Reduced-precision arithmetic in numerical weather prediction with an emphasis on data assimilation



Samuel Edward Hatfield



Jesus College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity Term 2019



Acknowledgements

Firstly, I would like to thank my supervisors, Tim Palmer and Peter Düben. Tim's advice was invaluable and he always encouraged me to see the bigger picture. Peter was a relentless source of optimism and always pushed me to apply to conferences and submit proposals, even when I doubted my own ability. I feel that my DPhil supervision has been in good hands.

I would like to thank my two de facto postdoc co-supervisors, Aneesh Subramanian and Matthew Chantry. Aneesh's help at the beginning of my DPhil gave me an excellent head start. After leaving Oxford, Aneesh continued to support me remotely and was kind enough to invite me to California for a conference talk and seminar. Mat's help was essential for conducting the IFS experiments and I consider him as an equal co-author on Chapter 6. Mat always allowed time to meet me, sometimes daily, and I am immensely grateful for his collaboration.

In Japan, I would like to thank Takemasa Miyoshi, who allowed me to intern in his group and helped me to arrange seminars in Japan, and Keiichi Kondo, who very patiently assisted me in my SPEEDY experiments. お世話になりました。

At ECMWF I would like to thank Glenn Carver and Paul Dando for technical support with the OpenIFS and IFS experiments, respectively. At NVIDIA I would like to thank Vishal Mehta for his help in developing the software emulator of the Tensor Core.

I would like to thank the Natural Environment Research Council (NERC), and by extension the UK taxpayer, for funding my tuition fees, my living costs and my research expenses. I would like to thank the Japan Society for the Promotion of Science (JSPS) for funding my living costs in Japan. I would also like to thank Jesus College for their generous graduate research allowance, which allowed me to attend many conferences.

Among the students of AOPP, I would like to thank Josh Dorrington and Milan Klöwer for providing a casual and friendly office environment, with many problem-solving interludes. The support from postdocs at the weekly Friday pub visit helped to keep me sane during the more difficult periods of the DPhil. I still owe many beers to Fenwick Cooper, Mio Matsueda, Yoshio Kawatani, Karsten Haustein, Andrew McRae and Jan Ackmann.

Finally, I would like to thank my parents for providing a home I can always return to in moments of doubt.

Abstract

The skill of weather forecasts has improved dramatically over the past 30 years. This improvement has depended to a large degree on developments in supercomputing, which have allowed models to increase in complexity and resolution with minimal technical effort. However, the nature of supercomputing is undergoing a significant change, with the advent of extremely parallel and heterogeneous architectures. This paradigm shift threatens the continual increase of forecast skill and prompts a reevaluation of how Earth-System models are developed. In this thesis we explore the notion of reduced-precision arithmetic to accelerate Earth-System models, specifically those used in data assimilation and in numerical weather prediction.

We first conduct data assimilation experiments with the Lorenz '96 toy atmospheric system, using the ensemble Kalman filter to perform assimilation. We reduce precision in the forecast and analysis steps of the ensemble Kalman filter and measure how this affects the quality of the data assimilation product, the analysis. We find that the optimal choice of precision is intimately linked with the degree of uncertainty from noisy observations and infrequent assimilation. We also find that precision can be traded for more ensemble members, and that this trade-off delivers a more accurate analysis than otherwise.

We then consider the SPEEDY intermediate complexity atmospheric general circulation model, again with the ensemble Kalman filter. In this case we find that, in a perfect model setting, reducing precision in the forecast model gives an unacceptable degradation in the data assimilation product. However, we then show that even a modest degree of model error can mask the errors introduced by reducing precision.

We consider also a precision reduction in the 4D-Var data assimilation scheme. We find that reducing precision increases the asymmetry between the tangent-linear and adjoint models, and that this retards the convergence of the minimisation scheme. However, with a standard reorthogonalisation procedure we are able to use single-precision, and even lower levels of precision, successfully.

Finally, we consider the use of reduced-precision arithmetic to accelerate the Legendre transforms of an operational global weather forecasting model. We find that, with a few considerations of the algorithmic structure of the transforms and the physical meaning of the different components, we are able to use even half-precision without affecting the forecast skill of the model.

In conclusion, we find that the errors introduced by reducing precision are negligible with respect to inherent errors in the forecasting system. In order to make optimal use of future supercomputers, reduced-precision arithmetic will be key.

