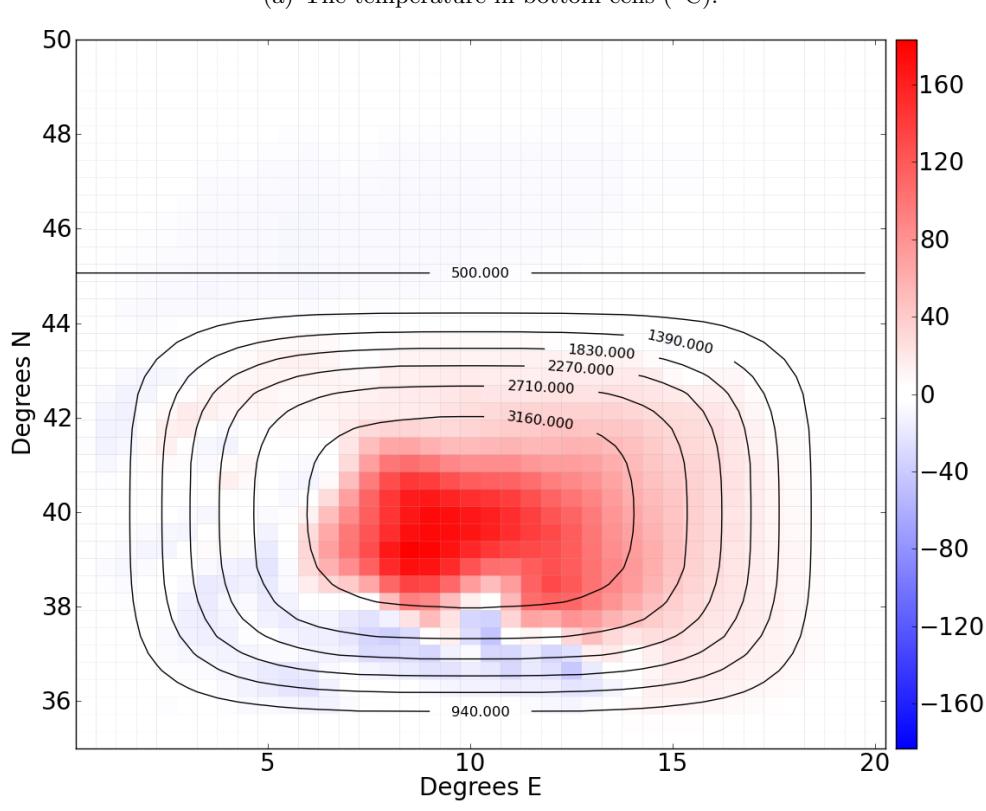
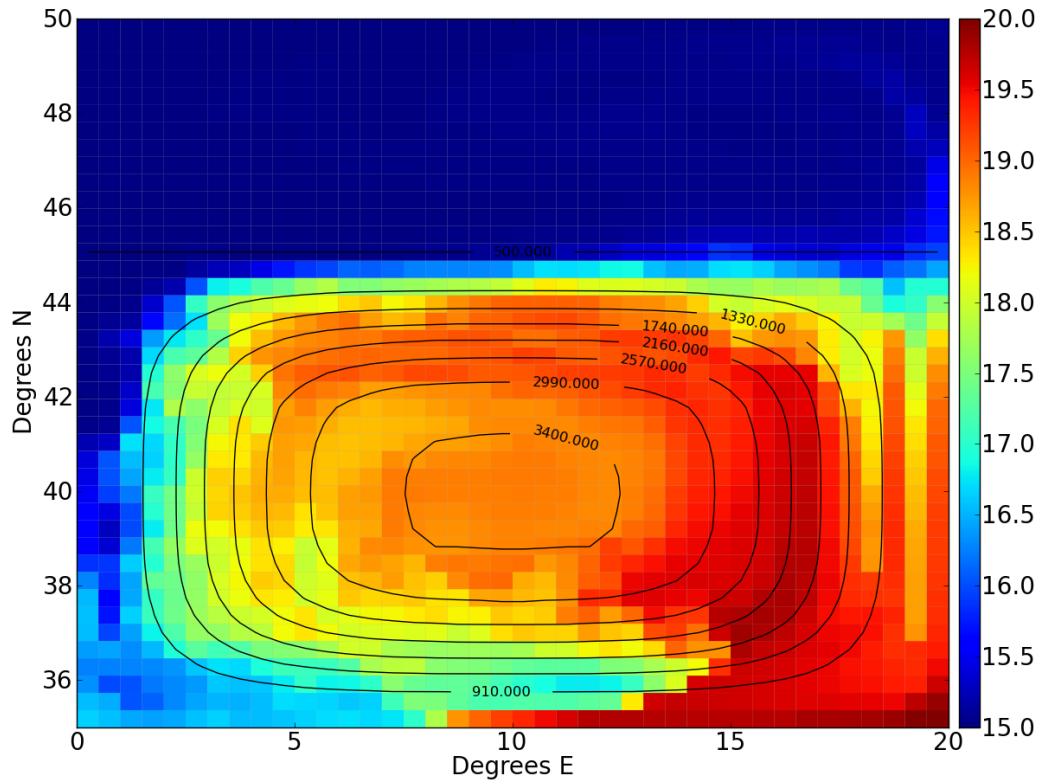


Figure 13.3: Properties of the `noblob` experiment at day 360. The quasi-streamfunction shows much stronger flow compared to the equivalent experiment for Winton et al. (1998) (Figure 13.2(a)), however, it does not have the same degree of coherence as their layered model results (Figure 13.2(b)).

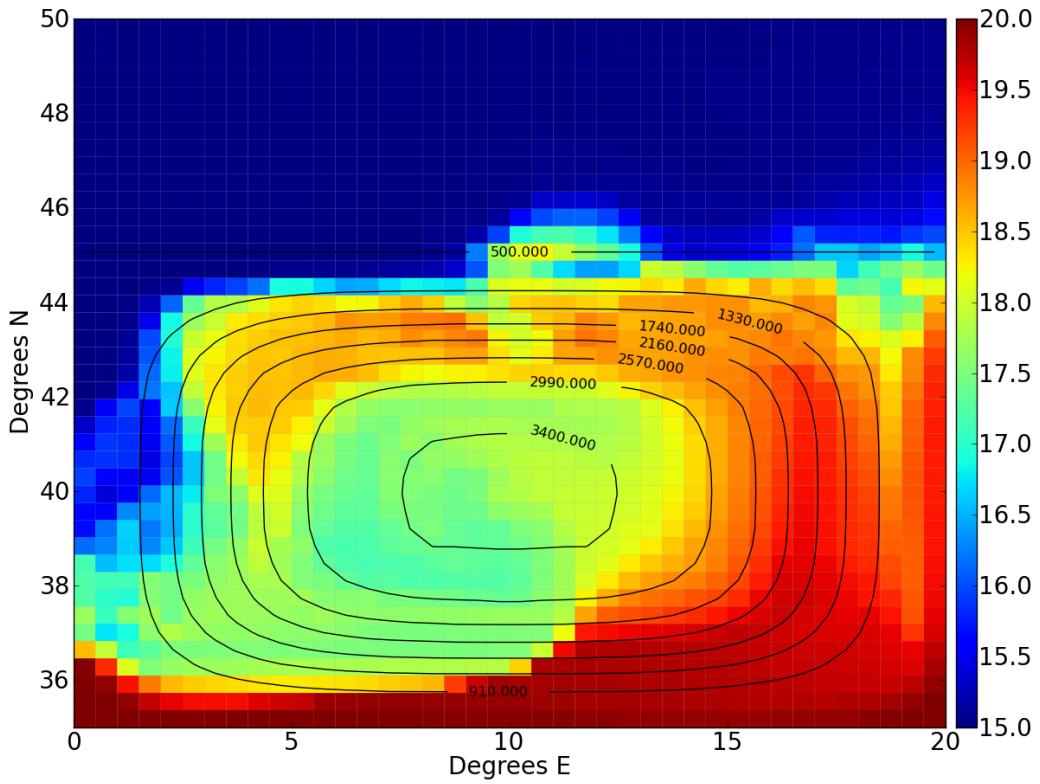
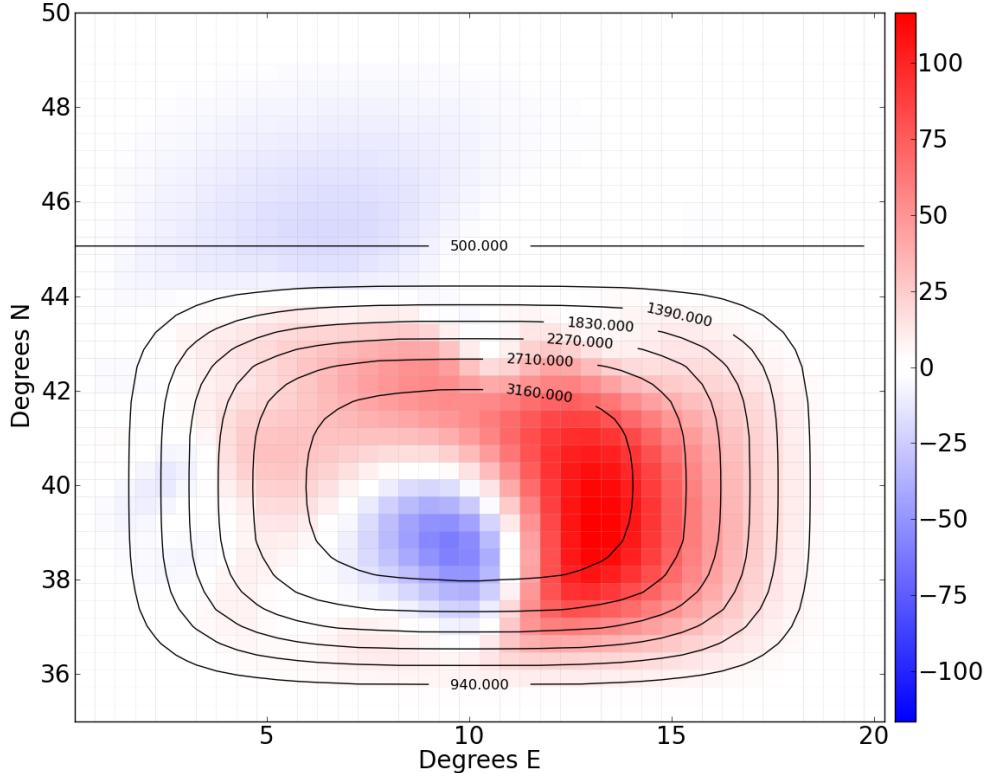
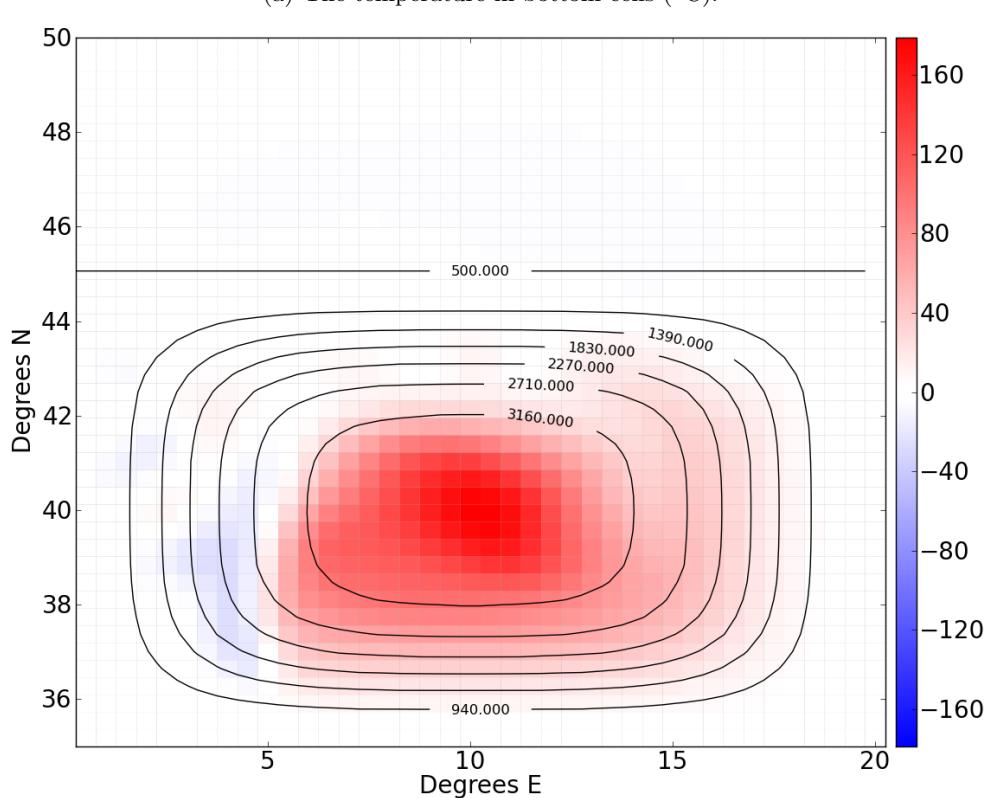
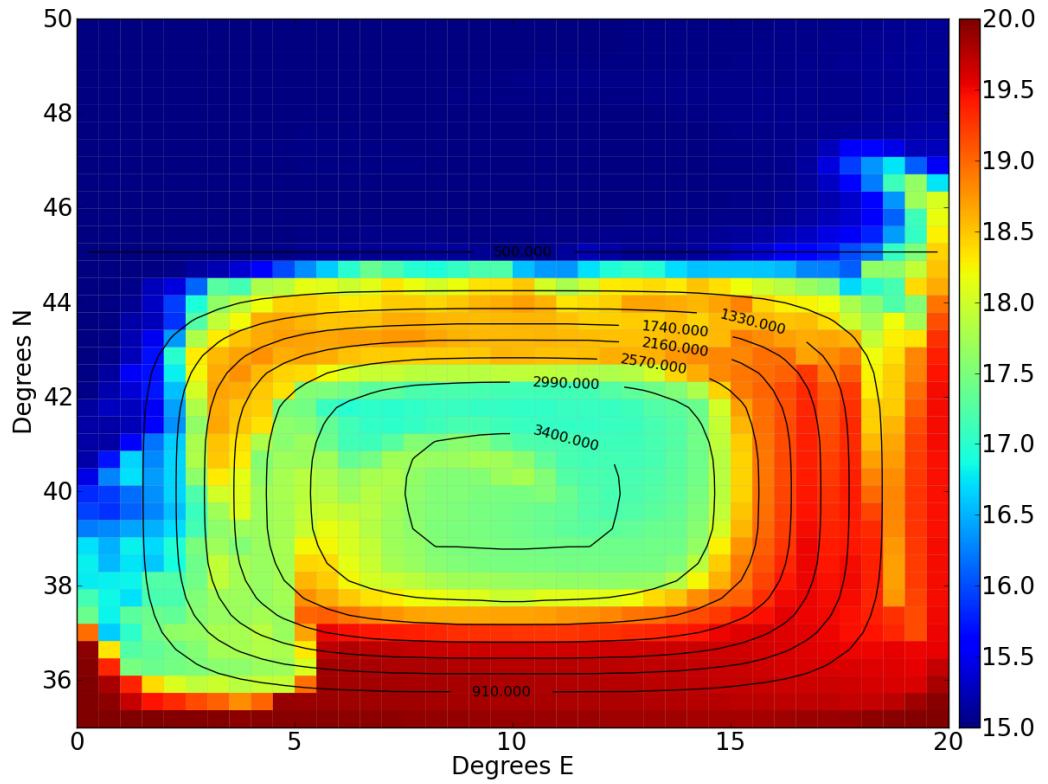
(a) The temperature in bottom cells ( $^{\circ}\text{C}$ ).(b) The barotropic quasi-streamfunction,  $\psi$  ( $10^9 \text{kg s}^{-1}$ ). Note the different colour scale to Figure 13.3(b).

Figure 13.4: Properties of the `ksmag2` experiment at day 360. The friction scheme appears to assist downslope transport of tracer properties when compared to the `noblob` experiment (Figure 13.3(a)). The quasi-streamfunction is also affected by these changes, with a strong cyclonic cell appearing.



(b) The barotropic quasi-streamfunction,  $\psi$  ( $10^9 \text{kg s}^{-1}$ ). Note the different colour scale to previous figures.

Figure 13.5: Properties of the `nomicom` experiment at day 360. Without the MICOM velocity scale, the Smagorinsky friction scheme is effective at getting material down the slope while largely maintaining a large anti-cyclonic cell, centred over the bottom of the bowl.

adverted along the western and southern walls, the scheme diffuses properties down the slope from the west and subsequently from the south as well. This creates a distinctively different tracer distribution to the other experiments.

In terms of the barotropic quasi-streamfunction, the circulation of the **BD97** experiment is much weaker than other experiments. It also has a number of cyclonic features on the western and southern sides of the bowl. If the results of the layered model of Winton et al. (1998) are taken as the benchmark, then, the **BD97** experiment degrades the solution of the circulation.

Using the Campin and Goosse (1999) parameterisation, the **CG99** experiment is qualitatively very similar to the **noblob** experiment both for the tracer distribution and for the quasi-streamfunction. This indicates that the parameterisation has little effect in th bowl configuration. Thus, results from the **CG99** experiment are not shown in this section.

The no return modified Campin and Goosse (1999) scheme, which was described in Section 4.5, is used in the **NoReturn** experiment. Results for the bottom temperature and barotropic quasi-streamfunction are shown in Figures 13.7. Unlike the standard Campin and Goosse (1999) parameterisation, the no return version is significantly different to the **noblob** experiment. Examining the temperature distribution, shown in Figure 13.7(a), it is apparent that the shelf waters have not advanced as far southward along the western wall as any of the previous experiments, with a reasonable amount of mixing apparent both in the bowl and on the shelf. The quasi-streamfunction in Figure 13.7(b) displays a large and strong anti-cyclonic structure centred over the middle of the bowl. This anti-cyclonic structure is similar in strength and shape to that found in the layer model of Winton et al. (1998).

The **ctrl** experiment, which uses the dynamically active bottom blobs (with parameters described in Table 13.2), has its bottom temperature, with blob positions (the white dots), shown in Figure 13.8(a) and the barotropic quasi-streamfunction for the E system only, shown in Figure 13.8(b). The first thing that is noticeable is that there is a large number of blobs formed, particularly when compared to the DOME experiments (for example, Figure 12.7(a)). The reason for the large number of blobs is that they can be formed all around the permieter of the bowl. So, the area over which blobs are formed is a significant proportion of the domain.

