

Balanced Local Data Assimilation with a Blended Numerical Model for Geophysical Flows

Dissertation

zur Erlangung des Grades

eines Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik

der Freien Universität Berlin

vorgelegt von

Ray Chew

Berlin, 2022

Erstgutachter und Betreuer:
Prof. Dr.-Ing. Rupert Klein
Fachbereich Mathematik und Informatik
Freie Universität Berlin

Zweitgutachter:
Prof. Dr. Sebastian Reich
Institut für Mathematik
Universität Potsdam

Tag der Disputation:
28.03.2022

Acknowledgments

I am tremendously grateful to my supervisor Prof. Rupert Klein. When I first started at AG Klein, I had not much of an idea about any of the scientific methods or content that make up this thesis. It was only through the kind and patient guidance by Prof. Klein and the stimulating scientific discussions we had that I managed to come so far in a relatively short time.

I would like to thank Prof. Sebastian Reich for the discussions, for his interest in my work, and for reviewing this thesis. I also thank my mentors Prof. Barbara Wagner and Prof. Andreas Münch, who have been supportive of me during my PhD. I am also grateful to Prof. Wagner whose course on mathematical modelling was what first inspired me to pursue a career in science.

Dr Tommaso Benacchio has been a fantastic senior and collaborator, and he has always been extremely kind and patient in answering my questions and providing suggestions on how I can do better. His attentive proofreading of this thesis has improved it significantly. Dr Mark Schlutow and Dr Nik Nüsken have also kindly proofread my thesis, and they have provided invaluable feedback and discussions that have made this thesis a much better piece of work.

The feedback provided by the four anonymous reviewers of the manuscript that form part of this thesis has been very helpful in improving and refining the blending strategies presented here. Their comments on blended data assimilation have led to additional studies that helped substantiate and validate the results presented in this thesis.

Gottfried Hastermann has helped me settle into AG Klein, and he has always been happy to answer my questions with his impressive repertoire of knowledge. He has also provided valuable discussions that have helped in developing the blending strategies.

Along with Nik, I would also like to mention Dr Liu Xin and Dr Débora Albanez for the friendship, encouragement, and support that they have shown me over the past years.

Finally, I thank Tabea and Tara for always being there for me.

Declaration of authorship

Surname: Chew

First name: Ray

I declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.

I agree to have my thesis examined by a plagiarism examination software.

Abstract

Local sequential Bayesian data assimilation introduces physical imbalances that pose a challenge for geophysical flows with implications for robust numerical weather prediction. The presence of fast-mode imbalances of the order of the relevant slower dynamics deteriorates solution quality. To negate this effect, dynamics-driven methods that suppress imbalances arising from data assimilation are introduced in this thesis. Specifically, a blended numerical model for the rotating compressible fluid flow equations under gravity is employed and equipped with access to sound-proof and hydrostatic dynamics. The blended numerical model is formally extended to support seamless transition between the shallow water equations and lake equations. Through careful numerical and asymptotic analysis, one-step blending strategies that enable seamless switching between model regimes within a simulation run are developed. Upon assimilation of data, the model configuration is switched for one time step to the limit regime. After that, the model configuration is switched back to the compressible or shallow water regime for the duration of the assimilation window. This switching between the model regimes is repeated for each subsequent assimilation window to eliminate the imbalances arising from the assimilation of data. Idealised experiments involving the travelling vortex, buoyancy-driven rising thermals, and inertia gravity waves demonstrate that the blending strategies successfully eliminate unphysical imbalances, yielding up to two orders-of-magnitude improvements in the error scores. This novel dynamics-conforming method of achieving balanced data assimilation can be extended to eliminate other forms of imbalances, and it has the potential to reduce data assimilation-generated spurious signals in numerical weather prediction simulations.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Contributions	4
1.3	Thesis structure	5
2	The analytical models	7
2.1	Compressible flow equations	7
2.2	Pseudo-incompressible equations	8
2.3	Hydrostatic primitive equations	9
2.4	Shallow water equations	10
2.5	Lake equations	12
2.6	Overview of the analytical models	14
3	The blended numerical model	15
3.1	Summary of the numerical scheme	15
3.2	Discretisation details	17
3.2.1	Obtaining the numerical fluxes	17
3.2.2	The advection routine	17
3.2.3	Explicit Euler update	22
3.2.4	Implicit Euler update	25
3.3	Implementation details	28
4	Blending strategies	29
4.1	Compressible–soundproof blending	29
4.1.1	Time level of the pressure-related variables	29
4.1.2	Conversion of the pressure-related variables	34
4.1.3	Choice of the perturbation variables	34
4.2	Nonhydrostatic–hydrostatic blending	35
4.2.1	Half time predictor step	36
4.2.2	Full time update	39
4.2.3	Enforcing the hydrostatic balance	43

Contents

4.2.4	Second-order pressure update	44
4.2.5	Nonhydrostatic-to-hydrostatic blending	44
4.2.6	Future work	45
4.3	Shallow water–lake blending	45
4.4	Overview of the blending strategies	47
4.5	Implementation details	47
5	Bayesian data assimilation	49
5.1	A short primer	49
5.1.1	Variational data assimilation methods	50
5.1.2	The Kalman filters	51
5.2	The local ensemble transform Kalman filter (LETKF)	54
5.3	Data assimilation and blending	55
5.4	Implementation details	58
6	Numerical results	59
6.1	The idealised test cases	59
6.1.1	The travelling vortices	59
6.1.2	Rising bubble	63
6.1.3	Inertia-gravity wave	64
6.2	Effectiveness of the blending schemes	66
6.2.1	Travelling vortex results	67
6.2.2	Rising bubble results	69
6.2.3	Inertia-gravity wave results	72
6.3	Ensemble data assimilation and blending (DAB): setup	74
6.3.1	The travelling vortices	74
6.3.2	Rising bubble: DAB setup	76
6.3.3	Inertia-gravity wave: DAB setup	78
6.3.4	Evaluation of data assimilation	79
6.4	Ensemble data assimilation and blending: results	80
6.4.1	The travelling vortices: DAB results	80
6.4.2	Rising bubble: DAB results	85
6.4.3	Inertia-gravity wave: DAB results	87

7 Conclusion	91
7.1 Overview of the appendices	93
7.2 Outlook and future work	94
A Modelling the observation error	97
B Scale analysis of the data assimilation error in the pressure-related fields	101
C Ensemble spread of the data assimilation experiments	105
D On the convergence of the data assimilation system	109
E Localisation region and imbalances	113
References	119

1 Introduction

The dynamics of the atmosphere evolve on a vast range of spatial and temporal scales, and various mathematical models have been developed to simulate these dynamics. The compressible Euler equations provide the most comprehensive description of atmospheric processes, including the effects of sound waves. Yet the effects of sound waves may not be of physical and meteorological interest in the simulation of small- to mesoscale phenomena. Moreover, in the low-Mach regime, phenomena of interest evolve much slower than the speed of sound, and resolving sound waves may introduce stiffness into the numerical scheme, limiting the time-step size of the simulation.

Limit models have been derived from the full compressible equations by means of scale analysis or asymptotic analysis. The limit soundproof pseudo-incompressible model suppresses the effects of sound waves and is suitable for investigating small- to mesoscale phenomena (Durran, 1989). The hydrostatic primitive equations suppress the effects of vertically propagating acoustic waves, and this model is best used to describe phenomena occurring over a large horizontal scale and a relatively small vertical scale, i.e., in the limit where the aspect ratio of the domain is large (White et al., 2005). More details on these models can be found in Klein (2010), Pedlosky (2013), and Vallis (2017).

Separate numerical discretisation schemes can be developed and implemented for the full compressible model and its limit models. However, discrepancies in the solutions arising from different discretisation schemes of the same equation set can be larger than the discrepancies in the solutions of different equation sets that have been discretised with the same numerical scheme (Smolarkiewicz and Dörnbrack, 2008; Klein, 2009). This justifies the development of a single numerical scheme that discretises the full compressible model and its limit models.

The discretisation and implementation of the full compressible and pseudo-incompressible equations within a single numerical framework were realised by Benacchio et al. (2014); Klein et al. (2014) and separately by Smolarkiewicz et al. (2014). In the numerical framework by Benacchio et al. (2014), continuously or discontinuously tuning an appropriate *blending parameter* switches the solution between the two model regimes, and results from acoustically balanced simulations with both model regimes are essentially identical. The authors further demonstrated that by solving the first time steps in the pseudo-incompressible regime before switching the solution to the compressible regime, acoustically balanced solutions can be obtained despite the initially acoustically imbalanced data. This indicates that, when accessible within a single numerical scheme, the dynamics of a limit model can be exploited to achieve balanced data initialisation.

The *blended numerical model* by Benacchio et al. (2014) was conceptually extended by Klein and Benacchio (2016) to support an additional blending parameter that tunes between the compressible nonhydrostatic equations and a variant of the hydrostatic primitive equations, and another blending parameter to tune between the full compressible model and the unified Arakawa and Konor (2009) model. Benacchio and Klein (2019) then developed a numerical scheme that supports blending between the hydrostatic and nonhydrostatic dynamics as well as between the pseudo-incompressible and compressible dynamics. Furthermore, they showed that for balanced initial data, simulation results between the full and the limit regimes are essentially identical.

1.1 Motivation

The blended numerical models in Benacchio et al. (2014) and Benacchio and Klein (2019) achieve the switching of model regimes by the tuning of an appropriate blending parameter. While this changes the equation set that is simulated, additional consideration must be taken to account for the differences in the numerical schemes and in the physical meaning of the quantities in the limit regimes. Therefore in addition to parameter tuning, blending strategies between different model regimes can be developed through careful numerical and asymptotic analysis. Such blending strategies drastically reduce the computational overhead of the blending process and the error induced by the blending procedure.

Aside from the computational efficiency gained, blending strategies that are based on dynamical and numerical considerations avoid the necessity of continuously tuning the blending parameter over multiple time steps. Such a continuous tuning was, for example, required to achieve the best blended solution in Benacchio et al. (2014). Analytically, the physics simulated by the intermediate model regimes accessed by a continuous tuning of the blending parameter is not well understood for the Euler equations. Development of dynamics-driven one-step blending strategies sidesteps this issue entirely, and only the full and the limit regimes are accessed by the blending procedure.

A first application of the improved blending method is in achieving balanced data initialisation. By employing the blending strategies developed in this thesis, one time step spent in the limit regime is shown to be sufficient to remove imbalances in the initial data. A strong reduction of the initial imbalances for simulations involving balanced data initialisation will establish the effectiveness of the blending strategies introduced. These are also the first results of blending between model regimes within a single simulation run for the numerical scheme developed by Benacchio and Klein (2019).

Extended with the improved blending strategies, the blended numerical model is then used to investigate the effectiveness of blending in achieving balanced data assimilation. Spatial localisation is usually applied to a data assimilation method, as it prevents spurious correlation of the observations, and the assimilation of observations may be computationally more efficient

(Kirchgessner et al., 2014; Hunt et al., 2007). Localisation is achieved either by considering only observations in the vicinity of the location where data is assimilated, or by applying a distance-dependent function that reduces the influence of the observations to zero beyond a certain distance from the location where data is assimilated, or by a combination of these two methods. Local data assimilation methods may introduce imbalances especially when the localisation is severe (Lorenc, 2003; Cohn et al., 1998; Mitchell et al., 2002). As the assimilation procedure does not take into account the physical dynamics of the flow processes, e.g. the conservation of mass, momentum, and energy, or of the structure and smoothness of the solution, initially balanced states may be destroyed by the assimilation procedure. The effects of localisation on balanced analysis fields are discussed in, e.g., Neef et al. (2006), Greybush et al. (2011), and Bannister (2015).

The fast-mode imbalances introduced through the data assimilation procedure may be of the same order of magnitude as the slower dynamics of interest and may potentially deteriorate the solution quality (Hohenegger and Schär, 2007). As with achieving balanced data initialisation, the blending procedure may suppress the imbalances by exploiting the dynamics of the appropriate limit model. After the assimilation of data, the solution is projected from the full model onto the limit model. For example, the suppression of acoustic imbalances is achieved by switching the numerical model from the compressible regime to the soundproof pseudo-incompressible regime. The solution is then solved for one time step in the limit regime and the numerical model is then switched back to simulate the full dynamics. The rest of the time steps in the assimilation window until the next assimilation time are solved with the full model.

The blending strategies developed here ensure that the solution from this method of switching between model regimes to suppress imbalances is dynamically consistent with the underlying model at each step of the process. Therefore, being dynamics-driven, this method of achieving balanced data assimilation is fundamentally different from existing methods to achieve balanced solutions after the assimilation of data. The digital filter initialisation (DFI, Lynch and Huang, 1992) and the incremental analysis update (IAU, Bloom et al., 1996) act as low pass filters, and the repeated application of these filters may have a detrimental effect on the slower dynamics that are of interest (Houtekamer and Zhang, 2016; Polavarapu et al., 2004).

Apart from the IAU and DFI, balance can be improved by the choice of localisation space (Kepert, 2009b), or by allowing observations outside of a localisation radius to relax to a certain climatological mean (Flowerdew, 2015). A study comparing the effectiveness of the blending method with parameter tuning against the effectiveness of a post-analysis penalty method in achieving balanced analysis fields for oscillatory system was done by Hastermann et al. (2021). Reviews on balanced data assimilation can be found in Zupanski (2009) and Houtekamer and Zhang (2016). Results presented in this thesis seek to establish the feasibility and effectiveness of blending as a means of achieving balanced data assimilation, and comparisons with existing balancing methods are deferred to future work.

1.2 Contributions

The main innovations and advances to existing methods in this thesis are listed below:

- The transition between all four combinations of the compressible / pseudo-incompressible, and nonhydrostatic / hydrostatic models within a single numerical scheme by the tuning of appropriate blending parameters was realised by Benacchio and Klein (2019). In this thesis, the blended numerical framework is formally extended to support solution of the horizontal rotating shallow water equations and the lake equations. This is done by careful identification of the parameters in the blended numerical framework with those of the shallow water equations and the lake equations. As with the other models accessible by the blended numerical model, the transition between the shallow water equations and the lake equations is also realised by an appropriate blending parameter.
- A one-step blending strategy is developed for the pseudo-incompressible and compressible model pair that enables the instantaneous switching between model regimes. This blending strategy is based on three innovations: 1) by accounting for the time-level difference in the pressure field between the full flow model and the limit model; 2) by judicious conversion of the pressure-related quantities at the blending interface, taking into consideration their multi-scale nature; and 3) by a careful choice of the perturbation variables in the conversion from the limit model to the full model. This leads to a substantial improvement over Benacchio et al. (2014), who needed additional time steps in the limit regime, in the ability of the blended model in suppressing unwanted signals.
- A one-step blending procedure is derived for the hydrostatic and nonhydrostatic model pair. This blending strategy is based on three innovations: 1) the realisation that the solution of a second-order time update requires a balanced vertical momentum field as its input for the solution to remain balanced; 2) by judiciously applying two first-order time updates, a vertical momentum field that is balanced with respect to the other quantities can be recovered; and 3) the choice of a second-order time update for the pressure perturbation variable leads to an improvement to the blended solution. This is a first study into the principal capability of hydrostatic and nonhydrostatic blending.
- The novel blending strategies are used to achieve balanced data assimilation in a compressible model, and the effects of data assimilation and blending on balanced solutions are investigated. Numerical results featuring blending as a means to achieve balanced local data assimilation are presented. Scale analysis shows that the error introduced by the assimilation of data quantifies the imbalances introduced.

Balanced data initialisation and balanced data assimilation are investigated in idealised two-dimensional numerical experiments. The travelling vortex test case on a horizontal plane (Kadioglu et al., 2008) is investigated for the shallow water / lake blending and the compressible

/ pseudo-incompressible blending. A warm rising bubble in a vertical slice is also investigated for the compressible / pseudo-incompressible blending (Mendez-Nunez and Carroll, 1994; Klein, 2009). For the nonhydrostatic / hydrostatic blending, vertical slice numerical experiments on the inertia-gravity wave in a channel by Skamarock and Klemp (1994) are conducted. For each of these test cases, employing an untuned and unbalanced data assimilation method alone destroys the solution quality, while data assimilation together with blending successfully suppresses imbalances. This establishes the feasibility of employing the blended numerical model in achieving balanced data assimilation that is consistent with the underlying dynamics.

The numerical experiments are conducted with a modular Python code that was implemented from the ground up. The code consists of four major components: a flow solver for the blended numerical model, a data assimilation engine, a blending module that interfaces the first two components, and a suite of tools to analyse, debug, and visualise the results of the simulations. The blended numerical flow solver component is based on a refactoring of the C code by Benacchio and Klein (2019). Due to extensive use of vectorisation and run-time compilation, the Python code is reasonably fast, and benchmarks involving an early optimised version of the Python code demonstrated that for simulations on a (64×64) grid, the Python code is about as fast as the C code.

1.3 Thesis structure

Chapter 1 is this chapter and serves as an introduction to the thesis. Chapter 2 describes the mathematical models accessible by the blended numerical framework, and the blended numerical model of Benacchio and Klein (2019) is also formally extended to support simulations of the shallow water equations and the lake equations. Chapter 3 elaborates on the discretisation and implementation details of the blended numerical framework, and chapter 4 develops the blending strategies for the compressible / pseudo-incompressible, the nonhydrostatic / hydrostatic, and the shallow water / lake model pairs. Chapter 5 presents a brief discussion on data assimilation and the ensemble data assimilation method used in the numerical experiments. The implementation details of the data assimilation algorithm are also elaborated upon. Finally, details are given on combining blending with data assimilation. Numerical results presented in chapter 6 are divided into four sections: 1) the configurations and setup of the idealised test cases used; 2) the effectiveness of the blending strategies in achieving balanced data initialisation; 3) the blended ensemble data assimilation setup; followed by 4) studies on the capabilities of blending in removing imbalances arising from the assimilation of data. Chapter 7 concludes this thesis.

Parts of the content in chapters 4, 5, and 6 can be found in the preprint Chew et al. (2021) that is currently under review.

2 The analytical models

The mathematical models accessible by the blended numerical model are briefly discussed in this chapter. The limit soundproof pseudo-incompressible and hydrostatic models are accessible by the toggling of the appropriate switches and further details can be found in Klein and Benacchio (2016); Benacchio and Klein (2019), while an extension of the blending framework to the shallow water equations and the lake equations is formally derived. An overview relating the models to one another can be found at the end of this chapter.

2.1 Compressible flow equations

The compressible nonhydrostatic Euler equations present a description of the dynamical processes in the atmosphere (Pedlosky, 2013; Vallis, 2017). The equations in a rotating Cartesian coordinate system with gravity are,

$$\rho_t + \nabla_{\parallel} \cdot (\rho \mathbf{u}) + (\rho w)_z = 0, \quad (2.1a)$$

$$(\rho \mathbf{u})_t + \nabla_{\parallel} \cdot (\rho \mathbf{u} \circ \mathbf{u}) + (\rho w \mathbf{u})_z = - (c_p P \nabla_{\parallel} \pi + f \mathbf{k} \times \rho \mathbf{u}), \quad (2.1b)$$

$$\alpha_w [(\rho w)_t + \nabla_{\parallel} \cdot (\rho \mathbf{u} w) + (\rho w^2)_z] = - (c_p P \pi_z + \rho g), \quad (2.1c)$$

$$\alpha_P P_t + \nabla_{\parallel} \cdot (P \mathbf{u}) + (P w)_z = 0. \quad (2.1d)$$

In (2.1), ρ is the density, $\mathbf{u} = (u, v)$ the vector of horizontal velocities, and w the vertical velocity. The variable P is the mass-weighted potential temperature and π is the Exner pressure. The quantity f is the Coriolis parameter on the horizontal (x, y) -plane, \mathbf{k} a unit vector in the vertical direction and \times the cross product. The quantity g is the acceleration of gravity acting in the direction of \mathbf{k} . The symbol \circ denotes the tensor product, ∇_{\parallel} denotes the horizontal gradient while the subscripts t and z denote the partial derivatives with respect to time t and the vertical coordinate z . The variables P and π are related to the thermodynamic pressure p by the equations of state,

$$\pi = \left(\frac{p}{p_{\text{ref}}} \right)^{R/c_p}, \quad P = \frac{p_{\text{ref}}}{R} \left(\frac{p}{p_{\text{ref}}} \right)^{c_v/c_p} = \rho \theta, \quad (2.2)$$

where p_{ref} is a reference pressure, c_p and c_v are the heat capacities at constant pressure and constant volume, $R = c_p - c_v$ is the ideal gas constant, and θ is the potential temperature. α_P and α_w are the *blending parameters*. α_P switches between the compressible model and the pseudo-incompressible model, and α_w switches between the hydrostatic model and the nonhydrostatic model (Klein and Benacchio, 2016).

Identifying χ with the inverse potential temperature

$$\chi = \frac{1}{\theta}, \quad (2.3)$$

the Exner pressure and inverse potential temperature can be decomposed as

$$\pi(t, x, y, z) = \bar{\pi}(z) + \pi'(t, x, y, z) \quad \text{and} \quad (2.4a)$$

$$\chi(t, x, y, z) = \bar{\chi}(z) + \chi'(t, x, y, z), \quad (2.4b)$$

where the bar denotes a hydrostatic background state, which depends only on the vertical coordinate, and the prime denotes the perturbation. From (2.2), P is solely a function of π ,

$$P(\pi) = \frac{p_{\text{ref}}}{R} \pi^{\frac{1}{\gamma-1}}, \quad (2.5)$$

where $\gamma = c_p/c_v$ is the isentropic exponent. With (2.5), (2.1d) can be written as

$$\frac{\partial P}{\partial \pi} \pi_t = -\nabla \cdot (P\mathbf{v}). \quad (2.6)$$

2.2 Pseudo-incompressible equations

The pseudo-incompressible set of equations was first derived by Durran (1989) and the formulation used in the blended numerical framework is discussed below.

Expression (2.4a) separates the background Exner pressure from its perturbation. For the low Mach number regime $\text{Ma} \ll 1$, where $\text{Ma} = u_{\text{ref}}/c_{\text{ref}}$ for reference velocity u_{ref} and speed of sound c_{ref} , an alternative formulation is the asymptotic expansion

$$\pi(t, x, y, z) = \pi^{(0)}(z) + \text{Ma}^2 \pi^{(1)}(t, x, y, z) + \dots, \quad (2.7)$$

with the superscript (0) representing the leading order expansion of π and (1) the next-to-leading order expansion. Substituting this expansion into (2.6),

$$\left(\frac{\partial P}{\partial \pi} \right) \text{Ma}^2 \pi_t^{(1)} = -\nabla \cdot (P\mathbf{v}), \quad (2.8)$$

and in the soundproof limit of $\text{Ma} \rightarrow 0$, the left-hand side tends to zero. This implies that $P_t = 0$ in (2.1d), and P takes the role of the leading order time-independent background quantity $P^{(0)}$, with $P^{(0)} = (\pi^{(0)})^{1/(\gamma-1)}$. The quantity π comprises a time-dependent component corresponding to the next-to-leading order $\pi^{(1)}$. Putting together these insights, the pseudo-incompressible set of equations becomes

$$\rho_t + \nabla_{\parallel} \cdot (\rho \mathbf{u}) + (\rho w)_z = 0, \quad (2.9a)$$

$$(\rho \mathbf{u})_t + \nabla_{\parallel} \cdot (\rho \mathbf{u} \circ \mathbf{u}) + (\rho w \mathbf{u})_z = - \left[c_p P^{(0)} \nabla \pi + f \mathbf{k} \times \rho \mathbf{u} \right], \quad (2.9b)$$

$$(\rho w)_t + \nabla_{\parallel} \cdot (\rho \mathbf{u} w) + (\rho w^2)_z = - \left[c_p P^{(0)} \pi_z + \rho g \right], \quad (2.9c)$$

$$\nabla_{\parallel} \cdot (P^{(0)} \mathbf{u}) + (P^{(0)} w)_z = 0, \quad (2.9d)$$

where (2.9d) enforces the soundproof divergence constraint. We note the similarity between (2.1) and (2.9), with $\alpha_P = 0$ in (2.1d) and P replaced with the leading order $P^{(0)}$ in the latter. These differences have to be taken into account in a blending strategy between the compressible model and the pseudo-incompressible model.

As is evident from its name, the soundproof pseudo-incompressible model suppresses acoustic effects, and sound waves are absent in simulations with this equation set. More details on this formulation of the pseudo-incompressible equations can be found in Klein (2009), Klein and Pauluis (2012), Benacchio et al. (2014), and Klein and Benacchio (2016).

2.3 Hydrostatic primitive equations

The hydrostatic balance

$$c_p P_h (\pi_h)_z = -\rho_h g \quad (2.10)$$

is achieved by setting the left-hand side of (2.1c) to zero. Here, the subscript h denotes the hydrostatically balanced quantities, specifically

$$P_h = \pi_h^{\frac{1}{\gamma-1}}, \quad P_h = \rho_h \theta. \quad (2.11)$$

The hydrostatic primitive equations are as follows (Vallis, 2017; Klein and Benacchio, 2016),

$$(\rho_h)_t + \nabla_{\parallel} \cdot (\rho_h \mathbf{u}) + (\rho_h w)_z = 0, \quad (2.12a)$$

$$(\rho_h \mathbf{u})_t + \nabla_{\parallel} \cdot (\rho_h \mathbf{u} \circ \mathbf{u}) + (\rho_h w \mathbf{u})_z = - \left[c_p P_h \nabla_{\parallel} \pi_h + f \mathbf{k} \times \rho_h \mathbf{u} \right], \quad (2.12b)$$

$$c_p P_h (\pi_h)_z = -\rho_h g, \quad (2.12c)$$

$$(P_h)_t + \nabla_{\parallel} \cdot (P_h \mathbf{u}) + (P_h w)_z = 0. \quad (2.12d)$$

Note that this is akin to setting $\alpha_w = 0$ in (2.1d) and ensuring that the quantities ρ_h , P_h , and π_h are hydrostatically balanced by the constraint in (2.12c). Again, a blending between the nonhydrostatic model and the hydrostatic model has to account for these differences, and such a blending strategy is developed later in this thesis. The hydrostatic model suppresses vertically propagating acoustic waves, and horizontally propagating acoustic waves, e.g. the Lamb waves, are left unaffected.

2.4 Shallow water equations

The horizontal rotating shallow water equations with no orography are

$$h_t + \nabla \cdot (h\mathbf{u}) = 0, \quad (2.13a)$$

$$(h\mathbf{u})_t + \nabla \cdot (h\mathbf{u} \circ \mathbf{u}) = - \left[\frac{g}{2} \nabla h^2 + f\mathbf{k} \times h\mathbf{u} \right], \quad (2.13b)$$

where h is the total water depth. From here on, the horizontal subscript \parallel in the nabla symbols is implied. More details on the shallow water equations can be found in Bresch (2009). In order to obtain a formulation for the blended numerical model, we non-dimensionalise the variables in (2.13) with

$$\begin{aligned} h &= h_{\text{ref}} h^*, & \nabla &= \frac{1}{l_{\text{ref}}} \nabla^* & t &= t_{\text{ref}} t^*, \\ \mathbf{u} &= \frac{l_{\text{ref}}}{t_{\text{ref}}} \mathbf{u}^*, & g &= \frac{h_{\text{ref}}}{t_{\text{ref}}^2} g^*, & f &= \frac{1}{t_{\text{ref}}} f^*, \end{aligned} \quad (2.14)$$

where both l_{ref} and h_{ref} have the units of length. Rewriting (2.13) with (2.14),

$$\frac{h_{\text{ref}}}{t_{\text{ref}}} h_{t^*}^* + \frac{h_{\text{ref}}}{t_{\text{ref}}} \nabla^* \cdot (h^* \mathbf{u}^*) = 0, \quad (2.15a)$$

$$\frac{h_{\text{ref}} l_{\text{ref}}}{t_{\text{ref}}^2} [(h^* \mathbf{u}^*)_{t^*} + \nabla^* \cdot (h^* \mathbf{u}^* \circ \mathbf{u}^*)] = - \left[\frac{h_{\text{ref}}^3}{l_{\text{ref}} t_{\text{ref}}^2} \frac{g^*}{2} \nabla^* (h^*)^2 + \frac{h_{\text{ref}} l_{\text{ref}}}{t_{\text{ref}}^2} f^* \mathbf{k} \times h^* \mathbf{u}^* \right]. \quad (2.15b)$$

Multiplying (2.15b) by $t_{\text{ref}}^2/(h_{\text{ref}} l_{\text{ref}})$ and identifying the Froude number squared

$$\text{Fr}^2 = \frac{l_{\text{ref}}^2}{h_{\text{ref}}^2} \frac{1}{g^*} = \frac{u_{\text{ref}}^2}{gh_{\text{ref}}}, \quad (2.16)$$

equation (2.15) in non-dimensionalised form is

$$h_t + \nabla \cdot (h\mathbf{u}) = 0, \quad (2.17a)$$

$$(h\mathbf{u})_t + \nabla \cdot (h\mathbf{u} \circ \mathbf{u}) = - \left[\frac{1}{2\text{Fr}^2} \nabla h^2 + f\mathbf{k} \times h\mathbf{u} \right], \quad (2.17b)$$

where the superscript $*$ denoting the non-dimensionalised quantities is dropped.

Let us now turn our attention to the compressible Euler equations in (2.1) and derive a form similar to (2.17). Rewriting the horizontal components of the compressible set of equations in (2.1) with the equations of state in (2.2), we obtain a set of equations in terms of the physical pressure p ,

$$\rho_t + \nabla \cdot (\rho\mathbf{u}) = 0, \quad (2.18a)$$

$$(\rho\mathbf{u})_t + \nabla \cdot (\rho\mathbf{u} \circ \mathbf{u}) = - [\nabla p + f\mathbf{k} \times \rho\mathbf{u}], \quad (2.18b)$$

$$\alpha_P(\rho\theta)_t + \nabla \cdot (\rho\theta\mathbf{u}) = 0. \quad (2.18c)$$

With $\Gamma = \gamma - 1/\gamma$ and $p_{\text{ref}}/\rho_{\text{ref}} = c_{\text{ref}}^2$, with c_{ref} being the speed of sound, we introduce the following reference quantities,

$$\begin{aligned} \rho &= \rho_{\text{ref}} \rho^*, & \nabla &= \frac{1}{l_{\text{ref}}} \nabla^*, & t &= t_{\text{ref}} t^*, & \mathbf{u} &= \frac{l_{\text{ref}}}{t_{\text{ref}}} \mathbf{u}^*, \\ p &= p_{\text{ref}} p^*, & f &= \frac{1}{t_{\text{ref}}} f^*, & \theta &= \frac{p_{\text{ref}}^\Gamma}{\rho_{\text{ref}} R} \theta^*. \end{aligned} \quad (2.19)$$

Rewriting (2.18) with (2.19),

$$\frac{\rho_{\text{ref}}}{t_{\text{ref}}} [\rho_{t^*}^* + \nabla^* \cdot (\rho^* \mathbf{u}^*)] = 0, \quad (2.20a)$$

$$\frac{\rho_{\text{ref}} l_{\text{ref}}}{t_{\text{ref}}^2} [(\rho^* \mathbf{u}^*)_{t^*} + \nabla^* \cdot (\rho^* \mathbf{u}^* \circ \mathbf{u}^*)] = - \left[\frac{p_{\text{ref}}}{l_{\text{ref}}} \nabla^* p^* + \frac{\rho_{\text{ref}} l_{\text{ref}}}{t_{\text{ref}}^2} f^* \mathbf{k} \times \rho^* \mathbf{u}^* \right], \quad (2.20b)$$

$$\frac{p_{\text{ref}}^\Gamma}{R t_{\text{ref}}} [(\rho^* \theta^*)_{t^*} + \nabla^* \cdot (\rho^* \theta^* \mathbf{u}^*)] = 0. \quad (2.20c)$$

Multiplying (2.20b) by $t_{\text{ref}}^2/(\rho_{\text{ref}} l_{\text{ref}})$, (2.20) in non-dimensionalised form is

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.21a)$$

$$(\rho \mathbf{u})_t + \nabla \cdot (\rho \mathbf{u} \circ \mathbf{u}) = - \left[\frac{1}{\text{Ma}^2} \nabla p + f \mathbf{k} \times \rho \mathbf{u} \right], \quad (2.21b)$$

$$\alpha_P (\rho \theta)_t + \nabla \cdot (\rho \theta \mathbf{u}) = 0, \quad (2.21c)$$

where

$$\text{Ma}^2 = \frac{l_{\text{ref}}^2 \rho_{\text{ref}}}{t_{\text{ref}}^2 p_{\text{ref}}} = \frac{u_{\text{ref}}^2}{c_{\text{ref}}^2} \quad (2.22)$$

is the Mach number squared. In order to relate (2.17) to (2.21), we make the following identifications,

$$p = \frac{gh^2}{2}, \quad \gamma = 2, \quad p_{\text{ref}} = \frac{h_{\text{ref}}^3}{t_{\text{ref}}^2}. \quad (2.23)$$

Indeed, inserting the identifications in (2.23) into the equations of state in (2.2),

$$\pi = \left(\frac{1}{p_{\text{ref}}} \right)^{\frac{1}{2}} \left(\frac{g}{2} h^2 \right)^{\frac{1}{2}} = \sqrt{\frac{t_{\text{ref}}^2}{h_{\text{ref}}^3}} \left(\sqrt{\frac{g}{2}} h \right), \quad (2.24a)$$

$$P = \frac{p_{\text{ref}}^{\frac{1}{2}}}{R} \left(\frac{g}{2} h^2 \right)^{\frac{1}{2}} = \frac{1}{R} \sqrt{\frac{h_{\text{ref}}^3}{t_{\text{ref}}^2}} \left(\sqrt{\frac{g}{2}} h \right), \quad (2.24b)$$

and setting the expansions in (2.24) into the first term on the right-hand side of the compressible horizontal momentum subequation (2.1b),

$$-c_p P \nabla \pi = -\frac{c_p g}{R} h \nabla h = -gh \nabla h = -\frac{g}{2} \nabla h^2, \quad (2.25)$$

2 The analytical models

which is the first term on the right-hand side of non-dimensionalised shallow water momentum subequation (2.17b) with g subsumed under the Froude number squared. Therefore, the shallow water equations can be solved by the blended numerical model with the identifications in (2.23) and

$$\rho \rightarrow h, \quad \text{Ma}^2 \rightarrow 2 \text{Fr}^2, \quad (2.26)$$

giving the governing non-dimensionalised rotating horizontal shallow water equations

$$\alpha_h h_t + \nabla \cdot (h \mathbf{u}) = 0, \quad (2.27a)$$

$$(h \mathbf{u})_t + \nabla \cdot (h \mathbf{u} \circ \mathbf{u}) = - \left[\frac{1}{2 \text{Fr}^2} \nabla h^2 + f \mathbf{k} \times h \mathbf{u} \right]. \quad (2.27b)$$

Here, α_h is the blending parameter that switches between the shallow water equations and the lake equations. Analogous to the $\alpha_P = 0$ switch to the pseudo-incompressible equations, $\alpha_h = 0$ toggles access to the lake equations.

2.5 Lake equations

The lake equation is derived from the vanishing Froude number limit of the governing non-dimensionalised shallow water equations in (2.27). Identifying 2Fr^2 with a small parameter $\varepsilon \ll 1$, we assume the following asymptotic expansions,

$$h = h^{(0)} + \varepsilon h^{(1)} + \dots, \quad (2.28a)$$

$$\mathbf{u} = \mathbf{u}^{(0)} + \varepsilon \mathbf{u}^{(1)} + \dots. \quad (2.28b)$$

Substituting the expansions in (2.28) into the non-dimensionalised shallow water equations in (2.27) and sorting by the order of expansion, the leading order expansion for the water depth equation (2.27a) is

$$O(\varepsilon^0): \quad h_t^{(0)} + \nabla \cdot (h^{(0)} \mathbf{u}^{(0)}) = 0. \quad (2.29)$$

For the momentum equation (2.27b), the leading order expansion is

$$O(\varepsilon^{-1}): \quad h^{(0)} \nabla h^{(0)} = 0. \quad (2.30)$$

For the non-trivial case where $h^{(0)} \neq 0$, the leading order momentum equation (2.30) implies that $h^{(0)}$ is solely time-dependent. The next-to-leading order expansion of the momentum equation

is

$$O(\varepsilon^0) : (h^{(0)}\mathbf{u}^{(0)})_t + \nabla \cdot (h^{(0)}\mathbf{u}^{(0)} \circ \mathbf{u}^{(0)}) = - \left[h^{(0)}\nabla h^{(1)} + h^{(1)}\nabla h^{(0)} + f\mathbf{k} \times h^{(0)}\mathbf{u}^{(0)} \right]. \quad (2.31)$$

As $h^{(0)}(t)$ depends only on time, the second term on the right is zero. Furthermore, integrating the leading order water depth equation (2.29) over the domain Ω and applying the divergence theorem,

$$h_t^{(0)} = -\frac{1}{|\Omega|} \int_{\Omega} h^{(0)}\mathbf{u}^{(0)} \cdot \mathbf{n} \, d\sigma = 0, \quad (2.32)$$

where $d\sigma$ is an infinitesimal segment of the domain boundary and \mathbf{n} is the corresponding outward-facing normal. The second equation to zero is due to the conservation of momentum in a doubly periodic domain.

Putting together the leading order water depth equation (2.29) and the next-to-leading order momentum equation (2.31) along with the boundary condition in (2.32) yields the horizontal rotating lake equations,

$$\nabla \cdot (h^{(0)}\mathbf{u}^{(0)}) = 0, \quad (2.33a)$$

$$(h^{(0)}\mathbf{u}^{(0)})_t + \nabla \cdot (h^{(0)}\mathbf{u}^{(0)} \circ \mathbf{u}^{(0)}) = - \left[h^{(0)}\nabla h^{(1)} + f\mathbf{k} \times h^{(0)}\mathbf{u}^{(0)} \right]. \quad (2.33b)$$

This is akin to setting

$$\rho \rightarrow h^{(0)}, \quad \mathbf{u} \rightarrow \mathbf{u}^{(0)}, \quad P \rightarrow h^{(0)}, \quad \pi \rightarrow h^{(1)}, \quad (2.34)$$

in the pseudo-incompressible equations (2.9), or $\alpha_h = 0$ in the governing shallow water equations (2.27). Similar derivations of the lake equations by means of formal asymptotics can be found in Bresch (2009) and Bresch et al. (2011). Details on the lake equations as a particular formulation of the pseudo-incompressible equation can be found in Lannes (2013).

This section and the preceding section extend the blended numerical model to support the switching between the shallow water and lake equations model pair. The aim here is to show that, as long as the appropriate switches and blending strategy are identified, the blending between model regimes may be employed to achieve balanced solutions. Therefore, the choice of this fairly simple model pair serves an illustrative purpose. Models that may be dynamically more interesting, such as the quasi- and semi-geostrophic equations, may be explored in future work.

2.6 Overview of the analytical models

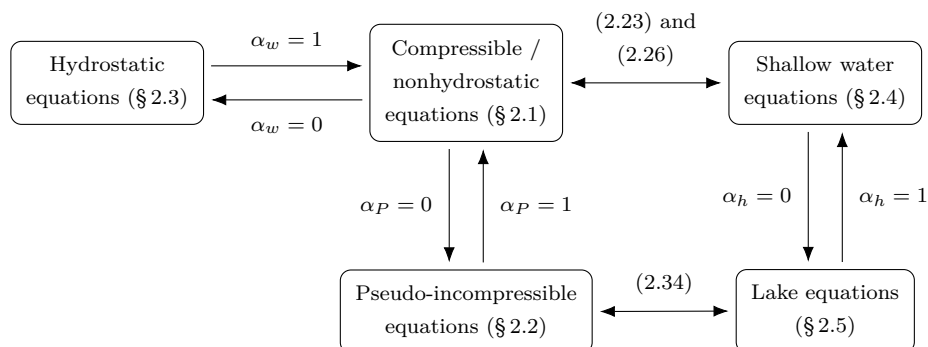


Figure 2.1: Overview of the analytical models and their relations within the blended numerical framework. Equation numbers are in brackets and sections are denoted with §.

Figure 2.1 gives an overview of how the models discussed in this chapter are related to one another within the blended numerical framework, summarising the content of this chapter. We note that this figure is incomplete. In order to enable accurate switching between model regimes within one simulation run, Figure 2.1 will be extended in chapter 4 to outline an *optimal* one-step blending strategy between the model regimes.

3 The blended numerical model

This chapter summarises the details of the blended numerical model from Benacchio and Klein (2019) along with elaborations on the discretisation and implementation details.

3.1 Summary of the numerical scheme

Rewriting the governing compressible equations (2.1) with χ as the inverse potential temperature in (2.3),

$$\rho_t + \nabla_{\parallel} \cdot (P\mathbf{u}\chi) + (Pw\chi)_z = 0, \quad (3.1a)$$

$$(\rho\mathbf{u})_t + \nabla_{\parallel} \cdot (P\mathbf{u} \circ \chi\mathbf{u}) + (Pw\chi\mathbf{u})_z = - [c_p P \nabla_{\parallel} \pi + f\mathbf{k} \times \rho\mathbf{u}], \quad (3.1b)$$

$$\alpha_w [(\rho w)_t + \nabla_{\parallel} \cdot (P\mathbf{u}\chi w) + (Pw\chi w)_z] = - (c_p P \pi_z + \rho g), \quad (3.1c)$$

$$\alpha_P P_t + \nabla_{\parallel} \cdot (P\mathbf{u}) + (Pw)_z = 0. \quad (3.1d)$$

We note that expanding the density subequation (3.1a) with the background $\bar{\chi}$ and perturbation χ' in (2.4b) yields the evolution equation of χ' , specifically

$$(P\chi')_t + \nabla_{\parallel} \cdot (P\mathbf{u}\chi') + (Pw\chi')_z = -Pw\bar{\chi}_z. \quad (3.2)$$

Introducing the notations by Smolarkiewicz et al. (2014) and Benacchio and Klein (2019),

$$\Psi = (\chi, \chi\mathbf{u}, \chi w, \chi'), \quad (3.3)$$

equations (3.1a)–(3.1c) may be written compactly alongside (3.1d) as

$$(P\Psi)_t + \mathcal{A}(\Psi; P\mathbf{v}) = Q(\Psi; P), \quad (3.4a)$$

$$\alpha_P P_t + \nabla \cdot (P\mathbf{v}) = 0, \quad (3.4b)$$

where $\mathbf{v} = (u, v, w)$ subsumes the three-dimensional velocity fields, $\mathcal{A}(\Psi; P\mathbf{v})$ denotes the advection of the quantity Ψ under the advective fluxes $P\mathbf{v}$, while $Q(\Psi; P)$ describes the effect on the right-hand side of (3.1) on Ψ given P .

The time update for (3.1d) is solved with an implicit midpoint method,

$$\alpha_P P^{n+1} = \alpha_P P^n - \Delta t \nabla \cdot (P\mathbf{v})^{n+1/2}. \quad (3.5)$$

We notice from (3.5) that for us to make a time integration of size Δt for the mass-weighted potential temperature P , we first have to determine the $(P\mathbf{v})$ -fields at the half-time level $n + 1/2$

3 The blended numerical model

before making the full time-step update. Therefore, to determine the advective fluxes at the half-time level, the time update for equations (3.4) is split into advective and non-advective terms. The advection terms on the left are updated by

$$(P\Psi)^\# = \mathcal{A}_{1\text{st}}^{\Delta t/2} [\Psi^n; (P\mathbf{v}^n)], \quad (3.6a)$$

$$\alpha_P P^\# = \alpha_P P^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^n, \quad (3.6b)$$

where $\tilde{\nabla}$ is the discrete nabla and $\mathcal{A}_{1\text{st}}$ is an advection scheme that is used in the half time predictor update. The terms on the right are then solved by an implicit Euler method,

$$(P\Psi)^{n+1/2} = (P\Psi)^\# + \frac{\Delta t}{2} Q(\Psi^{n+1/2}; P^{n+1/2}), \quad (3.7a)$$

$$\alpha_P P^{n+1/2} = \alpha_P P^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^{n+1/2}. \quad (3.7b)$$

The time updates (3.6) and (3.7) yield the advective fluxes at the half time level. As the implicit midpoint update for a given time step is equivalent to an implicit Euler method of half the time step followed by an explicit Euler update for the other half of the time step (Hairer et al., 2006), (3.7) completes the implicit half time step update required for the update of P in (3.5) and for the density $\rho = P/\chi$.

Having obtained the $(P\mathbf{v})$ -fields at the half-time level as required by the right-hand side of (3.5), we now update the quantities in Ψ by a full Δt time step. This is done by an explicit Euler half step, a full advection step, followed by an implicit Euler half step,

$$(P\Psi)^* = (P\Psi)^n + \frac{\Delta t}{2} Q(\Psi^n; P^n), \quad (3.8a)$$

$$(P\Psi)^{**} = \mathcal{A}_{2\text{nd}}^{\Delta t} \left[\Psi^*; (P\mathbf{v}^{n+1/2}) \right], \quad (3.8b)$$

$$(P\Psi)^{n+1} = (P\Psi)^{**} + \frac{\Delta t}{2} Q(\Psi^{n+1}; P^{n+1}), \quad (3.8c)$$

$$\alpha_P P^{n+1} = \alpha_P P^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v})^{n+1/2}. \quad (3.8d)$$

The second-order midpoint updates for ρ and P are completed with the explicit advection in (3.8b) and (3.8d). The other quantities are updated with a second-order trapezoidal rule for the right-hand side, comprising the explicit substep in (3.8a) and the implicit update in (3.8c), and the left-hand side is updated with the second-order advection substep in (3.8b). The interleaving of a second-order advection scheme by a first-order explicit and implicit Euler step in (3.8) yields an overall second-order accuracy for the full time update. More details are in section 2 of Smolarkiewicz (1991) and Smolarkiewicz and Margolin (1993).

We note that in the blended numerical scheme, the shallow water equations are analogous to the compressible flow equations given the identifications (2.23) and (2.26), and the lake equations are analogous to the pseudo-incompressible equations given the identifications (2.34).

3.2 Discretisation details

The time updates in (3.6), (3.8b), and (3.8d) involve an explicit advection routine that requires fluxes $(P\mathbf{v})^n$ or $(P\mathbf{v})^{n+1/2}$. (3.8a) is an explicit Euler update, and (3.7) and (3.8c) are implicit Euler updates. Furthermore, the quantities $(P\Psi)$ are stored as cell-centred values and the Exner pressure perturbation π' is stored on the nodes. The staggering of the pressure-related leading-order P and next-to-leading order π' on the cell and nodal grids ensures that the blended numerical solver works in the low-Mach limit. See, e.g., §8 in Schneider et al. (1999) for more details.