Abbreviations

4D-Var Four dimensional variational assimilation. 11, 14, 16, 21, 23–26, 71–75, 77–80, 83, 84, 87, 88, 114

CG Conjugate gradients. 72–75, 83–85

CPU Central processing unit. 3, 4

CRPS Continuous ranked probability score. 101–103

ECMWF European Centre for Medium-Range Weather Forecasts. 1, 3, 11, 13, 25, 26, 46, 73, 74, 78, 82, 85, 89–91, 94, 100, 101, 104, 111

EnKF Ensemble Kalman filter. 14–19, 21, 24, 25, 32, 58, 71, 114

EnSRF Ensemble square-root filter. 19, 27, 28, 31, 45, 48, 119, 120

FLOP Floating-point operation. 4, 109, 110

GMRES Generalized minimal residual. 84, 85

GPU Graphics processing unit. 4, 9–11, 46, 91, 92, 96, 97, 109

IFS Integrated Forecasting System. 12, 46, 89, 90, 92, 94, 96, 98, 100, 101, 104

LETKF Local ensemble transform Kalman filter. 20, 25, 53, 55, 61, 66, 68

MTU Model time unit. 31, 35–37, 45

NWP Numerical weather prediction. 2, 3, 11, 13, 23–25, 46, 82

OOPS Object Oriented Prediction System. 74, 76, 85

OSSE Observing system simulation experiment. 28, 49

PCG Preconditioned conjugate gradients. 85, 87

QG Quasigeostrophic. 76–78

RMS Root-mean-square. 35, 37–39, 47

RMSE Root-mean-square error. 6, 36, 37, 40, 43–45, 57, 58, 98, 104

rpe Reduced Precision Emulator. 9, 10, 35, 51, 77, 96

RTPP Relaxation-to-prior perturbations. 54, 55, 57, 62, 65

SPEEDY Simplified Parametrizations, primitivE-Equation DYnamics. 11, 49–53, 57, 58, 60, 61, 63, 65, 66, 68, 91

SPPT Stochastically perturbed parametrisation tendencies. 3, 98–101

Z500 Geopotential height at 500 hPa. 98

Contents

1 Introduction **1**
- 1.1 The state of the art in numerical weather prediction . . . 1
- 1.2 Exascale weather prediction . . . 3
- 1.3 Why reduce precision? . . . 4
- 1.4 Floating-point arithmetic . . . 7
- 1.5 The Reduced Precision Emulator (rpe) . . . 9
- 1.6 Outline of thesis . . . 11
- 1.7 Statement of originality . . . 12
- 1.8 Publications . . . 12

2 Data assimilation theory **13**
- 2.1 Introduction . . . 13
- 2.2 The data assimilation problem . . . 14
- 2.3 Ensemble Kalman filter . . . 16
  - 2.3.1 Ensemble square-root filter . . . 19
  - 2.3.2 Local ensemble transform Kalman filter . . . 20
- 2.4 4D-Var . . . 21
  - 2.4.1 Incremental 4D-Var . . . 23
- 2.5 Conclusion . . . 24

3 Experiments with the Lorenz '96 system **27**
- 3.1 Introduction . . . 27
- 3.2 The Lorenz '96 system . . . 28
- 3.3 Data assimilation setup . . . 31
- 3.4 Reduced-precision analyses . . . 35
- 3.5 Sensitivity to length of the assimilation window . . . 39
- 3.6 Trading precision for ensemble size . . . 43
- 3.7 Conclusion . . . 47

4 Experiments with the SPEEDY general circulation model **49**
- 4.1 Introduction . . . 49

4.2 The SPEEDY model . . . 49
4.3 Data assimilation setup . . . 53
4.4 Perfect model results . . . 57
4.5 Imperfect model results . . . 62
4.5.1 Introducing model error: decreased diffusion . . . 62
4.5.2 Introducing model error: high resolution nature run . . . 63
4.6 Conclusion . . . 66

5 Reducing precision in 4D-Var 71
5.1 Introduction . . . 71
5.2 Reducing precision in 4D-Var . . . 72
5.3 Experimental setup . . . 74
5.3.1 Data assimilation setup . . . 74
5.3.2 The two-layer quasigeostrophic system . . . 76
5.4 Tangent-linear/adjoint model asymmetry . . . 78
5.5 Effects of reducing precision on convergence . . . 79
5.6 Re-orthogonalising the residuals . . . 83
5.7 Conclusion . . . 87