The main direction of travel for the blobs is anti-clockwise (i.e. they are travelling in a cyclonic direction). However, as can be seen in the E system's barotropic quasi-streamfunction (Figure 13.8(b)), the circulation of the E system is strongly anti-cyclonic, the the magnitude of the flow being much larger than that found in any previous experiment, and is comparable to the strength found by the layer model of Winton et al. (1998, see Figure 13.2(b)).

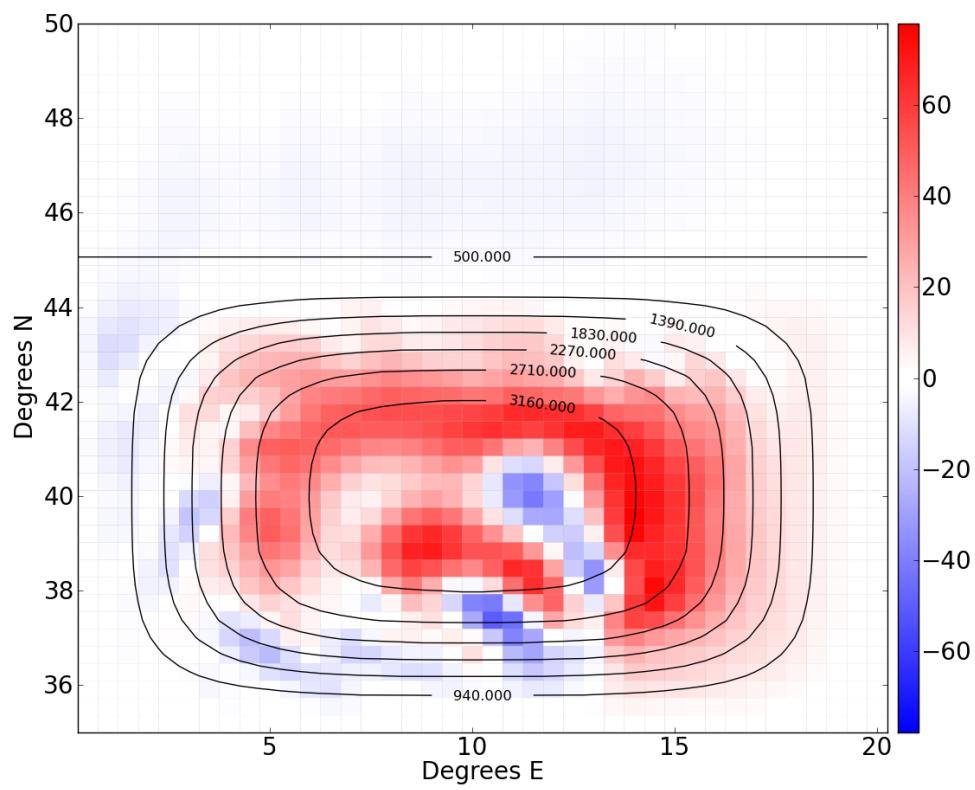
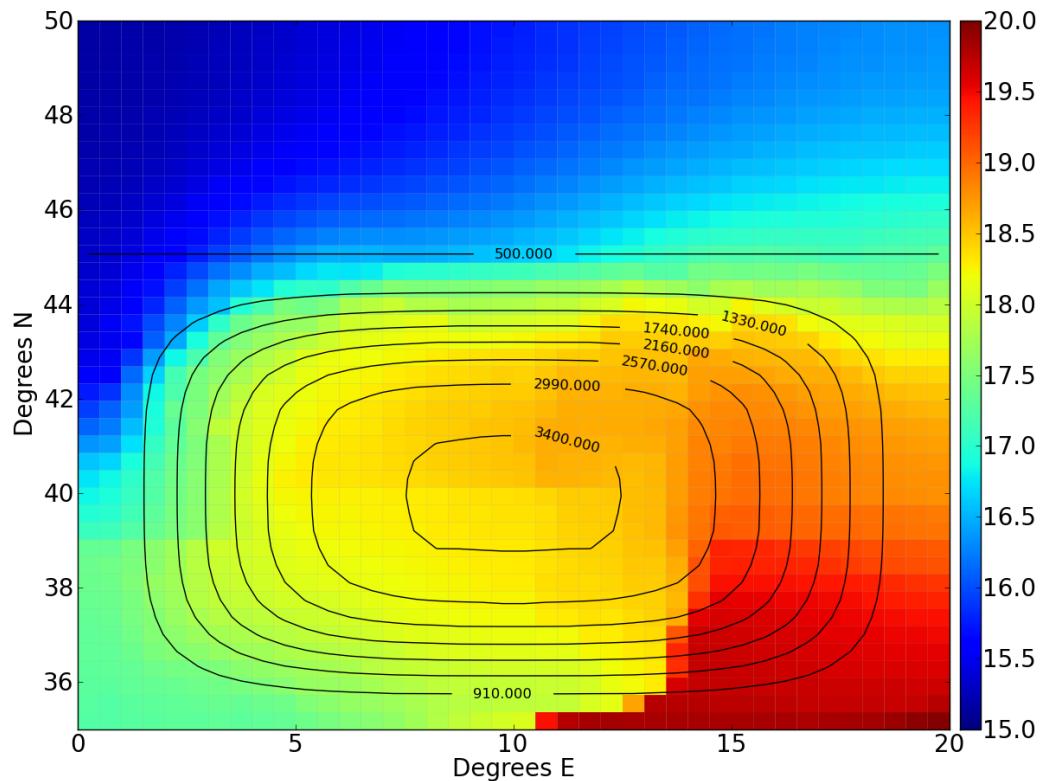


Figure 13.6: Properties of the BD97 experiment at day 360, which uses the Beckmann and Döscher (1997) parameterisation. The diffusive nature of the scheme is evident in the distribution of temperature in the bottom grid cells. The scheme also appears to weaken the strength of the circulation.

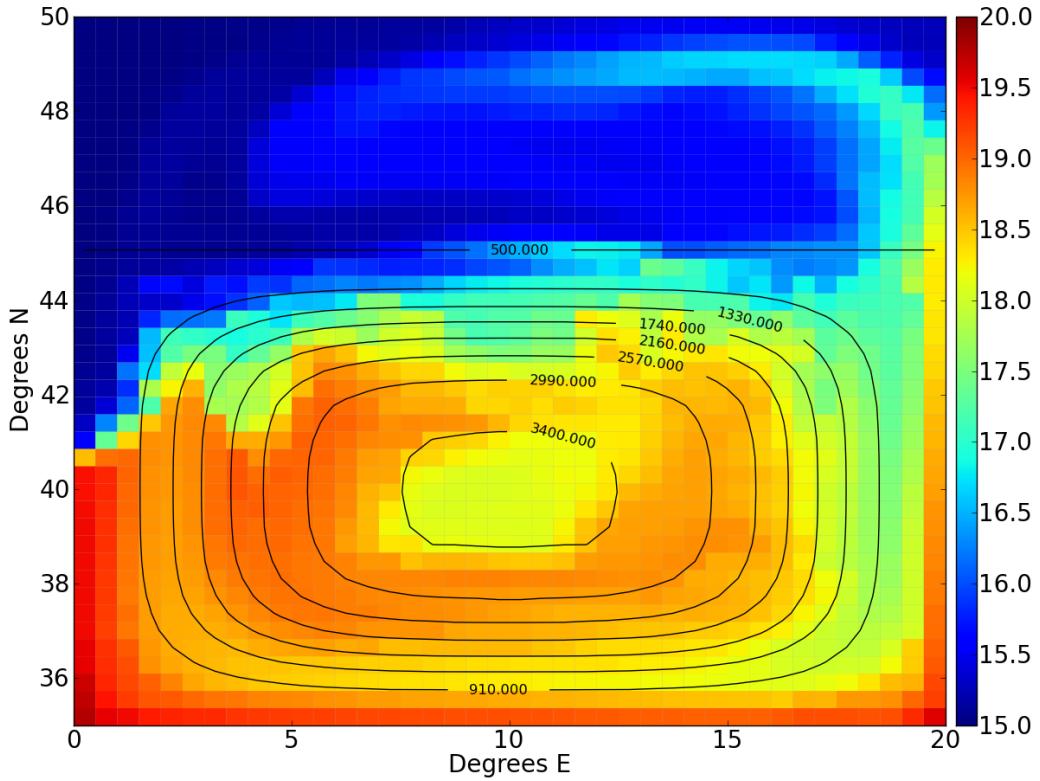
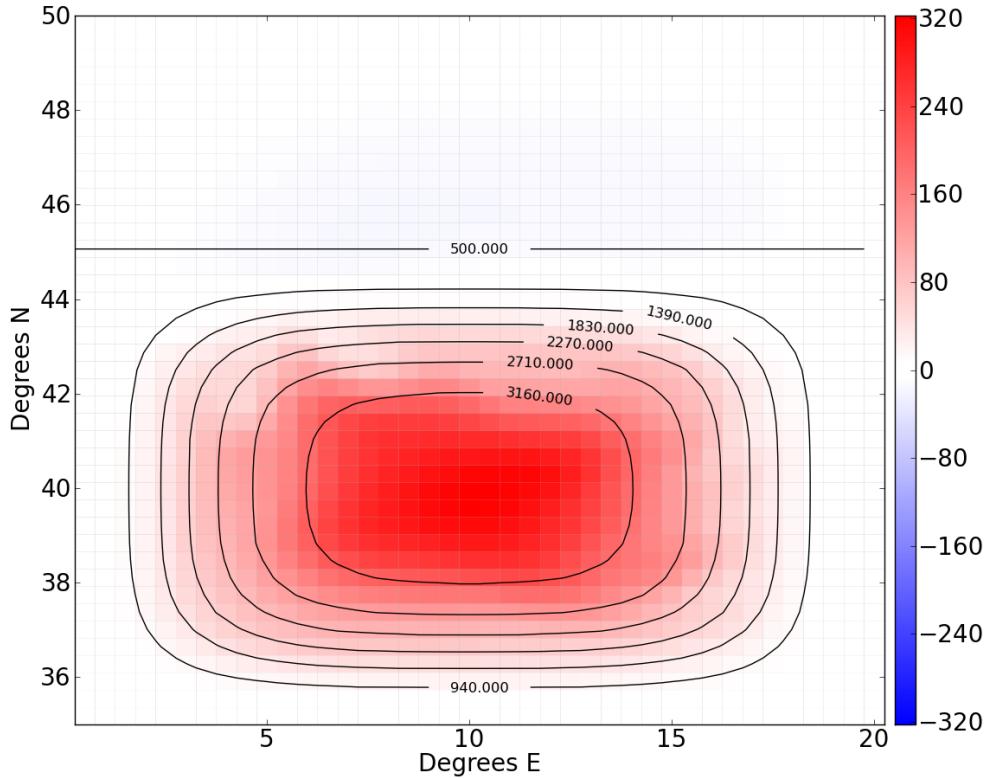
(a) The temperature in bottom cells ( $^{\circ}\text{C}$ ).(b) The barotropic quasi-streamfunction,  $\psi$  ( $10^9 \text{kg s}^{-1}$ ).

Figure 13.7: Properties of the NoReturn experiment at day 360, which uses the modified no-return Campin and Goosse (1999) parameterisation described in Section 4.5.

It can be seen in Figure 13.8(a) that blobs are roughly following topographic contours, and, not going downslope very much once they have reach geostrophic adjustment. Bottom drag breaks the geostrophic balance that makes parcels of water follow topographic contours, and increases the downslope component of their motion. To this effect, results from the `c3e-2` experiment, which has a dimensionless coefficient of drag that is an order of magnitude greater than `ctrl` experiment, are shown in Figures 13.9. It can be seen that the larger coefficient of drag is effective at breaking the geostrophic balance as there is a much greater number of blobs that are deeper than 3000m than in the `ctrl` experiment. The tracer distribution at shallow depths is similar between the `ctrl` and `c3e-2` experiments, however, it can be seen that the blobs make an appreciable difference at great depths. The other thing that is noticeable about the `c3e-2` experiment is that the pattern of the streamfunction, Figure 13.9(b), displays a similar pattern to the `ctrl` experiment, however, the magnitude of the flow is larger.

From the results presented in this section, it can be seen that some of the properties that Winton et al. (1998) found undesirable with their level coordinate model are present in the simulations here. Using other friction schemes, such as the Laplacian Smagorinsky scheme (Griffies and Hallberg, 2000) does encourage more downslope flow, however, there are still probelms with the flow pattern. Using the Smagorinsky scheme without the MI-COM velocity scale improves the flow pattern and strength.

The diffusive nature of the Beckmann and Döscher (1997) scheme is obvious in the results. While it is effective at getting tracer further down the slope, the diffusive nature of the scheme causes some non-physical results. Further, the scheme weakens the anti-cyclonic flow over the main basin and maintains the cyclonic cells to the west and south of the basin (both of which are undesirable). The Campin and Goosse (1999) scheme does very little to the tracer distribution or flow pattern and strength, however, the modified no return Campin and Goosse (1999) scheme (described in Section 4.5) makes a large difference to the solution at day 360. The anti-cyclonic circulation in the deep part of the basin is significantly strengthened by the scheme. The tracer distribution is markedly affected too, with less southward penetration of shelf waters along the western wall. It is also clear that the scheme assists in the downslope transport of properties.

Clearly, the dynamics of the bowl test case is very different to that of the DOME test case investigated in Chapter 12. The DOME test case is more simple inasmuch as it examines the dynamics of a plume on a uniform slope. The dynamics of the bowl test case is likely driven by the hypsometric effect (Winton et al., 1998). The hypsometric effect is driven by a source of dense water flowing down the slope of bowl-like basin. The heavy water flows in a cyclonic sense. The convergence of the dense water to the centre of the bowl causes a divergence of the bulk fluid. This divergence forces the bulk of the ocean to move in a anti-cyclonic sense (Rhines and MacCready, 1989). This result is clear in the

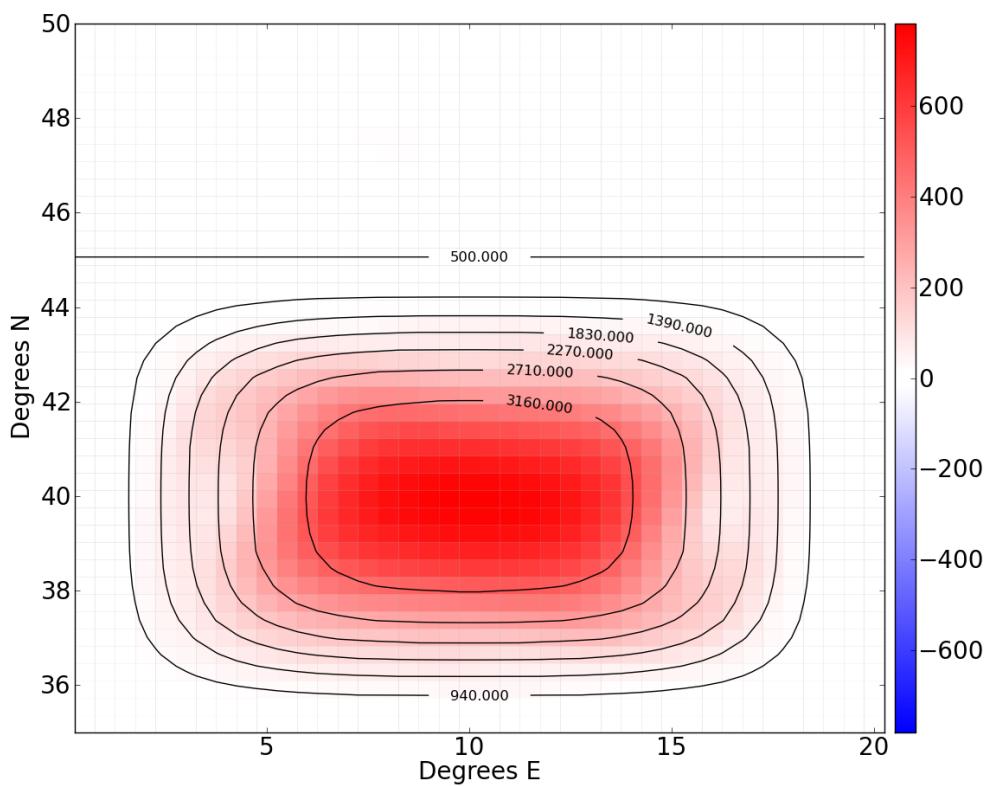
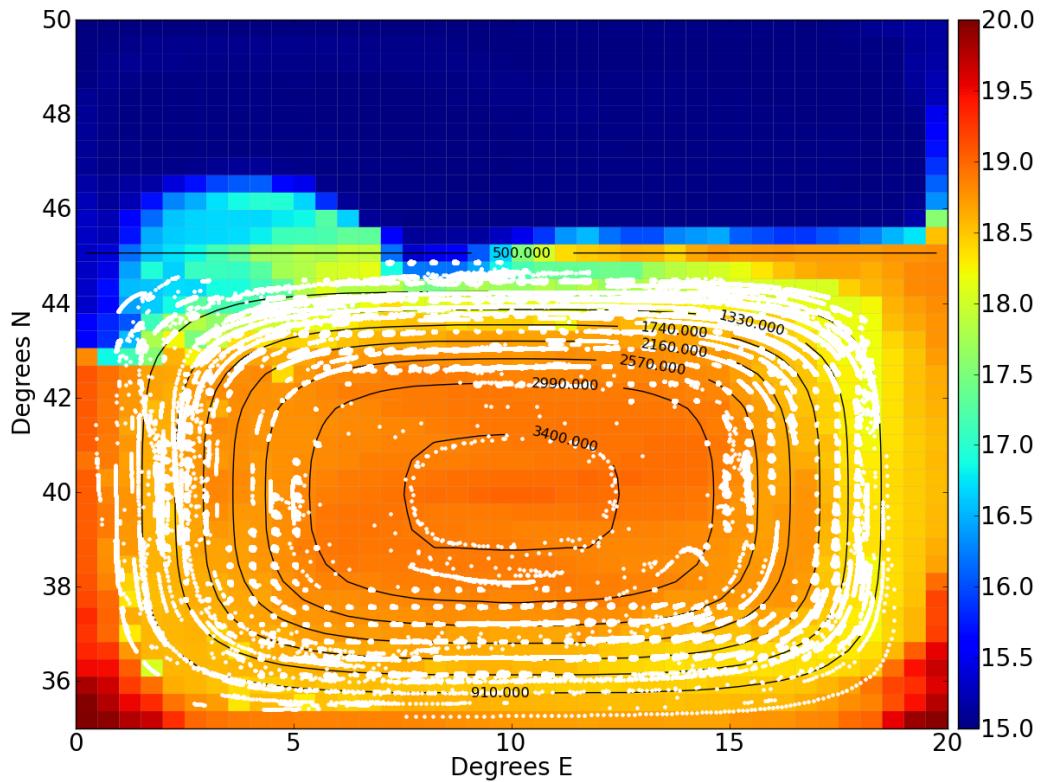


Figure 13.8: Properties of the `ctrl` experiment at day 360, which uses the dynamically active bottom blob parameterisation described in Section 5.3.