The two-dimensional discretisation of the time-update methods is elaborated upon in this section. From here on, we assume that the spatial domain is discretised by an equidistant Cartesian mesh.

3.2.1 Obtaining the numerical fluxes

The numerical fluxes on the cell interfaces are obtained by a cell-to-node averaging,

$$(Pu)_{i+1/2,j+1/2} = \frac{1}{4} [(Pu)_{i,j} + (Pu)_{i+1,j} + (Pu)_{i,j+1} + (Pu)_{i+1,j+1}], \quad (3.9a)$$

$$(Pu)_{i-1/2,j+1/2} = \frac{1}{4} [(Pu)_{i,j} + (Pu)_{i-1,j} + (Pu)_{i,j+1} + (Pu)_{i-1,j+1}], \quad (3.9b)$$

followed by a node-to-face averaging,

$$(Pu)_{i,j+1/2} = \frac{1}{2} [(Pu)_{i-1/2,j+1/2} + (Pu)_{i+1/2,j+1/2}]. \quad (3.10)$$

Fluxes on the cell interfaces for the other velocity components (v, w) are obtained in a similar fashion. Figure 3.1 depicts the cells and nodes involved in the cell-to-face averaging for the flux at $(i, j + 1/2)$ (left) and for the flux at $(i + 1/2, j)$ (right).

3.2.2 The advection routine

Recall that the quantities advected under the fluxes $(P\mathbf{v})$ are written with the compact form Ψ in (3.3). For the explicit full time step advection in equation (3.8b), assume that $(P\mathbf{v})^{n+1/2}$ has been obtained from the half time predictor step, i.e. after the implicit Euler update in (3.7), and that Ψ^n is available.

Let us consider the cells with indices $i - 1$, i , and $i + 1$. An application of the advection operator $\mathcal{A}^{\Delta t}$ is akin to

$$\mathcal{A}_x^{\Delta t} \mathcal{U}_i = \mathcal{U}_i - \frac{\Delta t}{\Delta x} \left[(Pu)_{i+1/2}^{n+1/2} \Psi_{i+1/2} - (Pu)_{i-1/2}^{n+1/2} \Psi_{i-1/2} \right], \quad (3.11)$$

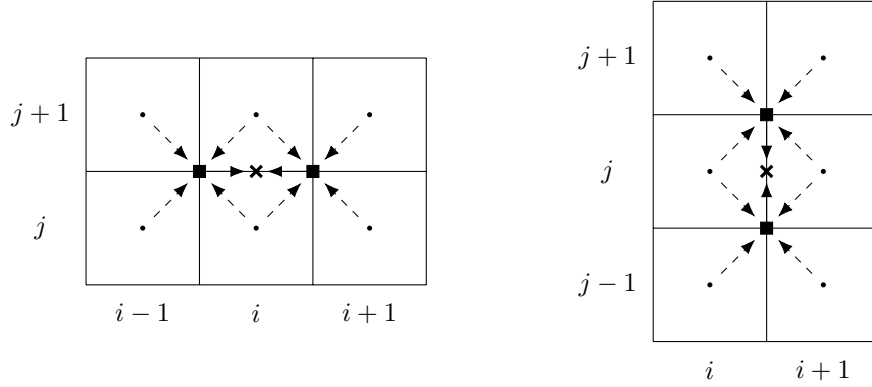


Figure 3.1: Cell-to-face averaging in the calculation of the numerical fluxes on the cell interface $(i, j + 1/2)$ (left) and $(i + 1/2, j)$ (right) represented by cross markers. Dot markers represent the cell-centres and square markers represent the nodes involved in the process. Dashed arrows represent the cell-to-node averaging, and the solid arrows represent the node-to-face averaging.

where

$$\mathcal{U}_i := \{\rho, \rho \mathbf{u}, \rho w, P, P\chi'\}_i \quad (3.12)$$

is the conserved quantity advected under the flux $(Pu)^{n+1/2}$, and $\Psi_{i-1/2}$ is the value of Ψ on the interface between the i -th and $(i-1)$ -th cells. In the limit as $\Delta t \rightarrow 0$, (3.11) can be written as a time update

$$\mathcal{A}_x^{\Delta t} \mathcal{U}_i = \mathcal{U}_i - \frac{1}{\Delta x} \left[\int_t^{t+\Delta t} \mathcal{F}(\mathcal{U}_i, \mathcal{U}_{i+1}) - \mathcal{F}(\mathcal{U}_{i-1}, \mathcal{U}_i) dt \right], \quad (3.13)$$

where $\mathcal{F}(\cdot, \cdot)$ is the flux across the cell interface of the respective quantity in \mathcal{U} . With the appropriate *reconstruction* for $\mathcal{F}(\cdot, \cdot)$, (3.13) is a Monotonic Upstream-centred Scheme for Conservation Laws (MUSCL) scheme that is second-order accurate in space. The method was developed by van Leer through a series of papers (Van Leer, 1973, 1974, 1977a,b, 1979).

Now let us consider a general advection equation,

$$U_t + F_x(U) = 0, \quad (3.14)$$

where U is the quantity advected, and F is a general flux function. Discretising the above with the quantity in (3.12) yields

$$\frac{\mathcal{U}^{n+1} - \mathcal{U}^n}{\Delta t} = -\frac{1}{\Delta x} [\mathcal{F}(\mathcal{U}_i, \mathcal{U}_{i+1}) - \mathcal{F}(\mathcal{U}_{i-1}, \mathcal{U}_i)], \quad (3.15)$$

where the right-hand side is a central difference scheme evaluated at the cell edges. Rewriting this,

$$\mathcal{U}^{n+1} = \mathcal{U}^n - \frac{\Delta t}{\Delta x} [\mathcal{F}(\mathcal{U}_i, \mathcal{U}_{i+1}) - \mathcal{F}(\mathcal{U}_{i-1}, \mathcal{U}_i)], \quad (3.16)$$

the first term in the square brackets on the right-hand side can be identified with $(Pu)_{i+1/2}^{n+1/2}\Psi_{i+1/2}$ and the second term with $(Pu)_{i-1/2}^{n+1/2}\Psi_{i-1/2}$ in (3.11). The quantities $(Pu)_{i\pm 1/2}^{n+1/2}$ are available at the onset of the advection routine as a cell-to-face averaging of the solutions from the half time predictor step. The missing pieces are then only the values $\Psi_{i\pm 1/2}^{n+1/2}$.

To obtain second-order accuracy in space of the MUSCL scheme in (3.13), we could use a slope limiter to approximate the cell averages instead of piecewise constant cell averages in a first-order Godunov's method. This also ensures that the scheme is *total variation diminishing* (Harten, 1997). More details can be found in the textbooks by LeVeque (1992) and LeVeque (2002).

Reconstruction

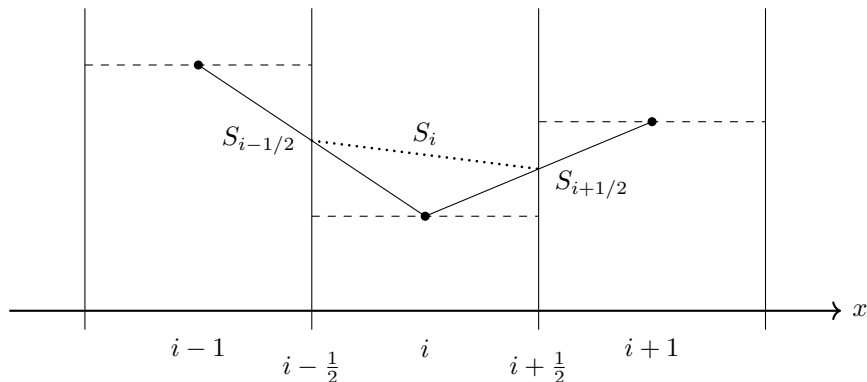


Figure 3.2: Construction of the slope in the i -th cell, S_i (dotted line). The vertical solid lines represent the cell interfaces. See text for the explanation of this figure.

We apply the following steps to obtain the missing piece $\Psi_{i+1/2}^{n+1/2}$. First, the slopes between adjacent cells are constructed, e.g. for the slope between the i -th and $(i + 1)$ -th cells,

$$S_{i+1/2} = \Psi_{i+1} - \Psi_i, \quad (3.17)$$

where Ψ_i is the cell-centred value of Ψ in the i -th cell. $S_{i-1/2}$ and $S_{i+1/2}$ are represented by the solid slopes in Figure 3.2. The piecewise constant cell value Ψ is represented by a horizontal dashed line and the cell-centred value by the black dot. The slope in the i -th cell, S_i (dotted line in Figure 3.2), is the average of the slopes constructed from the cell averages of the adjacent cells,

$$S_i = \frac{1}{2}(S_{i-1/2} + S_{i+1/2}). \quad (3.18)$$

We note that (3.18) is *not* a slope limiter in the i -th cell: in the limit of a discontinuity across the cell interfaces, $S_i \rightarrow \infty$. However, unlike the slope limiters, the averaging in (3.18) does not induce numerical dissipation, and this method of computing the slope is used in the numerical experiments of this thesis.

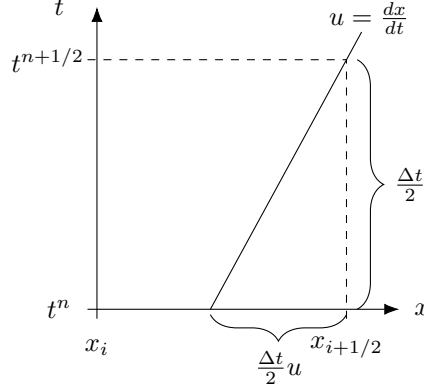


Figure 3.3: Backward tracing of the characteristic of Ψ from $(t^{n+1/2}, x_{i+1/2})$.

The characteristic of Ψ can be traced backwards in time as follows,

$$\Psi(t^{n+1/2}, x_{i+1/2}) = \Psi\left(t^n, x_{i+1/2} - \frac{\Delta t}{2} u_{i+1/2}^{n+1/2}\right), \quad (3.19)$$

and Figure 3.3 provides a graphical illustration of the idea behind the backward tracing. The quantity $u_{i+1/2}^{n+1/2}$ is obtained as follows,

$$u_{i+1/2}^{n+1/2} = \frac{(Pu)_{i+1/2}^{n+1/2}}{(P_i + P_{i+1})/2}. \quad (3.20)$$

Having obtained the slope in the cell, the value for Ψ anywhere in the cell can be obtained by linear interpolation,

$$\Psi(t^n, x) = \Psi_i^n + (x - x_i^n) S_i^n. \quad (3.21)$$

Using (3.21) together with the insight from the backward tracing of $\Psi_{i+1/2}^{n+1/2}$ in (3.19),

$$\begin{aligned} \Psi(t^{n+1/2}, x_{i+1/2}) &= \Psi_i^n + \left(x_{i+1/2}^n - \frac{\Delta t}{2} u_{i+1/2}^{n+1/2} - x_i^n\right) S_i^n \\ &= \Psi_i^n + \frac{\Delta x}{2} \left(1 - \frac{\Delta t}{\Delta x} u_{i+1/2}^{n+1/2}\right) S_i^n \\ &=: \Psi_{i+1/2}^-, \end{aligned} \quad (3.22)$$

where $x_{i+1/2} - x_i = \Delta x/2$ is used. $\Psi_{i+1/2}^-$ is the solution of Ψ on the left of the cell interface at $i + 1/2$ originating from the i -th cell.

The Riemann problem and the Harten-Lax-van Leer (HLL) solver

The derivation of $\Psi_{i+1/2}^-$ obtained in (3.22) supposes that the characteristic for $\Psi_{i+1/2}^{n+1/2}$ is traced from the i -th cell. Another possibility is the solution on the right of the $(i + 1/2)$ -th cell interface

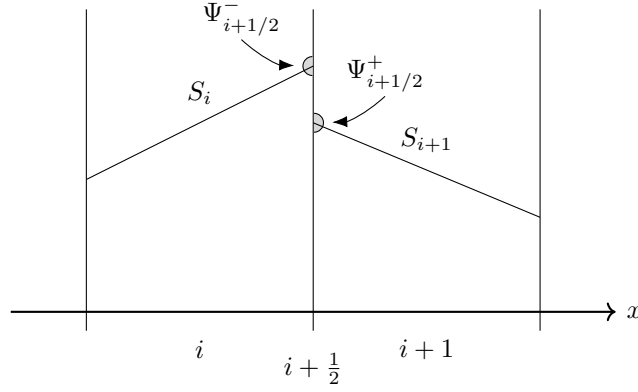


Figure 3.4: Discontinuity at the cell interface $(i + 1/2)$ arising from the recovery of $\Psi_{i+1/2}^{n+1/2}$ from the i -th and $(i + 1)$ -th cells. The slopes in the respective cell are represented by S .

originating from the $(i + 1)$ -th cell,

$$\begin{aligned} \Psi(t^{n+1/2}, x_{i+1/2}) &= \Psi_{i+1}^n - \frac{\Delta x}{2} \left(1 + \frac{\Delta t}{\Delta x} u_{i+1/2}^{n+1/2} \right) S_{i+1}^n \\ &=: \Psi_{i+1/2}^+. \end{aligned} \quad (3.23)$$

This leads to a discontinuity at the $(i + 1/2)$ cell interface, with $\Psi_{i+1/2}^-$ on the left of the interface and $\Psi_{i+1/2}^+$ on the right. Figure 3.4 illustrates this. The MUSCL-type time update in (3.16) along with the values of $\Psi^{n+1/2}$ recovered on the left and right of the interfaces in (3.22) and (3.23) constitute the Riemann problem (LeVeque, 1992). This is solved with the HLL solver by selecting the choice of (3.22) or (3.23) based on the direction of the advecting flux (Harten et al., 1983),

$$\Psi_{i+1/2} = \sigma \Psi_{i+1/2}^- + (1 - \sigma) \Psi_{i+1/2}^+, \quad (3.24)$$

where

$$\sigma = \text{sign} \left[(Pu)_{i+1/2}^{n+1/2} \right]. \quad (3.25)$$

With this selection of Ψ based on the upstream flux in (3.24), the missing pieces $\Psi_{i\pm 1/2}^{n+1/2}$ may be acquired, and the MUSCL scheme in (3.16) is updated, yielding the solution to our advection problem in (3.11).

Advection update

The advection problem in (3.11) represents an advection substep in the direction of the flux (Pu) . The full advection step consists of applying the advection substep for each coordinate direction. For example, a first-order Runge-Kutta update is used in the first advection routine

3 The blended numerical model

of the predictor half step, i.e.,

$$\mathcal{A}_{1\text{st}}^{\Delta t/2} \mathcal{U}_{i,j} = \mathcal{U}_{i,j} - \frac{\Delta t}{2} \left[\frac{(Pu)_{i+1/2}^n \Psi_{i+1/2} - (Pu)_{i-1/2}^n \Psi_{i-1/2}}{\Delta x} + \frac{(Pv)_{j+1/2}^n \Psi_{j+1/2} - (Pv)_{j-1/2}^n \Psi_{j-1/2}}{\Delta y} \right], \quad (3.26)$$

for a horizontal slice problem in the (x, y) -coordinate direction.

For the full time update, the advection operator $\mathcal{A}_{2\text{nd}}^{\Delta t}$ is split dimension-wise, and this leads to the second-order Strang splitting,

$$\mathcal{A}_{2\text{nd}}^{\Delta t} \mathcal{U}_{i,j} = \mathcal{A}_x^{\Delta t/2} \mathcal{A}_y^{\Delta t/2} \mathcal{A}_y^{\Delta t/2} \mathcal{A}_x^{\Delta t/2} \mathcal{U}_{i,j}. \quad (3.27)$$

Apart from Benacchio and Klein (2019), more details on the Strang splitting can be found in, e.g., MacNamara and Strang (2016).

3.2.3 Explicit Euler update

Equation (23) in Benacchio and Klein (2019), reproduced in (3.28) below, describes the time update of the right-hand side source terms in the implicit substep of the predictor step in (3.7), the explicit substep of the full time update in (3.8a), and the implicit substep of the full time update in (3.8c).

$$U_t = -c_p (P\theta)^\circ \pi'_x + fV, \quad (3.28a)$$

$$V_t = -c_p (P\theta)^\circ \pi'_y - fU, \quad (3.28b)$$

$$\alpha_w W_t = -c_p (P\theta)^\circ \pi'_z - g \frac{\tilde{\chi}}{\chi^\circ}, \quad (3.28c)$$

$$\tilde{\chi}_t = -W \frac{d\tilde{\chi}}{dz}, \quad (3.28d)$$

$$\alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'_t = -U_x - V_y - W_z, \quad (3.28e)$$

where $(U, W, W, \tilde{\chi}) = (Pu, Pv, Pw, P\chi')$, and the values $(P\theta)^\circ$, χ° , and $(\partial P/\partial \pi)^\circ$ are fixed from the solution of the preceding advection substep in the case of an implicit time update, and from the solution of the preceding time step in the case of the explicit time update. We note that the time update of ρ and P in (3.1a) and (3.1d) do not involve right-hand side source terms. The left-hand side of (3.28e) results from (2.5), i.e. P is dependent on π only. We also note that the second term on the right-hand side of the vertical momentum update (3.28c) is obtained from the right-hand side of (3.1c) as follows,

$$\begin{aligned}
-c_p(P\theta)^\circ \pi_z - P^\circ g &= -c_p(P\theta)^\circ \bar{\pi}_z - c_p(P\theta)^\circ \pi'_z - P^\circ g \\
&= (P\theta)^\circ g \bar{\chi} - c_p(P\theta)^\circ \pi'_z - P^\circ g \\
&= P^\circ g \left(\frac{\bar{\chi}}{\chi^\circ} - 1 \right) - c_p(P\theta)^\circ \pi'_z \\
&= -g \left(\frac{P^\circ \chi'}{\chi^\circ} \right) - c_p(P\theta)^\circ \pi'_z \\
&= -g \frac{\tilde{\chi}}{\chi^\circ} - c_p(P\theta)^\circ \pi'_z, \tag{3.29}
\end{aligned}$$

where we have used the hydrostatic background relation

$$\frac{d\bar{\pi}}{dz} = -\frac{g}{c_p} \bar{\chi}, \quad \bar{\pi}(0) = 1, \tag{3.30}$$

the fact that $\theta^\circ = 1/\chi^\circ$, and from the expansion of χ in (2.4b), $(\bar{\chi}/\chi^\circ - 1) = (\chi'/\chi^\circ)$.

Now, the explicit Euler substep in (3.8a) may be written as follows,

$$U^{n+1/2} = U^n - \frac{\Delta t}{2} [c_p(P\theta)^\circ \pi'_x{}^n - fV^n], \tag{3.31a}$$

$$V^{n+1/2} = V^n - \frac{\Delta t}{2} [c_p(P\theta)^\circ \pi'_y{}^n + fU^n], \tag{3.31b}$$

$$\alpha_w W^{n+1/2} = W^n - \frac{\Delta t}{2} \left[c_p(P\theta)^\circ \pi'_z{}^n + g \frac{\tilde{\chi}^n}{\chi^\circ} \right], \tag{3.31c}$$

$$\tilde{\chi}^{n+1/2} = \tilde{\chi}^n - \frac{\Delta t}{2} \left[W^n \frac{d\tilde{\chi}}{dz} \right], \tag{3.31d}$$

$$\alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'^{n+1/2} = \alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'^n - \frac{\Delta t}{2} [U_x^n + V_y^n + W_z^n]. \tag{3.31e}$$

The terms on the right of (3.31) are available at the onset of the n -th time step, although a node-to-cell differencing is required to obtain $\pi'_x{}^n$, $\pi'_y{}^n$, and $\pi'_z{}^n$, and a cell-to-node differencing for U_x^n , V_y^n , and W_z^n . Finally, because χ' is updated redundantly alongside χ , the value of $\tilde{\chi}^n$ in (3.31d) is reevaluated at every time step as

$$\tilde{\chi}^n = \rho^n \left(\frac{\rho^n}{P^n} - \bar{\chi} \right) \tag{3.32}$$

to eliminate any discrepancy between χ' and χ . Details of the node-to-cell differencing and the cell-to-node differencing are presented below.

Node-to-cell differencing

In two dimensions and for a particular Cartesian direction, the finite differences over the nodes are taken and the resulting discrete differences on the face centres are averaged to obtain values on the cell centres. In particular, and referring to Figure 3.5a, a node-to-cell differencing along the Cartesian x -direction is as follows: the finite differences of the nodal π' values (square markers)

3 The blended numerical model

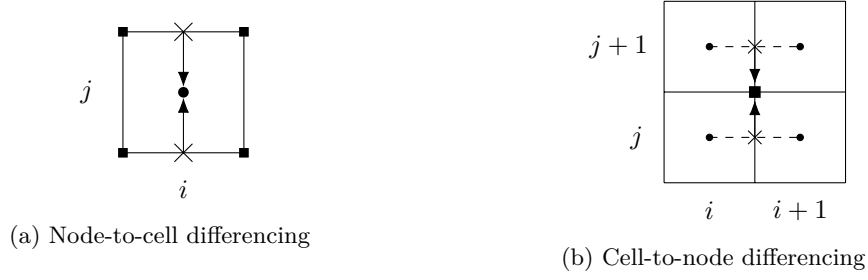


Figure 3.5: Computation of differentials on the cells and nodes. Square markers represent nodal values and dot markers cell-centred values.

are taken along the horizontal cell interfaces,

$$\left(\frac{\partial \pi'}{\partial x}\right)_{i,j-1/2} \approx \frac{1}{\Delta x} [\pi'_{i+1/2,j-1/2} - \pi'_{i-1/2,j-1/2}], \quad (3.33a)$$

$$\left(\frac{\partial \pi'}{\partial x}\right)_{i,j+1/2} \approx \frac{1}{\Delta x} [\pi'_{i+1/2,j+1/2} - \pi'_{i-1/2,j+1/2}]. \quad (3.33b)$$

Equation (3.33) yields the face-centred differences denoted by the cross markers. The averages of $(\partial \pi' / \partial x)_{i,j+1/2}$ and $(\partial \pi' / \partial x)_{i,j-1/2}$ are then taken along the arrows,

$$\left(\frac{\partial \pi'}{\partial x}\right)_{i,j} = \frac{1}{2} \left[\left(\frac{\partial \pi'}{\partial x}\right)_{i,j+1/2} + \left(\frac{\partial \pi'}{\partial x}\right)_{i,j-1/2} \right], \quad (3.34)$$

giving us the cell-centred $(\partial \pi' / \partial x)_{i,j}$ (the dot marker).

Cell-to-node differencing

An approach similar to the node-to-cell differencing is taken for the cell-to-node differencing. Differences of the cell-centred values are taken and the resulting discrete differences on the face centres are averaged to obtain nodal values. In particular, and referring to Figure 3.5b, finite differences of the cell-centred (Pu) values (dot markers) are taken along the dashed lines,

$$\left[\frac{\partial(Pu)}{\partial x}\right]_{i+1/2,j} = \frac{1}{\Delta x} [(Pu)_{i+1,j} - (Pu)_{i,j}], \quad (3.35a)$$

$$\left[\frac{\partial(Pu)}{\partial x}\right]_{i+1/2,j+1} = \frac{1}{\Delta x} [(Pu)_{i+1,j+1} - (Pu)_{i,j+1}]. \quad (3.35b)$$

Equation (3.35) gives us the face-centred differences (cross markers), and the averages of these values are taken along the vertical cell interfaces between the two cross markers (denoted by the arrows),

$$\left[\frac{\partial(Pu)}{\partial x}\right]_{i+1/2,j+1/2} = \frac{1}{2} \left\{ \left[\frac{\partial(Pu)}{\partial x}\right]_{i+1/2,j} + \left[\frac{\partial(Pu)}{\partial x}\right]_{i+1/2,j+1} \right\}. \quad (3.36)$$

This yields the nodal $(\partial(Pu)/\partial x)_{i+1/2,j+1/2}$ value (the square marker) in Figure 3.5b.

3.2.4 Implicit Euler update

Consider the implicit substep in the full time update in (3.8c). An implicit discretisation of equation (3.28) yields

$$U^{n+1} = U^{**} - \frac{\Delta t}{2} [c_p(P\theta)^\circ \pi_x'^{n+1} - fV^{n+1}], \quad (3.37a)$$

$$V^{n+1} = V^{**} - \frac{\Delta t}{2} [c_p(P\theta)^\circ \pi_y'^{n+1} + fU^{n+1}], \quad (3.37b)$$

$$\alpha_w W^{n+1} = W^{**} - \frac{\Delta t}{2} \left[c_p(P\theta)^\circ \pi_z'^{n+1} + g \frac{\tilde{\chi}^{n+1}}{\chi^\circ} \right], \quad (3.37c)$$

$$\tilde{\chi}^{n+1} = \tilde{\chi}^{**} - \frac{\Delta t}{2} \left[W^{n+1} \frac{d\tilde{\chi}}{dz} \right], \quad (3.37d)$$

$$\alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'^{n+1} = \alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'^{n+1/2} - \frac{\Delta t}{2} [U_x^{n+1} + V_y^{n+1} + W_z^{n+1}], \quad (3.37e)$$

where we recall that the superscript $**$ denotes the values obtained after the advection substep in (3.8b). We also note that while the explicit advection substep in (3.8b) completes the implicit midpoint update for P in (3.5), (3.37e) completes the synchronisation of the time levels of P and π .

Substituting the quantities at the new time level $n + 1$ with the appropriate equations and rearranging, (3.37a)–(3.37c) become

$$U^{n+1} = \frac{1}{1 + (\frac{\Delta t}{2} f)^2} \left[U^{**} + \frac{\Delta t}{2} fV^{**} - \frac{\Delta t}{2} c_p(P\theta)^\circ \left(\pi_x'^{n+1} + \frac{\Delta t}{2} f\pi_y'^{n+1} \right) \right], \quad (3.38a)$$

$$V^{n+1} = \frac{1}{1 + (\frac{\Delta t}{2} f)^2} \left[V^{**} - \frac{\Delta t}{2} fU^{**} - \frac{\Delta t}{2} c_p(P\theta)^\circ \left(\pi_y'^{n+1} - \frac{\Delta t}{2} f\pi_x'^{n+1} \right) \right], \quad (3.38b)$$

$$W^{n+1} = \frac{1}{\alpha_w + (\frac{\Delta t}{2} N)^2} \left[\alpha_w W^{**} - \frac{\Delta t}{2} g \frac{\tilde{\chi}^{**}}{\chi^\circ} - \frac{\Delta t}{2} c_p(P\theta)^\circ \pi_z'^{n+1} \right], \quad (3.38c)$$

where

$$N = \sqrt{-g \frac{1}{\chi^\circ} \frac{d\tilde{\chi}}{dz}} \quad (3.39)$$

is the local buoyancy frequency. Now, the time updates in (3.38) are explicit in (U, V, W) and implicit in π .

Inserting (3.38) into the right-hand side of (3.37e) and rearranging,

$$\begin{aligned} \alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'^{n+1} &- \left(\frac{\Delta t}{2} \right)^2 \frac{c_p(P\theta)^\circ}{1 + (\frac{\Delta t}{2} f)^2} \left[\left(\pi_{xx}'^{n+1} + \frac{\Delta t}{2} f\pi_{yx}'^{n+1} \right) + \left(\pi_{yy}'^{n+1} - \frac{\Delta t}{2} f\pi_{xy}'^{n+1} \right) \right] \\ &- \left(\frac{\Delta t}{2} \right)^2 \frac{c_p(P\theta)^\circ}{\alpha_w + (\frac{\Delta t}{2} N)^2} \pi_{zz}'^{n+1} \\ &= \alpha_P \left(\frac{\partial P}{\partial \pi} \right)^\circ \pi'^n - \frac{\Delta t}{2} \frac{1}{1 + (\frac{\Delta t}{2} f)^2} \left[\left(U_x^{**} + \frac{\Delta t}{2} fV_x^{**} \right) \right. \\ &\quad \left. + \left(V_y^{**} - \frac{\Delta t}{2} fU_y^{**} \right) \right] - \frac{\Delta t}{2} \frac{1}{\alpha_w + (\frac{\Delta t}{2} N)^2} \left[\alpha_w W_z^{**} - \frac{\Delta t}{2} g \frac{\tilde{\chi}^{**}}{\chi^\circ} \right]. \quad (3.40) \end{aligned}$$

3 The blended numerical model

The expression (3.40) yields a Helmholtz problem. The derivatives of $(U, V, W)^{**}$ on the right-hand side are prepared with the cell-to-node differencing illustrated in Figure 3.5b, while the left-hand side is updated with a 9-point stencil in two dimensions and a 27-point stencil in three dimensions.

27-point stencil

The implementation of the three-dimensional 27-point stencil is elaborated here. The 27 points involved in the stencil for the update of the nodal point $(\bar{i}, \bar{j}, \bar{k})$ are shown in Figure 3.6a. To keep the notations concise, the bar is introduced to index the nodal points.

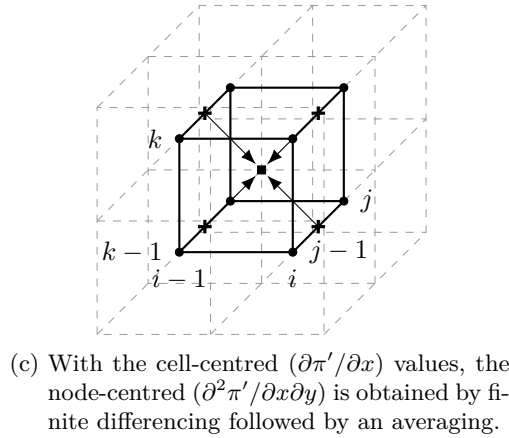
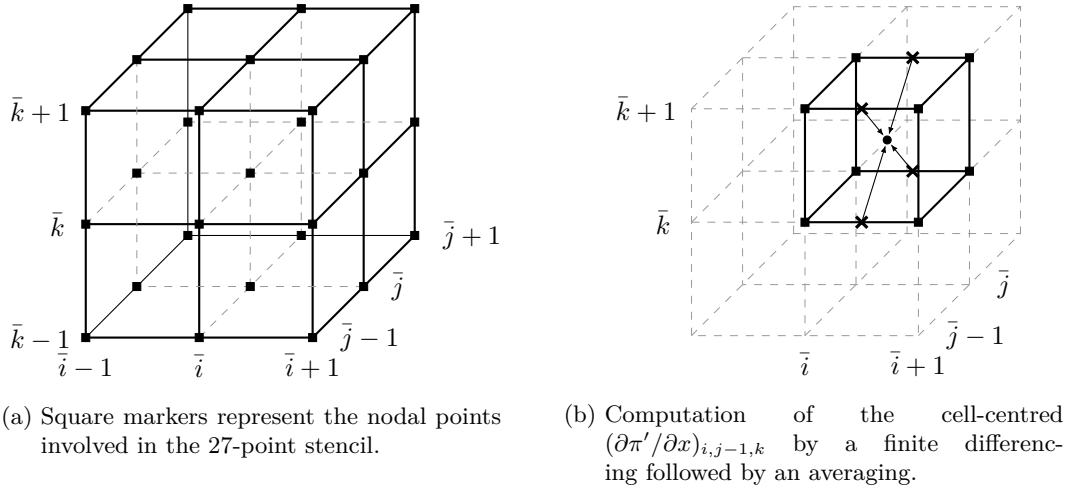


Figure 3.6: Method to compute the second-order derivative for π' with a 27-point stencil. This sets up a linear operator to solve (3.40).

Figures 3.6b and 3.6c illustrate the computation of π'_{xy} . Specifically, the cell-centred π'_x is obtained in Figure 3.6b by first applying a finite difference to the nodal values (square markers) along the cell interfaces in the x -direction (represented by the horizontal solid lines),

$$\left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j}-1,\bar{k}} \approx \frac{1}{\Delta x} \left[\pi'_{\bar{i}+1,\bar{j}-1,\bar{k}} - \pi'_{\bar{i},\bar{j}-1,\bar{k}} \right], \quad (3.41a)$$

$$\left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j},\bar{k}} \approx \frac{1}{\Delta x} \left[\pi'_{\bar{i}+1,\bar{j},\bar{k}} - \pi'_{\bar{i},\bar{j},\bar{k}} \right], \quad (3.41b)$$

$$\left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j}-1,\bar{k}+1} \approx \frac{1}{\Delta x} \left[\pi'_{\bar{i}+1,\bar{j}-1,\bar{k}+1} - \pi'_{\bar{i},\bar{j}-1,\bar{k}+1} \right], \quad (3.41c)$$

$$\left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j},\bar{k}+1} \approx \frac{1}{\Delta x} \left[\pi'_{\bar{i}+1,\bar{j},\bar{k}+1} - \pi'_{\bar{i},\bar{j},\bar{k}+1} \right]. \quad (3.41d)$$

The cell-centred $(\partial\pi'/\partial x)_{i,j-1,k}$ (dot marker) is then obtained by averaging the face-centred finite differences (cross markers),

$$\begin{aligned} \left(\frac{\partial\pi'}{\partial x}\right)_{i,j-1,k} &= \frac{1}{4} \left[\left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j}-1,\bar{k}} + \left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j},\bar{k}} \right. \\ &\quad \left. + \left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j}-1,\bar{k}+1} + \left(\frac{\partial\pi'}{\partial x}\right)_{\bar{i}+1/2,\bar{j},\bar{k}+1} \right]. \end{aligned} \quad (3.42)$$

This averaging is represented by the arrows from the cross markers to the dot marker. Repeating (3.41) and (3.42) to obtain the cell-centred $(\partial\pi'/\partial x)$ at $(i-1, j-1, k-1)$, $(i, j-1, k-1)$, $(i-1, j, k-1)$, $(i, j, k-1)$, $(i-1, j-1, k)$, $(i-1, j, k)$, and (i, j, k) brings us to Figure 3.6c. Note that the bar has been dropped for the indices of cell-centred values.

Referring to Figure 3.6c, finite differences are applied to the cell-centred $(\partial\pi'/\partial x)$ values (square markers) along the y -direction (represented by the solid diagonal lines in the y -direction). The face-centred $(\partial^2\pi'/\partial x\partial y)$ (cross markers) are averaged to obtain the nodal $(\partial^2\pi'/\partial x\partial y)_{\bar{i},\bar{j},\bar{k}}$ (square marker). This averaging is represented by the solid arrows. As the workings for these steps are similar to (3.41) and (3.42), the workings are not shown here.

The same principle applies to obtaining the second-order derivative for π' in the other Cartesian directions at $(\bar{i}, \bar{j}, \bar{k})$. By multiplying the stencil with the appropriate time-dependent coefficients present on the left-hand side of (3.40), we can set up a linear operator for the iterative method which solves for π'^{n+1} up to a given tolerance.

Correction terms

Having obtained π'^{n+1} , the time update for U^{n+1} , V^{n+1} , and W^{n+1} in (3.38) can be completed by applying a correction corresponding to the last term in the square brackets on the right-hand side of (3.38). Here, a node-to-cell differencing is applied (see Figure 3.5a for more details). This completes the implicit Euler update.

3.3 Implementation details

The cell-to-face averaging depicted in Figure 3.1 and the differencing between the cells and the nodes depicted in Figure 3.5 are akin to sliding window operations with overlaps. These operations are realised by convolution using the fast Fourier transform method with the appropriately chosen kernels. Specifically, the `fftconvolve` function from the `scipy` (Virtanen et al., 2020) `signal` toolbox is used in the Python code.

The advection routine in section 3.2.2 makes extensive use of `numpy` (Harris et al., 2020) vectorisation. The recovery of the fluxes at the cell interfaces and the solution from the HLL solver are computed by array-wide operations involving only the adjacent indices of the elements in the array that are along the direction of the advection. Combined with dimensional splitting, the advection routine implemented is dimensionally agnostic.

The Helmholtz problem in (3.40) is solved with a biconjugate gradient stabilised (BiCGStab) iterative method (Van der Vorst, 1992) that is available as part of the `scipy` linear algebra library. The linear operator is pre-initialised and compiled with the `numba` (Lam et al., 2015) Python runtime compiler. To update the time-dependent coefficients in run time, an anonymous function is used to interface the `scipy` BiCGStab solver with the precompiled operator. A 9-points 2D stencil and the 27-points 3D stencil from section 3.2.4 have been implemented.

Other components of the code include: a module that handles the boundary conditions in the ghost cells based on the user input; a class that outputs the simulation results into the HDF5 (The HDF Group, 1997) scientific data format; a module that generates an initial hydrostatic background state; and a class to instantiate the data structure for the solution arrays with methods to handle common operations on the solutions.

This chapter included a discussion on the blended numerical model introduced by Benacchio and Klein (2019). An overview of the numerical model was provided along with the discretisation and implementation details.

4 Blending strategies

Chapter 3 discussed the semi-implicit blended numerical model by Benacchio and Klein (2019). The model allows seamless transition between the full compressible and limit soundproof pseudo-incompressible models, and between the nonhydrostatic and limit hydrostatic models within a single numerical scheme.

In Benacchio et al. (2014), the blending between fully compressible and pseudo-incompressible dynamics is achieved by continuously adjusting the model parameter α_P in time. This chapter shows that in addition to parameter adjustment, careful accounting of the dynamical and the numerical differences in the models simulated by the blended numerical scheme provides substantial improvement to the blended solutions. Blending to a single time step in the limit regime is sufficient to recover a balanced state in the full model.

4.1 Compressible–soundproof blending

Apart from a plain parameter adjustment, accounting for the time level of the Exner pressure π and applying a conversion of the pressure-related quantities prior to the model transitions leads to an improvement of the compressible–soundproof blending. Parts of this section have appeared in Chew et al. (2021).

4.1.1 Time level of the pressure-related variables

Consider the simpler case without gravity and Coriolis forces, i.e. $g, f = 0$. Then (3.7a) and (3.8c) corresponding to the implicit time update of the right-hand side of the momentum equation at the half and full time stepping respectively can be written generally as

$$(P\mathbf{v})_t = -c_p(P\theta)^{\text{adv}}\nabla\pi, \quad (4.1)$$

where the superscript *adv* denotes the quantity that becomes available after the advection sub-steps (3.6a) and (3.8b). Equation (4.1) is multiplied by the potential temperature θ , in line with the discretisation details elaborated in section (4) of Benacchio and Klein (2019) and the general equations (3.28) for the time update of the right-hand side terms. Applying an implicit Euler discretisation to (4.1), we obtain

$$(P\mathbf{v})^{\text{out}} = (P\mathbf{v})^{\text{in}} - \delta t c_p(P\theta)^{\text{adv}}\tilde{\nabla}\pi^{\text{out}}, \quad (4.2)$$

where the superscript *in* denotes the quantities at the time level corresponding to the start of the time step and *out* at the end. Furthermore, $\delta t \leq \Delta t$ is an arbitrary time step size.

The compressible predictor step

For the case $\alpha_P = 1$, a discretisation of the left-hand side of (2.6) yields

$$P^{n+1} - P^n = \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{\text{adv}} (\pi^{n+1} - \pi^n), \quad (4.3)$$

recalling from (2.5) that P is a function of π only. At the half time level, (4.3) is

$$P^{n+1/2} - P^n = \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{\#} (\pi^{n+1/2} - \pi^n), \quad (4.4)$$

where $(\partial P/\partial\pi)^{\#}$ is obtained from P and π after the advection scheme at the half time level in (3.6). Substituting (4.4) into the implicit midpoint update (3.7b) gives

$$\left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{\#} (\pi^{n+1/2} - \pi^n) = -\frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^{n+1/2}. \quad (4.5)$$

Here, $(P\mathbf{v})^{\text{in}}$ in (4.2) is the solution of the advection terms in (3.6), i.e.,

$$(P\mathbf{v})^{\text{in}} = (P\mathbf{v})^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n, \quad (4.6)$$

with $\delta t = \Delta t/2$. Setting *out* to $n + 1/2$, inserting $(P\mathbf{v})^{\text{in}}$ from (4.6) into (4.2), and inserting the result into (4.5), we obtain

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{\#} \pi^{n+1/2} - \left(\frac{\Delta t}{2} \right)^2 \tilde{\nabla} \cdot [c_p(P\theta)^{\#} \tilde{\nabla} \pi^{n+1/2}] = \\ \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{\#} \pi^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^n + \left(\frac{\Delta t}{2} \right)^2 \tilde{\nabla} \cdot [\tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n], \end{aligned} \quad (4.7)$$

which fixes the time level of π after the half time predictor step in (3.6) and (3.7) at $n + 1/2$.

The compressible full time update

For the full time stepping of (3.8), the time update of P in (3.8d) may be written as

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{\#} (\pi^{n+1} - \pi^n) &= -\Delta t \tilde{\nabla} \cdot (P\mathbf{v})^{n+1/2} \\ &= -\frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^{n+1} - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^n, \end{aligned} \quad (4.8)$$

with

$$(P\mathbf{v})^{n+1} = (P\mathbf{v})^{\text{in}} - \frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\nabla} \pi^{n+1}], \quad (4.9)$$

and

$$(P\mathbf{v})^{\text{in}} = (P\mathbf{v})^n - \frac{\Delta t}{2} [c_p(P\theta)^n \tilde{\nabla} \pi^n] - \Delta t \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2}, \quad (4.10)$$

where the second term on the right arises from the explicit Euler substep in (3.8a) and the superscript $**$ denotes the solution to the advection substep in (3.8b). Inserting (4.10) into (4.9), and (4.9) into (4.8), we obtain

$$(P\mathbf{v})^{n+1} = (P\mathbf{v})^n - \frac{\Delta t}{2}[c_p(P\theta)**\tilde{\nabla}\pi^{n+1}] - \frac{\Delta t}{2}[c_p(P\theta)^n\tilde{\nabla}\pi^n] - \Delta t\tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2}. \quad (4.11)$$

Substituting (4.11) into (4.8) yields the following,

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^\# (\pi^{n+1} - \pi^n) = & -\Delta t\tilde{\nabla} \cdot (P\mathbf{v})^n + \frac{(\Delta t)^2}{2}\tilde{\nabla} \cdot \left[\frac{c_p(P\theta)^n\tilde{\nabla}\pi^n + c_p(P\theta)**\tilde{\nabla}\pi^{n+1}}{2} \right. \\ & \left. + \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} \right]. \end{aligned} \quad (4.12)$$

From (4.7), π is at time level $n + 1/2$ after the half time predictor step (3.7) while (4.12) starts with π at time level n for the full time stepping (3.8). Therefore, the time level of π has to be reset from $n + 1/2$ to n after the half time step (3.7) and before the full time step (3.8). Furthermore, the time level of π after the full time step (3.8) is $n + 1$, as expected.

The pseudo-incompressible predictor step

For $\alpha_P = 0$, the coupling between P and π in (2.6) no longer holds and the two variables decouple, leading to

$$-\nabla \cdot (P\mathbf{v}) = 0, \quad (4.13)$$

which takes the role of a divergence constraint. Identifying (4.13) with the left-hand side of (4.2),

$$\tilde{\nabla} \cdot (P\mathbf{v})^{\text{in}} = \tilde{\nabla} \cdot (\delta t c_p(P\theta)^{\text{adv}}\tilde{\nabla}\pi^{\text{out}}). \quad (4.14)$$

At the half time level, $(P\mathbf{v})^{\text{in}}$ is the solution of (3.6) comprising the half time advection. Therefore,

$$\tilde{\nabla} \cdot (P\mathbf{v})^{\text{in}} = \tilde{\nabla} \cdot \left[(P\mathbf{v})^n + \frac{\Delta t}{2}\tilde{\partial}_t(P\mathbf{v})^\# \right], \quad (4.15)$$

where $\tilde{\partial}_t$ is the discrete partial time derivative. As the second term is the solution to (3.6) starting at time level n , i.e. the left-hand side advection in the momentum equations (3.1b) and (3.1c) multiplied with θ , the following equation holds,

$$\tilde{\partial}_t(P\mathbf{v})^\# + \nabla \cdot (P\mathbf{v} \circ \mathbf{v})^n = 0, \quad (4.16)$$

and (4.15) is

$$\tilde{\nabla} \cdot (P\mathbf{v})^{\text{in}} = \tilde{\nabla} \cdot (P\mathbf{v})^n - \frac{\Delta t}{2}\tilde{\nabla} \cdot [\tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n]. \quad (4.17)$$

4 Blending strategies

Inserting (4.17) back into (4.14), with $\delta t = \Delta t/2$, the result is

$$\tilde{\nabla} \cdot \left(\frac{\Delta t}{2} c_p (P\theta)^n \tilde{\nabla} \pi^n \right) = \tilde{\nabla} \cdot (P\mathbf{v})^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot [\tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n], \quad (4.18)$$

where adv and π^{out} in the right-hand side of (4.14) have been fixed at time level n for (4.18) to be consistent.

On the unbalanced to balanced projection step

Equations (4.18) and (4.2) constitute a projection step that yields a divergence-free $(P\mathbf{v})^{n+1/2}$. However, the π^n solution in (4.18) is a compromise between controlling the divergence in $\tilde{\nabla} \cdot (P\mathbf{v})^n$ and making a time step of size $\Delta t/2$ to find a reasonable approximation of a pressure field obeying the nonlinear acoustic balance relation. Such a compromise would fail to achieve balanced solutions if the magnitude of the first term on the right-hand side of (4.18) is larger than that of the second term. An improvement that avoids this compromise at the expense of an additional projection step is provided below.

By setting $\Phi^n = \Delta t/2 \tilde{\nabla} \pi^n$ and taking the limit of $\Delta t \rightarrow 0$, the classical unbalanced to balanced projection is recovered,

$$\tilde{\nabla} \cdot [c_p (P\theta)^n \tilde{\nabla} \Phi^n] = \tilde{\nabla} \cdot (P\mathbf{v})^n, \quad (4.19a)$$

$$(P\mathbf{v})^{n*} = (P\mathbf{v})^n - c_p (P\theta)^n \tilde{\nabla} \Phi^n. \quad (4.19b)$$

The $(P\mathbf{v})^{n*}$ solution from (4.19) is then the divergence-free $(P\mathbf{v})$ -field closest to $(P\mathbf{v})^n$ in a suitably weighted L^2 -norm which may then be used as the input to the projection step comprising (4.18) and (4.2). When (4.18) is then invoked on the basis of $\tilde{\nabla} \cdot (P\mathbf{v})^{n*}$, it is ensured that $\tilde{\nabla} \cdot (P\mathbf{v})^{n*} = 0$, and the π^{n*} solution is a physically relevant pressure field that fully satisfies the nonlinear acoustic balance relation (which is also called the hidden constraint obtained by taking the time derivative of the primary constraint $\tilde{\nabla} \cdot (P\mathbf{v}) = 0$ and inserting into the momentum equation). Similar projection methods to achieve balanced solutions can be found in Hundertmark and Reich (2007) and Hundertmark and Reich (2011).