6 Accelerating Legendre transforms through reduced-precision arithmetic 89
6.1 Introduction . . . 89
6.2 The Legendre transform . . . 92
6.3 Reducing precision in the Legendre transforms . . . 95
6.4 Forecast tests . . . 97
6.4.1 Deterministic forecast skill . . . 98
6.4.2 Probabilistic forecast skill . . . 101
6.4.3 Operational resolution deterministic forecast skill . . . 104
6.5 Computational cost estimates . . . 109
6.6 Conclusion . . . 111

7 Conclusion 113
7.1 Summary of key findings . . . 113
7.2 Avenues of future research in Earth-System modelling . . . 115
7.2.1 Alternatives to IEEE 754 floating-point numbers . . . 116
7.2.2 Machine learning and data assimilation . . . 118
7.2.3 Reduced-precision communication . . . 121

Bibliography 125

Chapter 1

Introduction

1.1 The state of the art in numerical weather prediction

The skill of weather forecasts has improved dramatically over the past 40 years. Today's 7-day forecasts of 500-hPa height have approximately the same skill as 5-day forecasts produced 20 years ago (Bauer et al., 2015). Furthermore, forecasting skill of specific extreme events has also improved noticeably. For example, the European Centre for Medium-Range Weather Forecasts (ECMWF) successfully predicted Hurricane Sandy around 7 days before it made landfall (McNally et al., 2014). Such improvements, brought about by small but incremental and sustained improvements to models and observations as well as advances in supercomputing over the course of several decades, have obvious benefits for building a global society more resilient to extreme weather, especially within the context of a changing climate.

In order to produce a weather forecast, an operational centre such as ECMWF begins by collecting data on the atmospheric state over a period of time leading up to the beginning of the forecast. This data is processed by a data assimilation algorithm to produce a best guess of the atmospheric state at $t = 0$ — what we call the "diagnostic" or "analysis" step. The analysis step relies on a data assimilation algorithm to combine observations with a prior state estimate of the atmosphere. This is then fed into the numerical model which is run forward up to the desired lead time to produce the forecast — the "prognostic" or simply the "forecast" step. This paradigm, first outlined by Bjerknes (1904) and developed

further by Richardson (1922), has been the predominant form of weather prediction since the advent of the electronic computer.

The numerical model takes the meteorological fields and advances them forward in time by integrating the primitive equations, which describe the conservation of momentum, thermodynamic energy, mass and moisture following a fluid parcel. These equations are, in general, not solvable directly and so we must employ temporal and spatial discretisation schemes in order to make them amenable to numerical integration by a computer. The equations are discretised and then projected onto a spatial grid (with a spacing of around 20 km horizontally, in the case of operational numerical weather prediction (NWP)), with any higher order terms truncated. Any physical process or component of the flow occurring at a scale below this truncation is therefore not explicitly resolved.

Unresolved processes cannot simply be neglected, however. Convective clouds, for example, play an important role in the climate system by, among other things, coupling temperature and humidity changes diabatically through condensation and evaporation (Arakawa, 2004). We attempt to account for the effects of these unresolved processes on the resolved flow by using physical parametrization schemes, which make assumptions about the statistical properties of the unresolved processes. These schemes produce a deterministic estimate of the contribution to the tendency of a model variable from unresolved physical processes. However, because the underlying equations are scale-symmetric (Palmer, 2019), the notions of "resolved" and "unresolved" are nothing more than computational fictions — the resolved variables do not uniquely determine the tendency contributions from unresolved processes. Instead these contributions should be drawn as samples from a probability distribution centred on the deterministic scheme's output (Arnold et al., 2013). In other words, we should use stochastically perturbed physical parametrizations.

The use of stochastic physics is especially pertinent for the case of ensemble forecasting. Ensemble forecasts are produced by running many integrations of the model starting from slightly different initial conditions, with the spread of initial conditions ideally representing the initial uncertainty in the atmospheric state due to the imperfect process of

data assimilation. An ensemble forecasting system allows one to quantify one's uncertainty about a forecasted outcome and therefore provides more value to decision makers than a deterministic system (Palmer, 2017). In an ideal ensemble prediction system, the spread (the standard deviation about the ensemble mean) should equal the error of the ensemble mean with respect to the truth, on average (Palmer et al., 2006). Achieving this balance requires the inclusion of all possible sources of forecast uncertainty — if a source of error goes ignored then the ensemble spread will systematically underestimate the error in the mean. This is the case when deterministic parametrizations are employed. To account for this, ECMWF employ the Stochastically Perturbed Parametrization Tendencies (SPPT) scheme (Palmer et al., 2009) and as a result achieve an almost perfect spread-error ratio (Haiden et al., 2018). To reflect the inherently probabilistic nature of NWP, centres are increasingly placing emphasis on ensemble forecasts instead of deterministic forecasts, even when the latter uses a higher resolution model. The shift towards ensemble techniques is also occurring for the data assimilation algorithms used for producing the initial condition analyses, as we discuss in Chapter 2.