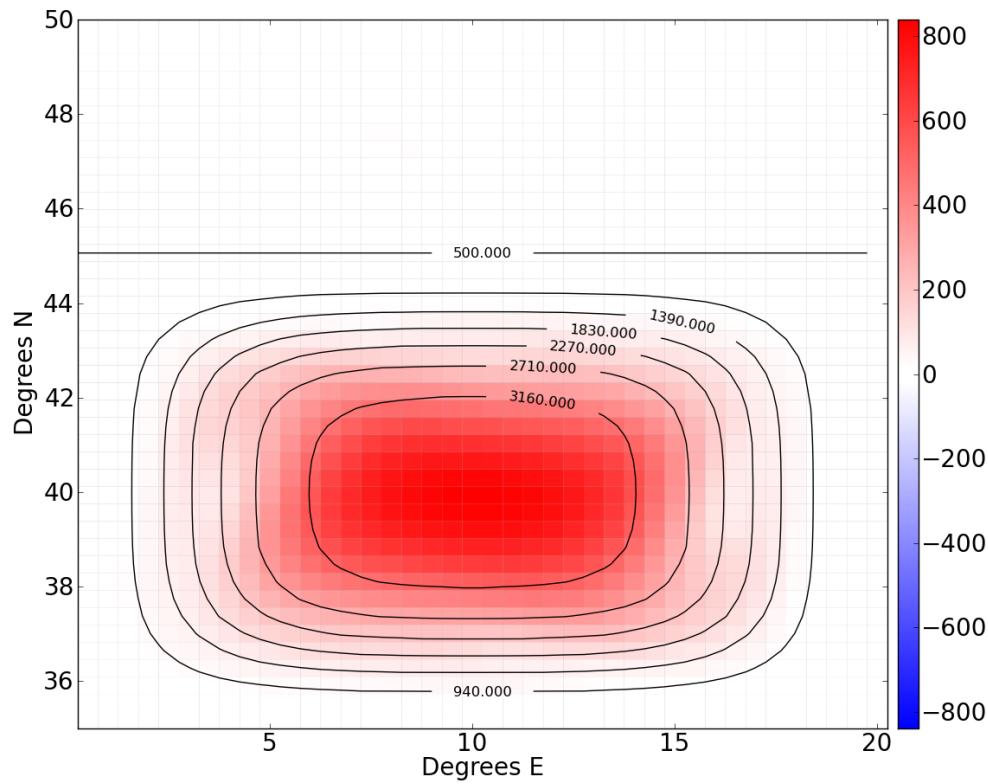
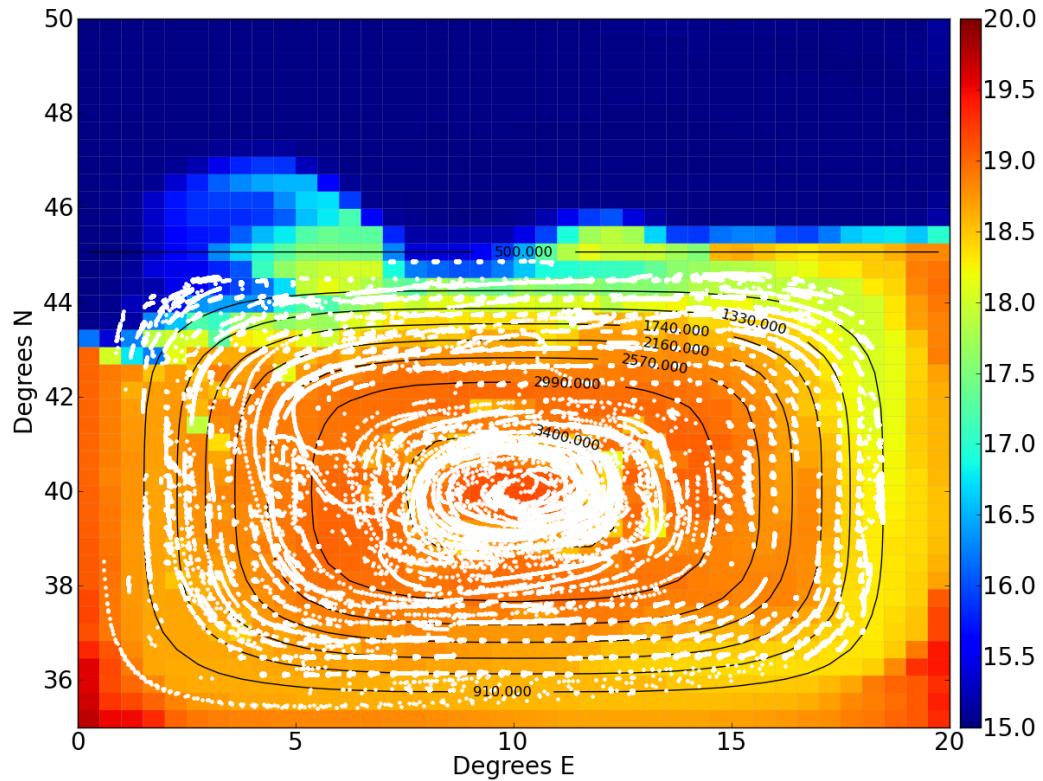


Figure 13.9: Properties of the `c3e-2` experiment at day 360, which uses the dynamically active bottom blob parameterisation described in Section 5.3 with a large bottom drag coefficient.

layered model results of Winton et al. (1998), reproduced here in Figure 13.2(b).

From the strength of the anti-cyclonic circulation, it can be deduced that the hypsometric effect is strongest in the experiments that use the dynamic bottom blobs. The effect is also more pronounced in the `NoReturn` experiment (which uses the no return overflow scheme described in Section 4.5) than other experiments. The reason why is because those schemes move mass down the slope, driving the vertical motion and creating the horizontal divergence required to drive the anti-cyclonic flow described by Rhines and MacCready (1989).

### 13.3.2 System Properties

Unlike the DOME configuration of Chapter 12, the bowl experiment is not forced. Thus, the mass of water in which the concentration of passive dye tracer is greater than 0.01 is, in some sense, a measure of the amount of mixing. As no additional dyed water is being added to the domain, mixing is the only mechanism that will increase the mass of water with a dye concentration greater than 0.01. The mass of water with a dye tracer concentration greater than 0.01 is shown in Figure 13.10. As can be seen, the initial mass of water containing dye tracer is approximately  $5 \times 10^{17}$ kg. The only experiment using the Lagrangian blob scheme shown is the `ctrl` experiment, which is largely representative of all experiments using the Lagrangian blob scheme.

The mean depth of water with a dye concentration greater than 0.01 is shown in Figure 13.11. The mean depth is a measure of how effectively material is transported downslope. The only experiment using the Lagrangian blob scheme shown is the `ctrl` experiment, which is largely representative of all experiments using the Lagrangian blob scheme.

In both the mass of the tagged water and the mean depth of the tagged water, it can be seen that most experiments have some initial rapid change, which then changes to a relatively more gradual change. The main exception to this is the `gm` experiment, which uses the Gent and McWilliams (1990) neutral physics scheme. In the `gm` experiment the mass increases virtually linearly throughout the experiment and the depth increases more or less at a similar rate throughout.

The experiment that causes the least increase in mass (equivalently, the least amount of mixing) over the simulation is the `ksmag2` experiment, which uses the Smagorinsky Laplacian friction scheme with the MICOM velocity scale (instead of a constant friction, as is used in `noblob`). The next least amount of mixing is the `nomicom` experiment, which is the same as the `ksmag2` experiment except it does not use the MICOM velocity scale. It can also be seen in the mean depth of the tagged water that the `ksmag2` and `nomicom` experiments have deeper penetration than other experiments, which is also supported by the bottom temperature plots in Figures 13.4(a) and 13.5(a).

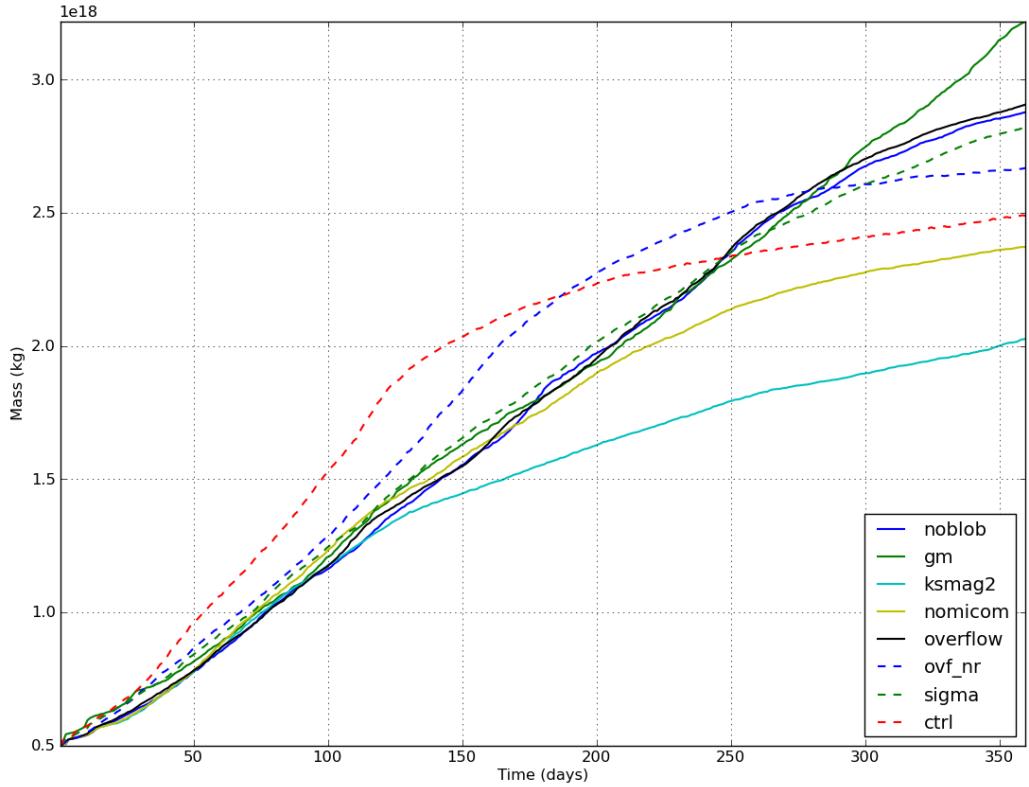


Figure 13.10: The mass of water with a dye tracer concentration of greater than or equal to 0.01. Since there is no forcing, the mass of tagged water is a measure of the amount of mixing. Initially, the `ctrl` experiment has the most mixing, but, that plateaus relatively quickly and is appreciably smaller than the `noblob` case. The experiments using the Smagorinsky friction have the smallest mass of tagged water.

The `CG99` experiment, which uses the scheme of Campin and Goosse (1999) tracks the trajectory of the `noblob` experiment in both the mass and the depth of tagged water. This indicates that in the bowl test case, the parameterisation does very little. The `BD97` experiment, which uses the Beckmann and Döscher (1997) parameterisation, has a similar trajectory to the `CG99` and `noblob` experiments with the mass of tagged water, indicating that they have similar amounts of mixing occurring. The mean depth of the tagged water is, however, rather different for the first half of the experiment. The mean depth of the tagged water in the `BD97` experiment rapidly deepens in the initial part of the experiment, making it appreciably deeper than the other experiments. However, as time goes on the mean depth of other experiments increases and the `BD97` experiment ends up with a similar mean depth to the `noblob` and `CG99` experiments. It is interesting to note that it was found in Section 13.3.1 that the barotropic circulation pattern for the three simulations are all quite similar.

The no return overflow scheme, described in Section 4.5, is used in the `NoReturn` ex-

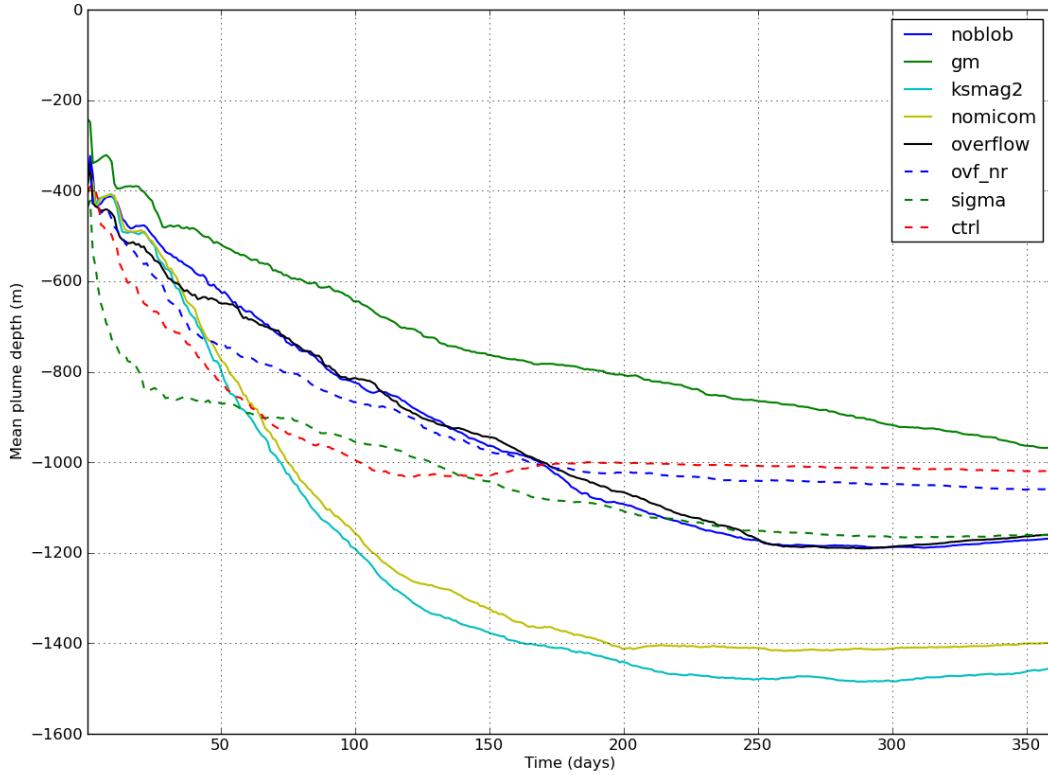


Figure 13.11: The mean depth of water with a dye tracer concentration of greater than or equal to 0.01. Experiments using the Smagorinsky friction scheme have the deepest mean depth, while the blob schemes actually have a more shallow mean depth than the case without blobs.

periment. It can be seen that the scheme initially increases the mass of the tagged water and lowers the mean depth of the tagged water more rapidly than the `noblob` scheme. At approximately day 170, the rate of increase of mass and the rate of deepening of the tagged water decreases such that eventually, the tagged water in the `noblob` experiment becomes more massive and deeper than the `NoReturn` experiment.

The `ctrl` experiment, which uses the Lagrangian blob scheme initially has the greatest rate of increase of tagged water mass and the second most rapid increase in the mean depth of tagged water. The rapid increase in the mass of the tagged water indicates rapid mixing initially, however, at the end of the experiment the `ctrl` experiment has a smaller mass than most of the other experiments shown (with the exception of `ksmag2` and `nomicom`). Interestingly, it has the second shallowest mean tagged water depth (after the `gm` experiment).

Examination of the bulk properties of the mass and mean depth of tagged water indicate that the Lagrangian scheme does limit mixing in the long term (although, enhances it initially), however, an unexpected result is that the mean depth of tagged water is

even more shallow than the experiments without any parameterisation at all. Without a “ground truth” it is difficult to say whether this result matters or not.

### 13.3.3 Blob Age and the Number of Blobs

The time evolution of the number of blobs in each experiment is shown in Figure 13.12. As can be seen, the number of blobs is very large, particularly when compared to the total number of E system computational points, which is 82,000. As can be seen, the number of blobs in most experiments eventually exceeds the number of E system grid points by quite a substantial amount. The `ctrl` experiment is largely representative of all of the experiments listed in Table 13.2 (i.e. experiments using the Lagrangian blob scheme) that are not shown in the figure.

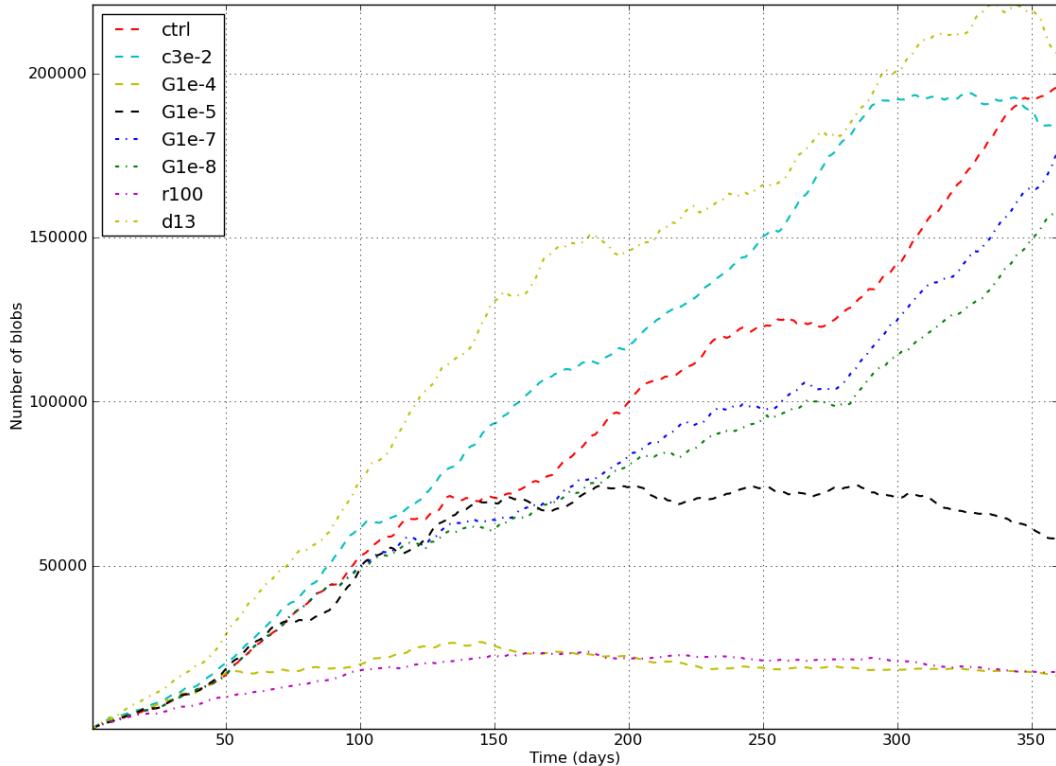


Figure 13.12: The number of blobs in each of the bowl experiments. The density threshold for forming a blob, the detrainment parameter, the fraction of a grid cell participating in an overflow event and the bottom drag coefficient all appear to influence the number of blobs in a simulation.