Experiments in this thesis are simulated with the projection step comprising (4.18) and (4.2) only, and studies with the double projection method involving (4.19) is deferred to future work.

The pseudo-incompressible full time update

For the full time stepping, $(P\mathbf{v})^{\text{in}}$ is the solution of (3.8b) and so (4.14) is

$$\tilde{\nabla} \cdot (P\mathbf{v})^{**} = \tilde{\nabla} \cdot \left(\frac{\Delta t}{2} c_p (P\theta)^{**} \tilde{\nabla} \pi^{\text{out}} \right), \quad (4.20)$$

with

$$\tilde{\nabla} \cdot (P\mathbf{v})^{**} = \tilde{\nabla} \cdot \left[(P\mathbf{v})^n - \frac{\Delta t}{2} c_p (P\theta)^n \tilde{\nabla} \pi^n + \Delta t \tilde{\partial}_t (P\mathbf{v})^{**} \right], \quad (4.21)$$

where the second term in the square brackets arises from the explicit Euler substep in (3.8a), and the third term is the solution of the advection substep at the full time level. Substituting (4.18) into (4.21), we obtain

$$\tilde{\nabla} \cdot (P\mathbf{v})^{**} = \tilde{\nabla} \cdot \left[\frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n + \Delta t \tilde{\partial}_t (P\mathbf{v})^{**} \right], \quad (4.22)$$

and noting again that advection substep (3.8b) solves the left hand side of the momentum equations (3.1b) and (3.1c) multiplied with θ , i.e, the following equation holds,

$$\tilde{\partial}_t (P\mathbf{v})^{**} + \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} = 0, \quad (4.23)$$

where the half time level on the right-hand side emerges from the solution of substep (3.8a) and (3.8b) under the advecting fluxes $(P\mathbf{v})^{n+1/2}$. Putting (4.22) and (4.23) together,

$$\begin{aligned} \tilde{\nabla} \cdot (P\mathbf{v})^{**} &= \frac{\Delta t}{2} \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n - \Delta t \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} \\ &= -\frac{\Delta t}{2} \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} - \frac{\Delta t}{2} \left[\tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} - \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n \right] \\ &= -\frac{\Delta t}{2} \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} - \left(\frac{\Delta t}{2} \right)^2 \left[\frac{\tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} - \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^n}{\Delta t/2} \right] \\ &= -\frac{\Delta t}{2} \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} - \left(\frac{\Delta t}{2} \right)^2 \tilde{\partial}_t \left(\tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} \right) \\ &= -\frac{\Delta t}{2} \left[1 + \frac{\Delta t}{2} \tilde{\partial}_t \right] \tilde{\nabla} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2}. \end{aligned} \quad (4.24)$$

Inserting (4.24) back into (4.20) fixes the time level of π^{out} at $n + 1$,

$$\tilde{\nabla} \cdot (c_p (P\theta)^{**} \tilde{\nabla} \pi^{n+1}) = - \left(1 + \frac{\Delta t}{2} \tilde{\partial}_t \right) \tilde{\nabla} \cdot \left[\tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{v})^{n+1/2} \right], \quad (4.25)$$

since the terms on the right is equivalent to making half a time step from the $n + 1/2$ time level.

Unlike in the compressible case, the results (4.18) and (4.25) imply that the solution of π after the half time predictor steps (3.6) and (3.7) is at the time level n , and could be used as the input to (3.8). Here, it is not necessary to reset π to the time level n for the pseudo-incompressible solve after the predictor step. After the full time update, the solution for π is at time level $n + 1$. Figure 4.1 summarises the time level analysis for π . As the solution of π depends nonlinearly on the other prognostic quantities, (4.18) and (4.25) constitute two separate elliptic equations that have to be solved for each time-step.

4.1.2 Conversion of the pressure-related variables

At the transition time interface between the compressible and pseudo-incompressible models and vice versa, judicious conversion of the pressure-related variables, accounting for the relevant scales, leads to improved blending results. The same applies for the shallow water and lake equations.

Equation (2.4a) separates the background Exner pressure from its perturbation. For the low Mach number regime $\text{Ma} \ll 1$, an alternative formulation is the asymptotic expansion

$$\pi = \pi^{(0)} + \text{Ma}^2 \pi^{(1)} + \dots \quad (4.26)$$

Substituting this expansion into the time update for π in (2.6), the equation becomes

$$\alpha_P \left(\frac{\partial P}{\partial \pi} \right) \text{Ma}^2 \pi_t^{(1)} = -\nabla \cdot (P \mathbf{v}). \quad (4.27)$$

In order to account for the $P \leftrightarrow \pi$ coupling in (4.27) for the compressible case of $\alpha_P = 1$, the P solution has to take the role of the full pressure-related quantity while the π solution is the next-to-leading order $\pi^{(1)}$ that has to be scaled by Ma^2 . The quantity π therefore takes the role of the Exner pressure perturbation, π' . On the other hand, for $\alpha_P = 0$, P and π decouple, leading to P taking on the role of the leading order pressure and π' remaining as the next-to-leading order term.

These insights motivate the following conversion strategy. To obtain the compressible P_{comp} , we use the relationship (2.5) between P and π and arrive at

$$P_{\text{comp}} = \left(P_{\text{psinc}}^{\gamma-1} + \text{Ma}^2 \pi'_{\text{psinc}} \right)^{\frac{1}{\gamma-1}}. \quad (4.28)$$

The subscript *comp* denotes quantities in the compressible regime and *psinc* in the pseudo-incompressible regime.

The pseudo-incompressible P_{psinc} is obtained from its compressible counterparts by inverting (4.28),

$$P_{\text{psinc}} = \left(P_{\text{comp}}^{\gamma-1} - \text{Ma}^2 \pi'_{\text{comp}} \right)^{\frac{1}{\gamma-1}}. \quad (4.29)$$

Therefore, at the blending interfaces, (4.28) or (4.29) is applied depending on the direction of the transition. Upon obtaining a converted P , and since $P = \rho\theta$, either ρ or θ and their respective conservative quantities have to be updated for consistency.

4.1.3 Choice of the perturbation variables

Time level analysis of π in subsection 4.1.1 demonstrated that, in a pseudo-incompressible solve, the solutions after the full time stepping and after the subsequent half time stepping are both

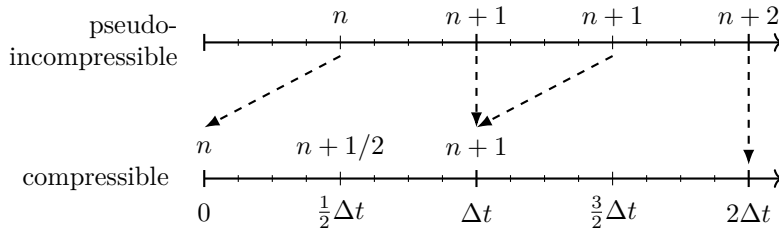


Figure 4.1: Summary of the time levels of π for the pseudo-incompressible (top) and the compressible (bottom) solutions in the numerical scheme. The dashed lines relate the π 's at the same time level between the two models.

at the same time level. This presents a choice of π' in the conversion (4.28) and similarly for the shallow water–lake blending, the choice of h' in (4.63). While this subsection deals with the compressible–pseudo-incompressible blending, the following discussion is analogous for the shallow water–lake case.

Consider the compressible to pseudo-incompressible blending at time $n+1$. The quantity P_{psinc}^{n+1} is obtained by inserting π'_{comp}^{n+1} into the right-hand side of (4.29). On the other hand, there are two valid choices for π' in a pseudo-incompressible to compressible transition:

- (1) π' after the full time stepping from n to $n+1$, abbreviated as π'_{full} , or
- (2) π' after the half time stepping from $n+1$ to $n+2$, abbreviated as π'_{half} .

Both π'_{full} and π'_{half} are at time level $n+1$. See Figure 4.1 for more details.

While both choices (1) and (2) are valid, choice (2) yields blended solutions that are slightly closer to the balanced solution compared to choice (1). π'_{half} is obtained from the solution of the implicit time update of the predictor step in (3.7) with the solution of (3.6) as its input. The input to (3.6) are Ψ^n and $(P\mathbf{v})^n$, and this means that π'_{half} is recovered from the other quantities. π'_{half} is independent of π at the previous time level, and so errors in the initialisation of π are not propagated. On the other hand, the quantity π'_{full} is obtained from the solution of (3.8). The explicit Euler update in (3.8a) has π' as an input to the right-hand side, $Q(\Psi^n; P^n)$. Therefore, π'_{full} propagates errors in the initialisation of π . While (2) results as a better choice than (1) in that it is more conforming to the underlying dynamics, it also entails solving, at the blending time interfaces, an additional half time step in the pseudo-incompressible regime in order to obtain π'_{half} . Apart from a comparison of simulation results with π'_{full} , the choice of π'_{half} will be used hereon.

4.2 Nonhydrostatic–hydrostatic blending

This section describes the development of the blending strategy between the compressible nonhydrostatic model and the hydrostatic model. A parameter adjustment, i.e. toggling α_w between 0 and 1, is used to switch between the nonhydrostatic regime and the hydrostatic regime. Aside

from applying a parameter adjustment, two innovations lead to a suppression of the vertically propagating acoustic mode in nonhydrostatic–hydrostatic blending results: 1) ensuring that the vertical momentum is balanced with respect to the other quantities at the onset of the full time update; while 2) employing a second-order update for the pressure perturbation variable. Through careful consideration of the hydrostatic balance relation, point 1) is achieved by applying two implicit time steps which yields a hydrostatically balanced solution regardless of the balancedness of the input. Point 2) arises from the observation that we have two options of π as an input to the full time update.

The following subsections develop this blending strategy. Specifically, subsection 4.2.1 derives the discrete implicit pressure update for the predictor step, and a similar derivation is given in subsection 4.2.2 for the full time update. Following the results from these two subsections, subsection 4.2.3 develops innovation point 1) above, and subsection 4.2.4 develops point 2). These innovations constitute a strategy for the hydrostatic-to-nonhydrostatic blending. To complete the nonhydrostatic-hydrostatic blending strategy, a brief discussion on the nonhydrostatic-to-hydrostatic blending direction is provided in subsection 4.2.5.

4.2.1 Half time predictor step

The difference between the hydrostatic and the nonhydrostatic solutions arises from the time update of the vertical momentum (Pw). The time update for the right-hand side of the vertical momentum equation in (3.37c) can be discretised as follows,

$$\alpha_w(Pw)^{n+1/2} = \alpha_w(Pw)^{\text{in}} - \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^\circ} \frac{\Delta t}{2} (P\chi)^{\text{out}}, \quad (4.30)$$

where we have

$$(Pw)^{\text{in}} = (Pw)^n + \frac{\Delta t}{2} \tilde{\partial}_t (Pw)^\#, \quad (4.31a)$$

$$(P\chi)^{\text{out}} = (P\chi)^{\text{in}} - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}}, \quad (4.31b)$$

$$(P\chi)^{\text{in}} = (P\chi)^n + \frac{\Delta t}{2} \tilde{\partial}_t (P\chi)^\#. \quad (4.31c)$$

(4.31a) and (4.31c) are the solutions of the advection terms on the left-hand side of the governing equations in (3.6), and (4.31b) is the implicit time update of the right-hand side of $(P\chi)$ corresponding to (3.37d). Substituting (4.31) into (4.30), we obtain

$$\begin{aligned} \alpha_w(Pw)^{n+1/2} &= \alpha_w(Pw)^{\text{in}} - \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^\circ} \frac{\Delta t}{2} \left[(P\chi)^{\text{in}} - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}} \right] \\ &= \alpha_w(Pw)^n + \alpha_w \frac{\Delta t}{2} \tilde{\partial}_t (Pw)^\# - \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^{\text{out}}] \\ &\quad - \frac{g}{\chi^\circ} \frac{\Delta t}{2} \left[(P\chi)^n + \frac{\Delta t}{2} \tilde{\partial}_t (P\chi)^\# - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}} \right] \end{aligned}$$

$$\begin{aligned}
 &= \alpha_w (Pw)^n + \alpha_w \frac{\Delta t}{2} \tilde{\partial}_t (Pw)^\# - \frac{\Delta t}{2} [c_p (P\theta)^\# \tilde{\partial}_z \pi^{\text{out}}] \\
 &\quad - \frac{g}{\chi^\circ} \frac{\Delta t}{2} (P\chi)^n - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_t (P\chi)^\# + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}}. \quad (4.32)
 \end{aligned}$$

Writing out the advection terms in (4.31a) and (4.31c) gives us,

$$\tilde{\partial}_t (Pw)^\# = -\tilde{\nabla}_\parallel \cdot (P\mathbf{u}w)^n - \tilde{\partial}_z (Pw^2)^n, \quad (4.33a)$$

$$\tilde{\partial}_t (P\chi)^\# = -\tilde{\nabla}_\parallel \cdot (P\mathbf{u}\chi)^n - \tilde{\partial}_z (Pw\chi)^n. \quad (4.33b)$$

Then rewriting (4.32) with (4.33), we have

$$\begin{aligned}
 \alpha_w (Pw)^{n+1/2} &= \alpha_w (Pw)^n - \alpha_w \frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u}w)^n - \alpha_w \frac{\Delta t}{2} \tilde{\partial}_z (Pw^2)^n - \frac{\Delta t}{2} [c_p (P\theta)^\# \tilde{\partial}_z \pi^{\text{out}}] \\
 &\quad - \frac{g}{\chi^\circ} \frac{\Delta t}{2} (P\chi)^n + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_\parallel \cdot (P\mathbf{u}\chi)^n + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z (Pw\chi)^n \\
 &\quad + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}}. \quad (4.34)
 \end{aligned}$$

Equation (4.34) comprises all the terms involved in the half time predictor step update of the vertical momentum (Pw).

The hydrostatic ($\alpha_w = 0$) case

For the hydrostatic case, setting $\alpha_w = 0$ and rearranging (4.34), we obtain

$$\frac{\Delta t}{2} [c_p (P\theta)^\# \tilde{\partial}_z \pi^{\text{out}}] = -\frac{g}{\chi^\circ} \frac{\Delta t}{2} \left[(P\chi)^n - \frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{v}\chi)^n - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}} \right], \quad (4.35)$$

which gives us the hydrostatic balance relation. The second term on the right requires $(Pw)^n$ as an input in the advection of χ while the third term on the right returns $(Pw)^{\text{out}}$ which satisfies the hydrostatic balance relation. The terms in the square brackets on the right correspond to the time update for $(P\chi)$. Note that, unlike in the compressible–soundproof blending, determining the actual time level of *out* is not crucial here. It suffices that $(Pw)^{\text{out}}$ satisfies the hydrostatic balance.

Next, we consider the time update of P at the half time predictor step, i.e.,

$$\begin{aligned}
 \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^\# (\pi^{n+1/2} - \pi^n) &= -\frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{v})^{n+1/2} \\
 &= -\frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u})^{n+1/2} - \frac{\Delta t}{2} \tilde{\partial}_z (Pw)^{n+1/2}. \quad (4.36)
 \end{aligned}$$

We know what $(P\mathbf{u})^{n+1/2}$ is from the time analysis of the compressible–soundproof blending,

$$-\frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u})^{n+1/2} = -\frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u})^n + \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_\parallel \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^n$$

$$+ \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_{\parallel} \cdot [c_p(P\theta)^{\#} \tilde{\nabla}_{\parallel} \pi^{n+1/2}], \quad (4.37)$$

and we obtain $(Pw)^{n+1/2}$ from the hydrostatic balance. Expanding and rearranging (4.35),

$$\begin{aligned} \frac{\Delta t}{2} [c_p(P\theta)^{\#} \tilde{\partial}_z \pi^{\text{out}}] &= -\frac{g}{\chi^{\circ}} \frac{\Delta t}{2} (P\chi)^n + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla} \cdot (P\mathbf{v}\chi)^n + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}}, \\ -\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}} &= -\frac{\Delta t}{2} [c_p(P\theta)^{\#} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^{\circ}} \frac{\Delta t}{2} (P\chi)^n + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla} \cdot (P\mathbf{v}\chi)^n, \\ (Pw)^{\text{out}} &= \frac{1}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \left\{ -\frac{\Delta t}{2} [c_p(P\theta)^{\#} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^{\circ}} \frac{\Delta t}{2} (P\chi)^n \right. \\ &\quad \left. + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla} \cdot (P\mathbf{v}\chi)^n \right\}. \end{aligned} \quad (4.38)$$

To substitute (4.38) into (4.36), take *out* to be $n + 1/2$, multiply (4.38) by $-\Delta t/2$, and take derivative with respect to z ,

$$\begin{aligned} -\frac{\Delta t}{2} \tilde{\partial}_z (Pw)^{n+1/2} &= \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^{\#} \tilde{\partial}_z \pi^{n+1/2}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{(P\chi)^n}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\ &\quad - \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^3 \tilde{\partial}_z \left\{ \frac{\tilde{\nabla} \cdot (P\mathbf{v}\chi)^n}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}. \end{aligned} \quad (4.39)$$

Substituting (4.37) and (4.39) into (4.36),

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^{\#} (\pi^{n+1/2} - \pi^n) &= \frac{\Delta t}{2} \tilde{\nabla}_{\parallel} \cdot \left\{ -(P\mathbf{u})^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^n \right\} \\ &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_{\parallel} \cdot [c_p(P\theta)^{\#} \tilde{\nabla}_{\parallel} \pi^{n+1/2}] \\ &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^{\#} \tilde{\partial}_z \pi^{n+1/2}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\ &\quad - \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}\chi)^n}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}, \end{aligned} \quad (4.40)$$

which is the half time predictor update for P in the case of $\alpha_w = 0$.

The nonhydrostatic ($\alpha_w = 1$) case

Setting $\alpha_w = 1$ in (4.34) and multiplying it by $-\Delta t/2$, taking the derivative w.r.t. z and taking *out* to be $n + 1/2$, and finally rearranging,

$$-\frac{\Delta t}{2} \tilde{\partial}_z (Pw)^{n+1/2} = \frac{\Delta t}{2} \tilde{\partial}_z \left\{ \frac{-(Pw)^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}w)^n}{1 - \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^{\#} \tilde{\partial}_z \pi^{n+1/2}}{1 - \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}$$

$$-\frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}\chi)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}. \quad (4.41)$$

Substituting (4.37) and (4.41) into (4.36),

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^\# (\pi^{n+1/2} - \pi^n) &= \frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot \left\{ -(P\mathbf{u})^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^n \right\} \\ &+ \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_\parallel \cdot \left[c_p(P\theta)^\# \tilde{\nabla}_\parallel \pi^{n+1/2} \right] \\ &+ \frac{\Delta t}{2} \tilde{\partial}_z \left\{ \frac{-(Pw)^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}w)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\ &+ \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^\# \tilde{\partial}_z \pi^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\ &- \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}\chi)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}, \end{aligned} \quad (4.42)$$

which is the time update for π with $\alpha_w = 1$. We note that in the absence of gravity, i.e. $g = 0$, (4.42) reduces to (4.7).

4.2.2 Full time update

Again, the difference between the hydrostatic solution and the nonhydrostatic solution comes from the update of the vertical momentum (Pw),

$$\alpha_w(Pw)^{n+1} = \alpha_w(Pw)^{\text{in}} - \frac{\Delta t}{2} [c_p(P\theta)^{\#*} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^\circ} \frac{\Delta t}{2} (P\chi)^{\text{out}}, \quad (4.43)$$

where

$$(Pw)^{\text{in}} = (Pw)^n - \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^n] + \Delta t \tilde{\partial}_t (Pw)^{\#*}, \quad (4.44a)$$

$$(P\chi)^{\text{out}} = (P\chi)^{\text{in}} - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}}, \quad (4.44b)$$

$$(P\chi)^{\text{in}} = (P\chi)^n - \frac{\Delta t}{2} (Pw)^n \frac{d\bar{\chi}}{dz} + \Delta t \tilde{\partial}_t (P\chi)^{\#*}, \quad (4.44c)$$

as in the half time predictor step. The additional terms that are in (4.44a) and (4.44c) but are not in 4.31a and (4.31c) respectively come from the explicit Euler substep in (3.8a). Substituting (4.44) into (4.43),

$$\begin{aligned} \alpha_w(Pw)^{n+1} &= \alpha_w(Pw)^{\text{in}} - \frac{\Delta t}{2} [c_p(P\theta)^{\#*} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^\circ} \frac{\Delta t}{2} \left[(P\chi)^{\text{in}} - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}} \right] \\ &= \alpha_w(Pw)^n + \alpha_w \Delta t \tilde{\partial}_t (Pw)^{\#*} - \alpha_w \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^n] - \frac{\Delta t}{2} [c_p(P\theta)^{\#*} \tilde{\partial}_z \pi^{\text{out}}] \end{aligned}$$

$$\begin{aligned}
 & -\frac{g}{\chi^\circ} \frac{\Delta t}{2} \left[(P\chi)^n + \Delta t \tilde{\partial}_t(P\chi)^{**} - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz}(Pw)^n - \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz}(Pw)^{\text{out}} \right] \\
 & = \alpha_w (Pw)^n + \alpha_w \Delta t \tilde{\partial}_t(Pw)^{**} - \alpha_w \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^n] - \frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\partial}_z \pi^{\text{out}}] \\
 & \quad - \frac{g}{\chi^\circ} \frac{\Delta t}{2} (P\chi)^n - \frac{g}{\chi^\circ} \frac{(\Delta t)^2}{2} \tilde{\partial}_t(P\chi)^{**} + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2} \right)^2 \frac{d\bar{\chi}}{dz}(Pw)^n \\
 & \quad + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2} \right)^2 \frac{d\bar{\chi}}{dz}(Pw)^{\text{out}}. \tag{4.45}
 \end{aligned}$$

Writing out the advection terms,

$$\tilde{\partial}_t(Pw)^{**} = -\tilde{\nabla}_\parallel \cdot (P\mathbf{u}w)^{n+1/2} - \tilde{\partial}_z(Pw^2)^{n+1/2}, \tag{4.46a}$$

$$\tilde{\partial}_t(P\chi)^{**} = -\tilde{\nabla}_\parallel \cdot (P\mathbf{u}\chi)^{n+1/2} - \tilde{\partial}_z(Pw\chi)^{n+1/2}. \tag{4.46b}$$

Inserting (4.46) into (4.45),

$$\begin{aligned}
 \alpha_w (Pw)^{n+1} & = \alpha_w (Pw)^n - \alpha_w \Delta t \tilde{\nabla}_\parallel \cdot (P\mathbf{u}w)^{n+1/2} - \alpha_w \Delta t \tilde{\partial}_z(Pw^2)^{n+1/2} \\
 & \quad - \alpha_w \frac{\Delta t}{2} [c_p(P\theta)^\# \tilde{\partial}_z \pi^n] - \frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^\circ} \frac{\Delta t}{2} (P\chi)^n \\
 & \quad + \frac{g}{\chi^\circ} \frac{(\Delta t)^2}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u}\chi)^{n+1/2} + \frac{g}{\chi^\circ} \frac{(\Delta t)^2}{2} \tilde{\partial}_z(Pw\chi)^{n+1/2} \\
 & \quad + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2} \right)^2 \frac{d\bar{\chi}}{dz}(Pw)^n + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2} \right)^2 \frac{d\bar{\chi}}{dz}(Pw)^{\text{out}}. \tag{4.47}
 \end{aligned}$$

Equation (4.47) is the general full time update for (Pw) .

The hydrostatic ($\alpha_w = 0$) case

For the hydrostatic case ($\alpha_w = 0$), rearranging (4.47) yields

$$\frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\partial}_z \pi^{\text{out}}] = -\frac{g}{\chi^\circ} \frac{\Delta t}{2} \left[(P\chi)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} - \Delta t \frac{\left(\frac{d\bar{\chi}}{dz}(Pw)^n + \frac{d\bar{\chi}}{dz}(Pw)^{\text{out}} \right)}{2} \right], \tag{4.48}$$

which gives us the hydrostatic balance relation for the full time update. Note that the $(Pw)^{\text{out}}$ update on the right is dependent on $(Pw)^n$.

Again, we consider the time update of P ,

$$\begin{aligned}
 \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi} \right)^{**} (\pi^{n+1} - \pi^n) & = -\frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^{n+1} - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v})^n \\
 & = -\frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u})^{n+1} - \frac{\Delta t}{2} \tilde{\nabla}_\parallel \cdot (P\mathbf{u})^n - \frac{\Delta t}{2} \tilde{\partial}_z(Pw)^{n+1} - \frac{\Delta t}{2} \tilde{\partial}_z(Pw)^n. \tag{4.49}
 \end{aligned}$$

And again, we know what $(P\mathbf{u})^{n+1}$ is from the time-level analysis of the pseudo-incompressible–

compressible blending (4.11),

$$\begin{aligned}
 -\frac{\Delta t}{2} \tilde{\nabla}_{\parallel} \cdot (P\mathbf{u})^{n+1} &= -\frac{\Delta t}{2} \tilde{\nabla}_{\parallel} \cdot (P\mathbf{u})^n + \frac{(\Delta t)^2}{2} \tilde{\nabla}_{\parallel} \cdot \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^{n+1/2} \\
 &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_{\parallel} \cdot [c_p(P\theta)^{\#} \tilde{\nabla}_{\parallel} \pi^n] + \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_{\parallel} \cdot [c_p(P\theta)^{**} \tilde{\nabla}_{\parallel} \pi^{n+1}],
 \end{aligned} \tag{4.50}$$

and we obtain $(Pw)^{n+1}$ from the hydrostatic balance. Expanding and rearranging (4.48),

$$\begin{aligned}
 \frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\partial}_z \pi^{\text{out}}] &= -\frac{g}{\chi^{\circ}} \frac{\Delta t}{2} (P\chi)^n + \frac{g}{\chi^{\circ}} \frac{(\Delta t)^2}{2} \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^n \\
 &\quad + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}}, \\
 -\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^{\text{out}} &= -\frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^{\circ}} \frac{\Delta t}{2} (P\chi)^n + \frac{g}{\chi^{\circ}} \frac{(\Delta t)^2}{2} \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} \\
 &\quad + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^n, \\
 (Pw)^{\text{out}} &= \frac{1}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \left\{ -\frac{\Delta t}{2} [c_p(P\theta)^{**} \tilde{\partial}_z \pi^{\text{out}}] - \frac{g}{\chi^{\circ}} \frac{\Delta t}{2} (P\chi)^n \right. \\
 &\quad \left. + \frac{g}{\chi^{\circ}} \frac{(\Delta t)^2}{2} \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^n \right\}.
 \end{aligned} \tag{4.51}$$

To substitute (4.51) into (4.49), take *out* to be $n + 1$, multiply (4.51) by $-\Delta t/2$ and take the derivative w.r.t. z ,

$$\begin{aligned}
 -\frac{\Delta t}{2} \tilde{\partial}_z (Pw)^{n+1} &= \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} + \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{(P\chi)^n}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\
 &\quad - \frac{g}{\chi^{\circ}} \Delta t \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{\tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} - \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^3 \tilde{\partial}_z \left\{ \frac{(Pw)^n \frac{d\bar{\chi}}{dz}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}.
 \end{aligned} \tag{4.52}$$

Substituting (4.50) and (4.52) into (4.49),

$$\begin{aligned}
 \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^{**} (\pi^{n+1} - \pi^n) &= \Delta t \tilde{\nabla}_{\parallel} \cdot \left\{ -(P\mathbf{u})^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^{n+1/2} \right\} \\
 &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_{\parallel} \cdot [c_p(P\theta)^{\#} \tilde{\nabla}_{\parallel} \pi^n] + \left(\frac{\Delta t}{2}\right)^2 \tilde{\nabla}_{\parallel} \cdot [c_p(P\theta)^{**} \tilde{\nabla}_{\parallel} \pi^{n+1}] \\
 &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\
 &\quad - \frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2}}{-\frac{g}{\chi^{\circ}} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}
 \end{aligned}$$

$$-\frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^3 \tilde{\partial}_z \left\{ \frac{(Pw)^n \frac{d\bar{x}}{dz}}{-\frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\} - \frac{\Delta t}{2} \tilde{\partial}_z (Pw)^n. \quad (4.53)$$

And since the last two terms on the right of (4.53) cancel out,

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^{**} (\pi^{n+1} - \pi^n) &= \Delta t \tilde{\nabla}_\parallel \cdot \left\{ -(P\mathbf{u})^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^{n+1/2} \right\} \\ &\quad + \frac{(\Delta t)^2}{2} \tilde{\nabla}_\parallel \cdot \left\{ \frac{c_p(P\theta)\# \tilde{\nabla}_\parallel \pi^n + c_p(P\theta)^{**} \tilde{\nabla}_\parallel \pi^{n+1}}{2} \right\} \\ &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{-\frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\} \\ &\quad - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2}}{-\frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\}, \end{aligned} \quad (4.54)$$

which is the time update of P for the hydrostatic $\alpha_w = 0$ case.

The nonhydrostatic ($\alpha_w = 1$) case

For the nonhydrostatic case ($\alpha_w = 1$), using equation (4.47) and identifying out with $n+1$ gives us the following,

$$\begin{aligned} (Pw)^{n+1} &= \left\{ \frac{(Pw)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}w)^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\} - \frac{\Delta t}{2} \left\{ \frac{c_p(P\theta)\# \tilde{\partial}_z \pi^n + c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\} \\ &\quad - \frac{g}{\chi^\circ} \frac{\Delta t}{2} \left\{ \frac{(P\chi)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} - \frac{\Delta t}{2} \frac{d\bar{x}}{dz} (Pw)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\}. \end{aligned} \quad (4.55)$$

Multiplying (4.55) with $-\Delta t/2$ and taking the derivative w.r.t. z yields

$$\begin{aligned} -\frac{\Delta t}{2} \tilde{\partial}_z (Pw)^{n+1} &= -\frac{\Delta t}{2} \tilde{\partial}_z \left\{ \frac{(Pw)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}w)^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\} \\ &\quad + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)\# \tilde{\partial}_z \pi^n + c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\} \\ &\quad + \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{(P\chi)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} - \frac{\Delta t}{2} \frac{d\bar{x}}{dz} (Pw)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{x}}{dz}} \right\}. \end{aligned} \quad (4.56)$$

Substituting (4.50) and (4.56) into (4.49),

$$\begin{aligned} \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^{**} (\pi^{n+1} - \pi^n) &= \Delta t \tilde{\nabla}_\parallel \cdot \left\{ -(P\mathbf{u})^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^{n+1/2} \right\} \\ &\quad + \frac{(\Delta t)^2}{2} \tilde{\nabla}_\parallel \cdot \left\{ \frac{c_p(P\theta)\# \tilde{\nabla}_\parallel \pi^n + c_p(P\theta)^{**} \tilde{\nabla}_\parallel \pi^{n+1}}{2} \right\} \end{aligned}$$

$$\begin{aligned}
 & -\frac{\Delta t}{2} \tilde{\partial}_z \left\{ \frac{(Pw)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}w)^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} + (Pw)^n \right\} \\
 & + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^\# \tilde{\partial}_z \pi^n + c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\
 & - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2} + \frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}, \tag{*}
 \end{aligned} \tag{4.57}$$

where

$$\begin{aligned}
 (*) & = -\frac{\Delta t}{2} \tilde{\partial}_z \left\{ \frac{2(Pw)^n - \Delta t \tilde{\nabla} \cdot (P\mathbf{v}w)^{n+1/2} - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz} (Pw)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\
 & = -\Delta t \tilde{\partial}_z \left\{ \frac{(Pw)^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}w)^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-\frac{\Delta t}{2} \frac{d\bar{\chi}}{dz} (Pw)^n}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}. \tag{4.58}
 \end{aligned}$$

Since the last term of (4.58) cancels with the last term in the last curly bracket of (4.57),

$$\begin{aligned}
 \left(\frac{\tilde{\partial}P}{\tilde{\partial}\pi}\right)^{**} (\pi^{n+1} - \pi^n) & = \Delta t \tilde{\nabla}_\parallel \cdot \left\{ -(P\mathbf{u})^n + \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v} \circ \mathbf{u})^{n+1/2} \right\} \\
 & + \frac{(\Delta t)^2}{2} \tilde{\nabla}_\parallel \cdot \left\{ \frac{c_p(P\theta)^\# \tilde{\nabla}_\parallel \pi^n + c_p(P\theta)^{**} \tilde{\nabla}_\parallel \pi^{n+1}}{2} \right\} \\
 & - \Delta t \tilde{\partial}_z \left\{ \frac{(Pw)^n - \frac{\Delta t}{2} \tilde{\nabla} \cdot (P\mathbf{v}w)^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\
 & + \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{c_p(P\theta)^\# \tilde{\partial}_z \pi^n + c_p(P\theta)^{**} \tilde{\partial}_z \pi^{n+1}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\} \\
 & - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \tilde{\partial}_z \left\{ \frac{-(P\chi)^n + \Delta t \tilde{\nabla} \cdot (P\mathbf{v}\chi)^{n+1/2}}{1 - \frac{g}{\chi^\circ} \left(\frac{\Delta t}{2}\right)^2 \frac{d\bar{\chi}}{dz}} \right\}. \tag{4.59}
 \end{aligned}$$

We note that in the absence of gravity, i.e. $g = 0$, (4.59) reduces to (4.12), the compressible π -update obtained in the compressible–pseudo-incompressible blending strategy.

4.2.3 Enforcing the hydrostatic balance

A comparison of the implicit time update for the pressure variables in the hydrostatic discretisation, i.e. equations (4.40) and (4.54), to those of the nonhydrostatic discretisation, i.e. equations (4.42) and (4.59), show that, with regards to the hydrostatic balance, the updates of the pressure variables are in line with expectations. This observation allows us to take the search for a discrepancy between the discretisation scheme for the compressible equations and that for the hydrostatic equations elsewhere. Below, we will focus our attention on the hydrostatic balance

4 Blending strategies

relations in (4.35) and its nonhydrostatic counterpart in (4.48).

The hydrostatic balance relations in the predictor step (4.35) recover a quantity $(Pw)^{\text{out}}$ that satisfies the hydrostatic balance relation. Starting with an imbalanced $(Pw)^n$ as the input, the first-order predictor time stepping yields a $(Pw)^{n+1}$ that satisfies the relation in (4.35) given $(P\chi)^{n+1}$ and π^{n+1} . As the input $(Pw)^n$ is not balanced, the hydrostatic balance relation is not *yet* satisfied at time level $n + 1$. However, as $(Pw)^{n+1}$ is recovered from $(P\chi)^{n+1}$ and π^{n+1} , we can now use this as a balanced input to (4.35) again to obtain a hydrostatically balanced relation involving $(Pw)^{n+2}$, $(P\chi)^{n+2}$, and π^{n+2} .

In the full time update, the hydrostatic balance is enforced by (4.48). In this equation, a balanced input for $(Pw)^{\text{in}}$ is required in the advection of χ and in the second-order update to obtain $(Pw)^{\text{out}}$. The hydrostatically balanced quantities at time level $n + 2$ can then be used as the input $(Pw)^{\text{in}}$ to the full time update in (4.48). This ensures that the second order solution remains hydrostatically balanced.

Summarising these insights, a hydrostatically balanced solution can be obtained from imbalanced initial conditions by two first-order time updates corresponding to (4.35) followed by a single second-order time update corresponding to (4.48). To achieve a one-step blending strategy, one could do a first-order time update from n to $n + 1/4$, another first-order time update from $n + 1/4$ to $n + 1/2$, and finally a second-order time update from $n + 1/2$ to $n + 1$. This is summarised in Figure 4.2.

4.2.4 Second-order pressure update

In the predictor hydrostatic update of π in (4.40), the time update of π does not involve π^n , and π^{out} is recovered from the update of the other quantities, namely $(P\mathbf{v})^n$ and $(P\chi)^n$. In the full time hydrostatic update of π in (4.54), the time update of π is second-order and involves π^n . This difference in how the quantity π is updated suggests that we have a choice in π as an input to the full time update. Empirically, the π -input to the full time update at $n + 1/2$ is best chosen as the solution of a second-order time update from n to $n + 1/2$ and this is depicted as the dotted arrow in Figure 4.2. An analytical argument for this observation is left for future work. Note that this additional second-order π update does not increase the computational effort substantially, as the predictor step from time n to $n + 1/4$ (the leftmost solid arrow in Figure 4.2) already prepares the input required by this full time update.

4.2.5 Nonhydrostatic-to-hydrostatic blending

The preceding subsections were concerned with deriving an optimal blending strategy in the hydrostatic-to-nonhydrostatic direction. At the blending time interface from the nonhydrostatic regime to the hydrostatic regime, α_w is set to zero. This freezes the time evolution of (Pw) , whereupon (Pw) takes on the role of enforcing the hydrostatic balance. Furthermore, as detailed

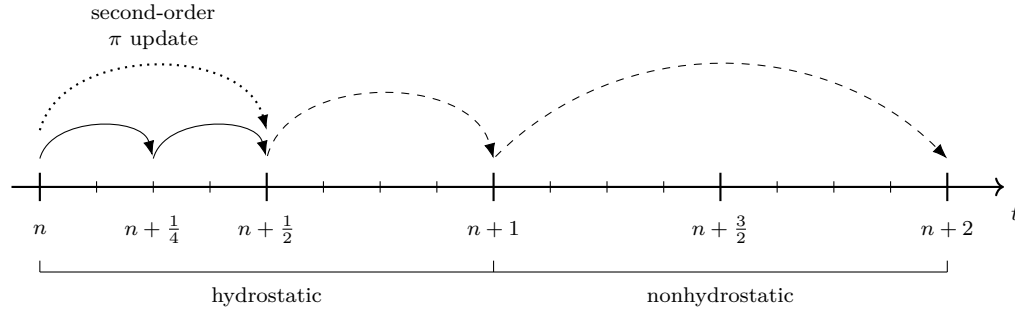


Figure 4.2: Summary of the nonhydrostatic–hydrostatic blending strategy. The solid arrows signify first-order time updates to obtain a balanced $(Pw)^{n+1/2}$, and the dashed arrows signify second-order time updates. The dotted arrow represents a second-order time update to obtain $\pi^{n+1/2}$. After the first-order predictor steps from time n to $n + 1/2$ and the second-order full time update from $n + 1/2$ to $n + 1$ in the hydrostatic regime, the rest of the time steps are solved in the nonhydrostatic regime with a second-order update.

in section 4.2.3, the nonhydrostatic vertical momentum field is essentially discarded in the hydrostatic regime. In its place, a hydrostatically balanced vertical momentum field is recovered through the other quantities by means of the two first-order time steps. Therefore, a nonhydrostatic solution can be blended to the hydrostatic regime without any additional conversion.

4.2.6 Future work

In the experiments involving balanced data initialisation of section 6.2.3, the nonhydrostatic–hydrostatic blending strategy developed in this section successfully suppresses vertically propagating acoustics. However, in experiments involving data assimilation, the ensemble data assimilation run with the blending strategy developed here is unable to obtain an error score lower than that of a plain ensemble data assimilation run for the density field despite successfully suppressing the vertically propagating acoustic mode. More details will be provided in section 6.4.3. Further work may therefore be required to develop a robust nonhydrostatic–hydrostatic blending strategy.

4.3 Shallow water–lake blending

The switch that toggles between the shallow water and lake equations, α_h , is analogous to α_P for the compressible–soundproof model pair. Apart from this parameter switching, a conversion of the full water depth in the shallow water to the leading order water depth in the lake model further improves quality of the blended solution. This section provides the details of this conversion.

Inserting the asymptotic expansion for the water depth (2.28a) into (2.6) with $P = \pi = h$, we obtain

$$\alpha_h 2 \text{Fr}^2 h_t^{(1)} = -\nabla \cdot (h\mathbf{u}). \quad (4.60)$$

4 Blending strategies

For the shallow water case of $\alpha_h = 1$, P takes the role of the full water depth h while π takes the role of the next-to-leading order $h^{(1)}$ scaled by 2Fr^2 , i.e. the water depth perturbation. For the lake equations arising from $\alpha_h = 0$, the right-hand side of (4.60) is (2.33a),

$$\nabla \cdot (h^{(0)} \mathbf{u}^{(0)}) = 0, \quad (4.61)$$

and therefore P is identified with the leading order $h^{(0)}$ and π remains as in the shallow water case, i.e., the next-to-leading order $h^{(1)}$. Therefore, the shallow water (SWE) to lake conversion is

$$h_{\text{lake}}^{(0)} = \bar{h}_{\text{SWE}}, \quad (4.62)$$

where \bar{h} is the mean water depth. Conversely,

$$h_{\text{SWE}} = h_{\text{lake}}^{(0)} + 2\text{Fr}^2 h_{\text{lake}}^{(1)}. \quad (4.63)$$

A conversion of the water depth h following (4.62) or (4.63) entails updating the other conservative quantities accordingly.

The choice of (4.62) is motivated by $\nabla h^{(0)} = 0$ from (2.30). The more natural choice of

$$h_{\text{lake}}^{(0)} = h_{\text{SWE}} - 2\text{Fr}^2 \delta \bar{h}_{\text{SWE}}^{(1)} \quad (4.64)$$

does not guarantee the constraint in general. Furthermore, while (2.33b) suggests that a conversion from the full velocity \mathbf{u} to the leading order $\mathbf{u}^{(0)}$ is necessary, this is not done in practice. Section 6.2 on the effectiveness of the blending scheme shows that the conversion of water depth h alone is sufficient to achieve blended results close to the balanced solution.

Finally, as in the compressible–soundproof blending case, we have the choices of $h'_{\text{half}}{}^n$ and $h'_{\text{full}}{}^n$ for a lake-to-shallow water blending at time level n , with $h'_{\text{half}}{}^n$ being the solution of the predictor half step from time level n to $n + 1/2$. See section 4.1.3 for more details.

4.4 Overview of the blending strategies

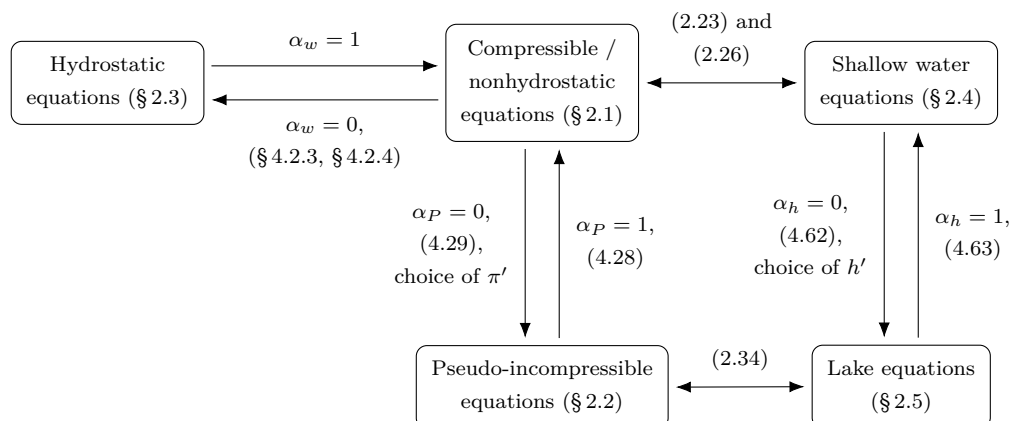


Figure 4.3: Overview of the blending strategies and how they relate the models within the blended numerical framework to one another. Equation numbers are in brackets and sections are denoted with §.

We now update Figure 2.1 with the blending strategies developed in this chapter. This is shown in Figure 4.3. The strategies summarised in Figure 4.3 ensure optimal blending for, e.g., the switching between different model regimes within a single simulation run. The blending strategies developed in this chapter require only one time step in the limit regime to obtain a balanced solution.

4.5 Implementation details

Switches were built into the blended numerical flow solver part of the Python code (more details in section 3.3), and based on user-defined parameters provided in an input file, the switches toggle the corresponding blending strategy. Furthermore, two blending scenarios are provided: blending for the first time step to enable balanced data initialisation, and blending after the assimilation of data to enable balanced data assimilation.

The blending strategies involve making additional time steps during a simulation without actually advancing the simulation in time, e.g. to recover π'_{half} in section 4.1.3. This is done by making recursive calls to the time stepping routine wherein the solution data containers are not updated and only the relevant quantities are returned.

This chapter included the development of the following blending strategies: between the compressible regime and the pseudo-incompressible regime, between the nonhydrostatic regime and the hydrostatic regime, and between the shallow water regime and the lake regime. Figure 4.3 summarised the blending strategies.

5 Bayesian data assimilation

This chapter provides a brief discussion on the ideas behind data assimilation followed by an elaboration on the local ensemble transform Kalman filter algorithm used in the numerical experiments of this thesis. The chapter then ends with a description of how the data assimilation method is interfaced with the blended numerical model. Some parts of this chapter can be found in Chew et al. (2021) in a similar form.

5.1 A short primer

Data assimilation is used in numerical weather prediction to improve forecasting by combining prior data from a numerical model with real-world observations. Suppose that the prior model data is represented by a *prior probability density function* (pdf) Π_{prior} . Then for a model state \mathbf{x} at a fixed time or over a time window, we are able to obtain a *posterior probability density function* Π_{post} that best combines the information encoded in $\Pi_{\text{prior}}(\mathbf{x})$ and the information encoded in the probability density function of the observations $\Pi(\mathbf{y}_{\text{obs}})$ by means of Bayes' theorem,

$$\Pi_{\text{post}}(\mathbf{x}) = \Pi(\mathbf{x}|\mathbf{y}_{\text{obs}}) = \frac{\Pi(\mathbf{y}_{\text{obs}}|\mathbf{x})}{\Pi(\mathbf{y}_{\text{obs}})} \Pi_{\text{prior}}(\mathbf{x}). \quad (5.1)$$

Here, $\Pi(\mathbf{x}|\mathbf{y}_{\text{obs}})$ represents the conditional probability of the state \mathbf{x} given the observations \mathbf{y}_{obs} and $\Pi(\mathbf{y}_{\text{obs}}|\mathbf{x})$ is the probability of the observation \mathbf{y}_{obs} given the state \mathbf{x} . More details on Bayesian data assimilation can be found in, e.g., Wikle and Berliner (2007) and Reich and Cotter (2013).