In the bowl test case, blobs are formed when the on shelf/off shelf density difference threshold is exceeded. For most experiments, this threshold is zero (i.e. if the off shelf water at the bottom of the deep ocean column is less dense than the on shelf water, a blob is formed – see Section 5.3.1). Blobs can be destroyed in multiple ways. One way is to fully detrain, that is, a blob’s properties have been returned to the E system via

detrainment, over the lifetime of the blob. Blobs may also ground (that is, they enter a water column with no wet grid cells). Grounding is not common in the bowl case. Blobs may also separate from the bottom if the ambient fluid that they are in becomes more dense than the blob. Blobs that separate are destroyed in all experiments in the bowl test case. Finally, a blob may also be destroyed if the grid cell mass constraint, described in Section 6.5.3, is violated. That is, if the mass of material in the L system exceeds some proportion of the total mass allowed in the grid cell, a blob is destroyed and the properties of the blob is returned to the E system. Due to the topography of the bowl test case, blobs do tend to converge towards the middle of the bowl. Thus, blob destruction via the violation of the grid cell mass constraint is not uncommon.

The number of blobs in simulations in the bowl test case is much greater than in the DOME test case (Figure 12.20), both in absolute terms and in terms of the number of blobs compared to the number of grid points. The reason for the much larger number in the bowl case is that the number of new blobs being formed (Figure 13.13) is much greater than in the DOME case (Figure 12.23). The reason for the large number of new blobs being formed in the bowl case is that there is a large area over which the instability condition for the formation of blobs is satisfied, and thus, there are a large number of blobs formed (e.g. Figure 13.8(a)) when compared to the DOME test case which forms blobs in a very limited region (e.g. Figure 12.7(a)).

It can be seen that the **r100** experiment forms many fewer blobs than the other experiments. This is due to the high density threshold used to decide when to form a new blob. The high threshold of  $\Delta\rho = 0.1\text{kg m}^{-3}$  in the **r100** experiment appears to have a much larger impact than the values of 0.01 and 0.001 used in the **r010** and **r001** experiments, which have a similar number of blobs to the **ctrl** experiment. Other parameters do not appear to greatly affect the number of new blobs formed.

The low number of blobs in the **r100** experiment in Figure 13.12 is explained by the lower number of new blobs formed. The lower number of blobs in the **G1e-4** and **G1e-5** experiments, which have a detrainment parameter of  $\Gamma = 1.0 \times 10^{-4}$  and  $1.0 \times 10^{-5}\text{kg m}^2\text{s}^{-1}$  respectively, also appears to greatly reduce the number of blobs relative to the **ctrl** experiment. The number of blobs that fully detrain is shown in Figure 13.14. As can be seen, the **G1e-4** and **G1e-5** experiments have a much greater number of blobs that fully detrain than other experiments. It can also be seen that the **G1e-7** and **G1e-8**, which have a detrainment parameter of  $\Gamma = 1.0 \times 10^{-7}$  and  $1.0 \times 10^{-8}\text{kg m}^2\text{s}^{-1}$  respectively, have far fewer blobs detrain fully when compared with the **ctrl** experiment. Thus, the number of blobs that detrain fully is closely related to the detrainment parameter,  $\Gamma$ . Despite having far fewer blobs fully detrain in the **G1e-7** and **G1e-8** experiments than the **ctrl** experiment, the number of blobs in the **ctrl** experiment is mostly greater than the number of blobs in the **G1e-7** and **G1e-8**. This is because the number of blobs that are being destroyed for other reasons (e.g. violating the grid cell mass constraint – see discussion in

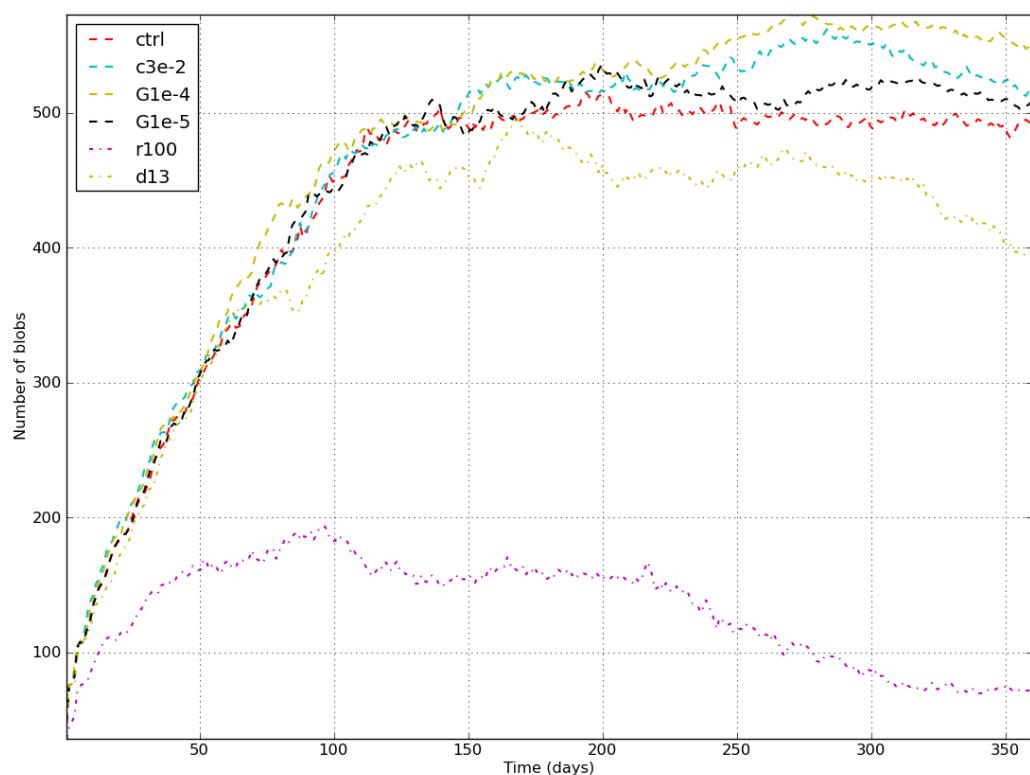


Figure 13.13: The evolution of the number of new blobs formed per timestep. The `r100` experiment, which has a high density threshold for the formation of a new blobs, produces many fewer new blobs than the other experiments as a result of the high density threshold.

Section 6.5.3) is greater in these experiments than in the `ctrl` experiment, as is shown in Figure 13.14.

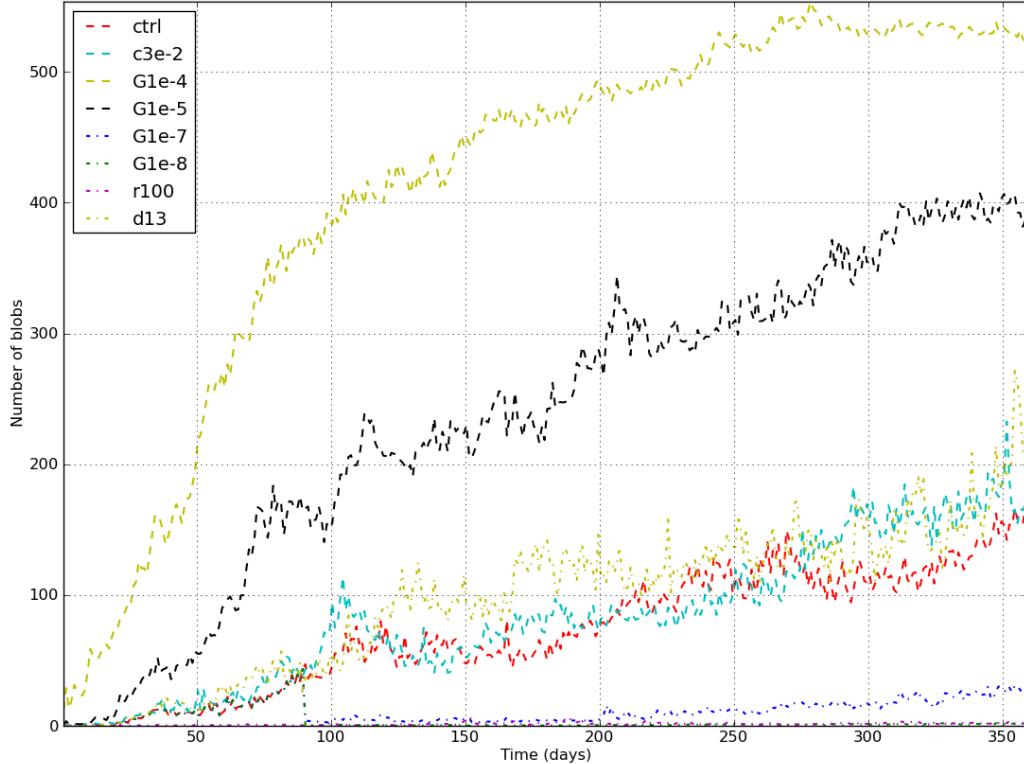


Figure 13.14: The number of blobs per time-step which fully detrain. There is a strong relationship with the detrainment parameter,  $\Gamma$ .

A smaller number of blobs being destroyed for reasons other than being fully detrained, as can be seen in Figure 13.15, helps to explain the reason why the `d13` experiment has appreciably more blobs than the `ctrl` experiment. In the bowl test case, the blobs that form up the slope and make their way down the slope cause a convergence of blobs toward the bottom of the domain. This convergence means that the likelihood of the grid cell mass constraint (see Section 6.5.3) being violated is more likely. So, experiments that tend to have a larger average blob mass will have more blobs destroyed this way. This explains why experiments like `G1e-7` and `G1e-8` (in which the mass of blobs does not decrease as quickly during their lifetime) have larger numbers being destroyed and experiments such as `d13` (which forms blobs of relatively small initial mass) does not have as many blobs destroyed.

The larger number of blobs in the `c3e-2` experiment may be due, in part, to longer living blobs as is shown by the average age of blobs in each experiment, Figure 13.16. The `G1e-5` and `G1e-4` experiments have shorter blob lifetimes than the other experiments, which is likely due to their more rapid detrainment. The `r100` experiment has the longest average blob lifetime for most of the experiment. Such a longer blob lifetime is likely

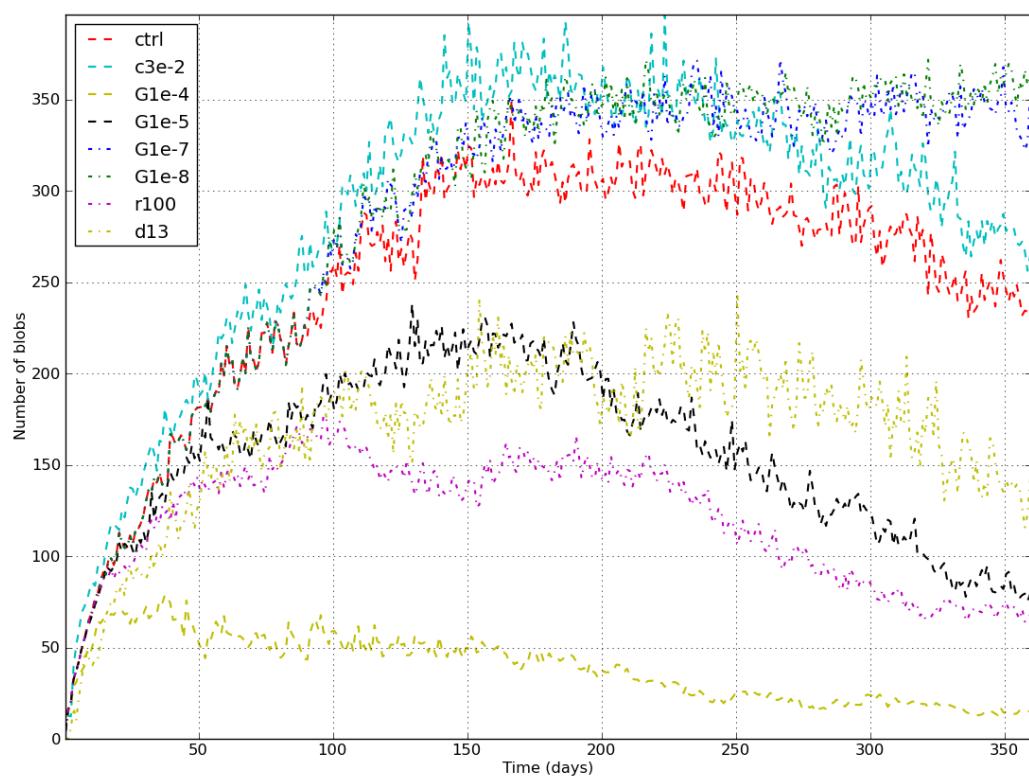


Figure 13.15: The number of blobs per time-step which are destroyed for violating the grid cell mass constraint. The grid cell mass constraint is that the mass of the L system cannot exceed a certain proportion of the total grid cell mass.

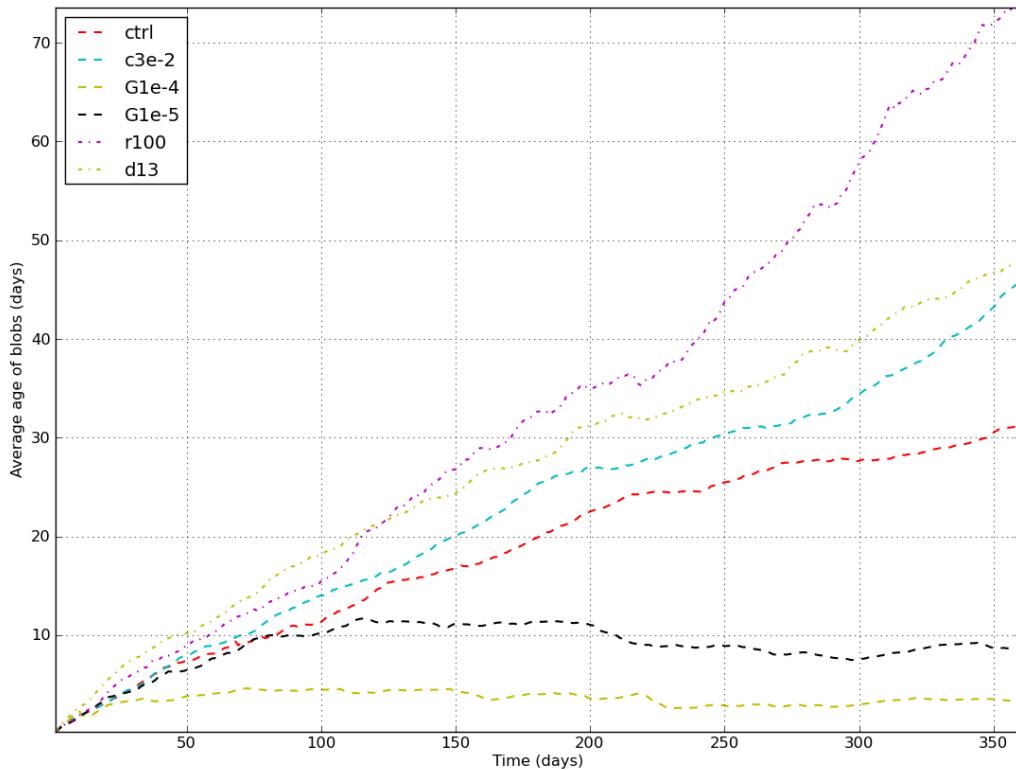


Figure 13.16: The average age of blobs. Blobs with a large detrainment parameter ( $G1e-4$  and  $G1e-5$ ) have a shorter lifetime due to the higher rate of detrainment. Blobs that are smaller ( $d13$ ), or, fewer in number ( $r100$ ) have longer lifetimes because fewer are destroyed due to the grid cell mass constraint.

due to a couple of factors, both of which stem from having a higher density threshold for the creation of a blob. Firstly, due to the formulation of the initial mass of a blob – Equation (5.22a) – a higher density difference between the on shelf and off shelf will create larger blobs. A larger initial mass means that it will take longer for a blob to fully detrain, thereby increasing the average age. Secondly, fewer blobs are formed (as shown in Figure 13.13), which means that there are fewer blobs converging in the bottom of the bowl, which results in fewer blobs being destroyed by the grid cell mass constraint being violated. The  $d13$  experiment also has a longer average lifetime. This is likely also due to the grid cell mass constraint not being violated as frequently, due to the smaller mass of the blobs converging toward the bottom of the bowl.

### 13.3.4 Blob Mass and Density

In all of the experiments shown in Figure 13.17 it is evident that the average blob mass is declining from initial maximum values. This phenomenon is due to the nature of the bowl experiment where the largest density difference between the shelf and abyssal waters is during the initial stages of the simulation. Without additional forcing, the density difference is gradually eroded.

The `ctrl` experiment is reasonably representative of the evolution of the average blob mass for the experiments that are not shown. The `r100` experiment, which has a density difference threshold for blob formation of  $\Delta\rho = 0.1\text{kg m}^{-3}$ , has the largest blob mass of the experiments conducted. The larger average blob mass is a result of the larger density threshold at which blobs are formed – Equation (5.22a). The density difference is also used in the calculation of the initial mass of the blobs, thus, having a higher density difference threshold means that blobs with a smaller initial mass are not formed. The `r100` experiment also has an appreciably higher average blob density (Figure 13.18) than the `ctrl` experiment. This is, again, due to the higher density threshold difference.

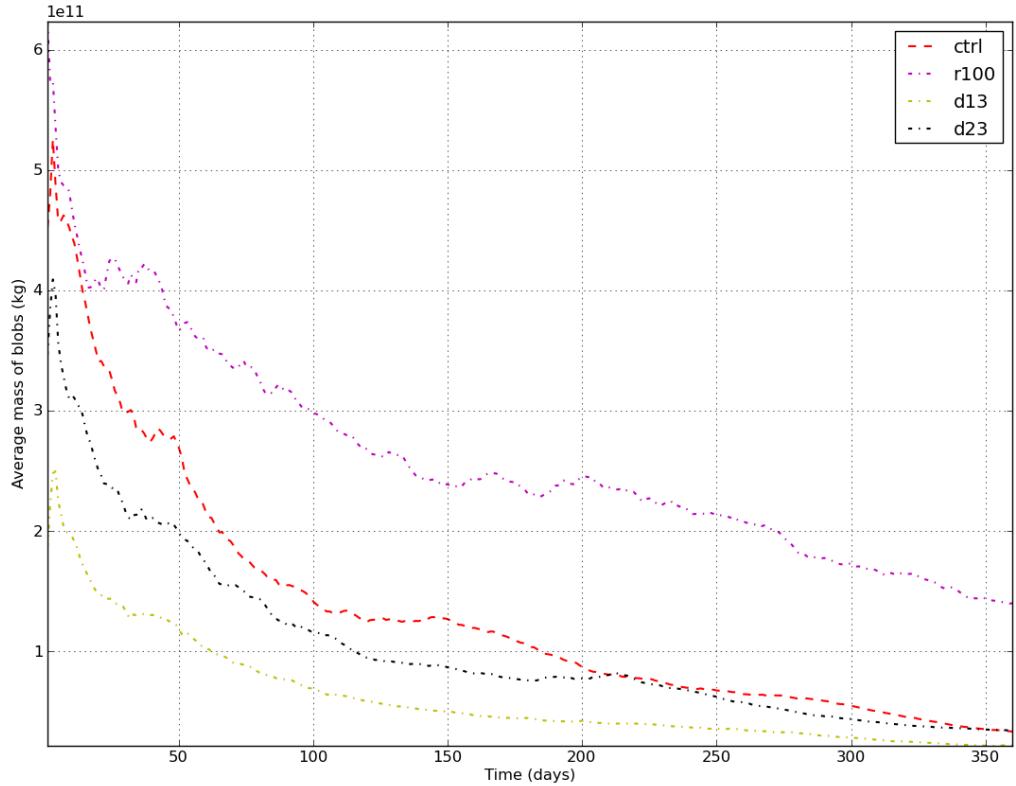


Figure 13.17: The average mass of blobs in a selection of experiments. The density threshold at which a blob is formed ( $\Delta\rho$ ), and, the proportion of a grid cell participating in an overflow event ( $\delta$ ) dictate the mass of blobs.

The `d13` and `d23` experiments, which have a lower value for  $\delta$  (the proportion of a grid cell participating in an overflow event), have blobs of a smaller average mass than the `ctrl` experiment. This is to be expected as the initial mass of blobs in those experiments is less due to the lower values for  $\delta$ . Interestingly, these two experiments also have a higher average density than the `ctrl` experiment.

The `G1e-4` and `G1e-5` experiments, which have higher values for the detrainment pa-

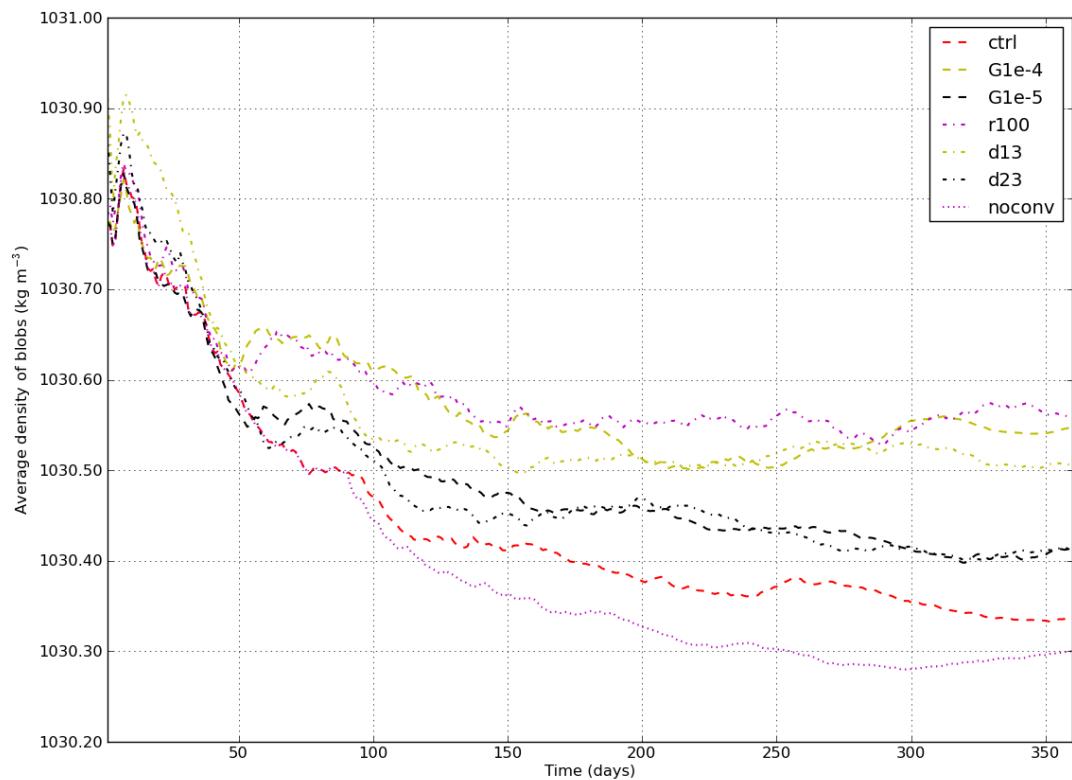


Figure 13.18: The average blob density. The density threshold at which a blob is formed ( $\Delta\rho$ ), and, the fraction of a blob participating in an overflow event ( $\delta$ ) are related to the density of the blobs.

rameter than the `ctrl` experiment, do not have tremendous differences in average blob mass to the `ctrl` experiment, but on average they do have appreciably more dense blobs than the `ctrl` experiment. The `noconv` experiment, which has no convective adjustment scheme, also does not have a significantly different average blob mass to the `ctrl` experiment, however, on average it has blobs with lesser densities.

In the bowl test case all of the parameters, with the exception of the coefficient of bottom drag, affects the average blob mass and density to some extent. It is largely dictated by the initial mass of the blob, however, it does appear that the average blob density is affected by the presence of convective adjustment.

### 13.3.5 Compute Time

The average compute time per core for each experiment is shown in Table 13.3. All experiments were run for 360 model days on 16 cores. The domain decomposition is the same across all experiments with there being four compute domains in the longitudinal direction (consisting of 10 grid points each) and four compute domains in the latitudinal direction (one with 11 grid points and three with 10 grid points), i.e. there are four domains that are  $10 \times 11$  points and twelve with  $10 \times 10$ . Thus, the smallest compute domain consists of 5000 grid points, while the largest compute domain consists of 5500 grid points (when taking into account that there are 50 vertical levels). The code was run on the Sun Constellation cluster, Vayu machine at the Australian National Computational Infrastructure National Facility. The Sun Constellation cluster has 1492 nodes in Sun X6275 blades. The code was compiled using the Intel Fortran Compiler version 12.0.4.191.

Unlike the DOME test case (Section 12.4.8), it can be seen in Table 13.3 that there is an appreciable difference in the total compute time (less the time taken for the E system diagnostics) for experiments that use the Lagrangian blobs than for experiments that do not. This is because of the large number of blobs formed in the bowl test case.

Using `noblob` as the benchmark (since it has the simplest friction scheme and the least number of other modules active) the Lagrangian blob scheme, the Gent and McWilliams (1990) neutral physics scheme or the Beckmann and Döscher (1997) overflow scheme all reduce the computational burden of the tracer module. The reduction in compute time required by the tracer module largely arises from the reduced burden of the tracer advection scheme. The `gm`, `BD97` and all experiments using the blobs all take approximately 30 seconds in the tracer advection scheme, while all the other experiments take approximately 200s. It is unclear why the use of these parameterisations has such a dramatic effect on the time taken to calculate tracer advection. Preliminary tests were run on another super computer and there was not a corresponding decrease in compute time in these modules. This leads to speculation that the changes could be compiler and/or hardware dependent.

It is also interesting to note that all of the overflow schemes reduces the computational

Experiment	Total	Tracer	Tracer Adv.	Velocity	Thickness	Barotropic	Overflows	Neutral Physics	Blobs
noblob	882.2	245.1	214.7	131.8	106.2	249.1	0.1	0.0	0.0
gm1	929.0	66.4	33.5	106.1	112.5	234.0	0.1	263.4	0.0
ksmag2	862.0	254.1	220.0	134.2	106.6	222.4	0.1	0.0	0.0
nomicom	888.1	234.0	203.9	139.3	105.4	238.7	0.1	0.0	0.0
overflow	828.5	237.9	207.8	98.4	108.6	234.9	4.5	0.0	0.0
ovf_nr	809.1	221.1	190.9	90.5	106.3	234.7	5.8	0.0	0.0
sigma	824.2	60.1	30.1	98.9	106.6	235.8	174.8	0.0	0.0
cml	1462.7	91.0	29.7	71.1	269.7	171.7	0.1	0.0	685.8
c3e-2	1545.8	97.7	29.9	71.5	258.4	172.6	0.1	0.0	741.6
c3e-4	1419.9	89.0	29.9	71.3	265.1	172.0	0.1	0.0	644.5
Gle-4	1082.1	81.1	30.9	72.9	206.1	171.5	0.1	0.0	398.3
Gle-5	1277.8	89.9	32.6	67.3	264.7	171.7	0.1	0.0	536.9
Gle-7	1435.8	93.7	30.0	72.0	322.9	171.7	0.1	0.0	617.4
Gle-8	1392.0	95.0	31.2	71.1	299.7	171.8	0.1	0.0	613.4
r001	1475.5	92.0	29.8	77.4	258.7	172.0	0.1	0.0	680.2
r010	1325.9	93.3	30.2	61.0	221.6	171.9	0.1	0.0	579.4
r100	1027.0	80.8	30.0	65.2	205.8	171.7	0.1	0.0	366.1
d13	1921.2	106.1	30.1	68.9	304.0	171.8	0.1	0.0	1031.7
d23	1530.9	92.6	31.0	72.4	288.5	171.2	0.1	0.0	725.7
noconv	1497.2	93.3	29.8	79.3	284.0	171.4	0.1	0.0	682.6

Table 13.3: The average amount of time taken (in seconds) per core for all experiments up to day 360. All experiments were run with 16 cores. The first column is the total time taken, less the time taken in the diagnostic module. Each column after that is the average time spent in routines associated with the column heading. Tracer Adv. stands for tracer advection, and, is a component of the tracer module. The list of modules is not comprehensive and the sum of the columns does not add up to the total.

burden of the velocity module. Again, it is unclear what causes the decrease in the time required by the velocity module. Use of the Lagrangian blobs increases the compute time taken by the thickness routines, but, reduces the compute time required by the barotropic routines. The increase in time required by the thickness routine is unsurprising as there are additional thickness arrays associated with the Lagrangian blobs that require additional calculations. On the other hand it is unclear as to the reason behind the reduction in the time taken by the barotropic module.

There is a large variation in the cost of individual experiments using the blob scheme – much more so than in the DOME test case (see Section 12.4.8). For instance, the **r100** experiment represents only a 16% increase over the **noblob** in compute time required, while the **d13** experiment is a 117% increase in compute time over the **noblob** experiment. The reason for the large discrepancy is the large differences in compute time required by the blob modules (366s and 1032s respectively). For a comparison with the widely used neutral physics parameterisation of Gent and McWilliams (1990), the neutral physics scheme in the **gm** experiment took 263s. So, in the bowl configuration, the blob scheme is generally significantly more expensive, however, as mentioned above, this is not necessarily applicable to other experimental setups where the propensity to form blobs is much lower.

It can be seen that the amount of time taken in the blob module is related to the number of blobs in the simulation. The experiments **G1e-4** and **r100** have the lowest compute time for the blob module, and, also generally have the lowest number of blobs (see Figure 13.12). On the other hand the experiment **d13** generally has the largest number of blobs (Figure 13.12) and it has the largest compute time out of the blob modules. The compute time required is not only related to the number of blobs, however, as the **c3e-2** experiment mostly has more blobs than the **ctrl** experiment ((Figure 13.12), but it has a lower amount of compute time required. In addition to the number of blobs, the compute time is also related to the time step of the individual blobs. As discussed in Section 7.3.2, the blob trajectories use an adaptive time step and thus, blob simulations where there is a small average blob time step used will be more computationally expensive than simulations that have a comparable number of blobs, but, are able to have a larger time step.

The compute time required by the blob scheme in the bowl test case is significant, however, the bowl test case is one in which a large number of blobs are formed due to the setup of the test case, as is evidenced by the number of blobs compared to the number of grid points. It would be reasonable to assume that the number of blobs (and thus the computational requirement of the blob modules) is likely to be much less in a test case that utilises realistic topography. Further decreases in the computational burden may be also obtained by decreasing the minimum timestep, or, increasing the desired error. Adjusting these parameters may have implications for the stability of the blobs – parameters which have not been tested in this study.

## 13.4 Summary

The dynamics of the bowl test case should be dominated by the hypsometric effect, in which dense water cascading down the side of a large bowl-like basin flows cyclonically toward the centre. The convergence of the dense water towards the centre of the bowl drives a divergence in the bulk ocean, and thus, causes a large scale anti-cyclonic motion. The physics of the bowl test case is therefore rather different to that of the DOME test case discussed in Chapter 12.

Winton et al. (1998) found that an isopycnal model's solution includes the large anti-cyclonic cell centred over the bottom of the bowl as predicted by the hypsometric theory of Rhines and MacCready (1989), while their level coordinate model solution really lacks this strong, coherent feature (Figures 13.2). The solutions obtained here were largely in qualitative agreement with the level model results of Winton et al. (1998), however, several factors were shown to greatly affect the solution.

### 13.4.1 Friction Scheme

Even without any overflow parameterisation, the choice of horizontal friction scheme has a tremendous bearing on the solution (Figures 13.3 to 13.5). The `nomicom` experiment, which uses the Laplacian Smagorinsky friction scheme without the MICOM velocity scale, had the strongest and most coherent anti-cyclonic flow of all the experiments that did not include an overflow parameterisation.

### 13.4.2 Traditional Overflow Schemes

The traditional overflow schemes of Beckmann and Döscher (1997) and Campin and Goosse (1999) have relatively little bearing on the circulation, although it is noted that the scheme of Beckmann and Döscher (1997) significantly alters the bottom cell tracer distribution (Figure 13.6(a)). The modified version of the Campin and Goosse (1999) scheme and all of the blob schemes, on the other hand, created a strong, coherent anti-cyclonic flow. It is hypothesised that the movement of mass affected by these schemes causes the necessary divergence of the bulk ocean to create the strong anti-cyclonic flow predicted by hypsometric theory.

### 13.4.3 The Lagrangian Blob Scheme

The ability of the lagrangian blobs to transport properties down the slope causes a convergence of mass toward the centre of the bowl in the bottom boundary layer. The bulk ocean responds to this convergence with a divergence, which then causes a strong anti-cyclonic flow in the bulk ocean. The ability of the blobs to model a gravity driven flow, and, the dynamic interaction with the Eulerian model means that the predictions made by hypsometric theory are represented in the simulations that use the dynamically active bottom blobs. This is a significant step toward overcoming the problems with level coordi-

nate models identified by Winton et al. (1998), and, gives a more physically sound solution.

In terms of more specific properties of the simulations: the mixing of properties between the dense shelf water and the light bulk ocean waters, the blob schemes initially had the greatest mixing, however, this tapered off so that at the end of 360 days, it was less than most other experiments, with the exception of the experiments using non-constant horizontal mixing. The experiments using blobs also had one of the most shallow mean depth of tagged water (the dense shelf water was initially tagged with a dye tracer concentration of 1.0).

The number of blobs in the bowl test case was very large, with most experiments having more blobs than Eulerian grid points. Virtually all of the blob parameters under examination, to a greater or lesser extent, affect the number of blobs in the simulation. The very large number of blobs formed was due to the large area over which blobs could be formed. In a simulation with realistic topography one would expect there to be far fewer blobs formed (relative to the number of Eulerian grid cells). A relatively large number of blobs were also destroyed due to the grid cell mass constraint. This was due to the nature of the topography in the bowl test case, in which large numbers of blobs were concentrated into a relatively small area. As a result, experiments that had fewer blobs, or, smaller blobs, tended to have older blobs.

While there is significant additional computational cost associated with running the Lagrangian blob scheme in the bowl test case and additional cell thickness calculations, there are also some reductions evident in the computational cost of other aspects of the code, including for tracer advection, velocity calculations and barotropic calculations. The reasons for the reductions in the cost of other parts of the code is unclear. There is a large variations in runtime between experiments that use the Lagrangian blob scheme. The variation is due to a combination of the number of blobs in a given experiment, and, the time step of the blobs in a simulation. Experiments that have many blobs and/or small blob time steps, require more computational time than those experiments that have fewer blobs and/or have smaller blob time steps. The bowl test case has a propensity to form many blobs due to its design. The number of blobs formed, relative to the number of Eulerian grid points, is not likely something that would be replicated with a realistic topography for the parameter regime that was investigated.



# Chapter 14

## Conclusions

The framework for an embedded Lagrangian model to represent subgrid-scale (SGS) phenomena in ocean climate models has been formulated and implemented in the Modular Ocean Model (Griffies, 2009). The framework has many potential applications. Here, the applications chosen were the representation of plumes in open ocean deep convection and gravity driven downslope flows. Other potential applications include the modelling of Argo floats, SGS eddies (using Lagrangian point vortices) or even a gravity wave ray tracing model.

The Lagrangian model calculates the trajectories of parcels, which are called “blobs.” For deep convection and downslope flows, the admission of the Lagrangian model requires the partitioning of the model into an Eulerian component and a Lagrangian component. Such a partitioning requires some careful accounting to ensure that seawater mass and tracer mass remain materially conserved.

It was shown that several existing schemes that represent convection and downslope flows can be regained in the blob framework. The blobs used to regain existing parameterisations exist for an infinitesimal amount of time. Their instantaneous existence means that they do not require the partitioning of mass. Thus, these blobs are dubbed dynamically passive blobs. It was shown that dynamically passive blobs were able to replicate the solutions of the original schemes.

Dynamically active schemes are those whose blobs exist for a finite amount of time. Two such schemes were developed. The first was a “bottom blob scheme” which was based on the streamtube model of Price and Baringer (1994) and the second was a “free blob” scheme which was designed to represent convective plumes. Both of these schemes require the partitioning of the mass of the model in order to admit them.

There were a number of non-trivial changes to the code in order to facilitate the partitioning of mass between the Eulerian system (the E system) and the Lagrangian system (the L system). These included changes to the calculation of grid cell thickness, hydrostatic pressure, tracer budgets and continuity.