Before discussing some data assimilation methods, an elaboration on the notion of the *truth* and *observations* along with their associated errors is provided below. Consider $\mathbf{x}_{\text{truth}}^n \in \mathbb{R}^m$, a state vector at time $t = t^n = t_0 + n \Delta t$ describing the true state of the atmosphere projected onto the numerical degrees of freedom, with m being the dimension of the discrete model state. We denote the forward-in-time numerical integration of the state vector \mathbf{x} as $\mathcal{F}(\mathbf{x})$, so that

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathcal{F}(\mathbf{x}^n) \quad (5.2)$$

describes our numerical time stepping procedure, with $\mathbf{x}^n \in \mathbb{R}^m$ a numerical approximation of \mathbf{x} at time t^n . A time-increment of $\mathbf{x}_{\text{truth}}$ then reads as

$$\mathbf{x}_{\text{truth}}^{n+1} = \mathbf{x}_{\text{truth}}^n + \Delta t \mathcal{F}(\mathbf{x}_{\text{truth}}^n) + \eta^n + \xi^n =: \mathcal{M}[\mathbf{x}_{\text{truth}}^n], \quad (5.3)$$

where η^n is the error associated with the imperfect model description of the true state and ξ^n is the truncation error due to the numerical discretisation. In reality, an exact description of the

true atmospheric state is impossible. Therefore,

$$\mathbf{x}^n = \mathbf{x}_{\text{truth}}^n + \delta^n, \quad (5.4)$$

where δ^n is the total deviation from the true state $\mathbf{x}_{\text{truth}}^n$ comprising the accumulation of ξ^q and η^q for $(q = 1, \dots, n)$ as well as uncertainties in the initial data.

Observations $\mathbf{y}_{\text{obs}}^n \in \mathbb{R}^l$ are obtained by applying an *approximate observation operator* $\mathcal{H} : \mathbb{R}^m \rightarrow \mathbb{R}^l$ to $\mathbf{x}_{\text{truth}}^n$,

$$\mathbf{y}_{\text{obs}}^n = \mathcal{H}(\mathbf{x}_{\text{truth}}^n) + \epsilon^n, \quad (5.5)$$

where l is the dimension of the observation space, and ϵ^n encompasses the errors incurred through (i) the discrete approximation of the observation process and (ii) the mapping of the true state of the atmosphere onto the numerical degrees of freedom. Applying the observation operator to \mathbf{x}^n yields $\mathbf{y}^n \in \mathbb{R}^l$, the state vector in observation space, *i.e.*

$$\mathbf{y}^n = \mathcal{H}(\mathbf{x}^n). \quad (5.6)$$

With \mathbf{y}^n and $\mathbf{y}_{\text{obs}}^n$, the distance between the forecast from the numerical model and observations of the true atmospheric state can then be quantified.

With some notion of what *truth* and *observation* mean, we move on to the two broad classes of Bayesian data assimilation methods. These are the *variational methods* and the *sequential methods*, and a short discussion on these methods is provided in the following subsections.

5.1.1 Variational data assimilation methods

Variational data assimilation methods include the 3D- and 4D-Var, with the latter taking into account the three spatial and one temporal dimensions, hence the “4D”. Identifying the prior with the term *forecast* (f), and the posterior with the term *analysis* (a), the idea behind variational methods is to find the most probable analysis state \mathbf{x}^a given the observations \mathbf{y}_{obs} . This is akin to finding the mode of $\Pi_{\text{post}}(\mathbf{x})$,

$$\mathbf{x}^a = \arg \max_{\mathbf{x}} \Pi_{\text{post}}(\mathbf{x}) = \arg \max_{\mathbf{x}} \Pi(\mathbf{x}|\mathbf{y}_{\text{obs}}). \quad (5.7)$$

By assuming Gaussian shape for the probability density functions and a perfect model scenario, we can write $\Pi(\mathbf{x}|\mathbf{y}_{\text{obs}}) \approx \exp[-J(\mathbf{x})]$ with the cost function

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^f)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^f) + \frac{1}{2}(\mathcal{H}(\mathbf{x}) - \mathbf{y}_{\text{obs}})^T \mathbf{R}^{-1}(\mathcal{H}(\mathbf{x}) - \mathbf{y}_{\text{obs}}), \quad (5.8)$$

where \mathbf{x}^f is the forecast model state, \mathbf{B} the error covariance matrix associated with the forecast, and \mathbf{R} the error covariance matrix associated with the observations. The maximisation problem

(5.7) can be solved by minimising (5.8). Taking the gradient of (5.8), we obtain

$$\nabla_{\mathbf{x}} J(\mathbf{x}) = \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^f) + \mathbf{H} \mathbf{R}^{-1}(\mathcal{H}(\mathbf{x}) - \mathbf{y}_{\text{obs}}), \quad (5.9)$$

where \mathbf{H} is the Jacobian of \mathcal{H} , i.e. $\partial \mathcal{H}_i / \partial \mathbf{x}_j$. The local zeroes of (5.9) can be found with an iterative algorithm, e.g. the gradient descent method.

Equation (5.8) can be extended to include the time dimension. Assume that we now have a time series of observations $(\mathbf{y}_{\text{obs}}^0, \dots, \mathbf{y}_{\text{obs}}^n)$ since the last available forecast \mathbf{x}^f , then the cost function of the 4D-Var is as follows,

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^f)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^f) + \frac{1}{2} \sum_{q=0}^n (\mathcal{H}^q(\mathcal{M}^q[\mathbf{x}]) - \mathbf{y}_{\text{obs}}^q)^T (\mathbf{R}^q)^{-1} (\mathcal{H}^q(\mathcal{M}^q[\mathbf{x}]) - \mathbf{y}_{\text{obs}}^q), \quad (5.10)$$

where we note that \mathcal{H}^q and \mathbf{R}^q are now time-dependent. $\mathcal{M}^q[\mathbf{x}]$ represents q repeated applications of (5.3), i.e. it evolves the model state \mathbf{x} to a model state at an observation time q .

While the 4D-Var is widely used in numerical weather prediction (Bannister, 2001), it has a few drawbacks. Solving (5.10) with an iterative gradient descent method requires computing the adjoint of the Jacobian of \mathcal{H}^q and \mathcal{M}^q , and the implementation of these terms can be technically challenging. Moreover, the probability density functions are assumed to be Gaussian-shaped, although this assumption may not hold in real-world applications. In the preceding discussion, the model is assumed to be perfect. In the imperfect model scenario, an additional term is needed in (5.10) to account for the model error, adding to the difficulty of minimising the cost function. Finally, gradient descent methods may not be guaranteed to find global minima and may instead be stuck in local minima.

More details on variational data assimilation techniques can be found in Kepert (2009a), Blayot et al. (2011), and Reich and Cotter (2015).

5.1.2 The Kalman filters

Some sequential data assimilation methods are the *particle filters* and the *Kalman filters*. This subsection focuses on the latter while an introduction to particle filters can be found in Fearnhead and Künsch (2018) and Reich and Cotter (2015).

Assume Gaussian shape for the probability density functions for the forecast and observations such that the pdfs can be described by their respective mean and covariance matrices, and assume linearity of the approximate observation operator \mathcal{H} such that $\mathcal{H}(x) = \mathbf{H}x$. Then, the Kalman filter is

$$\begin{aligned} \mathbf{x}^a &= \mathbf{x}^f + \mathbf{B} \mathbf{H}^T (\mathbf{H} \mathbf{B} \mathbf{H}^T + \mathbf{R})^{-1} (\mathbf{y}_{\text{obs}} - \mathbf{H} \mathbf{x}^f) \\ &= \mathbf{x}^f + \mathbf{K} (\mathbf{y}_{\text{obs}} - \mathbf{H} \mathbf{x}^f), \end{aligned} \quad (5.11)$$

where $\mathbf{B} \in \mathbb{R}^{m \times m}$ and $\mathbf{R} \in \mathbb{R}^{l \times l}$ are the covariance matrices associated with the forecast and observations, respectively. \mathbf{K} is the Kalman gain that rewards the forecast if $\|\mathbf{B}\|_F \ll \|\mathbf{R}\|_F$ and penalises it if $\|\mathbf{R}\|_F \ll \|\mathbf{B}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. A drawback of the Kalman filter is the problem of high-dimensionality, e.g. in the representation of the forecast covariance matrix \mathbf{B} .

The *ensemble Kalman filter* overcomes the problem of high dimensionality by representing the pdf of the forecast as an ensemble of particles. The ensemble comprises the set of possible model states, and as the ensemble size K is typically much smaller than the dimension of the model state, $K \ll m$, ensemble-based methods are computationally more tractable (Katzfuss et al., 2016). Specifically, for an ensemble forecast $\{\mathbf{x}_1^f, \dots, \mathbf{x}_K^f\}$, the ensemble mean and variance are updated by

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^f + \mathbf{K}^{\text{ens}}(\mathbf{y}_{\text{obs}} - \mathbf{H}\bar{\mathbf{x}}^f), \quad (5.12a)$$

$$\mathbf{P}_K^a = \mathbf{P}_K^f - \mathbf{K}^{\text{ens}}\mathbf{H}\mathbf{P}_K^f, \quad (5.12b)$$

$$\mathbf{K}^{\text{ens}} = \mathbf{P}_K^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_K^f\mathbf{H}^T + \mathbf{R})^{-1}, \quad (5.12c)$$

where $\bar{\mathbf{x}}^{a/f}$ is the (analysis / forecast) ensemble mean and $\mathbf{P}_K^{a/f} \in \mathbb{R}^{K \times K}$ is the (analysis / forecast) covariance associated with the ensemble.

The covariance \mathbf{P}_K is spanned by the spread of the ensemble, and it is systematically underestimated for an ensemble Kalman filter scheme (Whitaker and Hamill, 2002). *Ensemble inflation* may be applied to increase the spread of the ensemble. A simple method to achieve this is to multiply the forecast ensemble covariance by a constant factor larger than 1, thereby increasing the covariance in the direction of the ensemble spread (Anderson, 2007; Van Leeuwen et al., 2015).

Spatial *localisation* may be applied alongside ensemble Kalman filter techniques to improve the quality of the analysis. Localisation prevents spurious correlations with faraway observations by limiting the influence of observations outside of a prescribed correlation radius, and it has the positive side-effect that the observation covariance becomes more diagonal, reducing the computational complexity of the data assimilation problem (Hamill et al., 2001; Houtekamer and Mitchell, 1998). Furthermore, the analysis within each of the local regions is determined by a different linear (local) combination of the ensemble members. Localisation, therefore, allows the distribution represented by the ensemble to describe a higher-dimensional space than a space constrained by the ensemble size alone (Fukumori, 2002; Mitchell et al., 2002).

Two methods of achieving spatial localisation are *domain localisation* and *observation localisation*. Domain localisation (DL) is achieved by decomposing the global domain into smaller overlapping local domains so that only observations inside the local domains are considered in the data assimilation procedure, see Figure 5.1a for more details. Observation localisation (OL) is achieved by applying a weighting function which reduces the weight of observations that are

further away from the analysis grid point. Figure 5.1b depicts an example of observation localisation with the truncated Gaussian weighting function. See also Kirchgessner et al. (2014) for a study on the choice for an optimal localisation radius for the ensemble Kalman filter. Both DL and OL are used in this thesis.

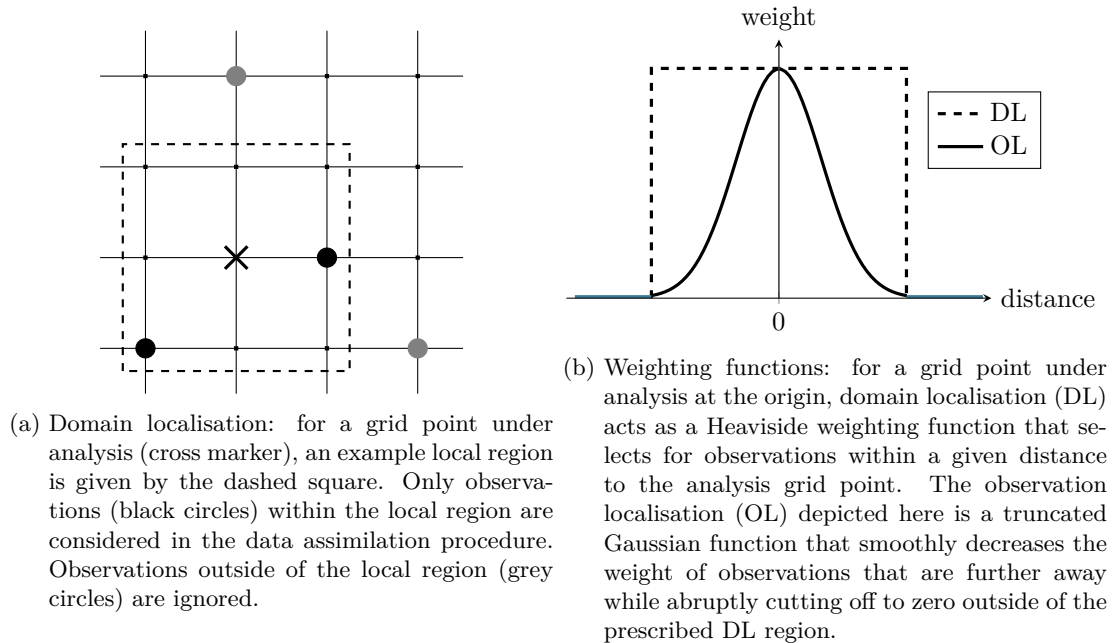


Figure 5.1: Illustration of the different localisation methods. (A similar figure appears in Kirchgessner et al. (2012).)

Assimilation of data may potentially introduce imbalances. The ensemble Kalman filter update in (5.12) assumes Gaussian-shaped probability density functions. On the other hand, processes modelled by the Euler equations in (2.1) are nonlinear. As the analysis ensemble is constructed from a linear combination of the forecast ensemble, nonlinear balance relations will not be fully preserved by the assimilation of data. Moreover, even if the dynamics are linear, a local data assimilation method may introduce imbalances. Within each of the local regions, the analysis is balanced. However, the global analysis solution is recovered by combining the local analysis solutions. This combination may result in non-physical discontinuities in the global analysis solution, and the linear balance relations are destroyed.

A very brief overview of some of the existing methods to suppress such imbalances was presented in chapter 1. The end of this chapter will introduce a novel dynamics-driven method that suppresses imbalances arising from sequential ensemble data assimilation. Imbalances are removed by a judicious projection of the imbalanced solution fields to reduced dynamics by means of the blending strategies developed in chapter 4.

5.2 The local ensemble transform Kalman filter (LETKF)

The ensemble Kalman filter update (5.12) in the previous subsection yields the analysis ensemble mean and covariance. However, the transformation to obtain the analysis ensemble from the forecast ensemble and the analysis ensemble mean is not unique (Reich and Cotter, 2015), and this gives rise to different flavours of the ensemble Kalman filter. A popular variation of the ensemble Kalman filter is the local ensemble transform Kalman filter (LETKF) introduced by Hunt et al. (2007), and this variation is used to conduct the data assimilation experiments in this thesis. The algorithm is adapted to work with the blended numerical framework from chapter 3, and the details are elaborated in Algorithm 1 below. More information on other variations of the ensemble Kalman filter can be found in the review by Houtekamer and Zhang (2016), and more theoretical background to the LETKF along with the original algorithm can be found in the publication by Hunt et al. (2007).

Algorithm 1 Local ensemble transform Kalman filter

Assume that we are given an ensemble of K state vectors, $\{\mathbf{x}_{k,[g]}^f\} \in \mathbb{R}^{m_{[g]}}$ for $k = 1, \dots, K$. Furthermore, we also assume that a set of observations $\mathbf{y}_{\text{obs},[g]} \in \mathbb{R}^{l_{[g]}}$ with a known covariance $\mathbf{R}_{[g]} \in \mathbb{R}^{l_{[g]} \times l_{[g]}}$ is available. The quantities m and l represent the dimension of the state and observation spaces, and the subscript $[g]$ represents the global state space before the application of a domain localisation method.

1. Apply the forward operator \mathcal{H} to obtain the state vectors in observation space,

$$\mathcal{H}(\mathbf{x}_{k,[g]}^f) = \mathbf{y}_{k,[g]}^f \in \mathbb{R}^{l_{[g]}}. \quad (5.13)$$

2. Stack the anomaly of the state and observation vectors to form the matrices,

$$\mathbf{X}_{[g]}^f = \begin{bmatrix} \mathbf{x}_{1,[g]}^f - \bar{\mathbf{x}}_{[g]} & \dots & \mathbf{x}_{K,[g]}^f - \bar{\mathbf{x}}_{[g]} \end{bmatrix} \in \mathbb{R}^{m_{[g]} \times K}, \quad (5.14)$$

$$\mathbf{Y}_{[g]}^f = \begin{bmatrix} \mathbf{y}_{1,[g]}^f - \bar{\mathbf{y}}_{[g]} & \dots & \mathbf{y}_{K,[g]}^f - \bar{\mathbf{y}}_{[g]} \end{bmatrix} \in \mathbb{R}^{l_{[g]} \times K}. \quad (5.15)$$

Here, $\bar{\mathbf{x}}_{[g]}$ ($\bar{\mathbf{y}}_{[g]}$) is the mean of the state vectors (in observation space) over the ensemble, e.g.,

$$\bar{\mathbf{x}}_{[g]} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_{k,[g]}^f \in \mathbb{R}^{m_{[g]}}. \quad (5.16)$$

3. From $\mathbf{X}_{[g]}^f$ and $\mathbf{Y}_{[g]}^f$, select the local \mathbf{X}^f and \mathbf{Y}^f . This selection depends on the choice of the domain localisation method, see §3 of Hunt et al. (2007) for more details.
 4. From the global observations $\mathbf{y}_{\text{obs},[g]}$ and observation covariance $\mathbf{R}_{[g]}$, select the corresponding local counterparts \mathbf{y}_{obs} and \mathbf{R} . The subscript $[g]$ is dropped when representing the local counterparts.
 5. Solve the linear system $\mathbf{RC}^T = \mathbf{Y}^f$ for $\mathbf{C} \in \mathbb{R}^{K \times l}$.
 6. Optionally, apply a localisation function to \mathbf{C} to modify the influence of the surrounding observations. This step corresponds to an application of an observation localisation method.
-

Algorithm 1 Local ensemble transform Kalman filter (continued)

7. Compute the $K \times K$ gain matrix,

$$\mathbf{K} = \left[(K-1) \frac{\mathbf{I}}{b} + \mathbf{C}\mathbf{Y}^f \right]^{-1}, \quad (5.17)$$

where $b > 1$ is an optional ensemble inflation factor.

8. Compute the $K \times K$ analysis weight matrix,

$$\mathbf{W}^a = [(K-1) \mathbf{K}]^{1/2}. \quad (5.18)$$

9. Compute the K -dimension vector encoding the distance of the observations from the forecast ensemble

$$\bar{\mathbf{w}}^a = \mathbf{K}\mathbf{C}(\mathbf{y}_{\text{obs}} - \bar{\mathbf{y}}^f), \quad (5.19)$$

and add $\bar{\mathbf{w}}^a$ to each column of \mathbf{W}^a to get a set of K weight vectors $\{\mathbf{w}_k^a\}$ with $k = 1, \dots, K$.

10. From the set of weight vectors, compute the analysis for each ensemble member,

$$\mathbf{x}_k^a = \mathbf{X}^f \mathbf{w}_k^a + \bar{\mathbf{x}}^f, \quad \text{for } k = 1, \dots, K. \quad (5.20)$$

11. Finally, we recover the global analysis ensemble $\{\mathbf{x}_{k,[g]}^a\}$, $k = 1, \dots, K$. This recovery depends on the domain localisation method applied in step 3.

5.3 Data assimilation and blending

This subsection goes into more details on how the LETKF data assimilation procedure is adapted to the blended numerical framework. Referring to Algorithm 1, we assume that, at time t^n , we have an ensemble forecast described by the state vector $\{\mathbf{x}_k^f\}$ for $k = 1, \dots, K$. Specifically, for the two-dimensional idealised experiments conducted with the Euler equations in this thesis, the ensemble state vector is

$$\{\mathbf{x}_1^f, \dots, \mathbf{x}_K^f\}^n = \{\rho, \rho u, \rho w, P, \pi'\}_{k=1, \dots, K}^n \in \mathbb{R}^{m \times K}, \quad (5.21)$$

where the choice of momenta ρu and ρw implies a vertical slice simulation. For an experimental setup with a two-dimensional grid of size $(N_x \times N_z)$, the state space of the problem becomes $m = (5 \times N_x \times N_z)$. Note that π' here is taken to be on the same underlying grid as the other quantities for simplicity. Recall from chapter 3 that for the blended numerical scheme, π' is solved on the dual node-based grid while the other quantities are solved on the primary cell-based grid.

We also assume that we have a set of observations $\mathbf{y}_{\text{obs}}^n$. Say we are only observing the momentum fields, then

$$\mathbf{y}_{\text{obs}}^n = \{(\rho u)_{\text{obs}}, (\rho w)_{\text{obs}}\}^n \in \mathbb{R}^l, \quad (5.22)$$

where the subscript *obs* indicates that the momenta observed are obtained from external sources.

These observations are sparse and noisy, and so $l = (2 \times N_{\text{obs}}(t^n))$, where $N_{\text{obs}}(t^n)$ is the time-dependent dimension of the sparse observation space. The observation covariance \mathbf{R}^n is determined by the observation noise which may or may not be time-dependent. In this thesis, the sparse and noisy observations are artificially generated and more details will be given in chapter 6.

Proceeding to step 1 of the algorithm, we apply the forward observation operator \mathbf{H} by selecting for each $\{\mathbf{x}_k^f\}$ in (5.21) the momentum fields $(\rho u)^n$ and $(\rho w)^n$ on the grid points for which there is an observation. This selection process maps $x_k^{f;n}$ from the state space \mathbb{R}^m to the observation space \mathbb{R}^l , and it furthermore assumes that, where there are observations, the observations spatially coincide with the grid points, and is therefore an idealised situation.

Steps 2–7 of the algorithm yield a Kalman gain similar to the Kalman gain update in (5.12c) while step 9 computes the distance of the ensemble forecast mean in observation space from the observations, similar to the right-hand side of (5.12a). Finally, step 10 along with the weight vectors obtained from steps 8–9 allow us to compute the analysis ensemble from the forecast ensemble.

Once this data assimilation procedure is completed, the full model regime is instantaneously switched to the limit regime according to the appropriate blending strategy developed in chapter 4 and summarised in Figure 4.3. The solution is then advanced for one time step in the limit regime before it is instantaneously switched back to the full model regime, again according to the appropriate blending strategy. Here, *instantaneous* means the switch occurs without a corresponding forward integration in time. The solution is computed with the full model up until the next assimilation time, upon which the process of data assimilation and blending is repeated. This method of switching between the full model and its limit regime uses the blended numerical model to remove imbalances arising from the data assimilation procedure and is hereon referred to as *blended data assimilation*.

In particular, if data is assimilated into the full compressible regime at time t^n , then a switching to the soundproof pseudo-incompressible regime involves setting the blending parameter α_P to 0 and converting the quantity P_{comp} to its pseudo-incompressible counterpart. The solution is then advanced for a single time step in the pseudo-incompressible regime and after that, α_P is set back to 1, and the quantity P_{psinc} is converted back to P_{comp} by using either π'_{half} or π'_{full} . The simulation then continues in the compressible regime for the duration of the assimilation window. This procedure for the compressible–pseudo-incompressible blending is summarised in Figure 5.2. We also recall from the time-level analysis in chapter 4 that unlike in the compressible solve, we do not reset π' after the half time predictor step in the soundproof pseudo-incompressible solve.

An overview of the components involved in achieving blended data assimilation and their respective locations in this thesis is given in Figure 5.3.

In the subsequent experiments involving blended data assimilation, the data assimilation parameters are not optimised or tuned in any way. While tuning the data assimilation parameters may lead to improved analysis scores and balance, the aim here is to establish the principal

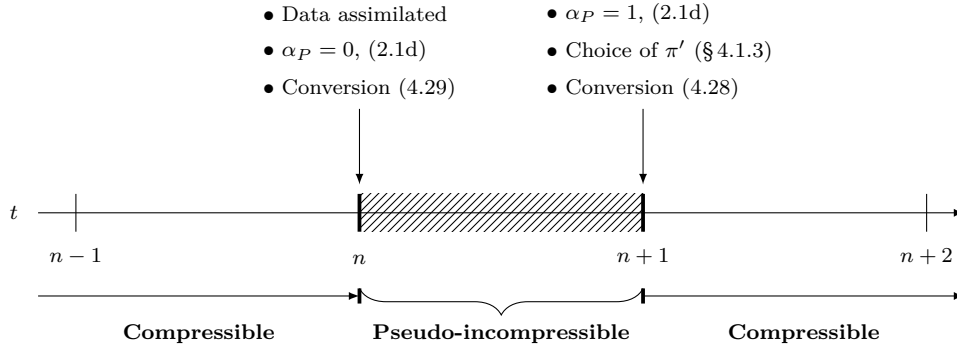


Figure 5.2: Blended data assimilation schematic at time t^n . *Blending time interfaces* where the model regime is switched instantaneously, i.e. at a fixed time, are represented with thick vertical lines. The time step spent in the pseudo-incompressible regime is shaded. See main text for more details.

capability of blended data assimilation to remove imbalances arising from sequential Bayesian data assimilation. As such, the effectiveness of blended data assimilation is evaluated with the additional challenge of an untuned data assimilation procedure. Demonstrating that the blending strategies work despite an untuned data assimilation system is advantageous in two regards: 1) it shows that measures of balancing the flow states can be separated from the data assimilation system, allowing for more control over the balancing mechanisms, and 2) the tuning of the data assimilation system can be used to achieve additional goals aside from suppressing imbalances, e.g. in the pursuit of better analysis scores that are not achievable by a plain data assimilation system alone.

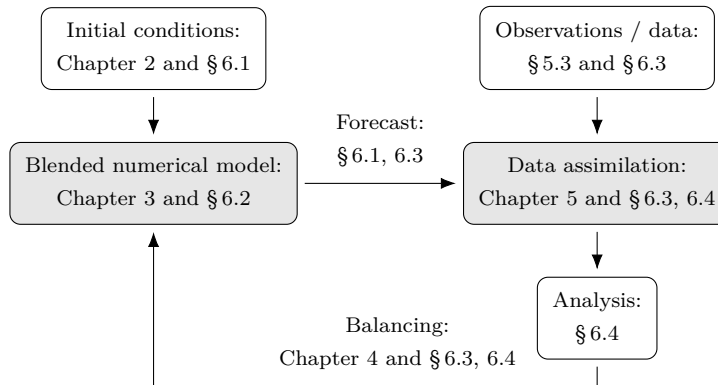


Figure 5.3: A summary overview linking the theoretical and algorithmic components required in achieving blended data assimilation and their corresponding chapters and sections (§) in this thesis.

5.4 Implementation details

A sketch of the LETKF implementation is provided in this section. For step 1 of the algorithm, masked arrays selecting for the grid points where an observation is available are applied to the ensemble of solution arrays. In steps 3 and 4, the local counterparts are generated by a sliding window view into the arrays with the `numpy.lib.stride_tricks.sliding_window_view` function. The boundary conditions are accounted for in this step. Although such a sliding window view operation may be slow, it is only done once for each data assimilation procedure. These local counterparts are then stored as chunks of `dask` arrays (Dask Development Team, 2016). Each chunk of local arrays is loaded sequentially into the memory, and the chunk size is chosen so as to avoid out-of-memory issues. For each local array in the chunk, steps 5–10 are computed in parallel using the `dask.delayed` function.

The linear system in step 5 is solved with the `spsolve` function available as part of the `scipy` sparse linear algebra submodule, and step 6 is computed with a straightforward matrix multiplication. An eigenvalue decomposition is used to compute the matrix inversion and square-root in steps 7 and 8, and specifically, the `eigh` function of the `scipy` linear algebra library is used. Vector and matrix multiplications are used to recover the weight vectors in step 9 and the analysis in step 10. Finally, the analysis ensemble in step 11 is obtained as follows: the analysis computed at each grid point of each ensemble member is directly reinserted into the ensemble of solution arrays.

This chapter included a brief discussion on data assimilation and the algorithm for the LETKF data assimilation method was reproduced here with some elaboration on the implementation details. Specifics on combining data assimilation with the blending strategies from chapter 4 was also provided.

6 Numerical results

In this chapter, results of numerical experiments with the one-step blending strategies for balanced initialisation and data assimilation are reported. In section 6.1, the description and experimental setup for the idealised test cases are provided; in section 6.2, the effectiveness of the balancing strategies developed in chapter 4 is established in the context of balancing initially imbalanced data; section 6.3 details the ensemble data assimilation setups; and in section 6.4, the blending strategies are applied to achieve balanced data assimilation.

The simulations were performed on a personal laptop with Intel® Core™ i7-8565U CPU @ 8 × 1.80GHz with 7.5 GB RAM in double machine precision. The results were visualised with a custom wrapper around the `matplotlib` APIs (Hunter, 2007), and `Jupyter notebooks` (Kluyver et al., 2016) are used to store and reproduce the workflows for visualisation and analysis. The Python code used to generate the results in this chapter, the HDF5 output of the results, and the notebooks are available upon request¹.

6.1 The idealised test cases

Idealised experiments are conducted to demonstrate the principal capability of the one-step blending strategies in achieving balanced solutions. The characteristics of the dynamics-driven balancing methods can be easier understood through experiments in such an idealised framework, and the use of relatively simple idealised tests also allows for the analysis of the sources of errors and imbalances.

6.1.1 The travelling vortices

A vortex test case with constant background velocity field is investigated for the shallow water equations (2.27) and the compressible flow equations (3.1). For brevity, the two setups will be referred to as the SWE vortex and Euler vortex respectively. Descriptions of the setups are given below.

SWE vortex

Starting with the gradient wind equation, we have

$$\frac{\partial p}{\partial r} = \frac{\rho u_\phi^2}{r} + f \rho u_\phi, \quad (6.1)$$

¹<mailto:ray.chew@fu-berlin.de>

where u_ϕ is the tangential velocity and r is the radial distance. Using identifications (2.23) and (2.26) to convert the gradient wind equation (6.1) to the shallow water context,

$$\begin{aligned}\frac{\partial}{\partial r} \left(\frac{gh^2}{2} \right) &= \frac{hu_\phi^2}{r}, \\ \frac{\partial h}{\partial r} &= \frac{1}{g} \frac{u_\phi^2}{r},\end{aligned}\tag{6.2}$$

for the case $f = 0$. Non-dimensionalising (6.2) with (2.14),

$$\begin{aligned}\frac{h_{\text{ref}}}{l_{\text{ref}}} \frac{\partial h^*}{\partial r^*} &= \frac{t_{\text{ref}}^2}{h_{\text{ref}}} \frac{l_{\text{ref}}^2}{t_{\text{ref}}^2} \frac{1_{\text{ref}}}{l_{\text{ref}}} \frac{1}{g^*} \frac{(u_\phi^*)^2}{r^*}, \\ \frac{\partial h^*}{\partial r^*} &= \frac{l_{\text{ref}}^2}{h_{\text{ref}}^2} \frac{1}{g^*} \frac{(u_\phi^*)^2}{r^*} \\ &= \text{Fr}^2 \frac{(u_\phi^*)^2}{r^*},\end{aligned}\tag{6.3}$$

where the definition of Fr^2 in (2.16) is used. Having obtained the dimensionless gradient wind equation for the shallow water vortex, the dimensionless expression for u_ϕ^* , following Kadioglu et al. (2008) for the Euler vortex, is used. Dropping the superscript $*$, this is given by

$$u_\phi(r) = 1024(1 - r^6)r^6,\tag{6.4}$$

and

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} / R,\tag{6.5}$$

with R the scaling factor for the radius of the vortex and (x_c, y_c) the initial centre position of the vortex. The initial velocities are given by

$$u(x, y, t) = \begin{cases} u_0 - u_\phi(r), & \text{if } r < 1, \\ u_0, & \text{otherwise,} \end{cases} \quad v(x, y, t) = \begin{cases} v_0 + u_\phi(r), & \text{if } r < 1, \\ v_0, & \text{otherwise,} \end{cases}\tag{6.6}$$

with $u_0 = 1.0$ and $v_0 = 1.0$ being the dimensionless horizontal background velocity. Integrating (6.3) with respect to r and inserting (6.4) into the result, the balanced distribution of the water height within the vortex structure is given by

$$\begin{aligned}H(r) = \text{Fr}^2 \left\{ \frac{r^{24}}{24} - \frac{12r^{23}}{23} + 3r^{22} - \frac{220r^{21}}{21} + \frac{99r^{20}}{4} - \frac{792r^{19}}{19} + \frac{154r^{18}}{3} - \frac{792r^{17}}{17} \right. \\ \left. + \frac{495r^{16}}{16} - \frac{44r^{15}}{3} + \frac{33r^{14}}{7} - \frac{12r^{13}}{13} + \frac{r^{12}}{12}, \right\}\end{aligned}\tag{6.7}$$

and for a dimensionless background water height $H_0 = 1.0$, the distribution of the water height over the entire domain is computed as follows,

$$h(x, y, t) = \begin{cases} H_0 + [H(r) - \max(H(r))], & \text{if } r < 1, \\ H_0, & \text{otherwise.} \end{cases} \quad (6.8)$$

Note that (6.8) leads to a *stable* vortex configuration wherein the centre of the vortex structure corresponds to a minimum in the water depth. The initial balanced state of the SWE vortex is depicted in Figure 6.1, and the lower right panel emphasises that the initial potential temperature θ is set to a constant of 1.0 everywhere such that $P \equiv h$ following the derivation of the shallow water equations in section 2.4. The SWE vortex is non-dimensionalised with

$$l_{\text{ref}} = 100.0 \text{ m}, \quad h_{\text{ref}} = 1.0 \text{ m}, \quad t_{\text{ref}} = 100.0 \text{ s}, \quad (6.9)$$

and this choice of reference units yields a Froude number $\text{Fr} \approx 0.319$. Within the compact vortex structure, however, the vortex attains a maximum velocity of 0.25 m s^{-1} , with $\text{Fr}_{\text{vort}} \approx 0.080$, and therefore the low Froude number analysis in section 4.3 is justified.

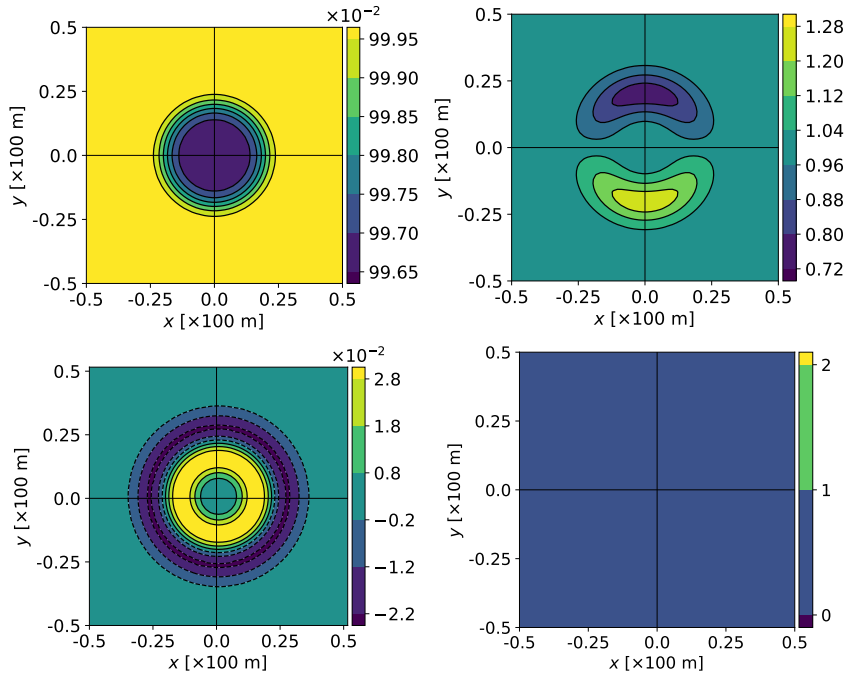


Figure 6.1: SWE vortex initial balanced states on a (64×64) grid: distribution of the water height h ; contours in the range $[99.65, 99.95] \times 10^{-2} \text{ m}$ with an interval of $0.05 \times 10^{-2} \text{ m}$ (top left), horizontal velocity u ; contours in the range $[0.72, 1.28] \text{ m s}^{-1}$ with a 0.08 m s^{-1} interval (top right), vorticity; contours in the range $[-2.2, 2.8] \times 10^{-2} \text{ s}^{-1}$ with a 10^{-2} s^{-1} interval (bottom left), and the equivalent of the potential temperature θ in the shallow water context that is set to a constant of 1.0 everywhere (bottom right). Negative contours dashed.

Experiments with the stationary steady-state shallow water vortex can be found in, for example, Michel-Dansac et al. (2016) and Bauer and Gay-Balmaz (2019), albeit with slightly different initial conditions.

Euler vortex

The setup for the cyclostrophic Euler vortex is similar to the SWE vortex. The initial distributions of the velocity fields follow equations (6.4) and (6.6). Furthermore, the initial stable vortex configuration is given as follows. For a vortex with radius R , the initial non-dimensionalised density distribution is

$$\rho(x, y, t) = \begin{cases} 1 - \frac{1}{2}(1 - r^2)^6 & \text{if } r < R, \\ 1 & \text{otherwise,} \end{cases} \quad (6.10)$$

where r is the radial distance given in (6.5). Integrating the gradient wind equation (6.1) with respect to r , the initial non-dimensionalised pressure distribution is as follows,

$$p(x, y, t) = \begin{cases} p(r) & \text{if } r < R, \\ 0 & \text{otherwise,} \end{cases} \quad (6.11)$$

where

$$\begin{aligned} p(r) = & -\frac{r^{36}}{72} + \frac{6r^{35}}{35} - \frac{15r^{34}}{17} + \frac{74r^{33}}{33} - \frac{57r^{32}}{32} - \frac{174r^{31}}{31} + \frac{269r^{30}}{15} - \frac{450r^{29}}{29} - \frac{153r^{28}}{8} \\ & + \frac{1564r^{27}}{27} - \frac{510r^{26}}{13} - \frac{204r^{25}}{5} + \frac{737r^{24}}{8} - \frac{1032r^{23}}{23} - \frac{477r^{22}}{11} + 64r^{21} - \frac{81r^{20}}{40} \\ & - \frac{1242r^{19}}{19} + \frac{731r^{18}}{9} - \frac{966r^{17}}{17} + \frac{219r^{16}}{8} - \frac{146r^{15}}{15} + \frac{18r^{14}}{7} - \frac{6r^{13}}{13} + \frac{r^{12}}{24}. \end{aligned} \quad (6.12)$$

Note that this is a slight modification of the initial vortex setup in Kadioglu et al. (2008), and the vortex setup presented here is of a stable configuration, i.e. the centre of the balanced vortex structure corresponds to the minimum of the density field. Figure 6.2 depicts the initial balanced Euler vortex structure.

The Euler vortex is non-dimensionalised as follows,

$$l_{\text{ref}} = h_{\text{ref}} = 10.0 \text{ km}, \quad t_{\text{ref}} = 100.0 \text{ s}, \quad p_{\text{ref}} = 100.0 \text{ kPa}, \quad \rho_{\text{ref}} = p_{\text{ref}}/c_{\text{ref}}^2 \approx 1.16 \text{ kg m}^{-2}, \quad (6.13)$$

and this yields a Mach number $\text{Ma} \approx 0.341$. However inside the compact vortex structure, the vortex attains a maximum velocity of 25.0 m s^{-1} which corresponds to $\text{Ma}_{\text{vort}} \approx 0.085$, and therefore the low Mach analysis in section 4.1.2 is justified. Experiments involving the Euler vortex have appeared in Chew et al. (2021).

In all the SWE and Euler vortex experiments, the time step size is constrained by $\text{CFL} = 0.45$

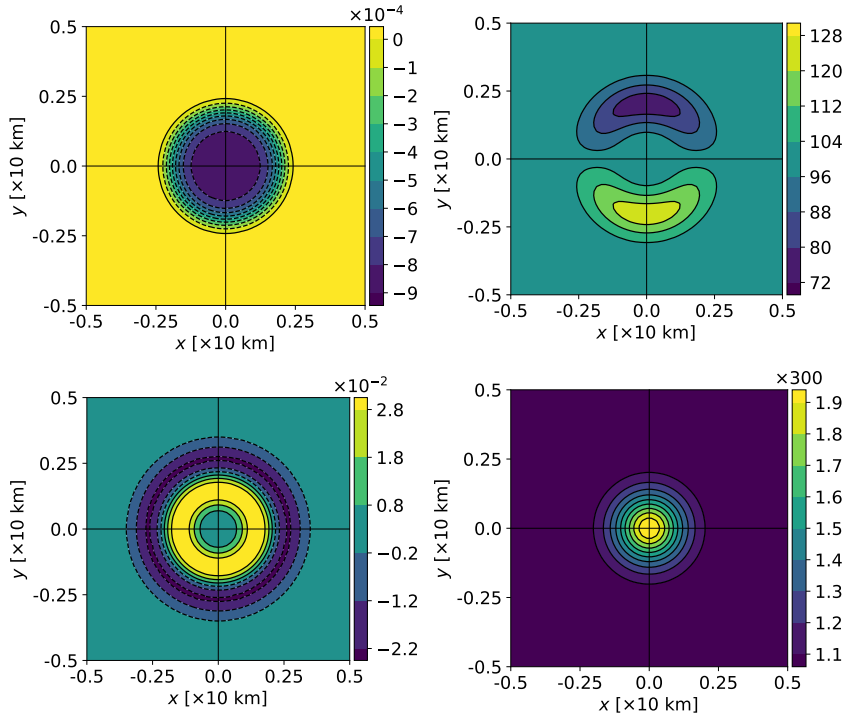


Figure 6.2: Euler vortex initial balanced states on a (64×64) grid: Exner pressure perturbation π' ; dimensionless contours in the range $[-9, 0] \times 10^{-4}$ with an interval of 10^{-4} (top left), horizontal velocity u ; contours in the range $[72, 128]$ m s^{-1} with a 8 m s^{-1} interval (top right), vorticity; contours in the range $[-2.2, 2.8] \times 10^{-2} \text{ s}^{-1}$ with a 10^{-2} s^{-1} interval (bottom left), and potential temperature θ ; contours in the range $[330, 570]$ K with a 30 K interval (bottom right). Negative contours dashed.

on a (64×64) grid. The dimensionless extent of the domain is $x = [-0.5, 0.5]$ and $y = [-0.5, 0.5]$ with doubly periodic boundary conditions. Furthermore, unless otherwise stated, $R = 0.4$ and $(x_c, y_c) = (0.0, 0.0)$.

6.1.2 Rising bubble

The rising bubble test case is a gravity-driven thermal flow with $f = 0.0$. The bubble is initialised as a positive perturbation of the background potential temperature θ_0 . For $\theta_0 = 300 \text{ K}$, the dimensionless perturbation $\delta\theta$ is defined by

$$\delta\theta = \frac{\theta'_0}{\theta_0} \cos\left(\frac{\pi}{2} r_B\right), \quad (6.14)$$

where

$$r_B = \frac{1}{r_0} \sqrt{x^2 + (z - z_0)^2}, \quad (6.15)$$

and $\theta'_0 = 2.0 \text{ K}$ is the initial magnitude of the potential temperature perturbation, $r_0 = 2.0 \text{ km}$ is the initial radius of the bubble, and $z_0 = 2.0 \text{ km}$ is the initial vertical displacement of the bubble. The dimensionless perturbation in the density field is initialised with $\rho = P/(1 + \delta\theta)$.

The extent of the domain is $x = [-10.0, 10.0]$ km and $z = [0.0, 10.0]$ km with periodic bound-

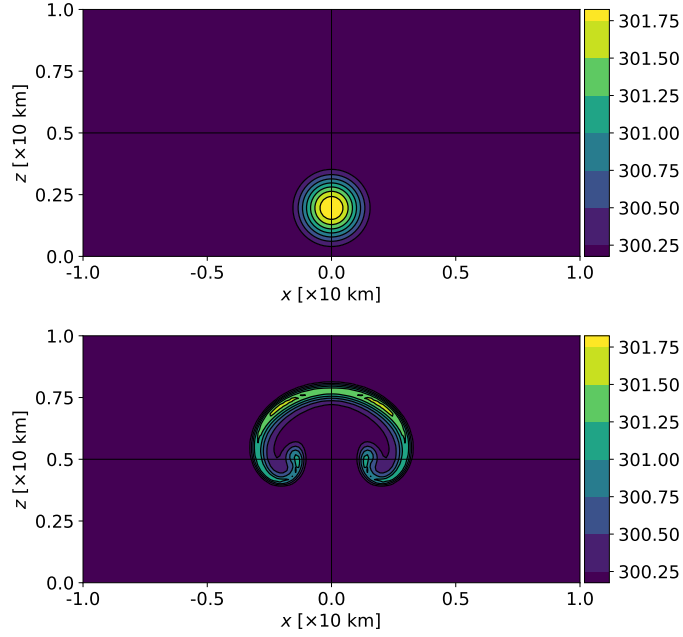


Figure 6.3: Rising bubble on a (160×80) grid: potential temperature at initial time, $t = 0.0$ s (top), and final time, $t_{\text{end}} = 1000.0$ s (bottom); contours in the range $[300.25, 301.75]$ K with a 0.25 K interval.

aries in x and no-flux in z (Mendez-Nunez and Carroll, 1994; Klein, 2009; Benacchio et al., 2014). Note that unlike the travelling vortex test cases which are defined on the horizontal xy -plane, the rising bubble is a vertical slice test defined on the vertical xz -plane. This test case is non-dimensionalised with the following reference units,

$$\begin{aligned}
 l_{\text{ref}} = h_{\text{ref}} = 10.0 \text{ km}, & & t_{\text{ref}} = 1000.0 \text{ s}, & & T_{\text{ref}} = 300.0 \text{ K}, \\
 p_{\text{ref}} = 86.1 \text{ kPa}, & & \rho_{\text{ref}} = p_{\text{ref}}/c_{\text{ref}}^2 \approx 1.0 \text{ kg m}^{-2}, & &
 \end{aligned} \tag{6.16}$$

yielding $\text{Ma} \approx 0.0341$. Unless otherwise stated, all the rising bubble experiments in this thesis are simulated with a time step that is constrained by $\text{CFL} = 0.5$ on a (160×80) grid.

The initial pressure fields are balanced with respect to a homogeneous hydrostatic pressure field $\bar{p}(z)$ based on θ_0 , with $\bar{p}(0) = 86.1$ kPa and $\pi' = 0.0$ everywhere. However, this initial pressure field is not balanced with respect to the initial potential temperature field $\theta_0 + \delta\theta$. Figure 6.3 depicts the potential temperature perturbation of the rising bubble at the initial time and at the simulation end time. Experiments involving the rising bubble have appeared in Chew et al. (2021).