The blobs themselves are represented in linked lists, which is a computer science technique not often applied in ocean climate modelling. Linked lists are a memory efficient means by which to store a widely varying number of nodes. Multi-threading of linked lists presents some unique challenges, particularly when maintaining bitwise reproducibility across restarts or an arbitrary number of processors.

The numerical scheme used to integrate the trajectories of the dynamically active blobs is an adaptive step Runge-Kutta method. Since the Lagrangian blobs are free to move about the domain, a point location scheme is required to figure out which grid cell a blob resides in order to properly calculate the interaction between the E system and that blob. The point location scheme is robust to cyclic grids and curvilinear grids, such as the tripolar grid of Murray (1996).

The dynamically active free blobs were tested in the deep convection test case (Jones and Marshall, 1993; Campin et al., 2011), which is used in process studies of open ocean deep convection. The test case uses a flat bottomed, doubly periodic domain with an intense cooling disc at the surface to examine a deep convective mixed patch. A very fine resolution (100m) non-hydrostatic “ground truth” simulation was run in the MITgcm (Marshall et al., 1997a,b, 1998), as well as a very fine resolution hydrostatic simulation and a coarse resolution (2km) hydrostatic simulation to examine the effects of the hydrostatic approximation and resolution. The introduction of the hydrostatic approximation had a profound effect on the nature of the simulation.

The introduction of pseudo-non-hydrostatic dynamics, via the Lagrangian blob scheme, represents a step toward a representation that is more similar to the non-hydrostatic case than the hydrostatic case. To examine the fidelity of the simulations, a passive tracer was initially set to one at the surface and was set to zero elsewhere. The tracer was then restored to one at the surface for the entire simulation. Diagnostics, using the passive tracer and the blobs show that the Lagrangian blob scheme makes the combined Lagrangian and Eulerian models susceptible to numerical instability. It appears that the model is stable within a relatively narrow parameter range. Some of the solutions that exhibit the numerical instability exhibit some desirable characteristics that are absent from the numerically stable simulations (although, the numerically stable simulations still represent a large step toward the non-hydrostatic solution). It is postulated that the root cause of the numerical instability is the simple representation of non-hydrostatic dynamics – further investigation on this topic is warranted.

The dynamically active bottom blobs were tested in the dynamics of overflows, mixing and entrainment (DOME) test case (e.g. Legg et al., 2006). The main objective of the DOME test case is to examine the properties of a dense plume on a uniform slope. It is known that coarse resolution level coordinate models have difficulty with gravity driven

---

currents not penetrating deep enough, and, becoming too dilute (thereby slowing their along slope progress). It was found that the traditional overflow schemes were able to overcome some of these issues, but, they would usually degrade the solution in some other way. Specifically, the scheme of Campin and Goosse (1999) improved the depth at which the plume would penetrate to, however, the reduced gravity of the plume is smaller, slowing the along slope velocity of the plume even more. On the other hand, the scheme of Beckmann and Döscher (1997) improved both the plume depth and the along slope velocity. The improvement in the along slope velocity, however, is due to the enhanced isotropic diffusion of the scheme. The isotropic diffusion transports tracer properties without reference to dynamical considerations, such as rotation, meaning that the plume pathway can be unphysical.

In the DOME test case, the Lagrangian blob scheme increased the depth of the plume, as well as the along slope velocity of the plume without any major drawbacks. The effect of various tunable parameters was also tested, namely, the coefficient of bottom drag for blobs, the detrainment parameter (which helps set the rate of detrainment), the proportion of a grid cell participating in an overflow event and the density threshold at which a blob is formed. The following conclusions can be drawn about the effects of these parameters on the scheme:

- A larger coefficient of drag slows the plume, but, it increases the angle at which the plume crosses isobaths and increases the plume's depth.
- A larger detrainment parameter decreases the average lifetime of a blob, which in turn decreases the number of blobs.
- There are drawbacks to having very massive blobs. One drawback is that having very massive blobs creates a very strong circulation due to the lateral movement of the (very massive) blobs. The other problem is associated with a limitation of the present formulation; mainly many blobs are destroyed because the dynamic bottom blobs are restricted to existing in the bottom grid cell. A number of very massive blobs violates the grid cell mass constraint, necessitating the premature destruction of some blobs to ensure that mass is conserved and the model remains stable.
- Having a large density threshold difference for blob formation produced less blobs since the formation condition was satisfied less frequently.

In the DOME test case, it was found that the computational cost of using the blobs was not significantly more than that of other experiments that did not use the blobs. This was partly due to the limited area in which blobs were formed (and therefore, the limited number of blobs formed). It was also partly due to the nature of the experimental setup, with many blobs being grounded (and thus destroyed) when they interacted with the western wall.

The bowl test case (Winton et al., 1998) also tested the dynamically active bottom blobs. The bowl test case is dominated by the dynamics of the hypsometric effect, in

which a descending bottom current in a bowl shaped basin causes an anti-cyclonic circulation in the basin (Rhines and MacCready, 1989). Winton et al. (1998) showed that level coordinate models struggle to capture this effect when compared to layered models. The experiments that did not use the dynamically active bottom blobs were largely in agreement with their results. The one exception to this is the experiment that used the modified Campin and Goosse (1999) scheme (described in Section 4.5) which did induce the strong anti-cyclonic circulation. The experiments that use the dynamically active bottom blobs created a strong anti-cyclonic circulation, indicating that the blobs improved the fidelity of the simulation.

The effect of the various blob parameters was largely consistent with the DOME test case. One major difference, however, was the compute time required. The large area over which blobs can be formed in the bowl test case means that a large number of blobs were formed, and thus, there was a relatively large increase in compute time required due to the very large number of blobs.

The validation process used to examine the fidelity of the embedded Lagrangian process, at its most basic, showed that the technique is viable. Test cases that have been studied extensively in the literature were used in order to compare the results with well known benchmarks. The test cases used were also the ideal test bed for examining the parameter space of the new model and this was examined in detail. A shortcoming of the validation process used was that there has been no direct comparison with nature, which is the ultimate litmus test. Tests against observations are planned for future studies. However, the ability to compare against existing and well known test cases does provide much insight into the fidelity of the system.

Overall, the use of the embedded Lagrangian blobs improved the representation of the physics that it was attempting to represent. The introduction of pseudo-non-hydrostatic dynamics to open ocean deep convection represents a significant step toward a representation of non-hydrostatic dynamics. The results indicate that there is utility in the Lagrangian blobs in the context of process studies involving open ocean deep convection. The utility of the Lagrangian blob scheme in coarse resolution models to examine large scale processes (such as the deep convection associated with the meridional overturning circulation) remains untested. While it has been argued by Send and Marshall (1995) that convective adjustment captures the integral effects of convection, however, Hughes et al. (2009) indicate that caution should be exercised when drawing conclusions about the overturning circulation when using traditional convective adjustment schemes because the energetics of the system is not represented accurately. The ability of the blobs to convert potential energy to vertical kinetic energy does, however, hold some promise in addressing the concerns raised by Hughes et al. (2009).

The use of the embedded Lagrangian scheme with gravity driven overflows in coarse

---

resolution models is well tested and it is clear that there are large improvements in the fidelity of the simulations. Depending on the number of blobs that are formed (which is largely influenced by forcing and topography), the increased computational cost can be quite modest and is comparable to other commonly used parameterisations. Given the ubiquity of the poor representation of overflows in the present generation of level coordinate ocean climate models, and the results presented here in idealised test cases, the embedded Lagrangian model clearly has the potential to represent a significant improvement in the representation of overflows and associated water masses in realistic climate models for a relatively modest increase in computational requirements. Perhaps the greatest potential that the dynamically active bottom blobs holds is in their dynamic nature. Unlike the embedded streamtube model of Danabasoglu et al. (2010), which has fixed source and product regions, the dynamic nature of the Lagrangian blobs means that the properties of the overflowing waters can change according to a set of physical rules in response to a change of forcing, making their application to climate change projections one of their greatest potential impacts.



## Appendix A

# Calculation of $m^a$ for the Diffusion-like Scheme

If we take Equation (4.9a) and divide by the volume of the grid cell of origin,  $V^k$ , this gives us the fraction of the mass of the grid cell that the blob is transporting in the non-Boussinesq case,

$$\frac{m^a}{m^k} = \gamma \frac{(\rho_i^k - \rho_i^{k+1})}{\rho_i^k}. \quad (\text{A.1})$$

The analogous equation for the Boussinesq case is

$$\frac{m^a}{m^k} = \gamma \frac{(\rho_i^k - \rho_i^{k+1})}{\rho_0}. \quad (\text{A.2})$$

If we then use the realistic values of  $\rho_i^k = 1035 \text{ kg m}^{-3}$ ,  $\rho_i^{k+1} = 1034 \text{ kg m}^{-3}$  and set the parameter  $\gamma = 1$  we get,

$$\begin{aligned} \frac{m^a}{m^k} &= 1 \cdot \frac{1035 - 1034}{1035} \\ &= 0.0966\% \\ &\approx 0.1\% \end{aligned}$$

Note, that because of the choice of  $\rho_i^k$  we get an identical answer for the Boussinesq case as a common choice for the reference density is  $\rho_0 = 1035 \text{ kg m}^{-3}$ .



## Appendix B

# Surface Area and Radius of a Sphere

The surface area and volume of a sphere are, respectively

$$A = 4\pi r^2 \quad (\text{B.1a})$$

$$V = \frac{4}{3}\pi r^3 \quad (\text{B.1b})$$

where  $A$  is the surface area,  $V$  is the volume and  $r$  is the radius. If we cube Equation (B.1a) and square Equation (B.1b) and rearrange in terms of  $r^6$

$$r^6 = \frac{1}{64\pi^3} A^3 \quad (\text{B.2a})$$

$$r^6 = \frac{9}{16\pi^2} V^2 \quad (\text{B.2b})$$

Now equating equations (B.2a) and (B.2b), solving for surface area  $A$

$$A = (36\pi)^{\frac{1}{3}} V^{\frac{2}{3}} \quad (\text{B.3})$$

Now to find the diameter ( $2r$ ) of a sphere in terms of its volume, we return to Equation (B.1b) and rearrange

$$\begin{aligned} V &= \frac{2^3}{3 \times 2} \pi r^3 \\ \Rightarrow 2r &= \left( \frac{6}{\pi} V \right)^{\frac{1}{3}} \end{aligned} \quad (\text{B.4})$$

APPENDIX B. SURFACE AREA AND RADIUS OF A SPHERE

---

## Appendix C

# List of Symbols and Abbreviations

### C.1 Abbreviations

Abbreviation	Meaning
AABW	Antarctic Bottom Water; page 12.
ALE	Arbitrary Lagrangian-Eulerian; page 6.
C	Degrees celcius; page 28.
CASL	Contour-Advective Semi-Lagrangian; page 7.
DOME	Dynamics of overflow mixing and entrainment; page 183.
E system	The Eulerian system; page 26.
J	Joules; page 28.
kg	kilogram; page 27.
L system	The Lagrangian system; page 26.
m	metre; 27.
MOM	Modular Ocean Model; page 19.
NADW	North Atlantic Deep Water; page 12.
OGCM	Ocean General Circulation Model; page 9.
s	second.
SGS	subgridscale; page 9.
SSOM	Slippery Sacks Ocean Model; page 5.
Sv	Sverdrup ( $10^9 \text{kg s}^{-1}$ ).
WSBW	Weddell Sea bottom water; page 13.
WSDW	Weddell Sea deep water; page 13.

### C.2 Latin Symbols

Symbol	Meaning
$A$	Surface area of a sphere; page 271.
$A_{\text{interface}}$	Surface area of the interface between a blob and the Eulerian model; page 53.

Continued on next page...

---

## APPENDIX C. LIST OF SYMBOLS AND ABBREVIATIONS

---

<b>Symbol</b>	<b>Meaning</b>
<i>a</i>	Label for an individual blob. When superscripted, denotes the property of an individual blob; page 42.
<i>b</i>	Label for an individual blob. When superscripted, denotes the property of an individual blob; page 42.
$C_d$	Coefficient of drag; page 59.
$C_p^0$	Heat capacity; page 28.
$\mathcal{C}$	Tracer concentration; page 27.
$\mathcal{C}_n$	Concentration of the $n^{\text{th}}$ seawater constituent in a seawater parcel; page 27.
$\mathcal{C}_w$	Tracer concentration of water crossing the surface boundary; page 75.
$\mathcal{D}$	Detrainment rate, $\text{ms}^{-1}$ ; page 52.
$dA$	Horizontal area of a grid cell; page 26.
$d\text{stry}$	Indicates the property of a destroyed blob when subscripted; page 30.
$dV$	Volume element; page 26.
$dx$	Zonal width of a grid cell; page 26.
$dx$	Zonal width of the north grid box face (when superscripted with grid cell indices); page 47.
$dy$	Meridional width of a grid cell; page 26.
$dy$	Meridional width of the eastern grid box face (when superscripted with grid cell indices); page 47.
$dz$	Height of a grid box; page 26.
$\frac{d}{dt}$	Material derivative; page 29.
$\mathbf{e}$	Global truncation error; page 126.
$E$	Algorithm efficiency; page 127.
$E$	When subscripted, indicates the Eulerain model property; page 26.
$\mathcal{E}$	The eastern tracer cell point; page 90.
$\mathcal{E}$	Entrainment; page 52.
(E2L)	Indicates, when subscripted, properties that are being transferred from the E system to a blob; page 30.
$\mathbf{F}$	The sub-gridscale tracer concentration flux vector; page 28.
$\mathbf{F}_N$	Normal force; page 60.
$F^{(s)}$	SGS dia-surface flux; page 29.
$\mathbf{F}^{(\mathbf{u})}$	Horizontal frictional force; page 38.
$F^{(x)}$	Horizontal SGS flux in the $x$ direction; page 29.
$F^{(y)}$	Horizontal SGS flux in the $y$ direction; page 29.
$f$	Final property (when subscripted); page 42.
$f$	Twice the horizontal component of the Earth's rotation vector, also known as the Coriolis parameter, $2\Omega \cos \phi$ , where $\phi$ is the latitude; page 32.
$f^*$	Vertical component of the Earth's rotation vector; page 32.
$\mathbf{G}$	Parameterised surface stress on a blob; page 3.4.

Continued on next page...

Symbol	Meaning
<b>G</b>	Vector of functions; page 125.
<i>g</i>	Acceleration due to gravity; page 27.
<b>g</b>	Vector for acceleration due to gravity, $(0, 0, -g)$ ; page 32.
<i>H</i>	Depth of topography (as measured from the ocean surface at rest, i.e. $z = 0$ ); page 18.
$\mathcal{H}$	The hash function; page 124.
$\mathcal{H}$	The heat content of a seawater parcel; page 28.
<b>h</b>	A metric or stretching function with components $(h_1, h_2)$ ; page 109.
<i>h</i>	Lagrangian blob step size; page 106.
$h_L$	Height of a blob; page 61.
$h_{\text{conv}}$	Depth of convection; page 45.
$h_{\text{st}}$	Streamtube height; page 59.
<i>i</i>	Initial property (when subscripted); page 42.
<i>i</i>	Zonal grid box index. When superscripted, denotes a property of the $i^{\text{th}}$ grid box; page 46.
<b>J</b>	A scalar flux; page 28.
<i>j</i>	Meridional grid box index. When superscripted, denotes a property of the $j^{\text{th}}$ grid box; page 46.
$\mathcal{K}$	Kinetic energy per unit mass, $(\text{m}^2 \text{s}^{-2})$ ; page 36.
$\mathcal{K}^{(h)}$	Horizontal kinetic energy per unit mass, $(\text{m}^2 \text{s}^{-2})$ ; page 37.
<i>k</i>	Vertical grid box index. When superscripted, denotes a property of the $k^{\text{th}}$ grid box; page 42.
$k_{\text{bot}}$	The bottom grid cell index; page 75.
<b>L</b>	When subscripted, indicates the Lagrangian model property; page 26.
<i>l</i>	Adjacent grid cell; Figure 4.2.
(L2E)	Indicates, when subscripted, properties that are being transferred from a blob to the E system; page 30.
<i>m</i>	Mass of a seawater parcel (e.g. grid cell, blob); page 27.
$m_n$	Mass of the $n^{\text{th}}$ seawater constituent in a seawater parcel; page 27.
<i>N</i>	The number of constituents that make up seawater; page 27.
<i>N</i>	The number of blob sub-cycles that make up an E system time step; page 106.
<i>N</i>	Buoyancy or (Brunt-Väisälä) frequency; page 54.
$\mathcal{N}$	The northern tracer cell point; page 90.
<i>n</i>	Tracer index; page 27.
$\hat{\mathbf{n}}$	Outward pointing unit normal vector to the bottom solid Earth surface; page 60.
$\hat{\mathbf{n}}$	Outward pointing unit normal vector to a volume, $V$ ; page 28.
new	Indicates the property of a new blob when subscripted; page 30.
<i>p</i>	Pressure; page 32.
<i>p</i>	Number of function evaluations; page 126.

Continued on next page...

---

APPENDIX C. LIST OF SYMBOLS AND ABBREVIATIONS

---

<b>Symbol</b>	<b>Meaning</b>
$p_A$	Pressure at the surface of the ocean; page 52.
$p_b^0$	A reference bottom pressure; page 96.
$p^*$	The pstar vertical coordinate; page 69.
$p'$	Anomalous pressure; page 85.
$Q$	Denotes the number of blobs in a particular domain (usually a water column); page 26.
$\mathcal{Q}$	Denotes the number of blobs in a particular domain (usually a single grid box); page 74.
$\mathcal{D}_{(C)}^{(\text{turb})}$	tracer due to turbulent fluxes; page 75.
$Q_{(\text{dstry})}$	The number of blobs destroyed in a grid cell; page 30.
$Q_{(\text{new})}$	The number of blobs created in a grid cell; page 30.
$Q_{(\text{in})}$	The number of blobs entering a grid cell; page 29.
$Q_{(\text{out})}$	The number of blobs entering a grid cell; page 29.
$q$	Denotes the $q^{\text{th}}$ blob; page 26.
$\mathcal{Q}_w$	The freshwater volume flux per horizontal unit area crossing the surface boundary with units $\text{ms}^{-1}$ ; page 75.
$\mathcal{R}$	Generating function; page 126.
$R$	Average radius of the Earth; page 110.
$\text{Ri}$	Richardson number; page 59.
$S$	Surface of a volume, $V$ ; page 28.
$s$	An arbitrary surface; page 29.
$\mathcal{S}^{(C)}$	Tracer source with units $\text{s}^{-1}$ ; page 28.
$\mathcal{S}^{(M)}$	Mass source with units $\text{s}^{-1}$ ; page 28.
$\mathcal{S}^{(\mathbf{u})}$	Horizontal momentum source/sink vector, with components $(\mathcal{S}^u, \mathcal{S}^v)$ ; page 33.
$\mathcal{S}^{(\mathbf{v})}$	Momentum source/sink vector, with components $(\mathcal{S}^u, \mathcal{S}^v, \mathcal{S}^w)$ ; page 32.
$\mathbf{T}$	Local truncation error; page 126.
$T$	The combined properties of the Eulerian model and the Lagrangian model; page 26.
$\mathcal{T}$	Tracer point; page 56.
$t$	Time; page 28.
$t_n$	Time at the $n^{\text{th}}$ step; page 126.
$t_{\text{mix}}$	Timescale of mixing; page 45.
$u$	$x$ component of velocity, $\text{dx}/\text{dt}$ ; page 28.
$\mathbf{u}$	Horizontal velocity, $(u, v)$ ; page 33.
$\mathbf{U}$	Vertically integrated horizontal velocity; page 88.
$\mathbf{U}^{(\rho)}$	Vertically integrated density weighted horizontal velocity; page 88.
$\mathcal{U}$	Horizontal velocity point; page 90.
$\mathbf{U}^{\text{slope}}$	Volume flux for an overflow, from Campin and Goosse (1999); page 47.