6.1.3 Inertia-gravity wave

The balanced inertia-gravity wave test case consists of an internal wave pulse spreading horizontally from its origin in a channel. This wave is generated by an initial perturbation of the

potential temperature $\delta\theta$, and its dimensionless form is given by

$$\delta\theta = \frac{\theta'_0}{\theta_0} \frac{\sin(\pi z/H)}{1 + [(x - x_c)/a]^2}, \quad (6.17)$$

where $\theta_0 = 300$ K is the reference background potential temperature at $z = 0.0$ km and $\theta'_0 = 0.01$ K is the magnitude of the initial potential temperature perturbation. Note that due to the stratification of $\theta = \theta_0 + \delta\theta$, the dynamics of the inertia-gravity waves differ from those of the previous test cases. Furthermore, $H = 10.0$ km, $x_c = 0.0$ km, and $a = 100.0$ km. The stably stratified isothermal background has a constant Brunt-Väisälä frequency of $N = 0.01$ s⁻¹, and there is no background flow.

The extent of the domain is $x = [-3000.0, 3000.0]$ km and $z = [0.0, 10.0]$ km with periodic boundary conditions in x and no-flux in z (Skamarock and Klemp, 1994; Benacchio and Klein, 2019). Experiments involving similar inertia-gravity wave tests can be found in Baldauf and Brdar (2013). To ensure that the initial potential temperature perturbation is horizontally bounded within the domain, a mollification function $m(x)$ is applied to x in (6.17), where

$$m(x) = \frac{1}{2} \min(1 - \cos(\pi \Xi_L(x)), 1 - \cos(\pi \Xi_R(x))), \quad (6.18)$$

and

$$\Xi_L(x) = \min\left(1.0, \frac{x - x_{\min}}{0.25 L}\right), \quad (6.19a)$$

$$\Xi_R(x) = \min\left(1.0, \frac{x_{\max} - x}{0.25 L}\right), \quad (6.19b)$$

with $L = x_{\max} - x_{\min} = 6000.0$ km being the total horizontal extension of the domain. As with the rising bubble test, this test develops on a vertical slice defined in the xz -plane. The test case is non-dimensionalised with the following reference units,

$$\begin{aligned} l_{\text{ref}} = h_{\text{ref}} = 10.0 \text{ km}, & & t_{\text{ref}} = 100.0 \text{ s}, & & T_{\text{ref}} = 300.0 \text{ K}, \\ p_{\text{ref}} = 100.0 \text{ kPa}, & & \rho_{\text{ref}} = p_{\text{ref}}/c_{\text{ref}}^2 \approx 1.16 \text{ kg m}^{-2}, & & \end{aligned} \quad (6.20)$$

yielding $\text{Ma} \approx 0.341$. However, the velocity of the inertia-gravity wave pulse is approximately $u_{\text{pulse}} \approx 30.33$ m s⁻¹, with $\text{Ma}_{\text{pulse}} \approx 0.10$. This justifies the use of the low Mach analysis in section 4.1.3 for the compressible-soundproof blending. Unless otherwise stated, a constant time step of size $\Delta t = 100.0$ s is used with a (300×10) grid for all inertia-gravity wave experiments in this thesis. The aspect ratio of the domain corresponds to the hydrostatic configuration in Skamarock and Klemp (1994). Figure 6.4 depicts the potential temperature perturbation of the balanced inertia-gravity wave at initial time and at the simulation end time.

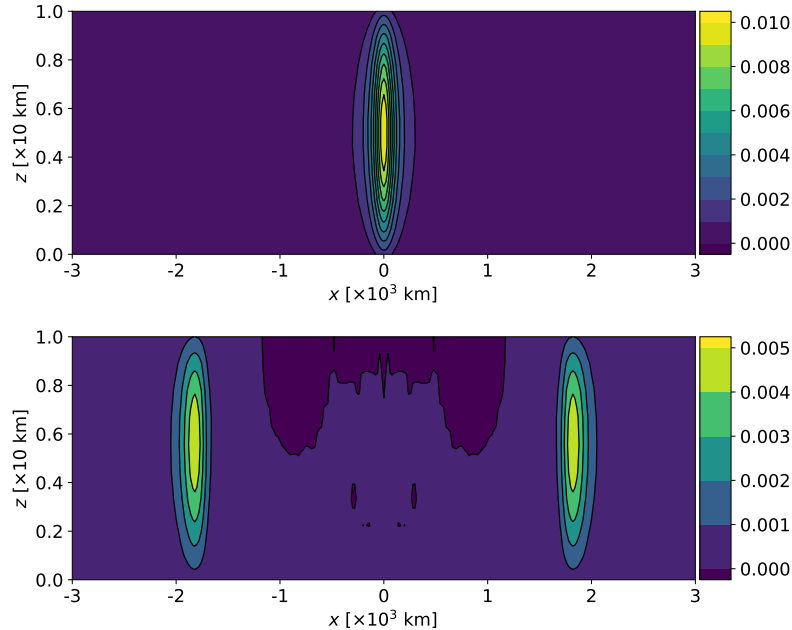


Figure 6.4: Hydrostatic inertia-gravity wave on a (300×10) grid: potential temperature perturbation at initial time, $t = 0.0$ s (top); contours in the range $[0.0, 0.01]$ K with a 0.001 K interval and final time $t_{\text{end}} = 60\,000.0$ s (bottom); contours in the range $[0.0, 0.005]$ K with a 0.001 K interval.

6.2 Effectiveness of the blending schemes

In this section, dynamically imbalanced initial data are used for compressible simulations of the idealised test cases. The one-step blending strategy from chapter 4 is applied for the first time step, and the solution of such a *blended run* is evaluated against the solution from the limit model.

Specifically, the performance of the one-step blending strategy is evaluated by taking probe measurements of the full pressure time-increment δp , where

$$\delta p^n = p^{n+1} - p^n, \quad (6.21)$$

with n indexing the time level. The first increment δp^0 is omitted as it corresponds to a spinup adjustment.

Having obtained a time series of δp , the distance of the pressure time-increment computed for a given compressible run from the reference run is evaluated by means of a relative error E_ν ,

$$E_\nu = \frac{\|\delta p_\nu - \delta p_{\text{ref}}\|_2}{\|\delta p_{\text{ref}}\|_2}, \quad (6.22)$$

where $\nu = b$ for the blended run and $\nu = c$ for the imbalanced compressible run. $\|\cdot\|_2$ is the 2-norm taken over the probe-measured time series of δp .

The example above on the relative error computation concerns a blending between the com-

pressible and pseudo-incompressible regimes, but this example can be straightforwardly extended to the blending between the shallow water and lake regimes as well as between the hydrostatic and nonhydrostatic regimes.

6.2.1 Travelling vortex results

SWE vortex

Recall from section 2.4 that in deriving the shallow water equations from the Euler equations, three quantities in the Euler equations were identified with a quantity related to the water height h , namely,

$$\rho \rightarrow h, \quad P \rightarrow h, \quad \pi' \rightarrow h'. \quad (6.23)$$

An artificial imbalanced initial state is created by setting the water depth at the initial time to a constant everywhere in the domain, i.e. $h(t=0) = 1.0$ m and $h'(t=0) = 0.0$ m. In this scenario, the water depth field is imbalanced with respect to the initial velocity fields, and this introduces fast mode imbalances in the form of shallow water waves. A probe measurement of the water depth h is taken at the centre of the domain at $(0,0)$ km. Here, a probe measurement means taking a measurement of the values of the flow variables at a fixed location in the domain and at selected time instances (Benacchio et al., 2014).

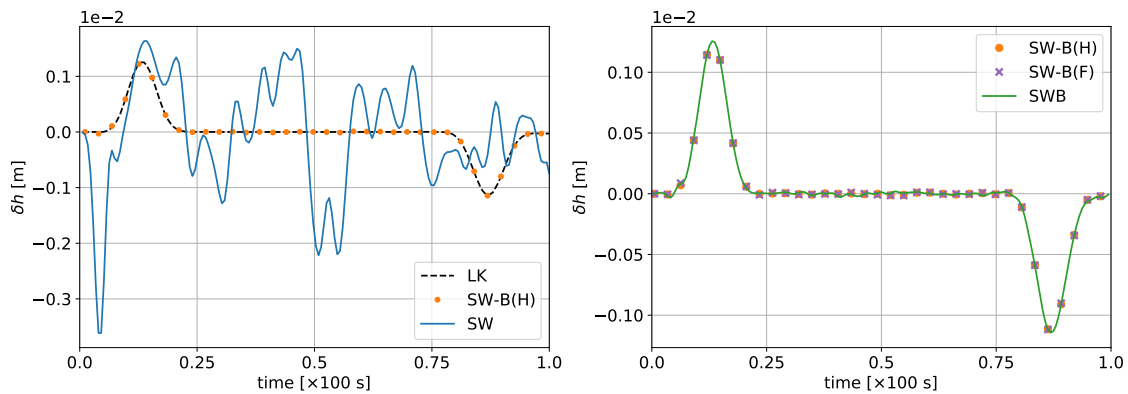


Figure 6.5: SWE vortex: effect of one-step blending for imbalanced initial states on the time series of temporal increments of the full water depth field δh at location $(x, y) = (0, 0)$ km. Left: comparison between a blended run using h'_{half} (orange dots, SW-B(H)), an imbalanced shallow water run without blending (solid blue line, SW), and the reference solution from the lake model (dashed black line, LK). Right: comparison of blended runs using h'_{half} (orange dots, SW-B(H)) and h'_{full} (purple crosses, SW-B(F)), and the shallow water solution with balanced initial states (solid green line, SWB).

The left panel of Figure 6.5 compares the time-increment of the water depth field h for three runs with this imbalanced initial state. Imbalances of the order of the slower vortex dynamics can be seen for the imbalanced shallow water solution (solid blue curve) and the solution quality is effectively destroyed. On the other hand, the solution of the blended run is close to the reference solution computed with the lake equations (compare the orange dot markers with the dashed

black line). This implies that the blending strategy for the shallow water and lake equations successfully removes the imbalances, and the method is able to recover the dynamics of the balanced state.

The right panel of Figure 6.5 depicts a close-up of the blended runs with the choices of h'_{half} and h'_{full} from section 4.3. These runs are compared against a run with the balanced initial state obtained from the known exact solution of the SWE vortex. Both the blended runs are almost indistinguishable from the reference balanced solution, although the h'_{half} run is slightly closer to the balanced solution. The relative error of the blended run with respect to the reference balanced run, E_b , is 0.0269 for the h'_{half} run and 0.0335 for the h'_{full} run. This result corroborates the insight from sections 4.1.3 and 4.3 that h'_{half} is a better choice. The relative error of the imbalanced compressible run with respect to the reference balanced run, E_c , is 2.42. Using the h'_{half} blending strategy improves the error score by approximately two orders of magnitude.

Euler vortex

For the Euler vortex, an imbalanced initial state is created by setting $P = 347.95 \text{ kg m}^{-2} \text{ K}$ and $\pi' = 0.0$ over the whole domain. Investigations similar to the ones presented for the SWE vortex are in Figure 6.6 for the probe measurement of the time-increment of the full pressure field.

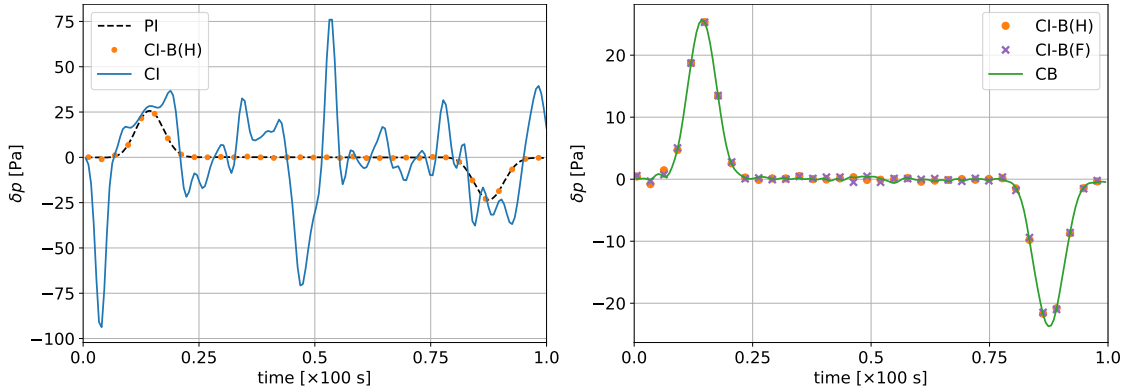


Figure 6.6: Euler vortex: effect of one-step blending for imbalanced initial states on the time series of temporal increments of the full pressure δp at location $(x, y) = (0, 0)$ km. Left: comparison between a blended run using π'_{half} (orange dots, CI-B(H)), a run without blending (solid blue line, CI), and the reference solution from the pseudo-incompressible model (dashed black line, PI). Right: comparison of blended runs using π'_{half} (orange dots, CI-B(H)) and π'_{full} (purple crosses, CI-B(F)), and the compressible solution with balanced initial states (green solid line, CB).

In the left panel of Figure 6.6, without blending, the solution quality of the imbalanced compressible run (solid blue curve) is destroyed as the model is unable to trace the dynamics of the vortex trajectory. On the other hand, the blended run recovers the balanced dynamics and is almost indistinguishable from the reference solution computed with the pseudo-incompressible model (compare the orange markers and the dashed black line). This result implies that the blending strategy between the compressible and pseudo-incompressible models is effective, and

the method is able to recover the balanced dynamics.

The right panel of Figure 6.6 depicts a close-up of the blended runs with π'_{half} and π'_{full} from section 4.1.3. Results of these blended runs are compared against a compressible run with the balanced initial state obtained from the known exact compressible solution of the Euler vortex. Both the blended runs are close to the reference balanced solution, with the π'_{half} run being slightly closer. The relative errors of the blended runs, E_b , are 0.0367 and 0.0467 for the π'_{half} and π'_{full} runs respectively. As with the SWE vortex case, this result corroborates the insight from section 4.1.3 that π'_{half} is a better choice. From here on, the choice of π'_{half} is used for all experiments. The relative error of the imbalanced compressible run, E_c , is 3.06, and therefore using the π'_{half} blended run improves the error score by approximately two orders of magnitude.

6.2.2 Rising bubble results

From the rising bubble setup in section 6.1.2, the initial pressure field is imbalanced with respect to the potential temperature field. Starting with this initial imbalanced state, the left panel of Figure 6.7 shows the propagation of acoustic pressure waves in the compressible solution, and this imbalance can be seen in the time series of the pressure perturbation increment $\delta p'$ (blue line in the right panel of Figure 6.7). The probe measurement is taken at $(x, z) = (-7.5, 5.0)$ km, and the location of this probe is indicated by the orange cross in the left panel. Here, $p' = p - \bar{p}$. As with the travelling vortex cases, the blended run (orange markers in the right panel) is able to recover a dynamics that is close to the acoustics-free reference pseudo-incompressible solution (dashed black line). A constant small time step $\Delta t_{AC} = 1.9$ s is used to generate these results. The results presented here aim to reproduce the results in Benacchio et al. (2014) within the Benacchio and Klein (2019) numerical framework.

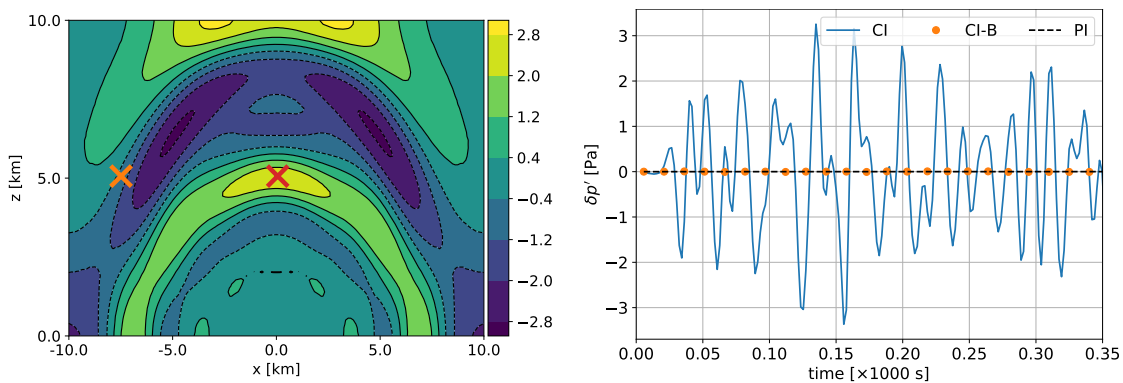


Figure 6.7: Left: $\delta p'$ value after the 14th time step ($t = 26.6$ s) for the compressible model; contours in the range $[-2.8, 2.8]$ Pa with an interval of 0.8 Pa, negative contours dashed. The orange cross marks $(x, z) = (-7.5, 5.0)$ km and the red cross $(x, z) = (0.0, 5.0)$ km. Right: history of $\delta p'$ over the first 350.0 s measured at $(x, z) = (-7.5, 5.0)$ km for the compressible model (solid blue line, CI), the pseudo-incompressible model (dashed black line, PI), and the blended model with one pseudo-incompressible time step (orange dots, CI-B).

6 Numerical results

Figure 6.8 shows the result of a blended run with probe measurements taken at two different locations. The blended run has a time step size of Δt_{AC} . Away from the trajectory of the rising bubble (left panel), the amplitude of the remnant imbalances from the blended run is significant, and it is comparable to the increment in the pressure perturbation induced by the dynamics of the rising bubble. On the bubble trajectory (right panel), the pressure perturbation increment due to the rising bubble dominates, and the remnant acoustic imbalances are relatively small. Here, the solution of the blended run is almost identical to the solution of the reference pseudo-incompressible run.

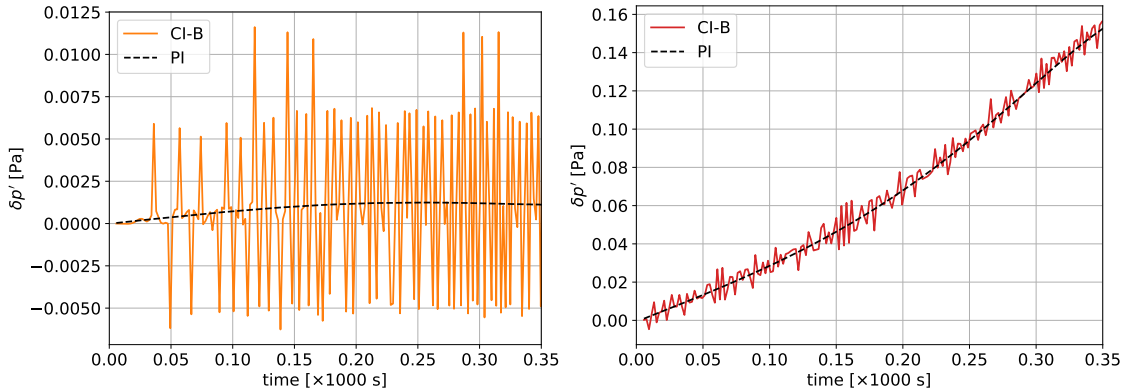


Figure 6.8: History of $\delta p'$ over the first 350.0 s measured at $(x, z) = (-7.5, 5.0)$ km (left) and at $(x, z) = (0.0, 5.0)$ km (right). A constant small time step $\Delta t_{AC} = 1.9$ s is used. Pseudo-incompressible solution (dashed black line, PI) and blended solution with one time step spent in the pseudo-incompressible regime (CI-B, solid orange or solid red corresponding to the probe marker in the left panel of Figure 6.7). The time-increment involving the first time step is omitted.

Figure 6.9 shows another two blended runs, but these runs are with a larger advective CFL-constrained time step, Δt_{ADV} . Specifically, Δt_{ADV} is constrained by $CFL = 0.5$ with $\Delta t = 21.69$ s for the first two time steps. The results are similar to the ones presented in Figure 6.8, although the amplitude of the remnant acoustic imbalances is slightly larger due to the larger time steps used. However when compared to imbalanced compressible solution without blending (blue line in the right panel of Figure 6.7), the amplitude is still relatively small: note the difference in the scale of the vertical axis between this panel and the panels of Figure 6.9.

The relative errors for the blended runs in Figures 6.8 and 6.9 are computed and presented in Table 6.1 as E_b , where the reference run is the pseudo-incompressible solution. Furthermore, E_c is computed as the relative error of the imbalanced compressible runs relative to the reference pseudo-incompressible run. Results for these imbalanced compressible runs are not shown in Figures 6.8 and 6.9, but the result of this run for Δt_{AC} can be found in Figure 6.7 (solid blue line in the right panel). Note that these relative errors are computed over the *full* simulation run, i.e. for $\delta p'$ over $[0.0, 1000.0]$ s. For the small time step Δt_{AC} run, the blending strategy outlined in chapter 4 is able to improve the solution quality by more than two orders of magnitude when compared to the relative error of the imbalanced compressible solution, and by a factor of ~ 28

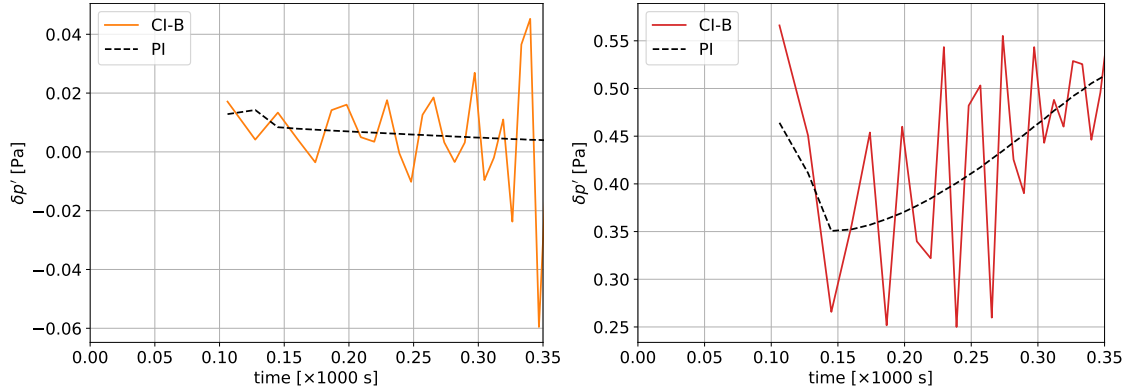


Figure 6.9: History of $\delta p'$ over the first 350.0 s measured at $(x, z) = (-7.5, 5.0)$ km (left) and at $(x, z) = (0.0, 5.0)$ km (right). A CFL-constrained time step, Δt_{ADV} , is used, where $\Delta t = 21.69$ s for the first two time steps and Δt is then determined by an advective CFL = 0.5. Pseudo-incompressible solution (dashed black line, PI) and blended solution with one time step spent in the pseudo-incompressible regime (CI-B, solid orange or solid red corresponding to the probe marker in the left panel of Figure 6.7). The time-increments involving the first two time steps are omitted in the plots.

Table 6.1: Errors E_c and E_b (see equation (6.22) and the related text for the definitions) of the time series of $\delta p'$ in $[0.0, 1000.0]$ s relative to the reference pseudo-incompressible run (dashed black lines in Figures 6.8 and 6.9). The acoustic time step size is $\Delta t_{AC} = 1.9$ s, while Δt_{ADV} is determined by advective CFL = 0.5 and $\Delta t_{ADV} = 21.69$ s for the first two time steps. Probe location $(-7.5, 5.0)$ km corresponds to the orange marker in Figure 6.7 and orange lines in Figures 6.8 and 6.9, and probe location $(0.0, 5.0)$ km to the red marker and red lines in the respective Figures.

Δt	probe location	E_c	E_b	E_c/E_b
Δt_{AC}	$(-7.5, 5.0)$ km	413.1849	1.4743	280.26
	$(0.0, 5.0)$ km	10.1787	0.0363	280.40
Δt_{ADV}	$(-7.5, 5.0)$ km	108.9313	3.8897	28.01
	$(0.0, 5.0)$ km	2.6447	0.0943	28.05

for the run with the larger CFL-constrained time step, Δt_{ADV} , see the E_c/E_b column in Table 6.1.

For the rising bubble test case, the *full* pressure time-increment δp differs slightly between the pseudo-incompressible solution and the solution of the blended runs. This difference is depicted in the left panel of Figure 6.10, and it is due to the time-dependence of the hydrostatically balanced background pressure \bar{p} in the blended run while \bar{p} is constant in time for the pseudo-incompressible run. However, the pressure perturbation time-increments $\delta p'$ are very similar between the two runs (see the right panel of Figure 6.10), and this suggests that the blending strategy is able to recover balanced dynamics regardless of small compressibility-induced variations in the background pressure \bar{p} .

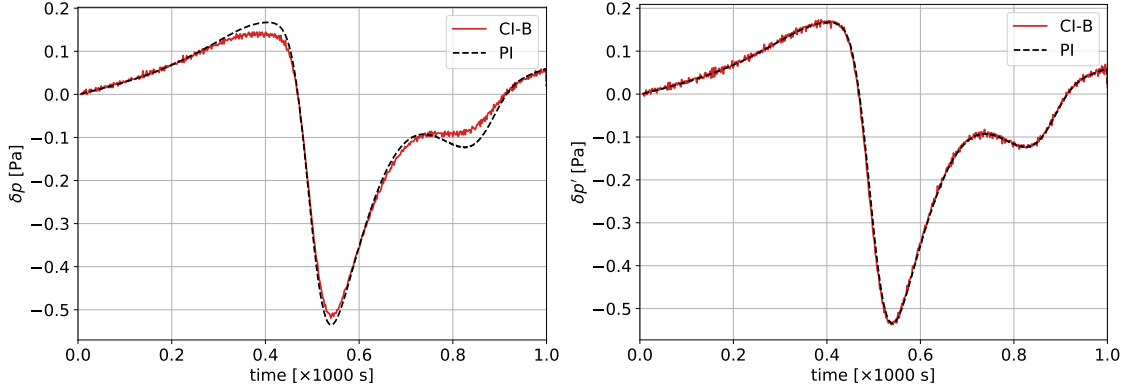


Figure 6.10: Probe measurement over $[0.0, 1000.0]$ s of the full pressure increment δp (left) and pressure perturbation increment $\delta p'$ (right). Pseudo-incompressible solution (black dashed line, PI) and the blended solution (solid red line, CI-B). The runs are with a time step size of Δt_{AC} and the probe is located at $(x, z) = (0.0, 5.0)$ km (indicated by the red marker in the left panel of Figure 6.7).

6.2.3 Inertia-gravity wave results

An artificial initially imbalanced state is created for the inertia-gravity wave by setting the Exner pressure perturbation $\pi' = 0.0$ everywhere in the domain. The imbalanced nonhydrostatic solution features an additional acoustic-gravity wave pulse alongside the inertia-gravity wave pulses. In this section, the blending between the compressible nonhydrostatic regime and the hydrostatic regime is investigated.

The long-time characteristic of the acoustic-gravity and inertia-gravity wave pulses are depicted in Figure 6.11. The acoustic-gravity wave pulse does not propagate, and it remains oscillating vertically in the centre of the domain (compare the dashed blue lines in the left and right panels). On the other hand, the inertia-gravity wave splits and propagates away from its initial location (black lines in both panels). Markers representing two additional blended runs, between the compressible and hydrostatic regimes (orange dots) and between the compressible and pseudo-incompressible regimes (green crosses), are depicted for comparison.

Probe measurements of the vertical velocity are taken, and the time series of such a measurement is documented in Figure 6.12. The probe is located at $(0.0, 5.0)$ km, and its location is indicated by the red cross in Figure 6.11. Measurements of the vertical velocity are taken as, empirically, this quantity best captures the features of the oscillating acoustic-gravity wave.

The results from Figure 6.12 show that blending for a single time step to the hydrostatic regime (orange dots) or to the pseudo-incompressible regime (green crosses) is sufficient in removing the dominant vertically propagating acoustic mode present in the imbalanced initial conditions (solid blue line in the left panel). The right panel is a close-up of the left panel without the imbalanced compressible solution, and in this panel, we see that the solutions of the blended runs are almost identical to the balanced compressible solution.

Finally, the errors relative to the balanced compressible solution over the full simulation time,

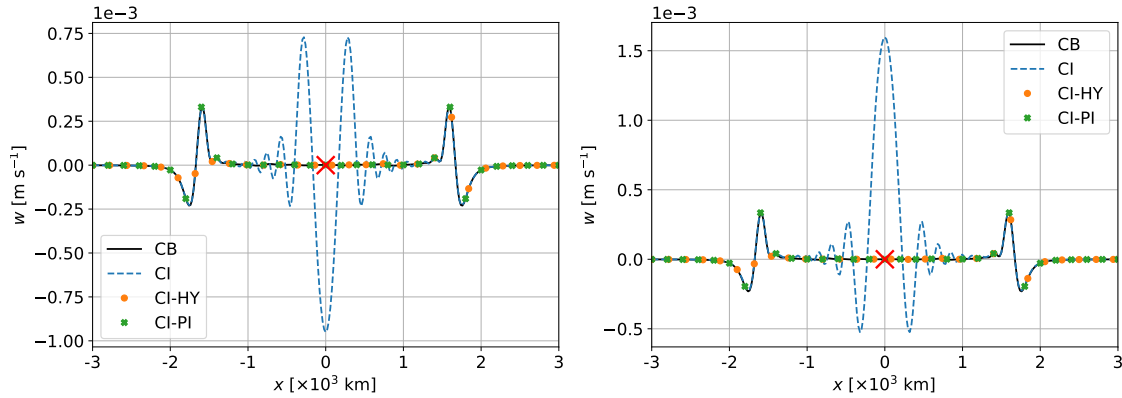


Figure 6.11: Inertia-gravity wave: slices of the vertical velocity w at $z = 5.0$ km after the 550th time step, i.e. at $t = 55\,000.0$ s, (left) and after the 551st time step, i.e. at $t = 55\,100.0$ s (right). Balanced compressible solution (solid black line, CB), imbalanced compressible solution (dashed blue line, CI), blended solution with one time step spent in the hydrostatic regime (orange dots, CI-HY), and blended solution with one time step spent in the pseudo-incompressible regime (green crosses, CI-PI). The red cross marks the location of the probe measurement in Figure 6.12.

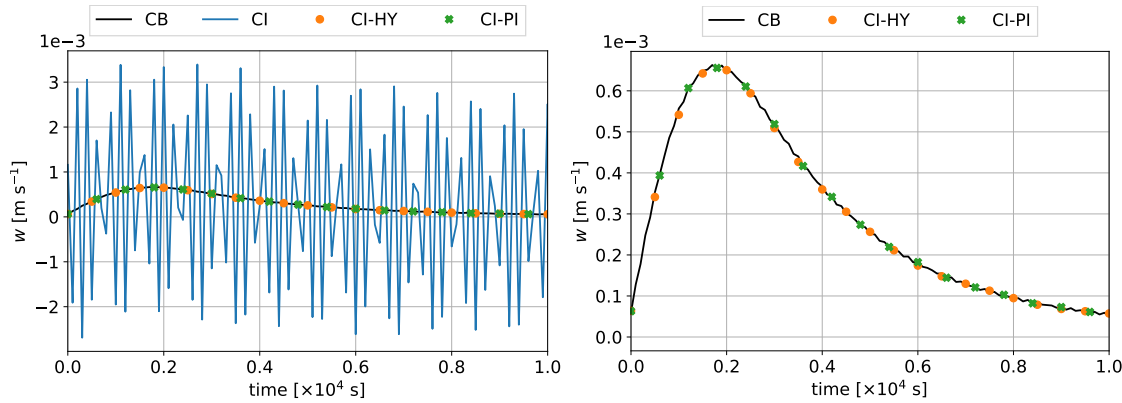


Figure 6.12: History of the probe measurement of the vertical velocity w over the first 10 000 s at $(x, z) = (0.0, 5.0)$ km. (left) Balanced compressible solution (black solid line, CB), imbalanced compressible solution (blue dashed line, CI), blended solution with one time step spent in the hydrostatic regime (orange dots, CI-HY), and blended solution with one time step spent in the pseudo-incompressible regime (green crosses, CI-PI). (right) A close-up of the left panel without the CI solution.

i.e. over $[0.0, 60\,000.0]$ s, are computed for the runs depicted in Figure 6.12. The relative errors are: 11.65 for the imbalanced compressible run; 0.02365 for the hydrostatic blended run; and 0.005527 for the pseudo-incompressible blended run. When compared with the relative error of the imbalanced compressible run, the hydrostatic blended run yields over two orders of magnitude improvement, and the pseudo-incompressible run is an improvement of over three orders of magnitude.

That both the blended runs work well in this configuration of the inertia-gravity wave test corroborates the results by Benacchio and Klein (2019). The authors demonstrated that for the hydrostatic configuration of the inertia-gravity wave test case used here, the solutions of both the pseudo-incompressible and the hydrostatic limit models are very close to that of the balanced

compressible run (cf. the bottom row of their Fig. 6). This promising result suggests that the blending strategy works for balanced gravity mode initialisation.

6.3 Ensemble data assimilation and blending (DAB): setup

Balanced data assimilation poses two additional challenges for the blending strategies. Unlike the artificially imbalanced initial states in section 6.2, the parameters chosen for the data assimilation procedure significantly affect the imbalances introduced. Furthermore, the balancing procedure via blending occurs after each point in time when data is assimilated, and any residual error that is not removed by the balancing procedure will accumulate and destroy the quality of the balanced solution over time. As a result, an effective dynamics-driven blending strategy must be capable of removing the imbalances regardless of the data assimilation parameters chosen.

Computationally, all the random processes mentioned in this section are seeded to ensure reproducibility of the experimental results.

6.3.1 The travelling vortices

To combine blending with data assimilation as detailed in section 5.3, an ensemble is generated by perturbing the initial vortex centre position (x_c, y_c) within the open half interval of $[-0.1 l_{\text{ref}}, 0.1 l_{\text{ref}})$ for both x_c and y_c . Recall that l_{ref} differs between the SWE and Euler vortices, see equations (6.9) and (6.13) for more details. The vortex is then generated around this centre position such that the full vortex structure is translated. Ten such samples are drawn, and they constitute the ensemble members. An additional sample is drawn and solved with the full model for the balanced initial condition. This run, denoted by *obs*, is used to generate the artificial observations. Another run with a setup identical to this additional *obs* sample is made. This time, blending for the first time step is applied, and it is used as the *truth*, i.e. the reference solution. The initial blended time step is to correct for any errors in the initialisation of π , as discussed in section 4.1.3.

This choice of generating the truth and *obs* through a perturbation of the initial condition is such that the ensemble mean does not coincide with the truth. Otherwise, ensemble deflation alone is sufficient to converge the ensemble towards the truth. A similar method of generating the truth and observation has been used in Lang et al. (2017). Note that although the vortex structure is translationally invariant, the setup presented here is not trivial for the experiments in this thesis, as the assimilation of data will destroy the compact vortex structure of the individual ensemble members and introduce imbalances.

The observations are taken from *obs* every $0.25 t_{\text{ref}}$. Only a tenth of the grid points are observed, and these are randomly drawn. The sparse observations are drawn as follows: a Boolean mask selecting for a tenth of the grid points is shuffled by the Fisher and Yates (1953)

algorithm and applied to the observed quantities in obs. This deviates from a more realistic situation where observations and grid points do not coincide. To simulate measurement noise, Gaussian noise is added independently to each observation grid point. The variances used in generating the zero-mean Gaussian noise and a short elaboration on how the observation noise is modelled are provided in Appendix A. A similar method of generating artificial noisy observations by addition of zero-mean Gaussian noise was used in, for example, Bocquet (2011) and Harlim and Hunt (2005) for the Lorenz-63 and Lorenz-96 models.

For the domain localisation, the local regions are of size (11×11) grid points and only observations within this radius are considered for analysis. For the observation localisation, a localisation function corresponding to a Gaussian bell is applied such that observations further from the grid point under analysis have less influence. This influence decays smoothly towards zero at the radius edge, and it is then abruptly truncated to zero at the edge. Finally, no ensemble inflation is applied.

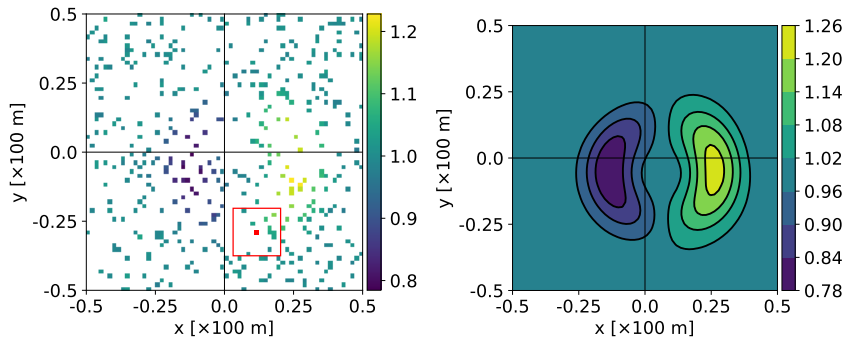


Figure 6.13: SWE vortex: sparse noisy observations (left panel) and truth (right panel) at $t = 300.0$ s. Horizontal momentum hv ; for the right panel, contours in the range $[0.78, 1.26]$ $\text{m}^2 \text{s}^{-1}$ with a $0.06 \text{ m}^2 \text{s}^{-1}$ interval. The red square illustrates, for an example grid point (in red), the observations considered in the local (11×11) grid points region.

Examples of the observations and truths used in the experiments with data assimilation are in Figures 6.13 for the SWE vortex and Figure 6.14 for the Euler vortex. Note that for the SWE vortex, only one experiment with observations drawn from the momentum fields is conducted, and for the Euler vortex, two sets of experiments are conducted: one with the observations drawn from the momentum fields only, and another with observations drawn from the full set of variables.

The ten ensemble members are initialised with balanced states and blending is applied for the first time step. This ensures that any imbalance present in the solution is the result of the data assimilation process only. The ensemble is then solved forward in time with the full model for three revolutions of the vortex across the domain, i.e. up to $3.0 t_{\text{ref}}$. Every $0.25 t_{\text{ref}}$, data from the observations generated are assimilated. The immediate time step after the assimilation procedure is solved with the limit model while the rest of the time steps in the assimilation window are solved with the full model. Conversions according to the blending strategies in chapter 4 are

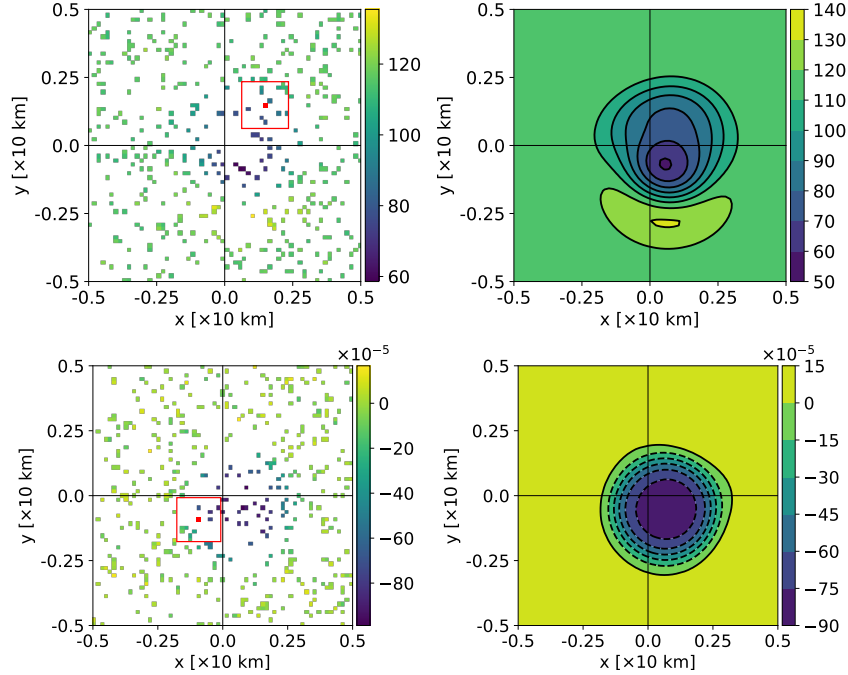


Figure 6.14: Euler vortex: sparse noisy observations (left panels) and truths (right panels) at $t = 300.0$ s. Top: horizontal momentum ρu ; for the right panel, contours in the range $[50.0, 140.0]$ $\text{kg m}^{-1} \text{s}^{-1}$ with a $10.0 \text{ kg m}^{-1} \text{s}^{-1}$ interval. Bottom: Exner pressure perturbation π' ; for the right panel, dimensionless contours in the range $[-90.0, 15.0] \times 10^{-5}$ with an interval of 15.0×10^{-5} . Negative contours are dashed. The red squares illustrate, for an example grid point (in red), the observations considered in the local (11×11) grid points region.

used to switch back and forth between the full and limit models. Furthermore, the choice of π'_{half} (or h'_{half}) is used, see sections 4.1.3 and 4.3 for more details. This ensemble solved with both data assimilation and blending is abbreviated as EnDAB.

The setup is repeated for two other ensembles: one ensemble where data is still assimilated but no blending is performed, and another ensemble where neither data assimilation nor blending is performed. The former is abbreviated as EnDA and the latter as EnNoDA. These two ensembles constitute an *identical twin experiment*, through which the effects of data assimilation can be evaluated. EnDA along with EnDAB constitute yet another identical twin experiment that evaluates the performance of blending. Table 6.2 summarises the details of the data assimilation experimental setup.

6.3.2 Rising bubble: DAB setup

The rising bubble ensemble spread is generated by drawing $\delta\theta'_0$ in the half-open interval of $[2.0 \text{ K}, 12.0 \text{ K})$ uniformly at random and then applied to equation (6.14). While the initial temperature perturbation is seemingly large, the ensemble spread of the bubble position at the end time of the simulation, $t_{\text{end}} = 1000.0$ s, is only moderate. More details will be provided in section 6.4.2.

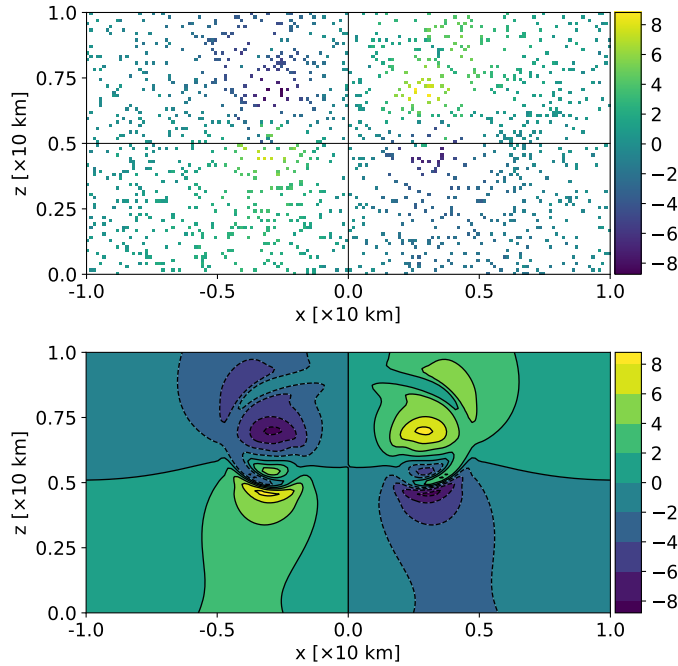


Figure 6.15: Rising bubble: horizontal momentum field ρu at $t = 1000.0$ s. Sparse and noisy observations (top) and truth (bottom); for the bottom panel, contours in range $[-8.0, 8.0]$ $\text{kg m}^{-1} \text{s}^{-1}$ with a $2.0 \text{ kg m}^{-1} \text{s}^{-1}$ interval, negative contours dashed.

Unlike the vortex experiments where the observations are drawn from an obs run, and the reference solution is from a separate truth run, the observations for the rising bubble data assimilation experiments are drawn from the truth run. As such, the obs and the truth for this experiment are obtained from an identical run. Blending is applied to the first time step of this combined obs/truth run, obtaining a balanced solution. After $t = 500.0$ s, observations are taken from obs every 50.0 s for the momentum fields only, and these observations are assimilated. As the dynamics of the rising bubble evolve slowly in the beginning, data is only assimilated after the bubble has been rising for half the simulation duration. As with the vortex experiments, only a tenth of the grid points are randomly observed, zero-mean Gaussian noise is added and localisation within a (11×11) grid points region is applied. More details on the observation noise are given in Appendix A. A truncated Gaussian localisation function corresponding to the Gaussian bell is applied and the ensemble is not inflated. Examples of the observations and the truth are given in Figure 6.15, and the data assimilation experimental parameters are summarised in Table 6.2.

As with the vortex experiments, three ensembles corresponding to the EnNoDA, EnDA, and EnDAB configurations, with balanced initial conditions and with 10 members each, are generated. For the rising bubble experiments, only the momentum fields are assimilated. Note that as the ensembles and the observations are generated with balanced initial conditions, any imbalance present in the simulation results has to be introduced by the data assimilation procedure.

6.3.3 Inertia-gravity wave: DAB setup

The ensemble for the inertia-gravity wave experiments is generated by randomly modifying the potential temperature perturbation in (6.17), specifically by drawing θ'_0 uniformly from the open half interval of $[0.005, 0.015)$ K. Similar to the rising bubble experiment, the truth and the obs are drawn from an identical run which in this case is a compressible run with balanced initial conditions and no blending is applied for the first time step. As the solutions from the blended runs are almost identical to balanced compressible solutions (cf. section 6.2.3), the ensemble data assimilation results obtained in the subsequent section do not differ much even if an additional initial blended time step is applied to this combined obs/truth run.

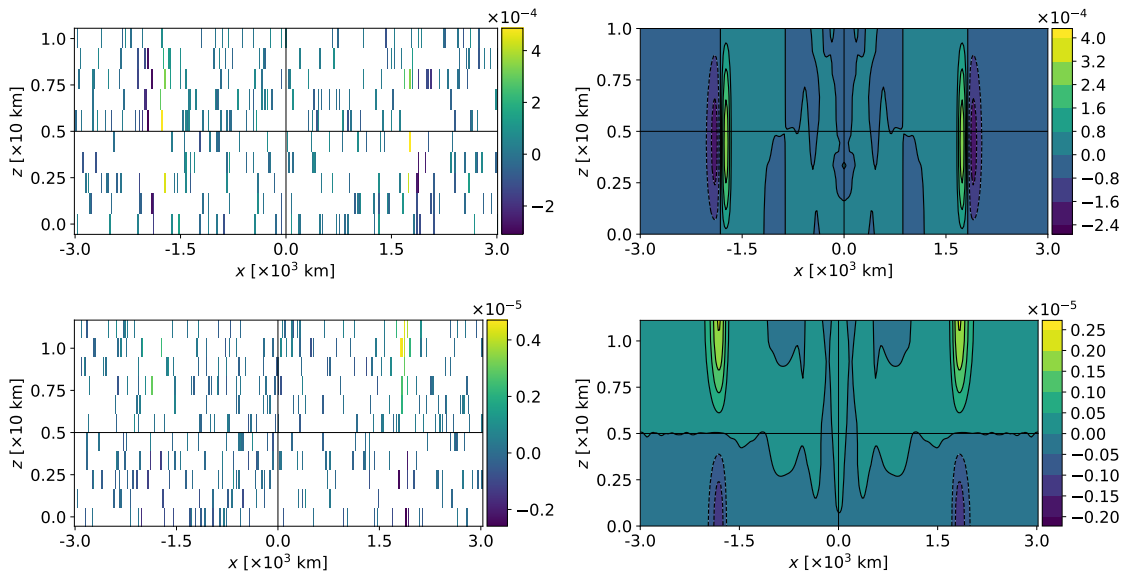


Figure 6.16: Inertia-gravity wave: sparse noisy observations (left panels) and truths (right panels) at $t = 60\,000.0$ s. Top: vertical momentum ρw ; for the right panel, contours in the range $[-2.4, 4.0] \times 10^{-4}$ $\text{kg m}^{-1} \text{s}^{-1}$ with a 0.8×10^{-4} $\text{kg m}^{-1} \text{s}^{-1}$ interval. Bottom: Exner pressure perturbation π' ; for the right panel, dimensionless contours in the range $[-0.20, 0.25] \times 10^{-5}$ with an interval of 0.05×10^{-5} . Negative contours are dashed.

To allow for the initial inertia-gravity wave pulse to separate and propagate in opposing directions, data is only assimilated after a simulation time of $t = 20\,000.0$ s, and subsequently, all the primary quantities are observed and assimilated every $5\,000.0$ s. As with the previous experiments, only a tenth of the grid points are randomly observed, and zero-mean Gaussian noise is used to model the observation error, see Table A.2 for the variances used. A truncated Gaussian localisation function is applied, whereby the influence of the observations decays smoothly towards zero within the localisation region, and the influence is abruptly truncated to zero at the edge. Unlike the previous experiments, the size of the localisation region is (9×9) grid points, since the inertia-gravity wave experiments are run on a (300×10) grid. Examples of the observations and truths are provided in Figure 6.16, and the data assimilation experimental parameters are summarised in Table 6.2.

As before, three ensembles corresponding to the EnNoDA, EnDA, and EnDAB configurations are generated with balanced initial conditions and with 10 members each. Note that for the EnDAB ensemble, the solution is switched to the hydrostatic limit regime for one time step after each assimilation procedure.

Table 6.2: Summary of the assimilation-related experimental parameters. K is the ensemble size, b the ensemble inflation factor, t_{first} the first assimilation time, Δt_{obs} the observation interval, $\psi_{\text{assimilated}}$ the set of quantities assimilated, $(N \times N)_{\text{local}}$ the size of the local region, f_{local} the type of localisation function, η_{obs} the observation noise, $\text{obs}_{\text{sparse}}$ the sparsity of the observations, and N_{blending} the number of time steps spent in the limit model regime after the assimilation of data. π' choice is used in the initialisation of N_{blending} , more details in section 4.1.3

Test case		Vortex	Bubble	IGWave
Ensemble	K		10 members	
	b		1.0	
Observations	t_{first} [s]	25.0	500.0	20 000.0
	Δt_{obs} [s]	25.0	50.0	5 000.0
	$\psi_{\text{assimilated}}$	$\{hu, hv\}$ and $\{\rho u, \rho v\}$ or $\{\rho, \rho u, \rho v, P, \pi'\}$	$\{\rho u, \rho v\}$	$\{\rho, \rho u, \rho v, P, \pi'\}$
	$(N \times N)_{\text{local}}$	(11×11) grid points		(9×9) grid points
	f_{local}		Truncated Gaussian	
	η_{obs}		Gaussian with zero mean and variance in Table A.2	
	$\text{obs}_{\text{sparse}}$		One in 10 grid points, randomly drawn	
Blending	N_{blending}	A single blended time step		
	π' choice	π'_{half} (or h'_{half})		–

6.3.4 Evaluation of data assimilation

The quality of data assimilation is evaluated by a spatially- and ensemble-averaged root mean square error (RMSE) from the truth. For the evaluation of a horizontal slice data assimilation experiment, the spatially- and ensemble-averaged RMSE is given by

$$\text{RMSE}(\psi) = \sqrt{\frac{1}{K} \frac{1}{N_I \times N_J} \sum_{k=1}^K \sum_{i,j=1}^{N_I, N_J} [\psi_k^{\text{ensemble}}(x_i, y_j) - \psi^{\text{truth}}(x_i, y_j)]^2}, \quad (6.24)$$

where $k = 1, \dots, K$ indexes the ensemble members and $i = 1, \dots, N_I$ and $j = 1, \dots, N_J$ the number of grid points in the (x, y) -coordinates. Here, ψ is the set of quantities $\{\rho, \rho u, \rho v, P, \pi'\}$ for the compressible flow equations and $\{h, hu, hv, h'\}$ for the shallow water equations. For the vertical slice experiments, the grid points are in the (x, z) -coordinates, and the vertical momentum ρw is evaluated instead of the horizontal ρv .

6.4 Ensemble data assimilation and blending: results

The ensemble data assimilation results for the setups in sections 6.3.1, 6.3.2, and 6.3.3 are provided below. Further investigations supporting the results in this section are provided in the appendices at the end of the thesis. See section 7.1 for an overview of these appendices.

6.4.1 The travelling vortices: DAB results

The SWE vortex results

Figure 6.17 depicts snapshots of the water depth for two ensemble members alongside the ensemble mean at the simulation end time $t_{\text{end}} = 300.0$ s. The vortex structure in the EnNoDA ensemble are perturbed around the origin $x, y = (0.0, 0.0)$ km, and the mean is centred around the origin. These results show that the EnNoDA ensemble is indeed in line with how the ensemble is generated. As the vortices are initialised with balanced data, the vortex structures remain compact and balanced through the simulation run. As we are interested in recovering the balanced solution, the EnNoDA ensemble takes on the role of the control ensemble.

Results of the EnDA ensemble show that by the end of the simulation, the vortices in the ensemble become imbalanced and the structure of the solution is destroyed. Plain local ensemble data assimilation alone is unable to retain the balanced structure of the solution. However, by switching to the lake equations for one time step after each assimilation process, the shallow water waves are suppressed, and the EnDAB result successfully recovers the balanced vortex structure. Moreover, the effects of data assimilation become obvious: the vortices in the ensemble no longer centre around the origin as in the EnNoDA case, but they have been nudged to the lower right quadrant, and this corresponds to the position of the observations and the truth, cf. Figure 6.14. Data assimilation is successful is nudging the ensemble towards the observation, while blending recovers the balanced solutions.

The results obtained by applying the spatial- and ensemble-averaged RMSE from (6.24) are depicted in Figure 6.18. The errors of the assimilated momentum fields decrease with each assimilation step, and the errors converge eventually to approximately 1.0. For the assimilated quantities, the errors of the EnDA and EnDAB experiments are substantially smaller than that of the EnNoDA experiment, and this suggests that the data assimilation procedure is effective.

The effect of blending has a profound impact on the quantities that are *not* assimilated. For the water depth and the water depth perturbation, the error of the EnDA ensemble jumps after the first assimilation procedure at $t = 25.0$ s, and the error remains roughly of the same order of magnitude for the rest of the simulation. For these quantities, the error score of the EnDA ensemble after the jump is higher than that of the EnNoDA ensemble, and this is due to the imbalances introduced (compare the top and middle rows of Figure 6.17). Appendix B quantifies the error jump as arising from the imbalances introduced by the data assimilation procedure.

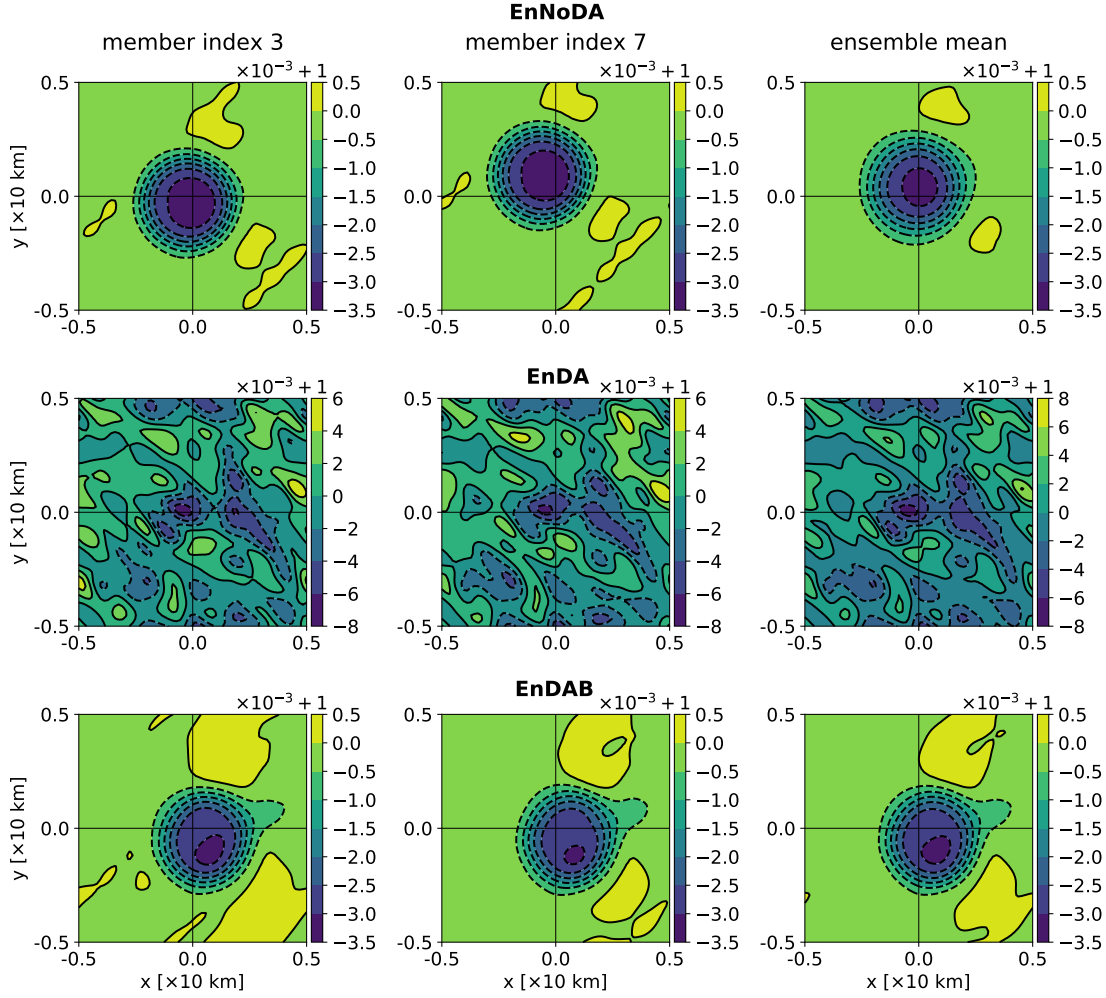


Figure 6.17: SWE vortex: snapshots of the water depth h . Ensemble members with index 3 (first column), 7 (second column), and ensemble mean (third column) at $t = 300.0$ s with the momentum fields $\{h_u, h_v\}$ assimilated. Top row: EnNoDA run; contours in range $[0.9965, 1.0005]$ m with a 0.0005 m interval. Middle row: EnDA run; contours in range $[0.992, 1.006]$ m with a 0.002 m interval. Bottom row: EnDAB run; contours in range $[0.9965, 1.0005]$ m with a 0.0005 m interval. Negative contours are dashed.

For the ensemble run with blending (EnDAB), the error scores for the quantities not assimilated drop after the first assimilation time and remain consistently below that of the control EnNoDA ensemble. The error jump corresponding to the imbalances introduced is absent, and this suggests that blending is an effective method to achieve balanced data assimilation for the shallow water equations.

These promising results also suggest that the one-step blending strategy developed in chapter 4 for the shallow water and the lake equations allows us to seamlessly switch back and forth between model regimes within a single simulation run without a buildup of any residual errors.

An additional encouraging detail is also present in Figure 6.18. In the blended numerical model, the water depth h and water depth perturbation h' are evolved as separate variables and yet, as we ought to expect from a physics-informed perspective, the errors depicted here are

6 Numerical results

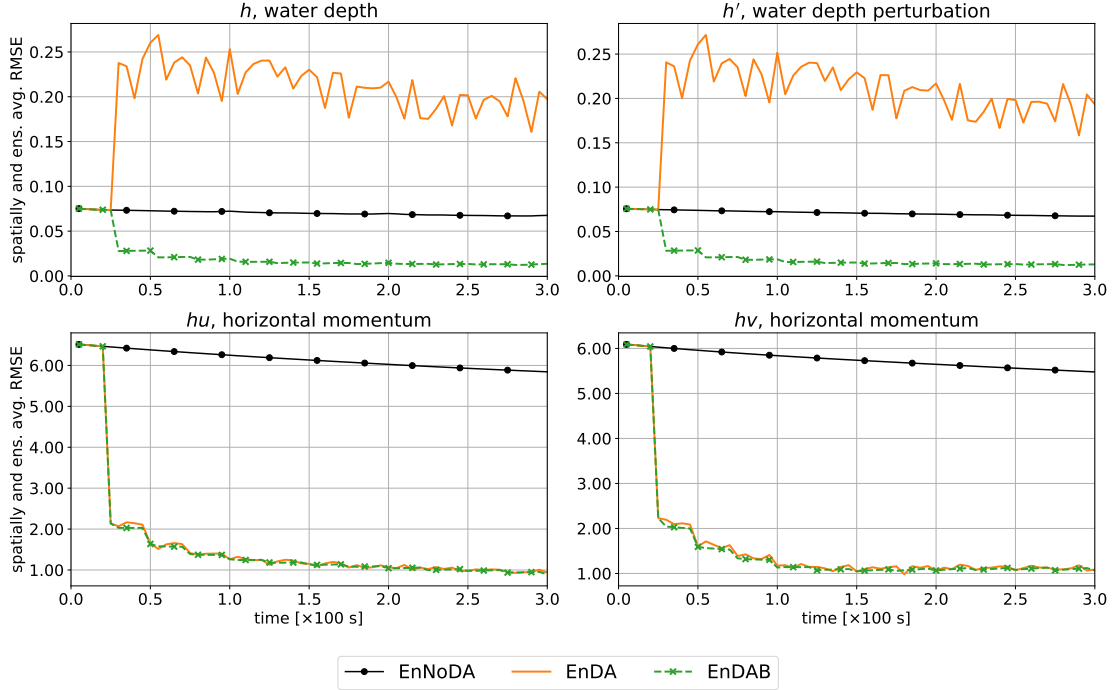


Figure 6.18: SWE vortex: EnNoDA run (dots, solid black), EnDA run (solid orange), EnDAB run (crosses, dashed green). Assimilated quantities are hu and hv . Spatially- and ensemble-averaged RMSE from $t = 0.0$ s to 300.0 s for the water depth h (top left, [m]), water depth perturbation h' (top right, [m]), and momenta hu , hv (bottom left and right, [$\text{m}^2 \text{s}^{-1}$]). The RMSE of the initial ensemble is omitted.

almost identical for both the quantities. This implies that the implementation of the blended numerical model works as intended.

The Euler vortex results

Unlike with the shallow water equations, the dynamics of the Euler equations is also dependent on the potential temperature θ . From section 6.1.1, we recall that this quantity is equivalent to unity in the shallow water context.

Figure 6.19 depicts the snapshots of the pressure perturbation for two ensemble members alongside the ensemble mean at the simulation end time $t_{\text{end}} = 300.0$ s with all quantities assimilated. As with the SWE vortex test case, the EnNoDA ensemble reflects the configuration by which the ensemble is generated: the vortex structures of the individual ensemble members are perturbed around the origin, while the ensemble mean is centred around the origin. The EnNoDA ensemble acts as the control ensemble with balanced solutions.

Without a balancing procedure, the solution structure of the EnDA ensemble with only plain local ensemble data assimilation becomes imbalanced and the compact vortex structure is destroyed. Recall that as the ensemble is generated with balanced initial conditions, any imbalance introduced may be attributed to the data assimilation process. On the other hand, application of blending as a balancing procedure recovers the balanced solution, and the compact vortex

structure is preserved for the EnDAB solutions. Furthermore, the effect of data assimilation is obvious with the vortex structure of the individual ensemble members and the ensemble mean being predominantly in the lower right quadrant, in line with the observations and the truth presented in section 6.3.1, Figure 6.14.

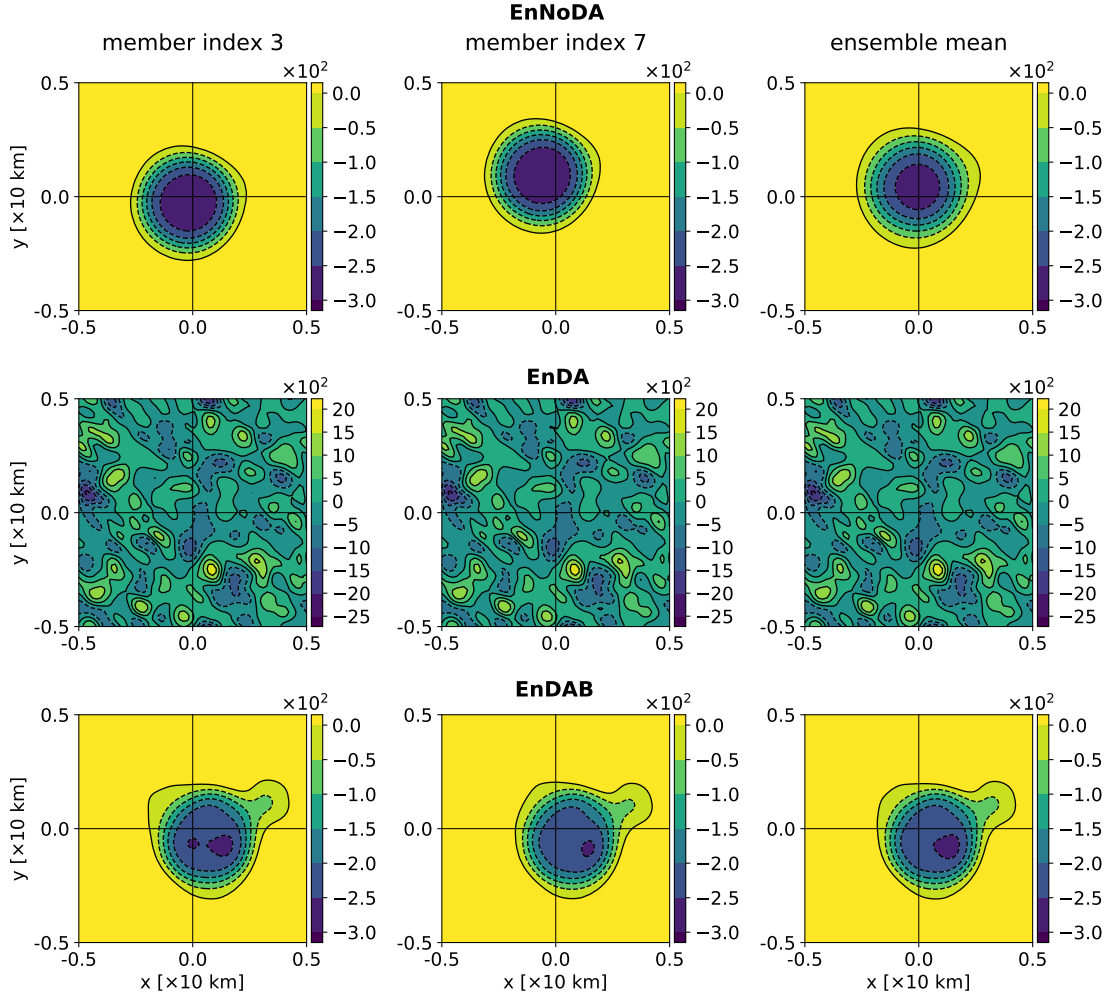


Figure 6.19: Euler vortex: snapshots of pressure perturbation p' . Ensemble members with index 3 (first column), 7 (second column), and ensemble mean (third column) at $t = 300.0$ s with all quantities $\{\rho, \rho u, \rho v, P, \pi'\}$ assimilated. Top row: EnNoDA run; contours in range $[-300.0, 0.0]$ Pa with a 50.0 Pa interval. Middle row: EnDA run; contours in range $[-2500.0, 2000.0]$ Pa with a 500.0 Pa interval. Bottom row: EnDAB run; contours in range $[-300.0, 0.0]$ Pa with a 50.0 Pa interval. Negative contours are dashed.

Figure 6.20 shows the spatial- and ensemble-averaged RMSE scores of the different ensemble configurations. The assimilation of the momentum field alone is insufficient for the Euler vortex test case (solid orange and solid green lines). For these ensemble runs, the errors of the quantities not assimilated, ρ , P , and π , increase after the first assimilation time at $t = 25.0$ s and remain consistently above that of the control EnNoDA run. For the quantities assimilated, ρu and ρv , the errors increase over time and are comparable to the EnNoDA run after three revolutions

of the vortex across the domain at $t = 300.0$ s. The poor performance of the data assimilation system is traced to the issue of *controllability* (Jazwinski, 1970): the dynamics of the Euler vortex involve a strong axisymmetric variation of the potential temperature field (cf. the lower right panel of Figure 6.14) that is not corrected by the assimilation of the momentum fields alone. Therefore, the correlation of the velocity and the potential temperature fields is destroyed as data is assimilated, and the solution becomes increasingly imbalanced. Nevertheless, the EnDAB run (solid green) presents an improvement over the plain EnDA run (solid orange), e.g. in the lower error scores in the P variable and in the smoother error profile over time for the assimilated momentum fields.

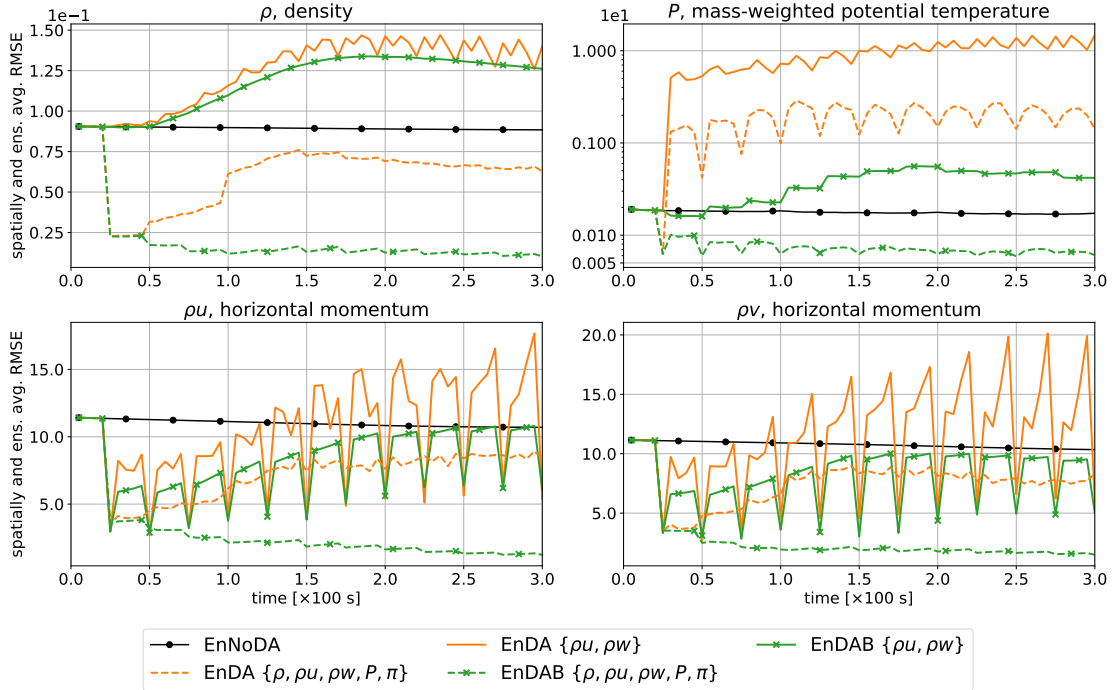


Figure 6.20: Euler vortex: EnNoDA run (dots, solid black), EnDA run (orange), EnDAB run (crosses, green). Assimilated quantities are ρu and ρv (solid lines) and ρ , ρu , ρv , P , π' (dashed lines). Spatially- and ensemble-averaged RMSE from $t = 0.0$ s to 300.0 s for density ρ (top left, $[\text{kg m}^{-2}]$), mass-weighted potential temperature P (top right log-linear scale, $[\text{kg m}^{-2} \text{K}]$), and momenta ρu , ρv (bottom left and right, $[\text{kg m}^{-1} \text{s}^{-1}]$). The RMSE of the initial ensemble is omitted.

The assimilation of all quantities circumvents the controllability issue. For the EnDA run (dashed orange line), the errors of the quantities ρ , ρu , and ρv decrease drastically after the first assimilation time and then increase gradually, but they remain below that of the control EnNoDA run. For the pressure-related P variable, the error jumps after the first assimilation time, and this corresponds to the imbalances introduced by the assimilation of data. This effect is quantified in Appendix B. The imbalances introduced do not increase or decrease over time, and this is reflected in the plateauing of the error in P .

The blended EnDAB run (dashed green line) recovers the balanced solution. After a drastic

decrease in the error after the first assimilation time that is visible for all quantities, the error either remains approximately of the same order of magnitude, e.g. in ρ and P , or decreases over time, e.g. in ρu and ρv . Note that this EnDAB run is the only run in which the error scores of the P variable are smaller than those of the control EnNoDA run with balanced solutions. Therefore, we not only obtain balanced solutions with the EnDAB run, but we also obtain an ensemble that is closer to the truth than that of the EnNoDA run. This corroborates the results depicted in the last row of Figure 6.19.

This set of results for the Euler vortex shows that the blending strategy is suitable to recover balanced solutions despite an untuned and suboptimal data assimilation scheme. However, the issue of controllability presents an additional problem for balanced data assimilation that is beyond the scope of the blending strategies developed in this thesis, and is therefore left for future work.

6.4.2 Rising bubble: DAB results

Snapshots of the pressure perturbation for two rising bubble ensemble members at the simulation end time $t_{\text{end}} = 1000.0$ s along with the ensemble mean are depicted in Figure 6.21. Recall that the ensemble is generated by randomly perturbing the initial potential temperature variation inside the bubble. As a result, the EnNoDA ensemble (top row) consists of bubbles that have risen to various heights at the simulation end time (bubbles with an initial temperature perturbation that is relatively warmer will rise faster).

The EnDA ensemble (second row from top) shows the presence of acoustic imbalances, and the features of the bubble rotors, captured by the regions of low pressure perturbation, are less distinct. Data assimilation appears effective in nudging the bubbles in the ensemble towards the truth, as the bubble rotors in the ensemble mean are not as spread out as in the EnNoDA case. The blended run, EnDAB (third row from top), recovers the balanced solutions and the acoustic imbalances have been eliminated. Furthermore, the bubble rotors in the ensemble mean is less diffused, which again points towards the effectiveness of data assimilation, and the ensemble converges. Finally, a difference is taken between the EnDA and the EnDAB results (bottom-most row), and this shows that the difference between these two solutions stems largely from the presence of acoustic imbalances. Data assimilation is just as effective in nudging the bubbles in both the ensembles, and this can be seen by the absence of the bubble rotor imprints in the difference of the ensemble means (bottom-most row, rightmost panel).

The results depicted in the snapshots of the ensemble members are corroborated by the RMSE profiles over time in Figure 6.22. That the data assimilation is effective in correcting the position of the bubbles in both the EnDA ensemble (solid orange line) and the EnDAB ensemble (dashed green line) can be seen in the RMSE plots for ρ , ρu , and ρw . With the first application of data assimilation at $t = 500.0$ s, the error scores drop below that of the control EnNoDA ensemble, and

6 Numerical results

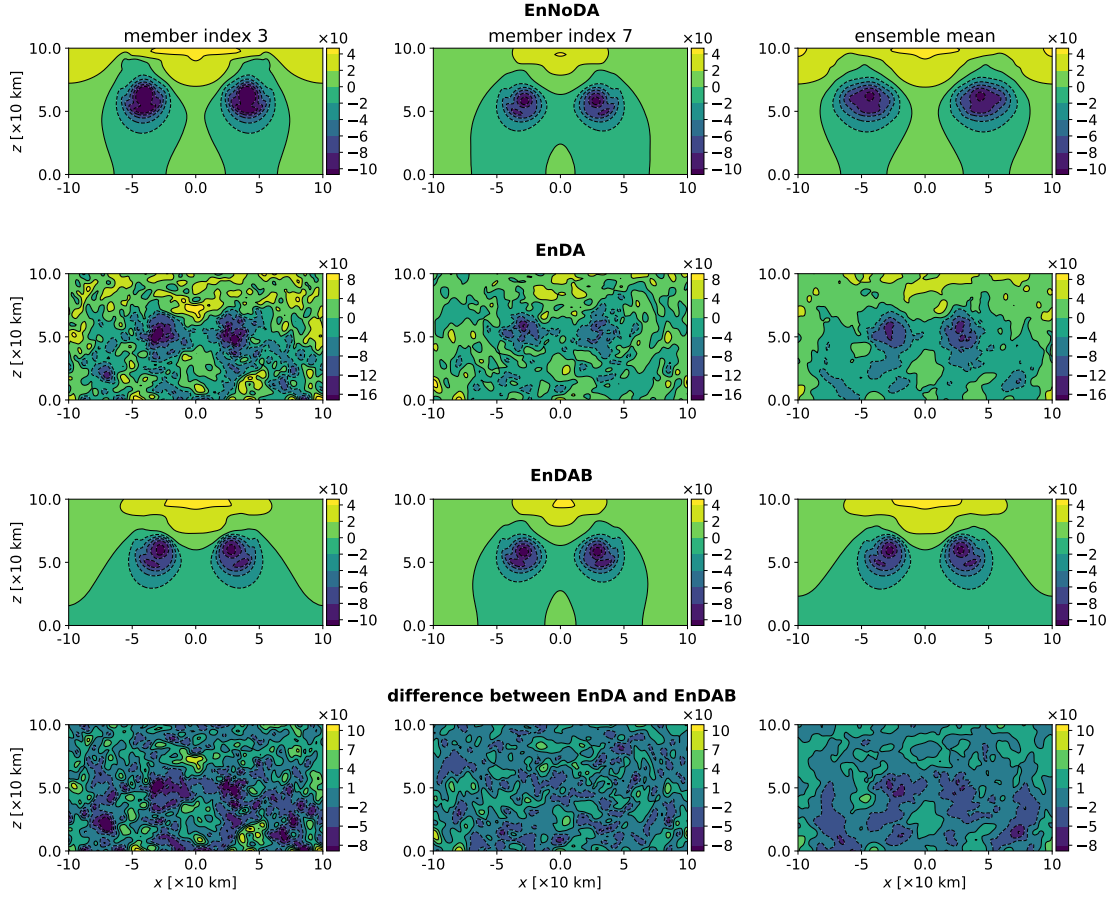


Figure 6.21: Rising bubble: snapshots of pressure perturbation p' . Quantities assimilated are ρu and ρw . Ensemble members with index 3 (first column) and 7 (second column) at $t = 1000.0$ s with the ensemble mean (third column). Row-wise: EnNoDA run, contours in range $[-100.0, 40.0]$ Pa with a 20.0 Pa interval (first row); EnDA run, contours in range $[-160.0, 80.0]$ Pa with a 40.0 Pa interval (second row); EnDAB run, contours in range $[-100.0, 40.0]$ Pa with a 20.0 Pa interval (third row); and difference between EnDA and EnDAB, contours in range $[-80.0, 100.0]$ Pa with a 30.0 Pa interval (fourth row). Negative contours are dashed.

the error in ρ continues to decrease over time. The effect of the imbalances present in the EnDA run can be seen in the slightly higher error in ρ and in the bumpy error profile over time when compared to the EnDAB run. The effect of the acoustic imbalances can be most drastically seen in the pressure-related quantity P . Here, the error of the EnDA ensemble jumps above that of the EnNoDA control run after the first assimilation time, and it remains of the same order of magnitude over time. As with the experiments before, this error jump characterises the imbalances introduced by data assimilation, and the effect is quantified in Appendix B. Finally, the balanced EnDAB run achieves an error score that is significantly lower than that of the EnNoDA run for all the quantities depicted in Figure 6.22.

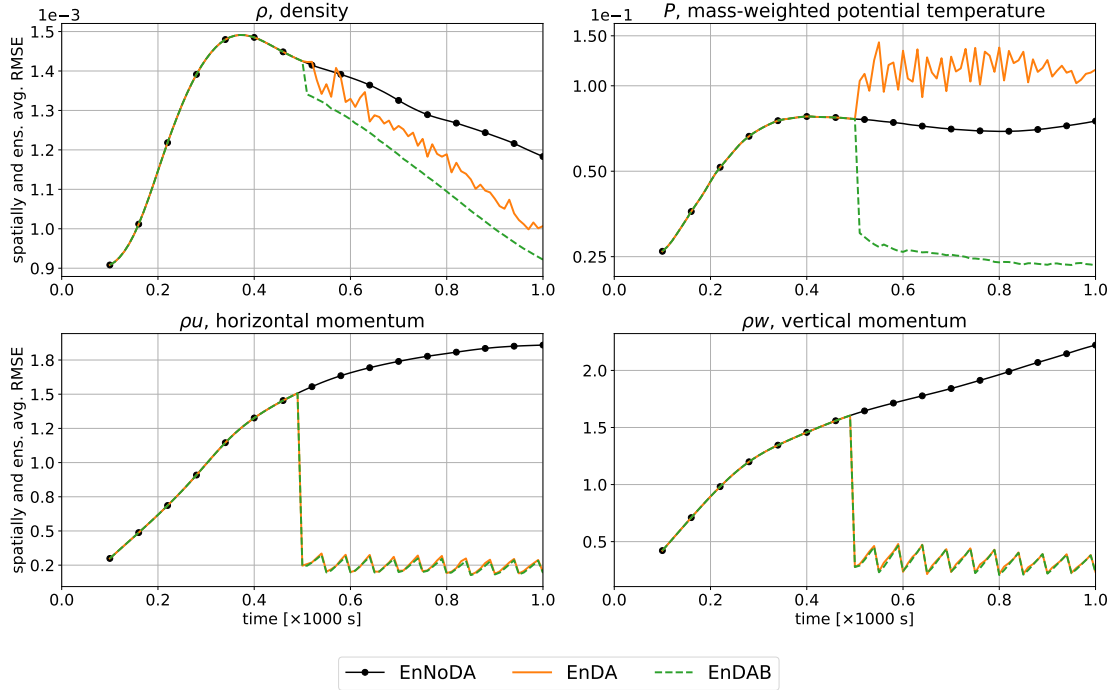


Figure 6.22: Rising bubble: EnNoDA run (dots, solid black), EnDA run (solid orange), EnDAB run (dashed green). Assimilated quantities are ρu and ρw . Spatially- and ensemble-averaged RMSE from $t = 100.0$ s to 1000.0 s for density ρ (top left, [kg m^{-2}]), mass-weighted potential temperature P (top right log-linear scale, [$\text{kg m}^{-2} \text{ K}$]), and momenta ρu , ρw (bottom left and right, [$\text{kg m}^{-1} \text{ s}^{-1}$]).

6.4.3 Inertia-gravity wave: DAB results

The ensemble snapshots of the vertical momentum field for two ensemble members of the inertia-gravity wave ensemble along with the ensemble mean are depicted in Figure 6.23. The snapshots are taken at the simulation end time, $t_{\text{end}} = 60\,000$ s, with all quantities assimilated. The ensemble spread can be seen in the amplitudes of the inertia-gravity wave pulses of the EnNoDA run (top row). For example, the amplitude of inertia-gravity wave pulses for ensemble member index 3 is larger than that of ensemble member index 7.

For the EnDA run (second row), the final amplitudes of the inertia-gravity wave pulses attain a value that is closer to the truth than those of the EnNoDA run (compare the amplitudes depicted in the colour bars to the one in the top right panel of Figure 6.16). On the other hand, presence of background noise is visible in the EnDA run, and the smooth and regular features in the background of the EnNoDA run are not retained here. The amplitudes of the inertia-gravity wave pulses for the EnDAB run (third row) attain a convergence similar to the EnDA run. Yet the background noise is successfully suppressed to an extent, and the background recovers, to a degree, the smooth and regular pattern seen in the background of the balanced EnNoDA run.

Finally, as in the rising bubble case, the difference between the EnDA and the EnDAB results is taken (bottom-most row). The absence of the imprints of the inertia-gravity wave pulses implies that data assimilation is as effective in nudging the ensemble towards the truth in both

6 Numerical results

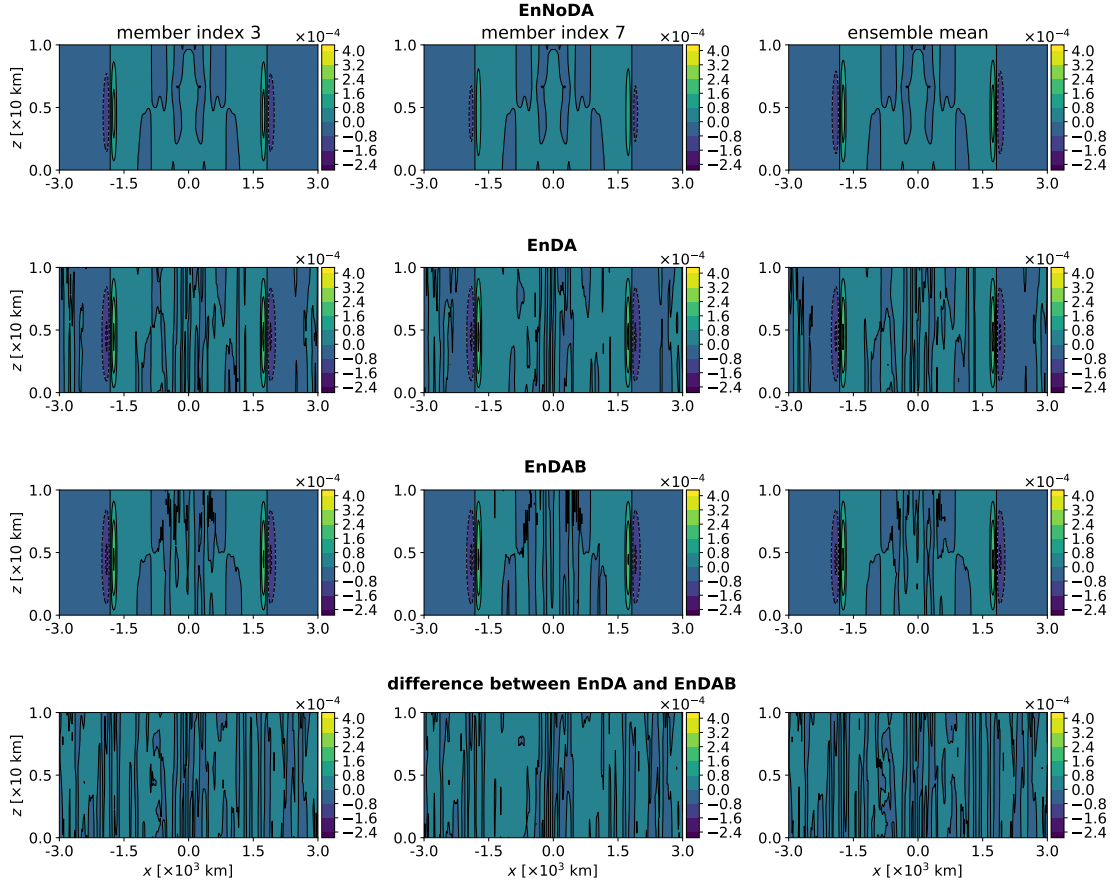


Figure 6.23: Inertia-gravity wave: snapshots of the vertical momentum ρw . Ensemble members with index 3 (first column) and 7 (second column) at $t = 60.0$ s with the ensemble mean (third column). Row-wise: EnNoDA run (first row); EnDA run (second row); EnDAB run (third row); and the difference between the EnDA and EnDAB runs (fourth row). For all rows, contours in range $[-2.4, 4.0] \times 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$ with a $0.8 \times 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$ interval. Negative contours are dashed.

cases. The difference in the solutions is largely in the background, and this suggests that the application of blending affects only the background noise, and blending does not alter the slower dynamics.

We note two further observations: 1) the full amplitude of the inertia-gravity pulses in the EnDA and the EnDAB ensembles slightly underestimates the amplitude of the truth, i.e., the minimum of the ensemble mean for the two ensembles is larger than that of the truth, and the maximum is smaller. 2) data assimilation introduces imbalances that are one order of magnitude smaller than the imbalance investigated in section 6.2.3. Here, the imbalances in the background are of the order $\sim 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$, but the artificial imbalances presented in Figure 6.12 are of order $\sim 10^{-3} \text{ kg m}^{-1} \text{ s}^{-1}$.

From section 6.2.3, we saw that the imbalances for the inertia-gravity wave take the form of a vertically propagating acoustic-gravity wave pulse in the centre of the domain. This is reflected in the RMSE profile of the vertical momentum in Figure 6.24. The error profile oscillates over

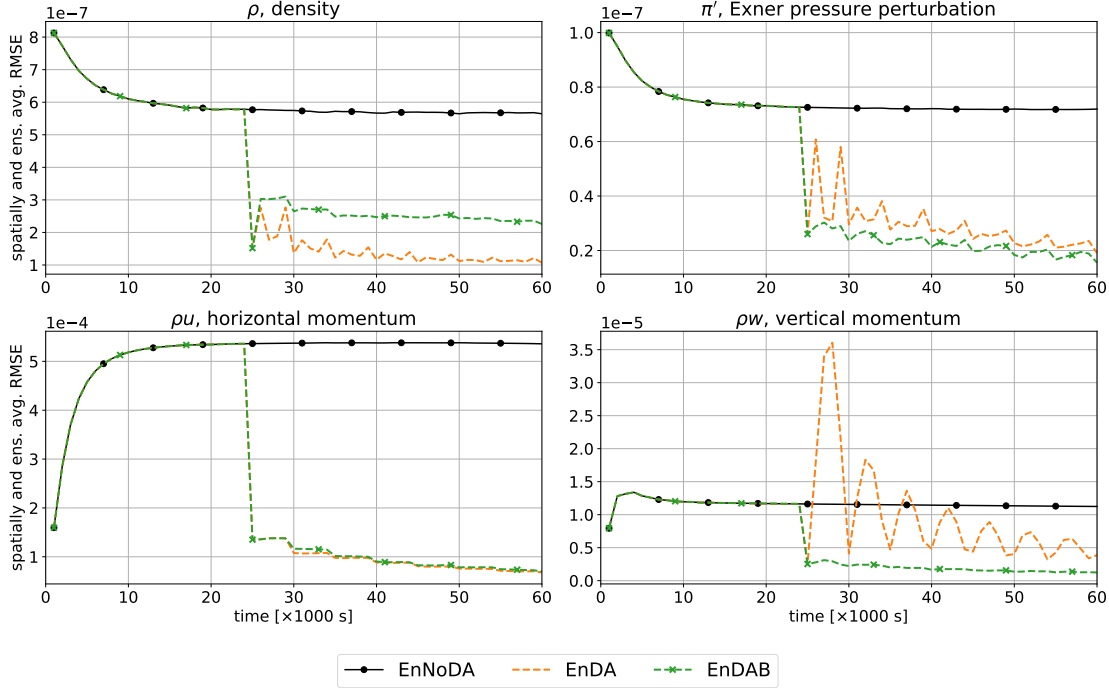


Figure 6.24: Inertia-gravity wave: EnNoDA run (dots, solid black), EnDA run (dashed orange), EnDAB run (crosses, dashed green). Assimilated quantities are $\{\rho, \rho u, \rho w, P, \pi'\}$. Spatially- and ensemble-averaged RMSE from $t = 0.0$ s to 60 000.0 s for density ρ (top left, [kg m^{-2}]), Exner pressure perturbation π' (top right, dimensionless), and momenta $\rho u, \rho w$ (bottom left and right, [$\text{kg m}^{-1} \text{s}^{-1}$]).

time for the EnDA run (dashed orange line) due to the imbalances introduced, although the error score ultimately decreases to below that of the EnNoDA run (dots, solid black line) as the inertia-gravity wave pulses are nudged towards the truth by the assimilation of data. This corroborates the results depicted in Figure 6.23. The imbalances introduced in the EnDA run have an effect on the density ρ and the Exner pressure perturbation π' , and the error profiles for these quantities exhibit a few kinks over time. We note that, by the end of the simulation, the error scores of the EnDA run are lower than those of the control EnNoDA run for all the quantities depicted.

The EnDAB run (dashed green line) successfully suppresses the vertically propagating acoustic, and this is visible in the smooth error profile for the vertical momentum ρw , with the error score of the EnDAB being the best of all three runs. The more balanced solutions of the EnDAB ensemble appear to have an effect on the density and Exner pressure perturbation. The error profiles of the EnDAB run for these quantities are smoother than those of the EnDA run. However, the error score of the density field is higher for the EnDAB run than that of the EnDA run.

These results suggest that the blending strategy for the hydrostatic and nonhydrostatic blending is rather successful in eliminating the imbalances introduced via data assimilation. However, further investigation is warranted on the cause of the error in the density field for the blended run. Such an investigation will be the subject of future work.

7 Conclusion

The blending strategies developed in this thesis present a novel dynamics-driven method to achieve balanced data initialisation and balanced data assimilation with a blended numerical model. The idea behind the blending strategy is conceptually rather simple, and it answers the following question: given a blended numerical model with seamless access to the compressible Euler equations and its limit regimes, how can we best ensure that the solution stays true to the underlying dynamics when switching between model regimes within a single simulation run? Yet, as we have seen in chapter 4 on deriving the blending strategies, understanding and then overcoming the subtle differences in the models, the dynamics, and the numerics can be a difficult and challenging task.