Continued on next page...

---

Symbol	Meaning
$u^{\text{slope}}$	Speed for an overflow, from Campin and Goosse (1999); page 46.
$V$	Volume; page 26.
$v$	$y$ component of velocity, $dy/dt$ ; page 28.
$\mathbf{v}$	Velocity, $(u, v, w)$ ; page 28.
$W$	Width of a streamtube; page 59.
$\mathcal{W}$	Vertical velocity point; page 57.
$w$	$z$ component of velocity, $dz/dt$ ; page 28.
$w^{(s)}$	Diasurface velocity component $z_s ds/dt$ ; page 29.
$\mathbf{X}$	Vector of computed solutions for position and velocity; page 125.
$\mathbf{x}$	Vector of position; page 125.
$\dot{\mathbf{x}}$	Blob velocity, with components $\dot{\mathbf{x}} = (\dot{x}, \dot{y}, \dot{z})$ ; page 32.
$\ddot{\mathbf{x}}$	Blob acceleration, with components $\ddot{\mathbf{x}} = (\ddot{x}, \ddot{y}, \ddot{z})$ ; page 32.
$x$	The zonal spatial coordinate (with east as positive).
$\hat{\mathbf{x}}$	The eastward pointing unit vector; page 34.
$\dot{\mathbf{x}}_{\perp}$	Velocity component perpendicular to the solid earth boundary; page 65.
$\dot{\mathbf{x}}_{\text{new}}$	Velocity vector after a blob interacts with topography; page 65.
$\dot{\mathbf{x}}_{\parallel}$	Velocity component parallel to the solid earth boundary; page 65.
$y$	The meridional spatial coordinate (with north as positive).
$\hat{\mathbf{y}}$	The northward pointing unit vector; page 34.
$z$	The vertical spatial coordinate (with up as positive).
$z^*$	The zstar vertical coordinate; page 69.
$\hat{\mathbf{z}}$	The upward pointing unit vector; page 32.

---

### C.3 Greek Symbols

---

Symbol	Meaning
$\alpha$	A Rayleigh drag coefficient with units $s^{-1}$ ; page 54.
$\beta$	A streamtube continuity parameter, from Price and Baringer (1994); page 59.
$\beta_s$	Free parameter; page 126.
$\Gamma$	Tunable proportionality parameter for background detrainment; page 58.
$\gamma$	Blob size scaling parameter; page 44.
$\gamma_s$	Free parameter; page 126.
$\Delta x$	Horizontal grid cell spacing; page 18.
$\Delta z$	Vertical grid cell spacing; page 18.
$\Delta\rho$	Difference in density between two horizontally adjacent grid cells; page 46.
$\Delta\rho_{\text{th}}$	Threshold density difference between the onshelf density and the deep ocean cell that the blob topography intersects; page 62.

---

Continued on next page...

---

## APPENDIX C. LIST OF SYMBOLS AND ABBREVIATIONS

---

Symbol	Meaning
$\Delta\tau$	A time interval (usually the E system's timestep); page 29.
$\delta$	Fraction of a grid box participating in an overflow, from Campin and Goosse (1999); page 46.
$\delta_s$	; page 126
$\varepsilon$	A parameter that dictates the elasticity of the interaction of a blob with topography; page 65.
$\chi$	Vector of exact solutions for position and velocity; page 125.
$\eta$	Sea surface height displace from its resting position at $z = 0$ ; page 52.
$\Theta$	Conservative temperature; page 27.
$\theta$	The clockwise angle between lines of longitude/latitude and the grid; page 116.
$\kappa$	Equivalent vertical diffusivity, from Klinger et al. (1996); page 45.
$\Lambda$	Size scaling for dynamic free blobs; page 56.
$\lambda$	Longitude; page 110.
$\mu$	A prescribed frictional dissipation ( $s^{-1}$ ), from Campin and Goosse (1999); page 46.
$\xi$	A generalised 2D orthogonal coordinate system with components $(\xi_1, \xi_2)$ ; page 109.
$\rho$	Density; page 27.
$\rho_0$	A reference density; page 27.
$\rho'$	Deviations from the average density; page 86.
$\rho_n$	Density of the $n^{\text{th}}$ seawater constituent in a seawater parcel; page 27.
$\rho_{\text{ocn}}$	Density of the bulk ocean in the vicinity of a streamtube; page 59.
$\rho_{\text{st}}$	Average density of a streamtube; page 59.
$\varrho$	Adjustable parameter for step size; page 129.
$\sigma$	Terrain following coordinates; page 69.
$\sigma^{(\text{p})}$	Pressure based terrain following coordinates; page 69.
$\tau$	Frictional stress tensor with components; page 32.
$\tau_{\text{bot}}$	Bottom surface stress with components $(\tau_{\text{bot}}^x, \tau_{\text{bot}}^y, \tau_{\text{bot}}^z)$ ; page 59.
$\phi$	Latitude; page 32.
$\phi_P$	Latitude of a bipolar grid join with a spherical grid; page 115.
$\varphi$	Angle that a blob's trajectory makes topography at the instant of collision; page 66.
$\Phi$	Gravitational potential energy per unit mass (geopotential; $m^2 s^{-2}$ ); page 36.
$\Phi'$	The anomalous geopotential; page 87.
$\Phi_b$	The geopotential at $z = -H$ ; page 87.
$\Psi$	An arbitrary, conserved scalar; page 26.
$\Omega$	Angular velocity of the Earth's rotation; page 32.
$\boldsymbol{\Omega}$	Earth's rotation vector; page 32.

---

## C.4 Model Variables

Symbol	Meaning
dat	Horizontal area of a tracer grid cell; page 91.
dau	Horizontal area of a velocity grid cell; page 91.
dst	Tracer grid cell thickness in the native vertical coordinate of the model; page 82.
dswt	Vertical distance between tracer grid cells in the native vertical coordinate of the model; page 83.
dzt	E system's contribution to the thickness of a tracer grid cell; page 82.
dztE	E system's contribution to the thickness of a tracer grid cell; page 81.
dztL	E system's contribution to the thickness of a tracer grid cell; page 81.
dztlo	E system's contribution to the distance from a tracer point to the bottom of a tracer grid cell; page 82.
dztloE	E system's contribution to the distance from a tracer point to the bottom of a tracer grid cell; page 81.
dztloL	E system's contribution to the distance from a tracer point to the bottom of a tracer grid cell; page 81.
dztloT	Total distance from a tracer point to the bottom of a tracer grid cell; page 81.
dztT	Total thickness of a tracer grid cell; page 82.
dztup	E system's contribution to the distance from a tracer point to the top of a tracer grid cell; page 82.
dztupE	E system's contribution to the distance from a tracer point to the top of a tracer grid cell; page 81.
dztupL	L system's contribution to the distance from a tracer point to the top of a tracer grid cell; page 81.
dztupT	Total distance from a tracer point to the top of a tracer grid cell; page 81.
dzu	E system's contribution to the thickness of a velocity grid cell; page 92.
dzuT	Total thickness of a velocity grid cell; page 91.
dzwt	E system's contribution to the vertical distance between vertical velocity grid points; page 82.
dzwtE	E system's contribution to the vertical distance between vertical velocity grid points; page 81.
dzwtL	L system's contribution to the vertical distance between vertical velocity grid points; page 81.
dzwtT	Total vertical distance between vertical velocity grid points; page 81.
dzwu	E system's contribution to the vertical distance between vertical velocity grid points on the velocity grid; page 92.
min	Minimum function; page 92.

Continued on next page...

Symbol	Meaning
<code>ni</code>	Total number of grid cells in the $i$ direction; page 114.
<code>nj</code>	Total number of grid cells in the $j$ direction; page 124.
<code>nk</code>	Total number of vertical grid cells; page 124.
<code>k</code>	Index for the vertical coordinate; page 80.
<code>kbot</code>	Index for the bottom grid cell; page 80.
<code>rho_dzt</code>	E system's Thickness weighted density of a grid cell; page 82.
<code>rho_dz_tE</code>	E system's Thickness weighted density of a grid cell; page 82.
<code>rho_dzu</code>	E system's Thickness weighted density of a velocity grid cell; page 92.

## C.5 Operators

Symbol	Meaning
$\nabla$	The three dimensional Eulerian gradient operator, $\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$ .

# Bibliography

- Adcroft, A., Hill, C., Marshall, J., 1997. Representation of topography by shaved cells in a height coordinate ocean model. *Monthly Weather Review* 125 (9), 2293–2315.
- Alam, J. M., Lin, J. C., 2008. Toward a fully Lagrangian atmospheric modeling system. *Monthly Weather Review* 136 (12), 4653–4667, doi: 10.1175/2008MWR2515.1.
- Arakawa, A., 1966. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part I. *Journal of Computational Physics* 1, 119–143.
- Ardhuin, F., Herbers, T. H. C., O'Reilly, W. C., 2001. A hybrid Eulerian-Lagrangian model for spectral wave evolution with applications to bottom friction on the continental shelf. *Journal of Physical Oceanography* 31 (6), 1498–1516.
- Aris, R., 1962. Vectors, Tensors and the Basic Equations of Fluid Mechanics. Dover Publications.
- Atkinson, K. E., 1989. An Introduction to Numerical Analysis 2nd Edition. John Wiley & Sons.
- Baines, P. G., Condie, S., 1998. Observations and modelling of Antarctic downslope flows: A review. In: Jacobs, S. S., Weiss, R. F. (Eds.), *Ocean, Ice and Atmosphere: Interactions at the Antarctic Continental Margin*. Vol. 75 of Antarctic Research Series. American Geophysical Union, Washington, D.C., pp. 29–49.
- Beckmann, A., Döscher, R., 1997. A method for improved representation of dense water spreading over topography in geopotential-coordinate models. *Journal of Physical Oceanography* 27 (4), 581–591.
- Beletsky, D., O'Connor, W. P., Schwav, D. J., Dietrich, D. E., 1997. Numerical simulation of internal Kelvin waves and coastal upwelling fronts. *Journal of Physical Oceanography* 27 (7), 1197–1215.
- Bigg, G. R., R, W. M., Stevens, D. P., 1996. Prediction of iceberg trajectories for the North Atlantic and Arctic Oceans. *Geophysical Research Letters* 23 (24), 3587–3590.
- Blumberg, A. F., Mellor, G. L., 1987. A description of a three-dimensional coastal ocean circulation model. In: Heaps, N. (Ed.), *Three-Dimensional Coastal Ocean Models*. Vol. 4 of Coastal and Estuarine Series. American Geophysical Union.

## BIBLIOGRAPHY

---

- Bogacki, P., Shampine, L. F., 1989. A 3(2) pair of Runge-Kutta formulas. *Applied Mathematical Letters* 2 (4), 321–325.
- Bokhove, O., 2000. On hydrostatic flows in isentropic coordinates. *Journal of Fluid Mechanics* 402, 291–310.
- Born, A., Levermann, A., Mignot, J., 2009. Sensitivity of the Atlantic Ocean circulation to a hydraulic overflow parameterisation in a coarse resolution model: Response of the subpolar gyre. *Ocean Modelling* 27 (3–4), 130–142, doi:10.1016/j.ocemod.2008.11.006.
- Briegleb, B. P., Danabasoglu, G., Large, W. G., 2010. An overflow parameterization for the ocean component of the Community Climate System Model. Tech. rep., National Center for Atmospheric Research, Boulder, Colorado.
- Bryan, K., October 1969. A numerical method for the study of the circulation of the world ocean. *Journal of Computational Physics* 4 (3), 347–379.
- Butcher, J. C., 1987. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*. John Wiley & Sons.
- Campin, J.-M., Goosse, H., 1999. Parameterization of density-driven downsloping flow for a coarse-resolution ocean model in z-coordinate. *Tellus* 51A (3), 412–430.
- Campin, J.-M., Hill, C., Jones, H., Marshall, J., 2011. Super-parameterization in ocean modelling: Application to deep convection. *Ocean Modelling* 36 (1–2), 90–101.
- Canuto, V. M., Cheng, Y., Howard, A. M., 2007. Non-local ocean mixing model and a new plume model for deep convection. *Ocean Modelling* 16, 28–46.
- Carmack, E. C., Foster, T. D., 1975. On the flow of water out of the Weddell Sea. *Deep-Sea Research* 22 (11), 711–724.
- Cash, J. R., Karp, A. H., 1990. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software* 16 (3), 201–222.
- Chorin, A. J., 1978. Vortex sheet approximation of boundary layers. *Journal of Computational Physics* 27 (3), 428–442.
- Christiansen, J. P., 1973. numerical simulation of hydrodynamics by the method of point vortices. *Journal of Computational Physics* 13 (3), 363–379.
- Colella, P., Woodward, P. R., 1984. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics* 54, 174–201.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. *Introduction to Algorithms*. The MIT Press.
- Cox, M. D., 1984. A Primitive Equation, 3-Dimensional Model of the Ocean. NOAA/Geophysical Fluid Dynamics Laboratory.

- Danabasoglu, G., Large, W. G., Breigleb, B. P., 2010. Climate impacts of parameterized Nordic Sea overflows. *Journal of Geophysical Research* 115 (C1105), 1–29, doi:10.1029/2010JC006243.
- de Szoeke, R. A., Samelson, R. M., 2002. The duality between the boussinesq and non-boussinesq hydrostatic equations of motion. *Journal of Physical Oceanography* 32 (7), 2194–2203.
- Denbo, D. W., Skjelkvåle, E. D., 1996. An ocean large-eddy simulation model with application to deep convection in the greenland sea. *Journal of Geophysical Research* 101 (C1), 1095–1110.
- Dickson, R., Brown, J., 1994. The production of North-Atlantic deep water-sources, rates and pathways. *Journal of Geophysical Research* 99 (C6), 12139–12341.
- Dietrich, D. E., 1997. Application of a modified Arakawa ‘a’ grid ocean model having reduced numerical dispersion to the Gulf of Mexico circulation. *Dynamics of Atmospheres and Oceans* 27 (1–4), 201–217.
- Dormand, J. R., Prince, P. J., 1980. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics* 6 (1), 19–26.
- Döscher, R., Beckmann, A., 2000. Effects of a bottom boundary layer parameterisation in a coarse-resolution model for the North Atlantic Ocean. *Journal of Atmospheric and Oceanic Technology* 17 (5), 698–707.
- Downes, S. M., Gnanadesikan, A., Griffies, S. M., Sarmiento, J. L., 2011. Water mass exchange in the southern ocean in coupled climate models. *Journal of Physical Oceanography* 41 (9), 1756–1771.
- Dritschel, D. G., Ambaum, M. H. P., 1997. A contour-advection semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields. *Quarterly Journal of the Royal Meteorological Society* 123 (540), 1097–1130.
- Duffy, P. B., Caldeira, K., 1997. Sensitivity of simulated salinity in a three-dimensional ocean model to upper ocean transport of salt from sea ice formation. *Geophysical Research Letters* 24 (11), 1323–1326.
- Durran, D. R., 1999. Numerical Methods for Wave Equations in Geophysical Fluid Dynamics. Springer Verlag.
- Ellis, T. M. R., Philips, I. R., Lahey, T. M., 1994. Fortran 90 Programming. Addison-Wesley Publishers Ltd.
- Ezer, T., Mellor, G. L., 2004. A generalized coordinate ocean model and a comparison of the bottom boundary layer dynamics in terrain-following and in z-level grids. *Ocean Modelling* 6 (3–4), 379–403.