To fully appreciate and test the capabilities of the blending strategies, we started in chapter 2 with a discussion of the various dynamical models, and with a derivation of the shallow water and lake equations from the compressible flow equations. Having acquired intuition behind the physics of the models, chapter 3 introduced the blended numerical model. Each full time update invokes a finite volume solver and an elliptic solver for problems with non-constant coefficients, and the solution of each substep is carefully tied together by the midpoint and the trapezoidal updates to ensure an overall second-order scheme. To develop the one-step blending strategies in chapter 4, numerical and asymptotic analyses were applied to the blended numerical model, and strategies to convert the solutions between model regimes were derived. Strategies to blend between the Euler equations and the limit soundproof and hydrostatic regimes, as well as between the shallow water equations and the lake equations, have been introduced in this thesis. The blending strategies ensure that, when converted from one model configuration to another, the solutions obey the dynamics of the respective model regime. Bayesian ensemble data assimilation, and in particular the LETKF algorithm, was discussed in chapter 5. Finally, the principal capabilities of the one-step blending strategies were tested in chapter 6.

Firstly, the blending strategies were applied to obtain balanced data initialisation in idealised tests. The results presented in chapter 6 on the principal capabilities of the blending strategies are encouraging. To achieve balanced data initialisation, blending was applied to the first time step. For the SWE and Euler vortex tests, the runs with blending achieved two orders of magnitude improvement in the error score compared to a run without balancing. For the rising bubble test, the blended run with a large advective CFL-constrained time step achieved more than one order of magnitude improvement in the error score compared to a run without balancing, and the blended run with a small time step achieved more than two orders of magnitude improvement. These scores for the rising bubble test are a significant improvement over the scores of the balanced data initialisation experiments in Benacchio et al. (2014) who, in addition, needed several time

7 Conclusion

steps in a hybrid compressible–soundproof configuration with non-integer values of the blending parameter to achieve their best reduction in the noise level. Moreover, the strategy of blending via multiple time steps in the hybrid compressible–soundproof regime is not discussed in this thesis for two reasons. First, Hastermann et al. (2021) showed that the intermediate model regimes in the context of highly oscillatory Hamiltonian ODE systems judiciously dissipate the unwanted fast-mode noise, but the role of such intermediate models is still not well understood for the Euler equations. The application of this method would therefore not be dynamics-driven. Second, the blended numerical model does not feature numerical dissipation, and this strategy of dissipating the fast-mode noise by intermediate models is less efficient in achieving balanced solutions.

Secondly, a novel nonhydrostatic–hydrostatic blending strategy was derived, and the effectiveness of this blending strategy was investigated with the inertia-gravity wave test case. In the balanced data initialisation experiments, the compressible–soundproof blended run achieved a three orders-of-magnitude improvement in the error scores over the imbalanced solution, and the nonhydrostatic–hydrostatic blended run achieved a two orders-of-magnitude improvement. These encouraging results demonstrated that in the presence of vertically propagating acoustic wave and for a setup in which both the pseudo-incompressible and the hydrostatic limit regimes work reasonably well, the two blending strategies were effective in suppressing imbalances arising from the initial data.

Thirdly, the one-step blending strategies were used to achieve balanced data assimilation. In the experiments with blending and data assimilation, the blending strategies successfully suppressed imbalances arising from the assimilation procedure in the shallow water vortex and the rising bubble test cases. For these two test cases, the application of the balancing method yielded the best error scores for all the quantities investigated. The characteristic error jump in the pressure-related variables, arising from the imbalances introduced by data assimilation and quantified in Appendix B, was also not present for these blended runs. Furthermore, the results demonstrated that, in line with the theory, the relevant slower dynamics of the rising bubble were not affected by the application of the blending strategy. Only the fast-mode imbalances were eliminated.

In experiments with the Euler vortex, when the potential temperature field was not assimilated, the issue of controllability dominated (see section 6.4.1 for more details), and the data assimilation system failed to correctly reproduce the dynamics of the vortex. In this scenario, the application of blending as a balancing method was insufficient to recover the balanced dynamics. However, by assimilating all the quantities, data assimilation with blending recovered the balanced solution. Furthermore and in agreement with these Euler vortex results, blended data assimilation with the momentum fields alone is sufficient to recover the balanced solution for the shallow water vortex case where the potential temperature distribution is a constant.

Nonhydrostatic–hydrostatic blending was investigated with data assimilation for the inertia-

gravity wave test, and promising results were visible in the error scores of the Exner pressure perturbation and the vertical momentum fields. For these two quantities, the run with blending yielded the lowest error scores, and the error profile was largely free of kinks or oscillations. These results suggest that the solution is indeed balanced for these two fields. However for the density field, the blended run had a larger error score than that of the run with a plain data assimilation scheme, and further studies are warranted to explain this observation. Nevertheless, by the end of the simulation, both runs with data assimilation were able to attain lower error scores than those of the control ensemble without data assimilation for all quantities investigated.

In all the data assimilation experiments above, no effort has been made to tune the data assimilation scheme. It is a testimony of the blending strategies' effectiveness that they work in spite of an untuned data assimilation scheme. More details on what happens in the scenario of more-optimal data assimilation schemes are provided in the appendices at the end of this thesis. In particular, results from appendices A, D, and E suggest that combining a tuned data assimilation scheme with the blending balancing method yields the best error scores that are not achieved by the tuning of the data assimilation parameters alone.

7.1 Overview of the appendices

Chapter 6 established the principal capabilities of the blending strategies. The appendices at the end of this thesis answer additional questions regarding the setup and consider several 'what-if' scenarios. The list below provides a summary of the content for each of the appendices.

Appendix A: The computation detail behind the variances used in generating the observation noise is elaborated upon, and the variances are listed in Table A.1 and A.2. A study of the effect of different variance choices on the error scores is then provided for the Euler vortex, and the results show that because the observations are close to the truth, strong assimilation of data is favoured for this test case.

Appendix B: A scale analysis argument is used to quantify the imbalances introduced by the data assimilation procedure for the shallow water-lake blending and the compressible-soundproof blending. Results from scale analysis reproduce the error jump visible in the water depth perturbation field for the shallow water vortex and in the mass-weighted potential temperature field for the Euler vortex and the rising bubble.

Appendix C: In a data assimilation experiment, the RMSE should be comparable to the ensemble spread (Fortin et al., 2014). Therefore, the ensemble spread of the travelling vortices and the rising bubble are provided here for comparison. In the case where all quantities are assimilated for the Euler vortex, the RMSE and the ensemble spread are not comparable, and an argument is provided to explain this observation which further supports the motivation behind developing the balancing techniques in this thesis.

Appendix D: Longer simulations are provided for the Euler vortex data assimilation experiment with all quantities assimilated. Notions behind the convergence of the data assimilation system are explored for these simulations, and three conclusions can be drawn. First, the error scores of a compressible run with data assimilation and blending is comparable to a pseudo-incompressible run with plain data assimilation alone; second, the data assimilation system converges when compared to a control ensemble run with imbalanced solutions; third, with a more favourable or optimal data assimilation setup, the data assimilation system converges when compared to a control ensemble run with balanced solutions.

Appendix E: Localisation is a source of imbalances arising from the data assimilation procedure. Here, the effect of the localisation region size on the balancedness of the solution is explored. For the idealised Euler vortex test, data assimilation runs without blending favour a moderate localisation region of approximately a quarter to half of the vortex diameter, and data assimilation runs with blending favour a relatively smaller localisation region.

7.2 Outlook and future work

Unphysical acoustic imbalances are a problem in numerical weather prediction, and various methods have been explored to damp or remove these imbalances (e.g. Daley, 1988; Skamarock and Klemp, 1992; Dudhia, 1995; Klemp et al., 2018). More generally, the assimilation of data in the compressible flow equations may excite all fast-mode oscillations, and various filtering techniques have been developed to eliminate unphysical imbalances (Ha et al., 2017). As such, the blending strategies developed in this thesis are a substantial improvement in this direction. The results demonstrated that the blending strategies were able to eliminate acoustic imbalances and recover balanced analysis fields for the idealised test cases studied here. Furthermore, the blending strategies developed here appear to be the first balancing method for the compressible Euler equations that is cleanly dynamics-conforming.

The research developed in this thesis suggests several avenues for improvement, further investigation, and new development. A few of the possibilities are listed as follows.

First, the choice of an untuned data assimilation method used in this thesis has been substantiated, although admittedly, the quality of the data assimilation method can be improved by tuning its parameters. Therefore, a prudent next step will be a study along the lines of Popov and Sandu (2019) on the multivariate tuning of the LETKF and the localisation parameters. Such a study should also include the comparison of the blending method with established methods of achieving balanced data assimilation, e.g. by extending the comparison study between the DFI and IAU in Polavarapu et al. (2004). Optimisation of the IAU, e.g. following Lei and Whitaker (2016) and He et al. (2020), may also have to be carried out. Finally, as the DFI and

IAU act as low-pass filters (Houtekamer and Zhang, 2016; Polavarapu et al., 2004), a detailed investigation of the impact of the various balancing methods on the slow dynamics will be of particular interest, and this investigation will establish the necessity of a dynamics-driven balancing method.

Second, future work may focus on a more comprehensive and careful treatment of the arguments and investigations presented in this thesis. In particular, the blending strategies in Chapter 4 are rather numerically focused and are based heavily on the discretisations detailed in Chapter 3. However, the underlying mathematical structure of the continuum formulations may be better explicated via the index of an algebraic differential equation or bounded derivative principles. These mathematical concepts would help link the blending strategies to existing work such as Hastermann et al. (2021). Furthermore, an in-depth study exploring the effects of data assimilation on the Euler travelling vortex dynamics may be conducted. Of interest may be the travelling vortex with non-zero Coriolis force, e.g. with the β -plane approximation, and in the presence of Rossby and Lamb wave.

Third, in order to progress beyond the idealised test cases explored in this thesis, the functionality of the blended numerical model implemented in the Python code has to be extended substantially. A natural extension of the existing code will be towards support for three-dimensional moist dynamics with bottom topography (Durran and Klemp, 1983; O’Neill and Klein, 2014; Duarte et al., 2015). Furthermore, the Python code could be better refactored, optimised, and documented in line with good scientific programming practices (Wilson et al., 2014, 2017). Full parallelisation of the code will be a worthy endeavour especially for the simulation of three-dimensional problems, and an easy next step will be to tackle the embarrassingly parallel problem of the ensemble simulation.

In addition, more complex idealised test cases may be constructed to further test the fidelity of the blending strategies to the underlying dynamics. An example would be the superposition of a Lamb wave with the imbalanced inertia-gravity wave. Blending to the hydrostatic regime should balance the inertia-gravity wave pulses without affecting the dynamics of the Lamb wave, and blending to the soundproof regime should eliminate the Lamb wave without affecting the dynamics of the superimposed inertia-gravity wave. Such a test would establish blending as an effective method to selectively eliminate unphysical dynamical phenomena.

Beyond the blending strategies developed here, the theoretical framework developed in Klein and Benacchio (2016) included the unified model by Arakawa and Konor (2009) as one of the reduced models. Therefore, with the appropriate extension to the blended numerical model, a blending strategy between the compressible equations and the Arakawa-Konor limit model may be developed. Such a blending strategy will allow for the filtering of smaller-scale acoustic noise while leaving the Lamb wave components dynamically unaffected. Recent development in this area can be found in Qaddouri et al. (2021) who introduced a numerical model that switches between a variant of the compressible Euler equations and the Arakawa-Konor unified model.

7 Conclusion

Furthermore, with an extension of the blended numerical model to support the appropriate limit regimes, blending strategies can be developed to filter other unphysical noises, e.g., topographic gravity wave noise (Zarzycki and Jablonowski, 2015) with an appropriate small Strouhal number limit regime.

An open question arising from the extension of the ideas presented in this thesis towards real-world numerical weather prediction is whether there will be a significant benefit in using the compressible equations for making predictions between observations. Suppose we can (a) formulate appropriate limit regimes that selectively eliminate undesired components of our solution while keeping the physically significant components, and (b) we can build the corresponding blending switches and strategies. Then the results in this thesis demonstrate that we can control which components of the solution to eliminate or keep through blending between the full and limit regimes. Verifying this for complicated real-world applications will be a crucial next step. As mentioned above, the compressible equations are currently our best description of the underlying dynamics. Therefore, the ability to judiciously select for physically relevant components of the simulation solution may provide substantial advantages.

Apart from extending the blended numerical model towards real-world applications relevant to numerical weather prediction, similar blending strategies may be developed for existing dynamical cores. The blending strategies rely upon a semi-implicit discretisation of the compressible Euler equations, and such a discretisation is used in various dynamical cores by weather centres worldwide, e.g. the operational hydrostatic IFS spectral transform model in use at the European Centre for Medium Range Weather Forecasts (ECMWF, Wedi et al., 2013), and the Met Office’s Unified Model (Davies et al., 2005; Wood et al., 2014), which has a hydrostatic / nonhydrostatic switch.

Of particular interest is the nonhydrostatic compressible IFS-FVM dynamical core at the ECMWF (Kühnlein et al., 2019). The numerical discretisation behind this dynamical core is similar to the blended numerical model used in this thesis, and therefore the IFS-FVM would be an ideal candidate for the implementation of the blending strategies in a next-generation semi-operational model. The blending strategies developed here may also be of particular relevance to operational compressible models that are able to selectively employ the dynamics of a limit model (Wood et al., 2014; Melvin et al., 2019; Voitius et al., 2019; Qaddouri et al., 2021).

The positive results presented in this thesis provide promising insights into the potential of blending strategies that make use of multimodel numerics to switch between model regimes. The impact will be considerable if these results can be replicated for real-world numerical weather predictions.

Appendix A

Modelling the observation error

For the data assimilation experiments in section 6.3, the observation noise is drawn from a zero-mean Gaussian distribution. The variance is computed as 5% of the variance of the sparsely observed field averaged over all observation time. Specifically, details on the computation of the variances are provided below.

For a sparsely observed quantity without measurement noise $\mathcal{Y}_{\text{obs}}^n$ at observation time t_{obs}^n , we compute the variance

$$(\sigma_{y_{\text{obs}}}^n)^2 = \frac{1}{N_{\text{obs}}} \sum_{s=1}^{N_{\text{obs}}} (\mathcal{Y}_{\text{obs}}^n(x, y)_s - \bar{\mathcal{Y}}_{\text{obs}}^n)^2, \quad (\text{A.1})$$

where s indexes the number of sparse observations N_{obs} , and $(x, y)_s$ represents the (x, y) spatial point for the s -th sparse observation. Here, $\bar{\mathcal{Y}}_{\text{obs}}^n$ is the mean of the sparsely observed field $\mathcal{Y}_{\text{obs}}^n$ at observation time t_{obs}^n ,

$$\bar{\mathcal{Y}}_{\text{obs}}^n = \frac{1}{N_{\text{obs}}} \sum_{s=1}^{N_{\text{obs}}} \mathcal{Y}_{\text{obs}}^n(x, y)_s. \quad (\text{A.2})$$

This gives us the variance $(\sigma_{y_{\text{obs}}}^n)^2$ for each sparsely observed quantity at each observation time $\{t_{\text{obs}}^1, t_{\text{obs}}^2, \dots, t_{\text{obs}}^N\}$. Finally, the time-average of 5% of the variance for each of the observed quantities is taken, i.e.,

$$\bar{\sigma}_{y_{\text{obs}}}^2 = \frac{1}{N} \sum_{n=1}^N 0.05 (\sigma_{y_{\text{obs}}}^n)^2. \quad (\text{A.3})$$

The variances computed via the method outlined above are displayed in Table A.1 for the SWE vortex test case and in A.2 for the rest of the experiments with the Euler equations.

For the rest of this appendix, we investigate the effect of the observation error covariance on the results of the Euler travelling vortex experiment with data assimilation. This is done by

Table A.1: Values of the error variance in the observations of the SWE vortex test case, computed as 5% of the variance of the sparsely observed fields averaged over all observation time.

Quantity	Variance
	SWE Vortex
Water depth h [m]	-
Horizontal momentum (hu) [$\text{m}^2 \text{s}^{-1}$]	1.9×10^{-2}
Horizontal momentum (hv) [$\text{m}^2 \text{s}^{-1}$]	1.8×10^{-2}
Water depth perturbation (h') [m]	-

Table A.2: Values of the error variance in the observations of the test cases involving the Euler equations, computed as 5% of the variance of the sparsely observed fields averaged over all observation time.

Quantity	Variance		
	Vortex	Bubble	IGWave
Density (ρ) [kg m ⁻²]	4.5×10^{-4}	-	2.0×10^{-3}
Horizontal momentum (ρu) [kg m ⁻¹ s ⁻¹]	0.064	0.022	2.1×10^{-9}
Horizontal momentum (ρv) [kg m ⁻¹ s ⁻¹]	0.064	-	-
Vertical momentum (ρw) [kg m ⁻¹ s ⁻¹]	-	0.032	9.3×10^{-13}
Mass-weight potential temperature (P) [kg m ⁻² K]	6.7×10^{-6}	-	0.52
Exner pressure perturbation (π') [dimensionless]	2.4×10^{-7}	-	3.6×10^{-13}

varying the choice of the variances used to generate the observation noise, and three sets of such variances are investigated below.

The deterministic method of modelling the variance for the observation noise detailed above assumes no prior knowledge of the observed dynamics. The idealised experiments in the manuscript have a relatively small region where interesting dynamics is happening, e.g., within the vortex structure or the rising bubble, and a larger constant background region. As such, the mean of an observed field is close to the background state in the experiments, and observations of the background are inadvertently close to this mean. This skews the computed variance towards a value that is smaller than a variance computed by considering only the dynamics within the regions of interest. We label this set of variances computed by the deterministic method outlined above as DET.

Another choice of the set of variances could be guided by the dynamics of the observed field. Considering the structure of the travelling Euler vortex, we empirically choose variances that are approximately 10% of the typical values in the centre of the vortex. We label this empirical set of variances as EMP.

Finally, as the difference between the variances in DET and EMP is large, we compute a third set of variances by taking the logarithmic middle point between the DET and EMP sets of variances. Specifically, for a variance σ_{DET} or σ_{EMP} in the set of variances DET or EMP respectively, we compute the logarithmic middle point

$$\log(\sigma_{\text{MID}}) = \frac{1}{2} [\log(\sigma_{\text{DET}}) + \log(\sigma_{\text{EMP}})], \quad (\text{A.4})$$

and σ_{MID} is recovered. We label this set of variances, obtained as the logarithmic middle point between DET and EMP, as MID. The three set of variances, DET, MID, and EMP, are listed in Table A.3.

Repeating the experiments in Figure 6.20 for all the quantities assimilated (i.e., the dashed

Table A.3: Variances in DET, MID, and EMP.

Quantity	Variance		
	DET	MID	EMP
Density (ρ) [kg m^{-2}]	4.5×10^{-4}	5.1×10^{-3}	5.8×10^{-2}
Horizontal momentum (ρu) [$\text{kg m}^{-1} \text{s}^{-1}$]	6.4×10^{-2}	6.9×10^{-1}	7.5
Horizontal momentum (ρv) [$\text{kg m}^{-1} \text{s}^{-1}$]	6.4×10^{-2}	6.9×10^{-1}	7.5
Mass-weight potential temperature (P) [$\text{kg m}^{-2} \text{K}$]	6.7×10^{-6}	1.1×10^{-3}	1.7×10^{-1}
Exner pressure perturbation (π') [dimensionless]	2.4×10^{-7}	1.4×10^{-5}	8.0×10^{-4}

curves) with these sets of variances yields the results in Figure A.1. We note that the DET results (blue lines with square markers) correspond to the dashed curves in Figure 6.20.

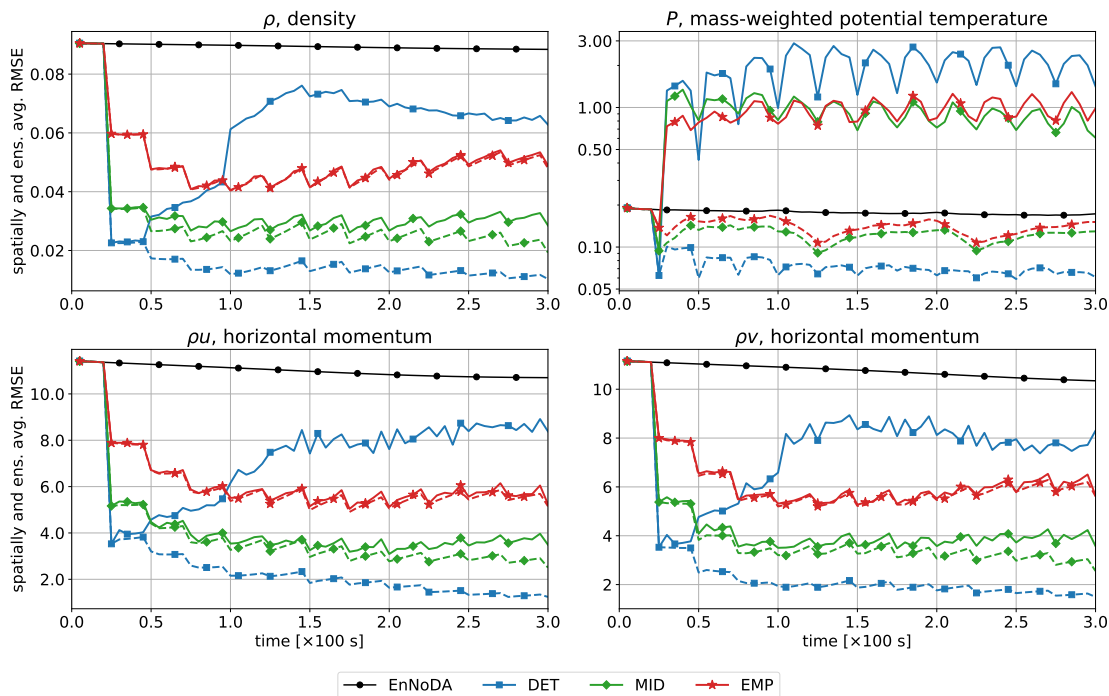


Figure A.1: Euler vortex: EnNoDA run (solid dotted, black), runs with observation error covariances DET (blue squares), MID (green diamonds), and EMP (red stars) given in Table A.3; EnDA runs (solid curves) and EnDAB runs (dashed curves). Assimilated quantities are ρ , ρu , ρv , P , π' . RMSE from $t = 0.0$ s to 300.0 s for density ρ (top left, [kg m^{-2}]), mass-weighted potential temperature P (top right log-linear scale, [$\text{kg m}^{-2} \text{K}$]), and momenta ρu , ρv (bottom left and right, [$\text{kg m}^{-1} \text{s}^{-1}$]). The RMSE of the initial ensemble is omitted.

After the first assimilation time at $t = 25.0$ s, runs with smaller observation error covariance experience more severe nudging of the vortex structure towards the truth. This can be seen in the magnitude of the initial decrease in the RMSE. However, as the observations assimilated are sparse and noisy, a drastic nudging of the ensemble towards the observations destroys the balanced vortex structure. This is exhibited in the DET EnDA run (solid blue curve), where

the error increases after the initial assimilation step. On the other hand, the DET EnDAB run (dashed blue curve) is able to cope with the drastic assimilation of data, and this run yields the best error scores of all runs.

For the more moderate nudging of the vortex due to comparatively larger observation error covariances, the RMSEs of the MID EnDA (solid green) and EMP EnDA (solid red) runs do not increase significantly after the first assimilation time. In each of these cases, the corresponding EnDAB run (dashed green and dashed red curves) performs better, albeit only marginally for the quantities ρ , ρu , and ρv in the EMP case.

In all cases, runs with blending (EnDAB) are able to remove the imbalances in the pressure-related P variable, and the error jump quantified in Appendix B is absent for these runs. Furthermore, the RMSE of all the EnDAB runs remain below that of the control EnNoDA ensemble.

The results suggest the following for the experiments with idealised tests. In the scenario that the observations are close to the truth, drastic assimilation of the observations may be beneficial in that it nudges the ensemble closer towards the truth. However, too drastic an assimilation can also be detrimental in that the procedure destroys the solution quality through imbalances introduced. Application of the blending method developed here allows us to circumvent this issue. With blending, the data assimilation procedure is more robust with regards to strong assimilation of data.

The DET case has been consciously employed in the data assimilation experiments of this thesis to highlight the effectiveness of the balancing strategies even when the results may be sub-optimal with a plain data assimilation setup alone.

Appendix B

Scale analysis of the data assimilation error in the pressure-related fields

Figures 6.18, 6.20, and 6.22 show that data assimilation introduces an error jump in the water depth perturbation h' or in the pressure-related P variable. This increase in the error occurs after the first assimilation time and remains of the same order of magnitude for the duration of the simulation. It quantifies the imbalance introduced by data assimilation. The following scale analysis corroborates this, and it is based on a similar scale analysis by Klein et al. (2001).

The assimilation of the momentum fields results in a change in the divergence of the velocity fields,

$$\nabla \cdot (\delta \mathbf{v}) = \frac{\partial \delta u}{\partial x} + \frac{\partial \delta v}{\partial y}, \quad (\text{B.1})$$

where $(\delta u, \delta v)$ are the changes in the velocity fields due to the assimilation of the horizontal momentum fields. Equation (B.1) has the units $[\text{s}^{-1}]$. The workings here can easily be adapted for the vertical slice rising bubble experiment involving the momentum fields ρu and ρw .

On the other hand, observe from the plots of the ensemble members and mean in Figures 6.17, 6.19, and 6.21 that the imbalances introduced are fast-mode acoustic or shallow water waves. These fast-mode imbalances are modelled as waves oscillating with peak amplitude right after the assimilation of data at the grid point under analysis. Introducing the acoustic timescale as t_{ac} , assume that the amplitude approaches zero after time t_{ac} . If the magnitude of the resulting acoustic waves is due to these oscillating waves, the maximum possible contribution to the magnitude from this setup corresponds to the period of the wave with positive amplitude.

Specifically, for an oscillating wave with amplitude arising from $\nabla \cdot (\delta \mathbf{v})_{(i,j)}$, where the subscript (i,j) denotes the grid point at (x_i, y_j) under analysis,

$$\begin{aligned} (\nabla \cdot \delta \mathbf{v})_{(i,j)} \int_0^{t_{\text{ac}}} \cos\left(\frac{\pi}{2} \frac{t}{t_{\text{ac}}}\right) dt &= \frac{2t_{\text{ac}}}{\pi} (\nabla \cdot \delta \mathbf{v})_{(i,j)} \left[\int_0^{\pi/2} \cos(\xi) d\xi \right] \\ &= \frac{2t_{\text{ac}}}{\pi} (\nabla \cdot \delta \mathbf{v})_{(i,j)} [\sin(\xi)]_0^{\pi/2} \\ &= \frac{2t_{\text{ac}}}{\pi} (\nabla \cdot \delta \mathbf{v})_{(i,j)}, \end{aligned} \quad (\text{B.2})$$

where $\xi = (\pi t)/(2t_{\text{ac}})$ is a change of variable. This yields the maximum possible magnitude of the fast-mode acoustic wave that characterises the imbalance.

To avoid underestimation, the acoustic timescale t_{ac} is chosen as the timescale of the largest

perturbations introduced. In the experiments, this corresponds to the (11×11) localisation region. As the analysis grid point is at the centre of this local region, a factor of $1/2$ is introduced,

$$t_{\text{ac}} = \frac{11}{2} \frac{dx}{c_{\text{ref}}}, \quad (\text{B.3})$$

where c_{ref} is the speed of sound, and dx is the grid-size which is a constant for the equidistant grids used in the travelling vortex and the rising bubble experiments. Equation (B.3) has units [s] and (B.2) is dimensionless.

In order to obtain an estimate for the contribution to the pressure from $\nabla \cdot (\delta \mathbf{v})_{(i,j)}$, recall that $p = \rho c_{\text{ref}}^2$. Multiplying this with (B.2),

$$\frac{2t_{\text{ac}}}{\pi} \nabla \cdot (\delta \mathbf{v})_{(i,j)} \rho_{(i,j)} c_{\text{ref}}^2 \sim \hat{p}_{(i,j)}, \quad (\text{B.4})$$

which has the units of [Pa]. The hat $\hat{}$ signifies that the quantity is obtained from the scale analysis of $\nabla \cdot (\delta \mathbf{v})$. Finally, we use the equation of state (2.2) to obtain an estimate for \hat{P} , the mass-weighted potential temperature computed via scale analysis.

Applying a similar argument for the shallow water equations, we arrive at

$$\frac{2t_{\text{ac}}}{\pi} \nabla \cdot (\delta \mathbf{v})_{(i,j)} h_{(i,j)} \sim \hat{h}'_{(i,j)}, \quad (\text{B.5})$$

where

$$t_{\text{ac}} = \frac{11}{2} \frac{dx}{\sqrt{gh_{\text{ref}}}}, \quad (\text{B.6})$$

with $\sqrt{gh_{\text{ref}}}$ being the shallow water speed.

The scale analysis is applied to each ensemble member and over the whole domain. In order to obtain a quantity that is comparable to the RMSEs in Figures 6.18, 6.20, and 6.22, the norm is taken for $\hat{\psi}$, where $\hat{\psi} = \hat{P}$ for the experiments with the Euler equations, and $\hat{\psi} = \hat{h}'$ for the shallow water vortex test. The norm is given by

$$|\hat{\psi}| = \frac{1}{K} \sum_k \left[\sqrt{\frac{1}{N_x \times N_y} \sum_{i,j}^{N_x, N_y} \left(\hat{\psi}_{(i,j)} \right)^2} \right]_k, \quad (\text{B.7})$$

where k indexes the K ensemble members and N_x and N_y are the number of grid-points in the x and y coordinates (or (x, z) for the rising bubble test case).

Figure B.1 shows the results of the scale analysis for the SWE vortex test case and Figure B.2 for the Euler vortex and the rising bubble test cases. Note that only contributions to h' or P from $\nabla \cdot (\delta \mathbf{v})$ to the flow fields are shown, i.e. outputs right after the assimilation of data and before the solution of the flow fields are omitted. Comparing the figures to the h' or P RMSE plot of the respective test cases, scale analysis yields results for \hat{h}' and \hat{P} that are of the same order of magnitude as the jumps in the RMSE plots with a similar profile over time. The

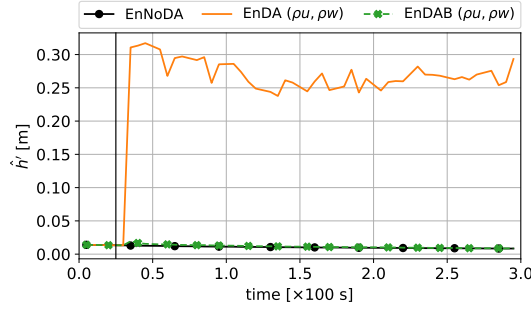


Figure B.1: Scale analysis of the contribution to water depth perturbation \hat{h}' from the divergence of the velocity fields for the SWE vortex ensemble. The first assimilation time is marked with a solid vertical black line.

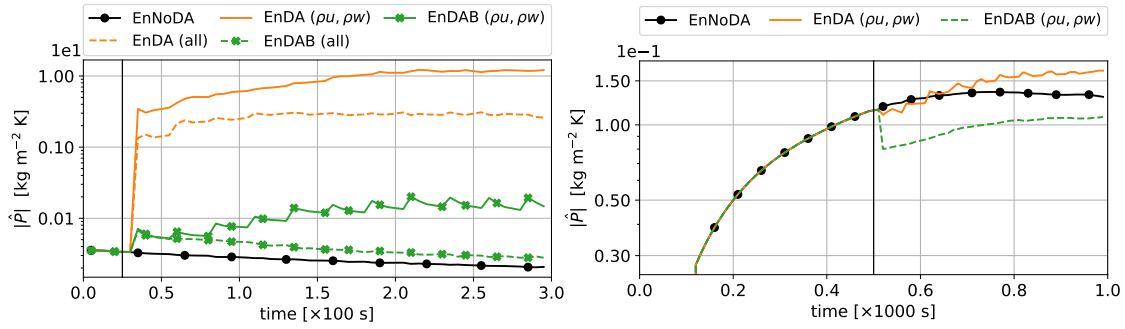


Figure B.2: Scale analysis of the contribution to the mass-weighted potential temperature \hat{P} from the divergence of the velocity fields for the Euler vortex ensemble (left) and the rising bubble ensemble (right). A semi-logarithmic scale is used here. In the legend, (all) represents the Euler vortex ensembles with all quantities, $\{\rho, \rho u, \rho w, P, \pi'\}$, assimilated. The first assimilation time is marked with a vertical solid black line.

quantities \hat{h}' and \hat{P} obtained from the scale analysis marginally overestimate the increase in the RMSEs. This is in line with the largest magnitude estimate used in (B.2) and with the largest choice of t_{ac} in (B.3). The results from scale analysis show that the error jumps seen in the pressure-related plots quantify the fast-mode imbalances introduced through the assimilation of data.

Appendix C

Ensemble spread of the data assimilation experiments

The RMSE and the ensemble spread should be comparable (Fortin et al., 2014), and we examine this statement here. The ensemble spread is calculated by

$$\text{Ens. spread}(\psi) = \sqrt{\frac{1}{K} \frac{1}{N_x \times N_y} \sum_{k=1}^K \sum_{i,j=1}^{N_x, N_y} [\psi_k^{\text{ensemble}}(x_i, y_j) - \bar{\psi}^{\text{ensemble}}(x_i, y_j)]^2}, \quad (\text{C.1})$$

where $k = 1, \dots, K$ indexes the ensemble members and $i = 1, \dots, N_x$ and $j = 1, \dots, N_y$ the number of grid points in the (x, y) coordinates. Here, ψ is the set of quantities $\{\rho, \rho u, \rho v, P, \pi\}$, and $\bar{\psi}^{\text{ensemble}}(x_i, y_j)$ is the ensemble mean at grid point (x_i, y_j) . For the vertical slice experiments, equation (C.1) has to be modified with the z coordinate direction instead of y and ρw instead of ρv .

The ensemble spread computed for the SWE vortex, Euler vortex, and the rising bubble experiments are provided in Figures C.1, C.2, and C.3 respectively. The results show that the data assimilation does what is to be expected. The data assimilation experiments do not experience catastrophic filter divergence and there is no ensemble collapse (Harlim and Majda, 2010). However, the filter diverges under certain circumstances, and being able to account for the filter divergence is at the core of the experiments with idealised tests in this thesis.

For ensemble data assimilation runs with only the momentum fields assimilated, the RMSE and the ensemble spread are qualitatively comparable. For Euler vortex run with only the momentum fields assimilated, i.e. the solid orange and green curves in in Figure 6.20, neither the plain data assimilation nor data assimilation with the blending strategy is able to track the vortex system. This is due to the dynamically relevant potential temperature field that is not corrected by the assimilation of the velocity fields alone. Indications of the failure of the data assimilation scheme to track the system can also be seen in the ensemble spread results (see the solid orange and solid green lines in Figure C.2). Note, however, that assimilation of all variables does allow the ensemble to closely track the reference vortex.

For the sparse observation of all quantities in the Euler vortex test case, the ensemble spreads for both the EnDA and EnDAB runs (dashed lines in Figure C.2) decrease substantially. Yet in the RMSE plots of Figure 6.20, the EnDA run is unable to recover the balanced state of the system while the EnDAB run is able to do so quite well (compare the dashed orange and green

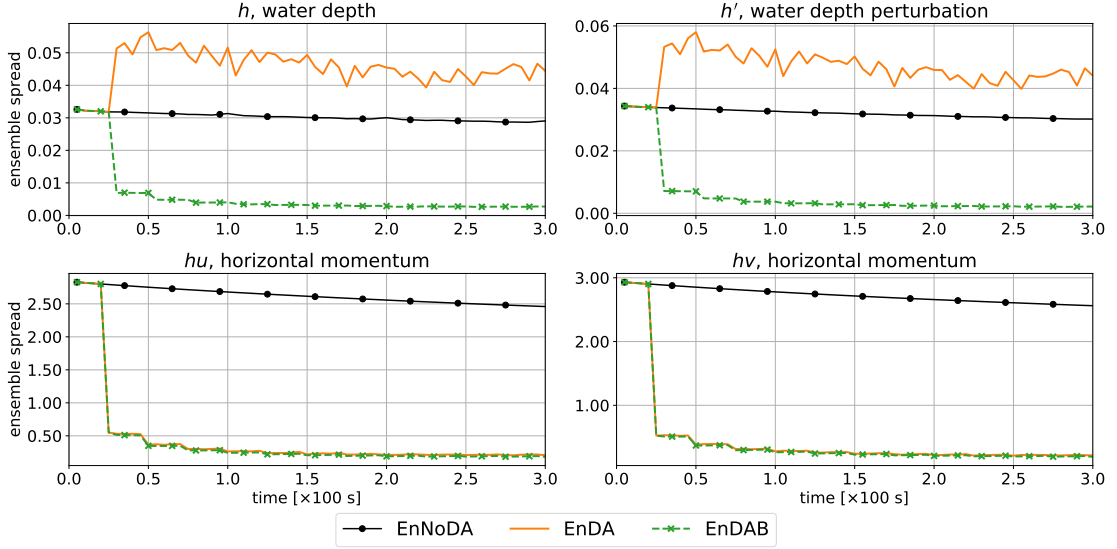


Figure C.1: Ensemble spread of the SWE vortex: EnNoDA run (dots, solid black), EnDA run (solid orange), EnDAB run (crosses, solid green). Assimilated quantities are hu and hv . The water depth h (top left, [m]), water depth perturbation h' (top right, [m]), and momenta hu , hv (bottom left and right, [$\text{m}^2 \text{s}^{-1}$]). The spread of the initial ensemble is omitted.

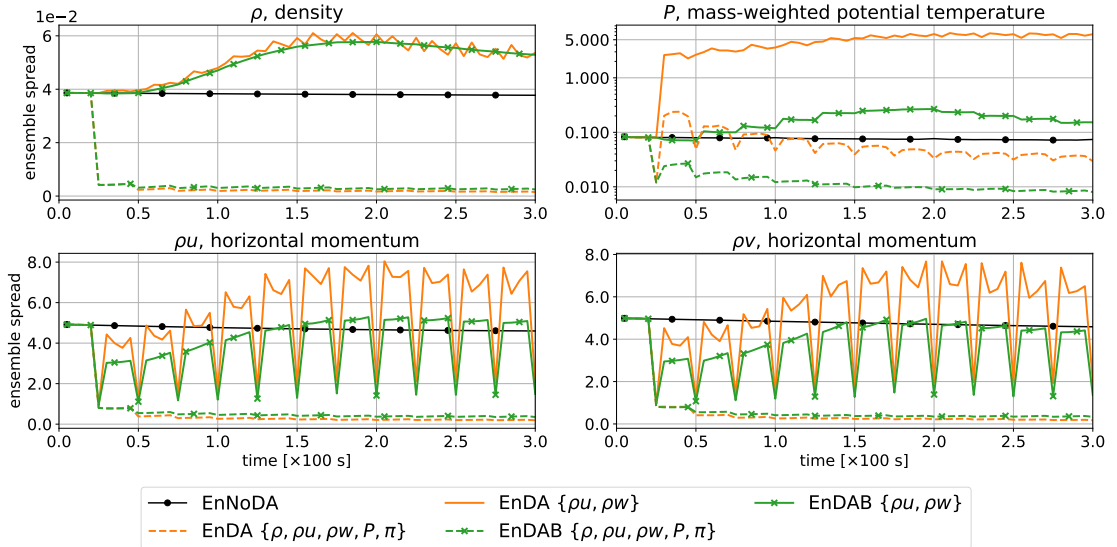


Figure C.2: Ensemble spread of the Euler vortex: EnNoDA run (dots, solid black), EnDA run (orange), EnDAB run (crosses, green). Assimilated quantities are ρu and ρv (solid lines) and ρ , ρu , ρv , P , π' (dashed lines). Density ρ (top left, [kg m^{-2}]), mass-weighted potential temperature P (top right log-linear scale, [$\text{kg m}^{-2} \text{K}$]), and momenta ρu , ρv (bottom left and right, [$\text{kg m}^{-1} \text{s}^{-1}$]). The spread of the initial ensemble is omitted.

lines in Figure 6.20). As shown in the snapshots of the ensemble members in Figure 6.19, the reason for this disparity between the RMSE and the ensemble spread becomes apparent. Data assimilation is effective, and the ensembles converge in both the EnDA and EnDAB experiments. As to be expected, this convergence is reflected in the ensemble spread. However, the EnDA run converges to an imbalanced state while the EnDAB converges to a balanced state. The former

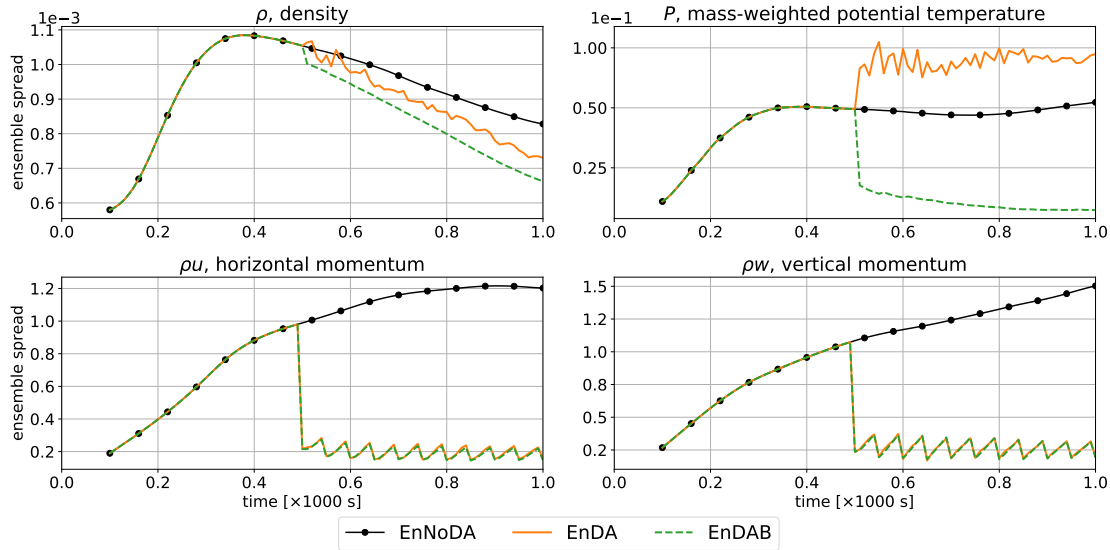


Figure C.3: Rising bubble ensemble spread: EnNoDA run (dots, solid black), EnDA run (solid orange), EnDAB run (dashed green). Assimilated quantities are ρu and ρw . Density ρ (top left, [kg m^{-2}]), mass-weighted potential temperature P (top right log-linear scale, [$\text{kg m}^{-2} \text{K}$]), and momenta ρu , ρw (bottom left and right, [$\text{kg m}^{-1} \text{s}^{-1}$]).

leads to a large error score and the latter to a relatively lower error score. Furthermore, this discrepancy implies the following. The ensemble spread as a metric alone is insufficient as an indication of the balance of the analysis ensemble.

Assimilation of the momentum fields alone with the plain data assimilation scheme is insufficient to recover balanced solutions for the SWE vortex test (solid orange line in Figure 6.18) and for the rising bubble test (solid orange line in Figure 6.22). The ensemble spread for these experiments exhibit a similar jump after the first assimilation time, and this can be seen in solid orange line in the top panels of Figure C.1 and in the top right panel of Figure C.3). In contrast, by applying the blending strategies as a balancing method on top of the plain data assimilation scheme, the balanced solutions were recovered for all the quantities in both the test cases (see the dashed green curves in Figures 6.18 and 6.22). The corresponding ensemble spread also decreases substantially for all quantities (dashed green curves in Figures C.1 and C.3).

In summary, the RMSE and the ensemble spread are comparable for all results investigated here except for the Euler vortex run with all quantities assimilated. Nevertheless, this discrepancy can be explained by considering the balance of the solution. A similar ensemble spread study for the nonhydrostatic-hydrostatic blending is left for future work.

Appendix D

On the convergence of the data assimilation system

The convergence of the data assimilation experiments over a longer simulation time is studied in this appendix. From experience and for the compressible-soundproof blending, the Euler vortex test case is the most sensitive to the assimilation of data. As such, we restrict our attention solely to the study of this test case here.

In Figure 6.20, the errors in the quantities ρ , ρu , and ρv for the ensemble simulation run with data assimilation only (EnDA, $\{\rho, \rho u, \rho v, P, \pi'\}$ observed, dashed orange curve) remain below that of the control EnNoDA ensemble. The data assimilation system converges, and the vortex structure remains compact for these quantities. The pressure-related quantities, P and π , in contrast, are rendered out of balance in the first assimilation step as indicated by a sudden increase of the error. This imbalance is analysed and quantified in Appendix B on the scale analysis. The compact vortex structure in the pressure field is also destroyed, see the second row of Figure 6.19. In summary, these results show that, over the simulation time horizon of 300.0 s, the data assimilation system evolves with an error that is lower than that of the control ensemble for $(\rho, \rho u, \rho v)$ while maintaining an error higher than that of the control ensemble for the pressure variables P and π .

However, as long as acoustic imbalances are introduced by the assimilation procedure, the error scores of the EnDA ensemble for the variables P and π will always be above those of the control EnNoDA ensemble as the latter consists of balanced solutions only. The reason is that acoustically balanced flows come with pressure variations of $O(\text{Ma}^2)$, whereas acoustic perturbations come with much larger pressure fluctuations of $O(\text{Ma})$ for low Mach number flows. Such acoustic fluctuations are triggered by the data assimilation procedure without balancing while they are absent from the EnNoDA ensemble.

Over longer simulation times, the imbalances introduced into the pressure-related fields by the data assimilation procedure cumulatively affect the balance of the other quantities, and the RMSE of the quantities ρ , ρu , and ρv is bound to increase eventually. Three additional results are provided below to demonstrate this effect, and an argument that the data assimilation system nevertheless converges will be made.