## BIBLIOGRAPHY

---

- Fahrbach, E., Rohardt, G., Scheele, N., Schröder, M., Strass, V., Wisotzki, A., 1995. Formation and discharge of deep and bottom water in the northwestern Weddell Sea. *Journal of Marine Research* 53 (4), 515–538.
- Foldvik, A., Gammelsrød, T., Østerhus, S., Fahrbach, E., Rohardt, G., Schröder, M., Nicholls, K. W., Padman, L., Woodgate, R. A., 2004. Ice shelf water overflow and bottom water formation in the southern Weddell Sea. *Journal of Geophysical Research* 109 (C02015), doi:10.1029/2003JC002008.
- Gent, P. R., McWilliams, J. C., 1990. Isopycnal mixing in ocean circulation models. *Journal of Physical Oceanography* 20 (1), 150–155.
- Gent, P. R., Willebrand, J., McDougall, T. J., McWilliams, J. C., 1995. Parameterizing eddy-induced tracer transports in ocean circulation models. *Journal of Physical Oceanography* 25 (4), 463–474.
- Gill, A. E., 1982. *Atmosphere-Ocean Dynamics*. Vol. 30 of International Geophysics Series. Academic Press.
- Girton, J. B., Sanford, T. B., 2003. Descent and modification of the overflow plume in the Denmark Strait. *Journal of Physical Oceanography* 33 (7), 1351–1364.
- Gordon, A., Visbeck, M., Huber, B., 2001. Export of Weddell Sea deep and bottom water. *Journal of Geophysical Research* 106 (C5), 9005–9017.
- Gordon, A. L., 1982. Weddell Deep Water variability. *Journal of Marine Research* 40, 199–217.
- Gordon, A. L., Orsi, A. H., Muench, R., Huber, B. A., Zambianchi, E., Visbeck, M., 2009. Western Ross Sea continental slope gravity currents. *Deep-Sea Research II* 56 (13–14), 796–817.
- Gordon, A. L., Zambianchi, E., Orsi, A., Visbeck, M., Giulivi, C. F., Whitworth III, T., Spezie, G., 2004. Energetic plumes over the western Ross Sea continental slope. *Geophysical Research Letters* 31 (L21302), doi:10.1029/2004GL020785.
- Griffies, S. M., 2004. *Fundamentals of Ocean Climate Models*. Princeton University Press.
- Griffies, S. M., 2009. Elements of MOM4p1: May 2009 release. Tech. Rep. No. 6, NOAA/Geophysical Fluid Dynamics Laboratory.
- Griffies, S. M., Adcroft, A. J., Banks, H., Böning, C. W., Chassignet, E. P., Danabasoglu, G., Danilov, S., Deleersnijder, E., Drange, H., England, M., Fox-Kemper, B., Gerdes, R., Gnanadesikan, A., Greatbatch, R. J., Hallberg, R. W., Hanert, E., Harrison, M. J., Legg, S., Little, C. M., Madec, G., Marsland, S. J., Nikurashin, M., Pirani, A., Simmons, H. L., Schröter, J., Samuels, B. L., Treguier, A.-M., Toggweiler, J. R., Tsujino, H., Vallis, G. K., White, L., 2010. Problems and prospects in large-scale ocean circulation models. In: Hall, J., Harrison, D. E., Stammer, D. (Eds.), *Proceedings of the OceanObs '09*

- Conference: Sustained Ocean Observations and Information for Society, Venice, Italy, 21-25 September 2009. Vol. 2. ESA Publication WPP-306, pp. 1–23.
- Griffies, S. M., Gnanadesikan, A., Dixon, K. W., Dunne, J. P., Gerdes, R., Harrison, M. J., Rosati, A., Russell, J. L., Samuels, B. L., Spelman, M. J., Winton, M., Zhang, R., 2005. Formulation of an ocean model for global climate simulations. *Ocean Science* 1, 45–79.
- Griffies, S. M., Hallberg, R. W., 2000. Biharmonic friction with a Smagorinsky-like viscosity for use in large-scale eddy-permitting ocean models. *Monthly Weather Review* 128 (8), 2935–2946.
- Griffies, S. M., Harrison, M. J., Pacanowski, R. C., Rosati, A., 2004. A technical guide to MOM4. Tech. Rep. No. 5, NOAA/Geophysical Fluid Dynamics Laboratory.
- Griffies, S. M., Pacanowski, R. C., Schmidt, M., Balaji, V., 2001. Tracer conservation with an explicit free surface method for z-coordinate ocean models. *Monthly Weather Review* 129 (5), 1081–1098.
- Griffiths, R. W., 1986. Gravity currents in rotating systems. *Annual Review of Fluid Mechanics* 18, 59–89.
- Haertel, P. T., Randall, D. A., 2002. Could a pile of slippery sacks behave like an ocean? *Monthly Weather Review* 130 (12), 2975–2988.
- Haertel, P. T., Randall, D. A., Jensen, T. G., 2004. Simulating upwelling in a large lake using slippery sacks. *Monthly Weather Review* 132 (1), 66–77.
- Haertel, P. T., Van Roekel, L., Jensen, T. G., 2009. Constructing an idealized model of the North Atlantic Ocean using slippery sacks. *Ocean Modelling* 27 (3–4), 143–159, doi:10.1016/j.ocemod.2008.12.003.
- Hansen, B., Østerhus, S., 2000. North Atlantic-Nordic Seas exchanges. *Progress in Oceanography* 45, 109–208.
- Hirt, C. W., Amsden, A. A., Cook, J. L., 1972. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics* 14, 227–253.
- Huang, R. X., Jin, X. Z., Zhang, X. H., 2001. An oceanic general circulation model in pressure coordinates. *Advances in Atmospheric Sciences* 18 (1), 1–22.
- Hughes, G. O., Hogg, A. M., Griffiths, R. W., 2009. Available potential energy and irreversible mixing in the meridional overturning circulation. *Journal of Physical Oceanography* 39 (12), 3130–3146.
- Hunter, J. R., Craig, P. D., Phillips, H. E., 1993. On the use of random-walk models with spatially-variable diffusivity. *Journal of Computational Physics* 106 (2), 366–376.
- Ilicak, M., Adcroft, A. J., Griffies, S. M., Hallberg, R. W., 2011a. Spurious dianeutral mixing and the orle of momentum closure. *Ocean Modelling* submitted.

## BIBLIOGRAPHY

---

- Ilicak, M., Legg, S., Adcroft, A., Hallberg, R., 2011b. Dynamics of a dense gravity current flowing over a corrugation. *Ocean Modelling* 38, 71–84.
- IOC, SCOR, IAPSO, 2010. The international thermodynamic equation of seawater-2010: calculation and use of thermodynamic properties. Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO, available from <http://www.TEOS-10.org>, 196pp.
- Jackett, D. R., McDougall, T. J., Feistel, R., Wright, D. G., Griffies, S. M., 2006. Algorithms for density, potential temperature, conservative temperature, and freezing temperature of seawater. *Journal of Atmospheric and Oceanic Technology* 23, 1709–1728.
- Johnson, G. C., 2008. Quantifying Antarctic Bottom Water and North Atlantic Deep Water volumes. *Journal of Geophysical Research* 113 (C05027), doi:10.1029/2007JC004477.
- Jones, H., Marshall, J., 1993. Convection with rotation in a neutral ocean: A study of open-ocean deep convection. *Journal of Physical Oceanography* 23 (6), 1009–1039.
- Kamenkovich, I., Cheng, W., Schmid, C., Harrison, D. E., 2011. Effects of eddies on an ocean observing system with profiling floats: Idealized simulations of the Argo array. *Journal of Geophysical Research* 116 (C06003), 1–14, doi:10.1029/2010JC006910.
- Killworth, P. D., 1977. Mixing on the Weddell Sea continental slope. *Deep-Sea Research* 24 (5), 427–448.
- Killworth, P. D., 1983. Deep convection in the world ocean. *Reviews of Geophysics* 21 (1), 1–26.
- Killworth, P. D., Edwards, N. R., 1999. A turbulent bottom boundary layer code for use in numerical ocean models. *Journal of Physical Oceanography* 29 (6), 1221–1238.
- Kim, S.-J., Stössel, A., 2001. Impact of subgrid-scale convection on global thermohaline properties and circulation. *Journal of Physical Oceanography* 31 (3), 656–674.
- Klinger, B. A., Marshall, J., Send, U., 1996. Representation of convective plumes by vertical adjustment. *Journal of Geophysical Research* 101 (C8), 18,175–18,182.
- Knuth, D. E., 1974. Structured programming with go to statements. *Computing Surveys* 6 (4), 261–301.
- Kösters, F., Käse, R. H., Schmittner, A., Herrmann, P., 2005. The effect of Denmark Strait overflow on the Atlantic Meridional Overturning Circulation. *Geophysical Research Letters* 32, L04602, doi:10.1029/2004GL022112.
- Large, W. G., McWilliams, J. C., Doney, S. C., 1994. Ocean vertical mixing: A review and a model with a nonlocal boundary layer parameterization. *Reviews of Geophysics* 32 (4), 363–403.

- Legg, S., Briegleb, B., Chang, Y., Chassignet, E. P., Danabasoglu, G., Ezer, T., Gordon, A. L., Griffies, S., Hallberg, R., Jackson, L., Large, W., Özgökmen, T. M., Peters, H., Price, J., Riemschneider, U., Wu, W., Xu, X., Yang, J., 2009. Improving oceanic overflow representation in climate models: The Gravity Current Entrainment Climate Process Team. *Bulletin of the American Meteorological Society* 90 (5), 657–670.
- Legg, S., Hallberg, R. W., Girton, J. B., 2006. Comparison of entrainment in overflows simulated by z-coordinate, isopycnal and non-hydrostatic models. *Ocean Modelling* 11 (1–2), 69–97.
- Lin, S.-J., 2004. A “vertically Lagrangian” finit-volume dynamical core for global models. *Monthly Weather Review* 132 (10), 2293–2307.
- MacDonald, A. E., Lee, J. L., Sun, S., 2000. QNH: Design and test of a quasi-nonhydrostatic model for mesoscale weather prediction. *Monthly Weather Review* 128 (4), 1016–1036, doi:10.1175/1520-0493(2000)128.
- Madec, G., Chartier, M., Delecluse, P., Crepon, M., 1991. A three-dimensional numerical study of deep-water formation in the northwestern Mediterranean Sea. *Journal of Physical Oceanography* 21 (9), 1349–1371.
- Marotzke, J., 1991. Influence of convective adjustment on the stability of the thermohaline circulation. *Journal of Physical Oceanography* 21 (6), 903–907.
- Marshall, J., Adcroft, A., Hill, C., Perelman, L., Heisey, C., 1997a. A finite-volume, incompressible navierstokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research* 102 (C3), 5753–5766.
- Marshall, J., Hill, C., Perelman, L., Adcroft, A., 1997b. Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling. *Journal of Geophysical Research* 102 (C3), 5733–5752.
- Marshall, J., Jones, H., Hill, C., 1998. Efficient ocean modeling using non-hydrostatic algorithms. *Journal of Marine Systems* 18, 115–134.
- Marshall, J., Schott, F., 1999. Open-ocean convection: Observations, theory, and models. *Reviews of Geophysics* 37 (1), 1–64.
- Marsland, S. J., Haak, H., Jungclaus, J. H., Latif, M., Röske, F., 2003. The Max-Plank-Institute global ocean/sea ice model with orthogonal curvilinear coordinates. *Ocean Modelling* 5, 91–127.
- Martin, T., Adcroft, A., 2010. Parameterizing the fresh-water flux from land ice to ocean with interactive icebergs in a coupled climate model. *Ocean Modelling* 34 (3–4), 111–124.
- Mauritzen, C., Price, J., Sanford, T., Torres, D., 2005. Circulation and mixing in the Faroese Channels. *Deep-Sea Research I* 52, 883–913.

## BIBLIOGRAPHY

---

- McDougall, T. J., 2003. Potential enthalpy: a conservative oceanic variable for evaluating heat content and heat fluxes. *Journal of Physical Oceanography* 33 (5), 945–963.
- McWilliams, J. C., 2007. Irreducible imprecision in atmospheric and oceanic simulations. *Proceedings of the National Academy of Sciences* 104 (21), 8709–8713.
- Metcalf, M., Reid, J., Cohen, M., 2004. Fortran 95/2003 Explained. Oxford University Press.
- Mohammadian, A., Marshall, J., 2010. A “vortex in cell” model for quasi-geostrophic, shallow water dynamics on the sphere. *Ocean Modelling* 32 (3–4), 132–142.
- Muench, R. D., Wåhlin, A. K., Özgökmen, T. M., Hallberg, R., Padman, L., 2009. Impacts of bottom corrugations on a dense Antarctic outflow: NW Ross Sea. *Geophysical Research Letters* 36 (L23607).
- Munk, W. H., 1950. On the wind-driven circulation. *Journal of Meteorology* 7 (2), 79–93.
- Murray, R. J., 1996. Explicit generation of orthogonal grids for ocean models. *Journal of Computational Physics* 126, 251–273.
- Nakano, H., Sugino, N., 2002. Effects of bottom boundary layer parameterization on reproducing deep and bottom waters in a world ocean model. *Journal of Physical Oceanography* 32 (4), 1209–1227.
- Newell, A., Shaw, J. C., February 1957. Programming the logic theory machine. In: *Proceedings of the Western Joint Computer Conference*. pp. 230–240.
- Nof, D., 1983. The translation of isolated cold eddies on a sloping bottom. *Deep-Sea Research* 30 (2A), 171–182.
- Oliveira, S., Stewart, D., 2006. Writing Scientific Software: A Guide to Good Style. Cambridge University Press.
- Orsi, A. H., Johnson, G. C., Bullister, J. L., 1999. Circulation, mixing and production of Antarctic Bottom Water. *Progress in Oceanography* 43 (1), 55–109.
- Özgökmen, T. M., Fischer, P. F., 2008. On the role of bottom roughness in overflows. *Ocean Modelling* 20 (4), 336–361.
- Pacanowski, R. C., Griffies, S. M., 1999. MOM3.0 manual. Tech. rep., NOAA/Geophysical Fluid Dynamics Laboratory.
- Palusziewicz, T., Romea, R. D., 1997. A one-dimensional model for the parameterization of deep convection in the ocean. *Dynamics of Atmospheres and Oceans* 26, 95–130.
- Pedlosky, J., 1987. Geophysical Fluid Dynamics. Springer-Verlag.

- Piggott, M. D., Gorman, G. J., Pain, C. C., Allison, P. A., Candy, A. S., Martin, B. T., Wells, M. R., 2007. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. *International Journal for Numerical Methods in Fluids*Doi: 10.1002/fld.1663.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1992. *Numerical Recipes in Fortran 77: The Art of Scientific Computing* (2nd Edition). Cambridge University Press.
- Price, J. F., Baringer, M. O., 1994. Outflows and deep water production by marginal seas. *Progress in Oceanography* 33 (3), 161–200.
- Rahmstorf, S., 1993. A fast and complete convection scheme for ocean models. *Ocean Modelling* Nr. 101.
- Rainville, L., Pinkel, R., 2006. Propagation of low-mode internal waves through the ocean. *Journal of Physical Oceanography* 36 (6), 1220–1236.
- Rew, R., Davis, G., Emmerson, S., Davies, H., 2010. The NetCDF Fortran 77 Interface Guide. Tech. Rep. NetCDF 4.1.1, Unidata Program Center.
- Rew, R. K., Davis, G. K., 1990. NetCDF: an interface for scientific data access. *IEEE Computer Graphics and Applications* 10 (4), 76–82.
- Rhines, P. B., MacCready, P., 1989. Boundary control over the large scale circulation. In: *Proceedings of the Fifth 'Aha Huliko'a Hawaiian Winter Workshop*. Hawaii Institute for Geophysics, pp. 75–97.
- Roberts, M. J., Wood, R. A., 1997. Topographic sensitivity studies with a Bryan-Cox-type ocean model. *Journal of Physical Oceanography* 27 (5), 823–836.
- Semtner, A. J., 1974. An oceanic general circulation model with bottom topography. *Numerical Simulation of Weather and Climate*. Tech. Rep. 9, UCLA Department of Meteorology.
- Send, U., Marshall, J., 1995. Integral effects of deep convection. *Journal of Physical Oceanography* 25 (5), 855–872.
- Shampine, L. F., 1994. *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall.
- Shampine, L. F., Reichelt, M. W., 1997. The MATLAB ODE suite. *SIAM Journal on Scientific Computing* 18 (1), 1–22.
- Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM national conference*. ACM '68. ACM, New York, NY, USA, pp. 517–524.  
URL <http://doi.acm.org/10.1145/800186.810616>

## BIBLIOGRAPHY

---

- Sheremet, V. A., 2004. Laboratory experiments with tilted convective plumes on a centrifuge: a finite angle between the buoyancy force and the axis of rotation. *Journal of Fluid Mechanics* 506, 217–244.
- Smagorinsky, J., 1963. General circulation experiments with the primitive equations. *Monthly Weather Review* 91 (3), 99–164.
- Smith, P. C., 1975. A streamtube model for bottom boundary currents in the ocean. *Deep-Sea Research* 22, 853–873.
- Spall, M. A., 2008. Buoyancy-forced downwelling in boundary currents. *Journal of Physical Oceanography* 38 (12), 2704–2721.
- Spiegel, E. A., Veronis, G., 1960. On the Boussinesq approximation for a compressible fluid. *The Astrophysical Journal* 131 (2), 442–447.
- Stommel, H., 1948. The westward intensification of wind-driven ocean currents. *Transactions of the American Geophysical Union* 29, 202–206.
- Straneo, F., Kawase, M., Riser, S. C., 2002. Idealized models of slantwise convection in a baroclinic flow. *Journal of Physical Oceanography* 32 (2), 558–572.
- Tailleux, R., 2010. Identifying and quantifying nonconservative energy production/destruction terms in hydrostatic Boussinesq primitive equation models. *Ocean Modelling* 34 (3–4), 125–136.
- Tseng, Y.-H., Dietrich, D. E., 2006. Entrainment and transport in idealized three-dimensional gravity current simulation. *Journal of Atmospheric and Oceanic Technology* 23 (9), 1249–1269.
- Turner, J. S., 1973. Buoyancy effects in fluids. Cambridge University Press.
- Turner, J. S., 1986. Turbulent entrainment: the development of the entrainment assumption and its applications to geophysical flows. *Journal of Fluid Mechanics* 173, 431–471.
- Umlauf, L., Burchard, H., Bolding, K., 2005. Gotm: Source code and test case documentation: Version 3.2. Tech. Rep. 63, Leibnitz-Institute for Baltic Sea Research, Warnemünde, Germany.
- Vallis, G. K., 2006. Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation. Cambridge University Press.
- White, A. A., Bromley, R. A., 1995. Dynamically consistent, quasi-hydrostatic equations for global models with a complete representation of the coriolis force. *Quarterly Journal of the Royal Meteorological Society* 121, 399–418.
- White, L., Adcroft, A., 2008. A high-order finite volume remapping scheme for nonuniform grids: The piecewise quartic method (PQM). *Journal of Computational Physics* 227 (15), 7394–7422.

- White, L., Adcroft, A., Hallberg, R., 2009. High-order regridding-remapping schemes for continuous isopycnal and generalized coordinates in ocean models. *Journal of Computational Physics* 228 (23), 8665–8692.
- Whitworth, III, T., Orsi, A. H., 2006. Antarctic Bottom Water production and export by tides in the Ross Sea. *Geophysical Research Letters* 33 (L12609), doi:10.1029/2006GL026357.
- Winters, K. B., Lombard, P. N., Riley, J. J., D'Asaro, E. A., 1995. Available potential energy and mixing in density-stratified fluids. *Journal of Fluid Mechanics* 289, 115–128.
- Winton, M., Hallberg, R., Gnanadesikan, A., 1998. Simulation of density-driven downslope flow in z-coordinate ocean models. *Journal of Physical Oceanography* 28 (11), 2163–2174.
- Wirth, A., Barnier, B., 2006. Tilted convective plumes in numerical experiments. *Ocean Modelling* 12, 101–111.
- Wirth, A., Barnier, B., 2008. Mean circulation and structures of tilted ocean deep convection. *Journal of Physical Oceanography* 38 (4), 803–816.
- Wu, W., Danabasoglu, G., Large, W. G., 2007. On the effects of parameterized Mediterranean overflow on North Atlantic ocean circulation and climate. *Ocean Modelling* 19 (1–2), 31–52.