In Figure D.1, the EnDA and EnNoDA experiments from Figure 6.20 are repeated in the pseudo-incompressible regime. These runs are labelled as EnNoDA (psinc) and EnDA (psinc) in

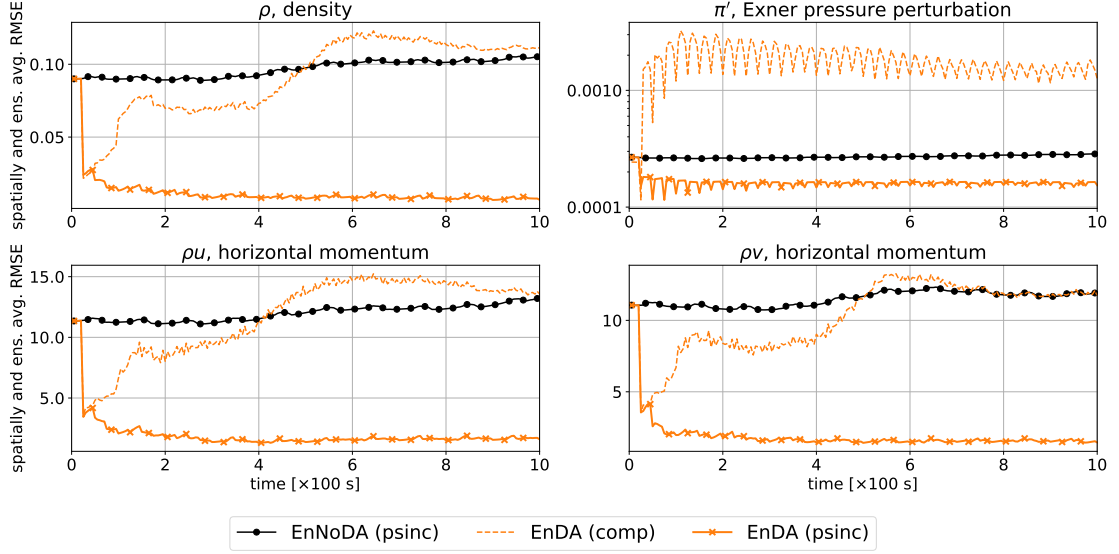


Figure D.1: Travelling vortex: pseudo-incompressible EnNoDA run (solid dotted, black), compressible EnDA run (dashed orange), and pseudo-incompressible EnDA run (solid orange, crosses). Assimilated quantities are ρ , ρu , ρv , P , π' . Spatially and ensemble averaged RMSE from $t = 0.0$ s to 1000.0 s for density ρ (top left, [kg m^{-2}]), dimensionless Exner pressure perturbation π' (top right, log-linear scale), and momenta ρu , ρv (bottom left and right, [$\text{kg m}^{-1} \text{s}^{-1}$]). The RMSE of the initial ensemble is omitted.

the Figure. Acoustic imbalances are not present in the pseudo-incompressible solution, and as seen in Figure D.1, the error jump characterised by the acoustic imbalances and introduced by the data assimilation procedure is indeed not present in the Exner pressure perturbation π' for the EnDA (psinc) run. Thus, in the absence of acoustic imbalances, the data assimilation system converges for all quantities assimilated over ten periods of the travelling vortex cycle (1000.0 s). This conclusion supports the use of the blending strategies, as this balancing method is able to dynamically remove acoustic imbalances. A longer simulation of the compressible EnDA result with all quantities assimilated (dashed orange graph from Figure 6.20), labelled EnDA (comp) here, is provided for comparison.

Figure D.1 shows that the error of the EnDA (comp) does not actually converge below that of the EnNoDA control ensemble over a longer simulation time. This is due to the feedback from the imbalances in the pressure-related fields, as mentioned above. Figure D.2 includes results from an additional *EnOneDA* run alongside the compressible EnNoDA, EnDA (comp), and EnDAB runs. The pseudo-incompressible EnDA run from Figure D.1, labelled EnDA (psinc), is included for comparison. The *EnOneDA* run comprises the travelling vortex ensemble with only one assimilation procedure at time $t = 25.0$ s. This data assimilation step introduces an imbalance similar in magnitude to that in the EnDA run, so that the subsequent forward simulations create a new ensemble without data assimilation and with imbalanced initial data.

Figure D.2 reveals that over a longer simulation period, the errors of the EnDA run converge

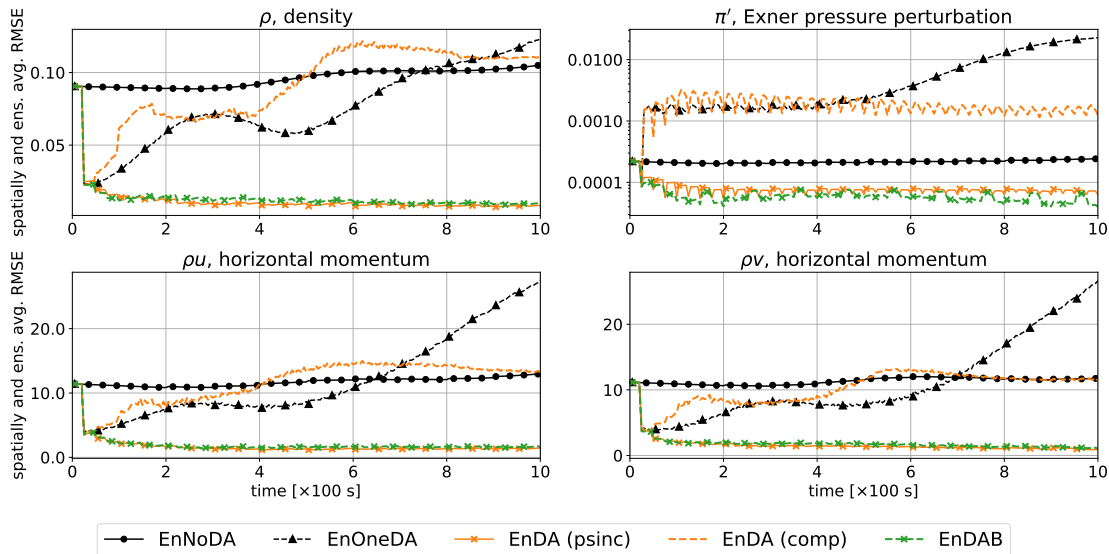


Figure D.2: Travelling vortex: EnNoDA run (solid black, dots), EnOneDA run (dashed black, triangles; see text for run details), pseudo-incompressible EnDA run (solid orange, crosses), compressible EnDA run (dashed orange), and EnDAB run (dashed green, crosses). Assimilated quantities are ρ , ρu , ρv , P , π' . Spatially and ensemble averaged RMSE from $t = 0.0$ s to 1000.0 s for density ρ (top left, [kg m^{-2}]), Exner pressure perturbation π' (top right log-linear scale, dimensionless), and momenta ρu , ρv (bottom left and right, [$\text{kg m}^{-1} \text{s}^{-1}$]). The RMSE of the initial ensemble is omitted.

below those of the EnOneDA run for all quantities. Therefore, if the control ensemble is chosen such that it consists of vortices with imbalanced initial data, then the error of the EnDA run converges below that of the imbalanced ensemble run for all the quantities assimilated, albeit over a longer simulation time. In the experiments of chapter 6, however, we are interested in a more challenging scenario: we wish to demonstrate that a data assimilation ensemble balanced by the blending strategies is able to beat the scores of a freely evolving ensemble with balanced solutions in all variables, including the pressure variables in particular. This is why an ensemble with balanced solutions, EnNoDA, was chosen as the control ensemble. Moreover, comparison of the EnDA (psinc) and EnDAB results demonstrates that the blending strategy achieves error scores of comparable magnitude to those obtained with the pseudo-incompressible model which suppresses fast acoustic degrees of freedom entirely, even though the EnDAB run is solved with the compressible model that does allow for fast dynamics.

For Figure D.3, the EnDA setup from Figure 6.20 is modified such that 90% of the grid points are observed (as opposed to 10%). Moreover from $t = 25.0$ s onward, observations are assimilated every 5 s (as opposed to every 25.0 s), and a moderate localisation region size of (21×21) grid points is used (as opposed to a (11×11) localisation region). With these settings, the RMSE of the quantities ρ , ρu , and ρv for the EnDA run converge below the RMSE of the EnNoDA ensemble with balanced solutions. The imbalance introduced in the pressure-related quantities P and π is mitigated by the larger localisation region and the large number of grid points observed,

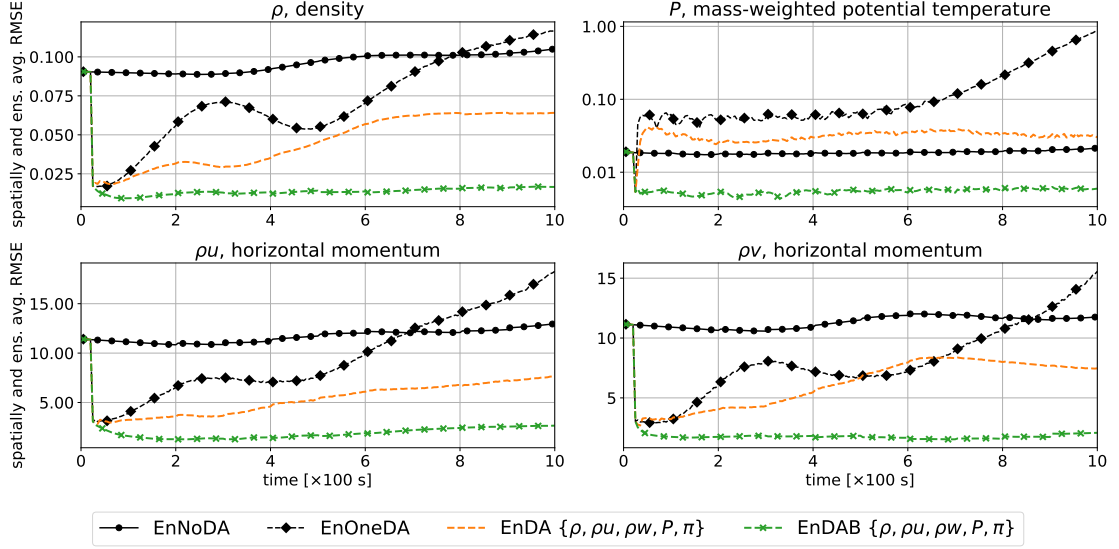


Figure D.3: Compressible travelling vortex: EnNoDA run (solid black, dots), EnOneDA run (dashed black, triangles), EnDA run (dashed orange), and EnDAB run (dashed green, crosses). Quantities ρ , ρu , ρv , P , π' are assimilated every 5s, 90% of the grid points are observed, and a localisation region of (21×21) grid points is applied. Spatially and ensemble averaged RMSE from $t = 0.0$ s to 1000.0 s for density ρ (top left, $[\text{kg m}^{-2}]$), mass-weighted potential temperature P (top right log-linear scale, $[\text{kg m}^{-2} \text{K}]$), and momenta ρu , ρv (bottom left and right, $[\text{kg m}^{-1} \text{s}^{-1}]$). The RMSE of the initial ensemble is omitted.

and the corresponding increase in the RMSE in P is smaller in comparison to that of Figure 6.20. An EnOneDA run similar to the one in Figure D.2, but with the modified data assimilation setup detailed in this paragraph, is depicted for comparison. It is seen that if the data assimilation setup is tuned so as to largely mitigate the imbalances, the data assimilation system would converge for the quantities ρ , ρu , and ρv . The RMSE scores of the EnDA run for P are also lower than those for the EnOneDA run, which is the “fair” comparison. As explained above by the Mach number scaling argument, however, even with the tuned parameters the EnDA run is unlikely to beat scores of the EnNoDA run (with balanced solutions) due to the imbalances introduced by the data assimilation procedure.

In the longer simulation runs above, the truth and observations are obtained from the analytical solution of the stable travelling vortex.

Appendix E

Localisation region and imbalances

In this appendix, the effect of the localisation region on the imbalances introduced is investigated, specifically for the Euler vortex test case. Similar results can be seen for the SWE vortex and the rising bubble tests, but these results are not shown nor discussed here. Note that this study appeared in Chew et al. (2021).

Varying localisation radii for the EnDA and EnDAB ensembles are investigated, and the goal of this study is not to obtain the optimal choice of the localisation radius but to illustrate its effect on the imbalances. All the quantities are assimilated for the Euler vortex test case, and localisation regions of (5×5) , (11×11) , (21×21) and (41×41) grid points are used in addition to a run without localisation (EnNoLoc). Otherwise, the setup follows the parameters laid out in section 6.3.1 and Table 6.2.

For the EnDA experiments in Figure E.1, the balanced vortex structure is preserved for the quantities ρ , ρu , and ρv , and here, the size of the localisation region used plays an important role in the quality of the data assimilation.

For small localisation regions, fewer observations are assimilated for each grid point under analysis, and the effect of data assimilation is less severe. This leads to a more gradual nudging of the ensemble towards the truth, and the drop in the RMSE after the first assimilation time at $t = 25.0$ s is less drastic. This can be seen in the top left and the bottom panels of Figure E.1 where the run with a (5×5) localisation region (solid magenta with square markers) has the least drastic decrease in the RMSE at $t = 25.0$ s of all the runs, and the RMSEs continue to decrease gradually over the subsequent assimilation steps. However, the small localisation region also leads to severe imbalances, and this can be seen in the pressure-related quantity P where the run with a (5×5) localisation region exhibits the largest error jump after $t = 25.0$ s.

For moderately small localisation regions, e.g. (11×11) (solid orange line with triangle markers in Figure E.1), more observations are assimilated for each grid point under analysis and the effect of data assimilation becomes more pronounced. This leads to a drastic decrease in the RMSE of the quantities ρ , ρu , and ρv after the first assimilation time at the expense of destroying the balanced compact vortex structure. At the same time, the imbalances introduced by the localisation remains severe, and the jump in the error of the P variable is large. In such a case, contributions from these two sources of errors overlap, and the increase in the RMSE over time becomes large. As a result, the run with a (11×11) localisation region has the largest error at the simulation end time among all the runs with data assimilation.

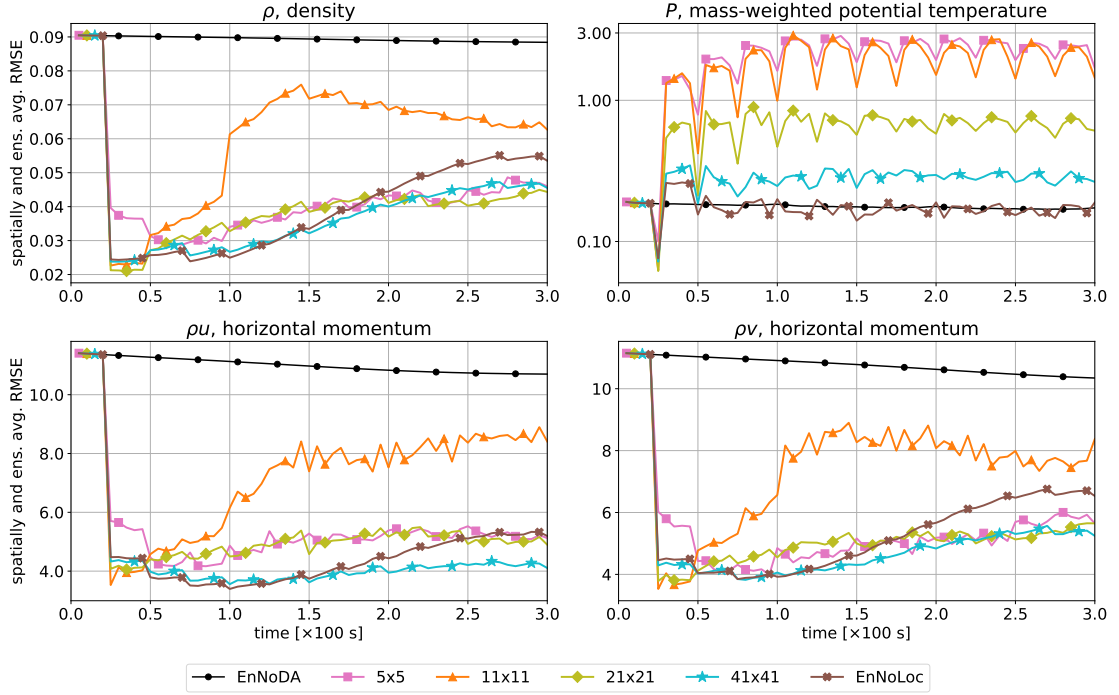


Figure E.1: Travelling vortex: EnNoDA (black dots) and EnDA experiments corresponding to localisation regions with (5×5) (magenta squares), (11×11) (orange triangles), (21×21) (yellow diamonds), and (41×41) (cyan stars) grid points, and EnNoLoc without localisation (brown crosses). Assimilated quantities are ρ , ρu , ρv , P , π' . Spatially and ensemble averaged RMSE from $t = 0.0$ s to 300.0 s for density ρ (top left, $[\text{kg m}^{-2}]$), mass-weighted potential temperature P (top right log-linear scale, $[\text{kg m}^{-2} \text{K}]$), and momenta ρu , ρv (bottom left and right, $[\text{kg m}^{-1} \text{s}^{-1}]$). The RMSE of the initial ensemble is omitted.

In comparison, a small localisation region does not exhibit such an overlapping of the two error sources, as the gentle nudging of the ensemble towards the truth better preserves the compact vortex structure. As such, the run with a (5×5) localisation region has only moderate increase in the RMSE towards the end of the simulation, and its error scores for the quantities ρ , ρu , and ρv are significantly below that of the run with a (11×11) localisation region.

For large localisation regions, e.g. (21×21) (solid yellow line with diamond markers in Figure E.1) and (41×41) (solid cyan line with star markers), the imbalances introduced by local data assimilation is less severe, and this can be seen in the moderate error jumps in the P variable after $t = 25.0$ s. For the run without localisation (EnNoLoc, solid brown line with cross markers), the error jump in P is almost non-existent, although a residual imbalance is introduced, and this is seen in the fluctuation of the error profile over time when compared to the smoother error profile of the balanced EnNoDA run (solid black dotted line). As expected, EnDA runs with larger localisation regions tend to have a lower error score.

However, a localisation region that is too large leads to an oversampling of the dynamics. Grid points under analysis inside of the compact vortex structure are influenced by observations of the background and vice versa. As a result, the vortex becomes more diffused or spread out as the

assimilation procedure is repeated. The effect of this oversampling can be seen in, for example, the EnNoLoc run in Figure E.1, and the (41×41) and the EnNoLoc runs in Figure E.2, where the error scores towards the end of the simulation are higher than that of a run with only a moderate localisation region.

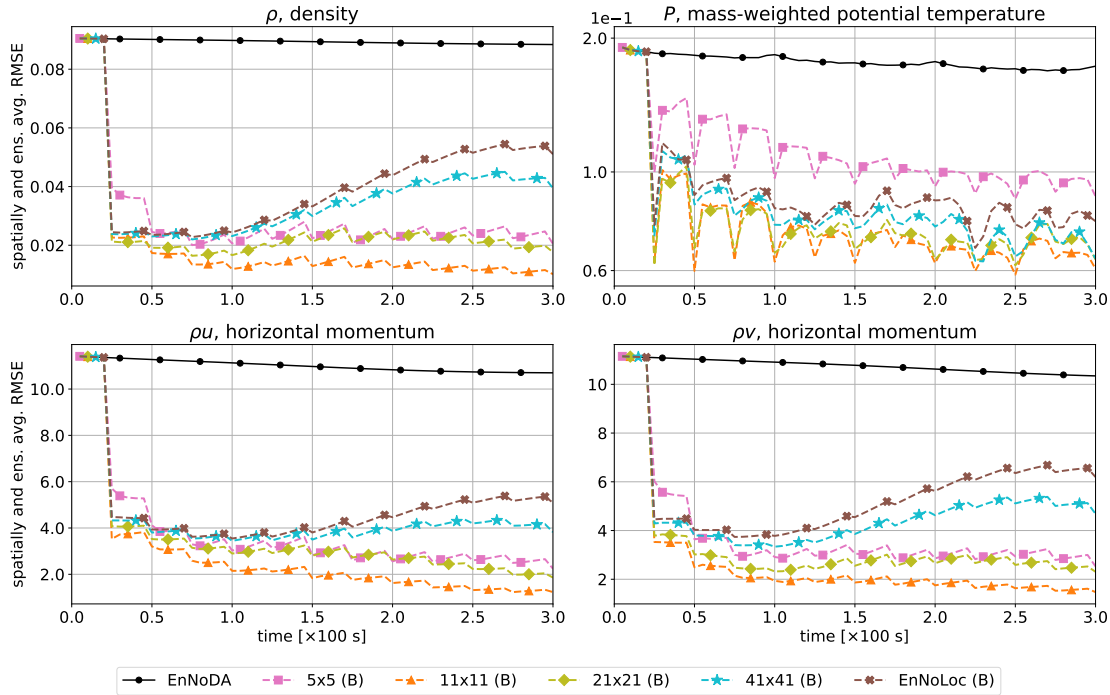


Figure E.2: Travelling vortex: EnNoDA (black dots) and EnDAB experiments corresponding to localisation regions with (5×5) (magenta squares), (11×11) , (orange triangles), (21×21) (yellow diamonds), and (41×41) (cyan stars) grid points, and EnNoLoc without localisation (brown crosses). See Figure E.1 for the details of the panels.

Application of the blending strategies to data assimilation as a balancing mechanism eliminates the imbalances from a local data assimilation procedure. As a result, smaller localisation regions may be used with fewer detrimental effects, and the best error scores are achieved by an EnDAB run with a localisation region of (11×11) grid points (dashed orange line with triangle markers in Figure E.2). The higher RMSE in the (5×5) run (dashed magenta line with square markers) may be due to the undersampling of the vortex dynamics. This is the opposite of the oversampling effect described above.

As the run with a localisation region of size (11×11) grid points best highlights the positive benefits of the blending strategy, this choice of the localisation region is used for Euler vortex experiments in chapter 6.

Zusammenfassung

Die lokale sequentielle Bayes'sche Datenassimilation verursacht unphysikalische Imbalancen. Dies stellt eine Herausforderung für die Modellierung geophysikalischer Strömungen dar und hat Auswirkungen auf robuste numerische Wettervorhersagen. Das Auftreten sich schnell ausbreitender Imbalancen von der Größenordnung der betreffenden langsameren Dynamik vermindert die Lösungsqualität. Diese Arbeit führt dynamikgesteuerte Methoden ein, die Imbalancen aus der Datenassimilation unterdrücken und so dem oben beschriebenen Effekt entgegenwirken. Konkret wird für die kompressiblen Strömungsgleichungen mit Schwer- und Corioliskraft ein numerisches Modell in gemischter Form verwendet, welches auf schalldichte und hydrostatische Dynamik zurückgreifen kann. Dieses Modell wird dann formal erweitert, um einen nahtlosen Übergang zwischen den Flachwassergleichungen und deren inkompressiblem Analogon, den sogenannten "lake-equations", zu ermöglichen. Durch sorgfältige numerische und asymptotische Analysen werden ferner einstufige Blending-Strategien entwickelt, die einen nahtlosen Wechsel zwischen den Modellregimen während des Simulationslaufs ermöglichen. Nach der Assimilation der Daten wird die Modellkonfiguration für einen Zeitschritt in das Grenzregime überführt. Anschließend wird die Modellkonfiguration für die Dauer des Assimilationsfensters wieder auf das Flachwasser- oder das kompressible Modell eingestellt. Dieser Wechsel zwischen den Modellregimen wird für jedes weitere Assimilationsfenster wiederholt, um die durch die Assimilation der Daten entstehenden Imbalancen zu beseitigen. Idealisierte Experimente mit dem wandernden Wirbel, der auftriebsgetriebenen aufsteigenden atmosphärischen Blase und den Trägheitsschwerewellen zeigen, dass die Blending-Strategien unphysikalische Imbalancen erfolgreich beseitigen und Abweichungen um bis zu zwei Größenordnungen verkleinern. Diese neuartige dynamikkonforme Methode zur Erlangung einer balancierten Datenassimilation kann erweitert werden, um andere Formen von Imbalancen zu eliminieren. Sie hat das Potenzial, durch Datenassimilation erzeugte Störsignale in numerischen Wettervorhersagesimulationen zu reduzieren.

References

- J. L. Anderson. An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus*, 59(2):210–224, 2007. doi:10.1111/j.1600-0870.2006.00216.x.
- A. Arakawa and C. S. Konor. Unification of the anelastic and quasi-hydrostatic systems of equations. *Mon. Wea. Rev.*, 137(2):710–726, 2009. doi:10.1175/2008MWR2520.1.
- M. Baldauf and S. Brdar. An analytic solution for linear gravity waves in a channel as a test for numerical models using the non-hydrostatic, compressible Euler equations. *Quart. J. Roy. Meteor. Soc.*, 139(677):1977–1989, 2013. doi:10.1002/qj.2105.
- R. N. Bannister. Elementary 4D-Var. *Reading: University of Reading*, 2001. <http://www.met.rdg.ac.uk/~ross/Documents/Var4d.pdf>.
- R. N. Bannister. How is the balance of a forecast ensemble affected by adaptive and nonadaptive localization schemes? *Mon. Wea. Rev.*, 143:3680–3699, 2015. doi:10.1175/MWR-D-14-00379.1.
- W. Bauer and F. Gay-Balmaz. Towards a geometric variational discretization of compressible fluids: The rotating shallow water equations. *J. Comput. Nonlinear Dyn.*, 6(1):1–37, 2019. doi:10.3934/jcd.2019001.
- T. Benacchio and R. Klein. A semi-implicit compressible model for atmospheric flows with seamless access to soundproof and hydrostatic dynamics. *Mon. Wea. Rev.*, 147:4221–4240, 2019. doi:10.1175/MWR-D-19-0073.1.
- T. Benacchio, W. P. O’Neill, and R. Klein. A blended soundproof-to-compressible numerical model for small-to mesoscale atmospheric dynamics. *Mon. Wea. Rev.*, 142:4416–4438, 2014. doi:10.1175/MWR-D-13-00384.1.
- E. Blayo, E. Cosme, M. Nodet, and A. Vidart. Introduction to data assimilation, 2011.
- S. Bloom, L. Takacs, A. Da Silva, and D. Ledvina. Data assimilation using incremental analysis updates. *Mon. Wea. Rev.*, 124(6):1256–1271, 1996. doi:10.1175/1520-0493(1996)124<1256:DAUIAU>2.0.CO;2.
- M. Bocquet. Ensemble Kalman filtering without the intrinsic need for inflation. *Non-linear Processes Geophys.*, 18:735–750, Oct. 2011. doi:10.5194/npg-18-735-2011.
- D. Bresch. Shallow-water equations and related topics. *Handbook of differential equations: evolutionary equations*, 5:1–104, 2009. doi:10.1016/S1874-5717(08)00208-9.

References

- D. Bresch, R. Klein, and C. Lucas. Multiscale analyses for the shallow water equations. In *Computational Science and High Performance Computing IV*, pages 149–164. Springer, 2011. doi:10.1007/978-3-642-17770-5_12.
- R. Chew, T. Benacchio, G. Hastermann, and R. Klein. A one-step blended soundproof-compressible model with balanced data assimilation: theory and idealised tests, 2021. <https://arxiv.org/abs/2103.11861>.
- S. E. Cohn, A. Da Silva, J. Guo, M. Sienkiewicz, and D. Lamich. Assessing the effects of data selection with the DAO physical-space statistical analysis system. *Mon. Wea. Rev.*, 126(11):2913–2926, 1998. doi:10.1175/1520-0493(1998)126<2913:ATEODS>2.0.CO;2.
- R. Daley. The normal modes of the spherical non-hydrostatic equations with applications to the filtering of acoustic modes. *Tellus*, 40(2):96–106, 1988. doi:10.3402/tellusa.v40i2.11785.
- Dask Development Team. *Dask: Library for dynamic task scheduling*, 2016. <https://dask.org>.
- T. Davies, M. J. P. Cullen, A. J. Malcolm, M. Mawson, A. Staniforth, A. A. White, and N. Wood. A new dynamical core for the Met Office’s global and regional modelling of the atmosphere. *Quart. J. Roy. Meteor. Soc.*, 131(608):1759–1782, 2005. doi:10.1256/qj.04.101.
- M. Duarte, A. S. Almgren, and J. B. Bell. A low Mach number model for moist atmospheric flows. *J. Atmos. Sci.*, 72:1605–1620, 2015. doi:10.1175/JAS-D-14-0248.1.
- J. Dudhia. Reply-"A nonhydrostatic version of the Penn state-NCAR mesoscale model: validation tests and simulation of an Atlantic Cyclone and Cold Front". *Mon. Wea. Rev.*, 123(8):2573–2575, 1995. doi:10.1175/1520-0493(1995)123<2573:R>2.0.CO;2.
- D. R. Durran. Improving the anelastic approximation. *J. Atmos. Sci.*, 46:1453–1461, 1989. doi:10.1175/1520-0469(1989)046<1453:ITAA>2.0.CO;2.
- D. R. Durran and J. B. Klemp. A compressible model for the simulation of moist mountain waves. *Mon. Wea. Rev.*, 111(12):2341–2361, 1983. doi:10.1175/1520-0493(1983)111<2341:ACMFTS>2.0.CO;2.
- P. Fearnhead and H. R. Künsch. Particle filters and data assimilation. *Annu. Rev. Stat. Appl.*, 5:421–449, 2018. doi:10.1146/annurev-statistics-031017-100232.
- R. A. Fisher and F. Yates. *Statistical tables for biological, agricultural and medical research*. Hafner Publishing Company, 3 edition, 1953.
- J. Flowerdew. Towards a theory of optimal localisation. *Tellus*, 67A:25257, 2015. doi:10.3402/tellusa.v67.25257.

- V. Fortin, M. Abaza, F. Anctil, and R. Turcotte. Why should ensemble spread match the RMSE of the ensemble mean? *J. Hydrometeorol.*, 15(4):1708 – 1713, 2014. doi:10.1175/JHM-D-14-0008.1.
- I. Fukumori. A partitioned Kalman filter and smoother. *Mon. Wea. Rev.*, 130(5):1370–1383, 2002. doi:10.1175/1520-0493(2002)130<1370:APKFAS>2.0.CO;2.
- S. J. Greybush, E. Kalnay, T. Miyoshi, K. Ide, and B. R. Hunt. Balance and ensemble Kalman filter localization techniques. *Mon. Wea. Rev.*, 139:511–522, 2011. doi:10.1175/2010MWR3328.1.
- S. Ha, C. Snyder, W. C. Skamarock, J. Anderson, and N. Collins. Ensemble Kalman filter data assimilation for the Model for Prediction Across Scales (MPAS). *Mon. Wea. Rev.*, 145(11):4673–4692, 2017. doi:10.1175/MWR-D-17-0145.1.
- E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich. Geometric numerical integration. *Oberwolfach Rep.*, 3(1):805–882, 2006. doi:10.4171/OWR/2006/14.
- T. M. Hamill, J. S. Whitaker, and C. Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Mon. Wea. Rev.*, 129(11):2776–2790, 2001. doi:10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2.
- J. Harlim and B. R. Hunt. Local ensemble transform Kalman filter: An efficient scheme for assimilating atmospheric data. Preprints, 2005. https://www.meto.umd.edu/~ekalnay/pubs/harlim_hunt05.pdf.
- J. Harlim and A. J. Majda. Catastrophic filter divergence in filtering nonlinear dissipative systems. *Commun. Math. Sci.*, 8(1):27–43, 2010. doi:10.4310/CMS.2010.v8.n1.a3.
- C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi:10.1038/s41586-020-2649-2.
- A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 135(2):260–278, 1997. doi:10.1016/0021-9991(83)90136-5.
- A. Harten, P. D. Lax, and B. v. Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.*, 25(1):35–61, 1983. doi:10.1007/978-3-642-60543-7_4.
- G. Hastermann, M. Reinhardt, R. Klein, and S. Reich. Balanced data assimilation for highly oscillatory mechanical systems. *Comm. App. Math. Comp. Sci.*, 16(1):119–154, 2021. doi:10.2140/camcos.2021.16.119.

References

- H. He, L. Lei, J. S. Whitaker, and Z.-M. Tan. Impacts of assimilation frequency on ensemble Kalman filter data assimilation and imbalances. *J. Adv. Model. Earth Syst.*, 12(10):1–14, 2020. doi:10.1029/2020MS002187.
- C. Hohenegger and C. Schär. Predictability and error growth dynamics in cloud-resolving models. *J. Atmos. Sci.*, 64(12):4467–4478, 2007. doi:10.1175/2007JAS2143.1.
- P. L. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Mon. Wea. Rev.*, 126(3):796–811, 1998. doi:10.1175/1520-0493(1998)126<0796:DAUAEK>2.0.CO;2.
- P. L. Houtekamer and F. Zhang. Review of the ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, 144:4489–4532, 2016. doi:10.1175/MWR-D-15-0440.1.
- T. Hundertmark and S. Reich. A regularization approach for a vertical-slice model and semi-Lagrangian Störmer–Verlet time stepping. *Quart. J. Roy. Meteor. Soc.*, 133:1575–1587, 2007. doi:10.1002/qj.118.
- T. Hundertmark and S. Reich. On the use of slow manifolds in molecular and geophysical fluid dynamics. *Eur. Phys. J.: Spec. Top.*, 200(1):259–270, 2011. doi:10.1140/epjst/e2011-01527-7.
- B. R. Hunt, E. J. Kostelich, and I. Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D*, 230:112–126, 2007. doi:10.1016/j.physd.2006.11.008.
- J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.
- A. H. Jazwinski. Linear filtering theory. In *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*, pages 194–265. Elsevier, 1970. doi:https://doi.org/10.1016/S0076-5392(09)60376-3.
- S. Y. Kadioglu, R. Klein, and M. L. Minion. A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics. *J. Comput. Phys.*, 227:2012–2043, 2008. doi:10.1016/j.jcp.2007.10.008.
- M. Katzfuss, J. R. Stroud, and C. K. Wikle. Understanding the ensemble Kalman filter. *Amer. Statist.*, 70(4):350–357, 2016. doi:10.1080/00031305.2016.1141709.
- J. Kepert. 4D-Var for dummies. In *8th Adjoint Workshop*, 2009a. https://gmao.gsfc.nasa.gov/events/adjoint_workshop-8/present/Sunday/Kepert1.pdf.
- J. D. Kepert. Covariance localisation and balance in an ensemble Kalman filter. *Quart. J. Roy. Meteor. Soc.*, 135:1157–1176, 2009b. doi:10.1002/qj.443.

- P. Kirchgessner, L. Nerger, and A. Bunse-Gerstner. Localization in ensemble data assimilation. In *International Symposium on data assimilation*, 2012. <https://epic.awi.de/id/eprint/31369/1/Offenbach.pdf>.
- P. Kirchgessner, L. Nerger, and A. Bunse-Gerstner. On the choice of an optimal localization radius in ensemble Kalman filter methods. *Mon. Wea. Rev.*, 142(6):2165–2175, 2014. doi:10.1175/MWR-D-13-00246.1.
- R. Klein. Asymptotics, structure, and integration of sound-proof atmospheric flow equations. *Theor. Comput. Fluid Dyn.*, 23:161–195, 2009. doi:10.1007/s00162-009-0104-y.
- R. Klein. Scale-dependent models for atmospheric flows. *Annu. Rev. Fluid Mech.*, 42:249–274, 2010. doi:10.1146/annurev-fluid-121108-145537.
- R. Klein and T. Benacchio. A doubly blended model for multiscale atmospheric dynamics. *J. Atmos. Sci.*, 73:1179–1186, 2016. doi:10.1175/JAS-D-15-0323.1.
- R. Klein and O. Pauluis. Thermodynamic consistency of a pseudo-incompressible approximation for general equations of state. *J. Atmos. Sci.*, 69:961–968, 2012. doi:10.1175/JAS-D-11-0110.1.
- R. Klein, N. Botta, T. Schneider, C.-D. Munz, S. Roller, A. Meister, L. Hoffmann, and T. Sonar. Asymptotic adaptive methods for multi-scale problems in fluid mechanics. *J. Eng. Math.*, 39:261–343, 2001. doi:10.1023/A:1004844002437.
- R. Klein, T. Benacchio, and W. O’Neill. Using the sound-proof limit for balanced data initialization. In *Proc. ECMWF Seminar on Numerical Methods*, pages 227–236, Reading, United Kingdom, ECMWF, 2014. <https://www.ecmwf.int/sites/default/files/elibrary/2014/10483-using-sound-proof-limit-balanced-data-initialization.pdf>.
- J. B. Klemp, W. C. Skamarock, and S. Ha. Damping acoustic modes in compressible horizontally explicit vertically implicit (HEVI) and split-explicit time integration schemes. *Mon. Wea. Rev.*, 146(6):1911–1923, 2018. doi:10.1175/MWR-D-17-0384.1.
- T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing. Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016. doi:10.3233/978-1-61499-649-1-87.
- C. Kühnlein, W. Deconinck, R. Klein, S. Malardel, Z. P. Piotrowski, P. K. Smolarkiewicz, J. Szmelter, and N. P. Wedi. FVM 1.0: A nonhydrostatic finite-volume dynamical core formulation for IFS. *Geosci. Model Dev.*, 12:651–676, 2019. doi:10.5194/gmd-12-651-2019.

References

- S. K. Lam, A. Pitrou, and S. Seibert. Numba: A LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, 2015. doi:10.1145/2833157.2833162.
- M. Lang, P. Browne, P. J. Van Leeuwen, and M. Owens. Data assimilation in the solar wind: Challenges and first results. *Space Wea.*, 15:1490–1510, 2017. doi:10.1002/2017SW001681.
- D. Lannes. *The water waves problem: Mathematical analysis and asymptotics*, volume 188. American Mathematical Soc., 2013. doi:10.1090/surv/188.
- L. Lei and J. S. Whitaker. A four-dimensional incremental analysis update for the ensemble Kalman filter. *Mon. Wea. Rev.*, 144(7):2605–2621, 2016. doi:10.1175/MWR-D-15-0246.1.
- R. J. LeVeque. *Numerical methods for conservation laws*. Springer, 1992. doi:10.1007/978-3-0348-8629-1.
- R. J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge university press, 2002. doi:10.1017/CBO9780511791253.
- A. C. Lorenc. The potential of the ensemble Kalman filter for NWP—A comparison with 4D-Var. *Quart. J. Roy. Meteor. Soc.*, 129:3183–3203, 2003. doi:10.1256/qj.02.132.
- P. Lynch and X.-Y. Huang. Initialization of the HIRLAM model using a digital filter. *Mon. Wea. Rev.*, 120:1019–1034, 1992. doi:10.1175/1520-0493(1992)120<1019:IOTHMU>2.0.CO;2.
- S. MacNamara and G. Strang. Operator splitting. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 95–114. Springer, 2016. doi:10.1007/978-3-319-41589-5_3.
- T. Melvin, T. Benacchio, B. Shipway, N. Wood, J. Thuburn, and C. Cotter. A mixed finite-element, finite-volume, semi-implicit discretization for atmospheric dynamics: Cartesian geometry. *Quart. J. Roy. Meteor. Soc.*, 145(724):2835–2853, 2019. doi:10.1002/qj.3501.
- L. R. Mendez-Nunez and J. J. Carroll. Application of the MacCormack scheme to atmospheric nonhydrostatic models. *Mon. Wea. Rev.*, 122:984–1000, 1994. doi:10.1175/1520-0493(1994)122<0984:AOTMST>2.0.CO;2.
- V. Michel-Dansac, C. Berthon, S. Clain, and F. Foucher. A well-balanced scheme for the shallow-water equations with topography. *Comput. Math. Appl.*, 72(3):568–593, 2016. doi:10.1016/j.camwa.2016.05.015.
- H. L. Mitchell, P. L. Houtekamer, and G. Pellerin. Ensemble size, balance, and model-error representation in an ensemble Kalman filter. *Mon. Wea. Rev.*, 130(11):2791–2808, 2002. doi:10.1175/1520-0493(2002)130<2791:ESBAME>2.0.CO;2.

- L. J. Neef, S. M. Polavarapu, and T. G. Shepherd. Four-dimensional data assimilation and balanced dynamics. *J. Atmos. Sci.*, 63:1840–1858, 2006. doi:10.1175/JAS3714.1.
- W. O’Neill and R. Klein. A moist pseudo-incompressible model. *Atmos. Res.*, 142:133–141, 2014. doi:10.1016/j.atmosres.2013.08.004.
- J. Pedlosky. *Geophysical Fluid Dynamics*. Springer, 2 edition, 2013. doi:10.1007/978-1-4612-4650-3.
- S. Polavarapu, S. Ren, A. M. Clayton, D. Sankey, and Y. Rochon. On the relationship between incremental analysis updating and incremental digital filtering. *Mon. Wea. Rev.*, 132(10):2495–2502, 2004. doi:10.1175/1520-0493(2004)132<2495:OTRBIA>2.0.CO;2.
- A. A. Popov and A. Sandu. A Bayesian approach to multivariate adaptive localization in ensemble-based data assimilation with time-dependent extensions. *Nonlinear Process. Geophys.*, 26(2):109–122, 2019. doi:10.5194/npg-26-109-2019.
- A. Qaddouri, C. Girard, S. Z. Husain, and R. Aider. Implementation of a semi-Lagrangian fully implicit time integration of the Unified Soundproof System of Equations for numerical weather prediction. *Mon. Wea. Rev.*, 149(6):2011–2029, 2021. doi:10.1175/MWR-D-20-0291.1.
- S. Reich and C. Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, 2015. doi:10.1017/CBO9781107706804.
- S. Reich and C. J. Cotter. Ensemble filter techniques for intermittent data assimilation. In S. K. M. Cullen, Melina A Freitag and R. Scheichl, editors, *Large Scale Inverse Problems: Computational Methods and Applications in the Earth Sciences*, volume 13 of *Radon Series on Computational and Applied Mathematics*, pages 91–134. De Gruyter, 2013. doi:10.1515/9783110282269.91.
- T. Schneider, N. Botta, K. Geratz, and R. Klein. Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flows. *J. Comput. Phys.*, 155(2):248–286, 1999. doi:10.1006/jcph.1999.6327.
- W. C. Skamarock and J. B. Klemp. The stability of time-split numerical methods for the hydrostatic and the nonhydrostatic elastic equations. *Mon. Wea. Rev.*, 120(9):2109–2127, 1992. doi:10.1175/1520-0493(1992)120<2109:TSOTSN>2.0.CO;2.
- W. C. Skamarock and J. B. Klemp. Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique. *Mon. Wea. Rev.*, 122:2623–2630, 1994. doi:10.1175/1520-0493(1994)122<2623:EAAOTK>2.0.CO;2.
- P. K. Smolarkiewicz. On forward-in-time differencing for fluids. *Mon. Wea. Rev.*, 119:2505–2510, 1991. doi:10.1175/1520-0493(1991)119<2505:OFITDF>2.0.CO;2.

References

- P. K. Smolarkiewicz and A. Dörnbrack. Conservative integrals of adiabatic Durran's equations. *Int. J. Numer. Methods Fluids*, 56:1513–1519, 2008. doi:10.1002/flid.1601.
- P. K. Smolarkiewicz and L. O. Margolin. On forward-in-time differencing for fluids: Extension to a curvilinear framework. *Mon. Wea. Rev.*, 121:1847–1859, 1993. doi:10.1175/1520-0493(1993)121<1847:OFITDF>2.0.CO;2.
- P. K. Smolarkiewicz, C. Kühnlein, and N. P. Wedi. A consistent framework for discrete integrations of soundproof and compressible PDEs of atmospheric dynamics. *J. Comput. Phys.*, 263:185–205, 2014. doi:10.1016/j.jcp.2014.01.031.
- The HDF Group. *Hierarchical Data Format, version 5*, 1997. <https://www.hdfgroup.org/HDF5/>.
- G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. Cambridge University Press, 2nd edition, 2017. doi:10.1017/9781107588417.
- H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 13(2):631–644, 1992. doi:10.1137/0913035.
- B. Van Leer. Towards the ultimate conservative difference scheme I. The quest of monotonicity. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pages 163–168. Springer, 1973. doi:10.1007/BFb0118673.
- B. Van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.*, 14(4):361–370, 1974. doi:10.1016/0021-9991(74)90019-9.
- B. Van Leer. Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow. *J. Comput. Phys.*, 23(3):263–275, 1977a. doi:10.1016/0021-9991(77)90094-8.
- B. Van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *J. Comput. Phys.*, 23(3):276–299, 1977b. doi:10.1016/0021-9991(77)90095-X.
- B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(1):101–136, 1979. doi:10.1016/0021-9991(79)90145-1.
- P. J. Van Leeuwen, Y. Cheng, and S. Reich. *Nonlinear Data Assimilation*. Springer, 2015. doi:10.1007/978-3-319-18347-3.
- P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman,

- N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.
- F. Voitus, P. Bénard, C. Kühnlein, and N. P. Wedi. Semi-implicit integration of the unified equations in a mass-based coordinate: model formulation and numerical testing. *Quart. J. Roy. Meteor. Soc.*, 145(725):3387–3408, 2019. doi:10.1002/qj.3626.
- N. P. Wedi, M. Hamrud, and G. Mozdzyński. A fast spherical harmonics transform for global NWP and climate models. *Mon. Wea. Rev.*, 141(10):3450–3461, 2013. doi:10.1175/MWR-D-13-00016.1.
- J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Mon. Wea. Rev.*, 130(7):1913–1924, 2002. doi:10.1175/1520-0493(2002)130<1913:EDAWPO>2.0.CO;2.
- A. A. White, B. J. Hoskins, I. Roulstone, and A. Staniforth. Consistent approximate models of the global atmosphere: shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic. *Quart. J. Roy. Meteor. Soc.*, 131(609):2081–2107, 2005. doi:10.1256/qj.04.49.
- C. K. Wikle and L. M. Berliner. A Bayesian tutorial for data assimilation. *Physica D*, 230:1–16, 2007. doi:10.1016/j.physd.2006.09.017.
- G. Wilson, D. A. Aruliah, C. T. Brown, N. P. C. Hong, M. Davis, R. T. Guy, S. H. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, et al. Best practices for scientific computing. *PLoS biology*, 12(1):e1001745, 2014. doi:10.1371/journal.pbio.1001745.
- G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, and T. K. Teal. Good enough practices in scientific computing. *PLoS computational biology*, 13(6):e1005510, 2017. doi:10.1371/journal.pcbi.1005510.
- N. Wood, A. Staniforth, A. White, T. Allen, M. Diamantakis, M. Gross, T. Melvin, C. Smith, S. Vosper, M. Zerroukat, et al. An inherently mass-conserving semi-implicit semi-Lagrangian discretization of the deep-atmosphere global non-hydrostatic equations. *Quart. J. Roy. Meteor. Soc.*, 140(682):1505–1520, 2014. doi:10.1002/qj.2235.
- C. M. Zarzycki and C. Jablonowski. Experimental tropical cyclone forecasts using a variable-resolution global model. *Mon. Wea. Rev.*, 143(10):4012 – 4037, 2015. doi:10.1175/MWR-D-15-0159.1.
- M. Zupanski. Theoretical and practical issues of ensemble data assimilation in weather and climate. In S. K. Park and L. Xu, editors, *Data Assimilation for Atmospheric, Oceanic and*

References

Hydrologic Applications, volume 1, pages 67–84. Springer, 2009. doi:10.1007/978-3-540-71056-1_3.