1.2 Exascale weather prediction

As mentioned earlier, the skill of weather forecasts has increased dramatically over the past 30 years, and there are still plenty of gains to be made in the coming years. Most research activity is concentrated on increasing the horizontal resolution of weather models. ECMWF, for example, are planning to have a 5 km resolution ensemble prediction system by 2025. This is a significant upgrade over the current (at the time of writing) resolution of 18 km and is expected to improve predictions of high-impact weather (ECMWF, 2016). Such an upgrade would also allow assimilation of higher resolution observations.

Meanwhile, in the background of these developments in weather prediction, the nature of supercomputing is changing. Future supercomputers will be more *heterogeneous* — they will employ a variety of different kinds of hardware, instead of relying entirely on standard central processing units (CPUs). For example, a significant fraction of a supercomputer's

computational ability will come from graphics processing units (GPUs). Programmable hardware, such as field-programmable gate arrays (Düben et al., 2015) may also play a significant role. Furthermore, supercomputers, the largest of which are already comprised of several million cores, will likely become even more parallel in the coming years (Schulthess, 2015). We will enter the era of "exascale" supercomputing as computers are able to perform 10^{18} floating-point operations (FLOPs) per second. Meeting the goals of next generation weather prediction outlined above while supercomputing undergoes such a change is a major scientific and technical challenge (Lawrence et al., 2017), and one in which the choice of numerical precision will play a central role.

1.3 Why reduce precision?

The conventional wisdom is that, when performing a numerical computation, the highest numerical precision possible should be chosen to minimise the impact of rounding errors on the result. This then raises the question of why we would deliberately choose to use a lower precision than is available. We might expect the rounding errors from reducing precision to degrade our model's performance and therefore affect the skill of the forecast that it produces. Nonetheless, there are two main motivations for reducing precision as much as possible: a computational motivation and a physical motivation.

To understand the first motivation, we must first introduce the concepts of "memory-bound" and "compute-bound". Numerical computation at its simplest consists of two operations, data transfer and data processing. In order for the CPU to carry out a numerical operation on some data, it must first be transferred from memory to the CPU's registers. After the CPU has finished with the data, the result must then be transferred back to memory so that the next numerical operation can commence. If there are many operations for each datum transferred, such that there is always something for the CPU to do at any given moment, then we say that the operation is compute-bound. That is to say, if we want to accelerate the operation, we must focus on the CPU (e.g. increase the clock frequency). On the other hand, if there are relatively few operations compared with the amount of data

that must be transferred, then we say that the operation is memory-bound. In this case, the only way to accelerate the operation is to increase the memory bandwidth.

Weather and climate models are composed ultimately of many millions of these simple transfer-compute-transfer operations. Additionally, modern operational systems use supercomputers with potentially tens of thousands of computing nodes, and in this scenario we must also consider node-to-node communication. However, this is still fundamentally a transfer of data and so the same arguments apply.

Generally, weather and climate models are memory-bound (Schulthess et al., 2018), meaning that the computer spends more time transferring data between its various components than actually doing useful computations. This data-transfer bottleneck provides the computational motivation for reducing precision. Lower precision numbers have a smaller memory footprint. Therefore, when we lower precision we are effectively increasing the memory bandwidth. As a result, the overall run time of a weather or climate simulation is reduced.

The second motivation for reducing precision comes from physical intuition. There are two sources of error within weather forecasting: initial error and model error[1]. Initial error arises due to an imperfect knowledge of the starting state of the atmosphere due to both limited observational coverage and also imperfect use of these observations by data assimilation systems. Due to the nonlinear nature of error growth within the atmosphere, any initial error will grow, eventually rendering the forecast no better than a random draw from climatology. Model error is due to an imperfect representation of the atmosphere within the computer because of a finite spatial resolution (i.e. the discretisation error) and a simplification of modelled physical processes. Not all of the physical processes that are known to influence the weather, such as small-scale diabatic processes taking place within clouds, can be resolved by the model and so they must be represented by inherently imperfect parametrizations, as discussed earlier.

Both of these error sources are unavoidable and limit the horizon of time within which

[1]These two error sources are not strictly independent. In producing the initial condition, observations are assimilated into the model. Therefore, if model error is reduced then the initial error will also be reduced.

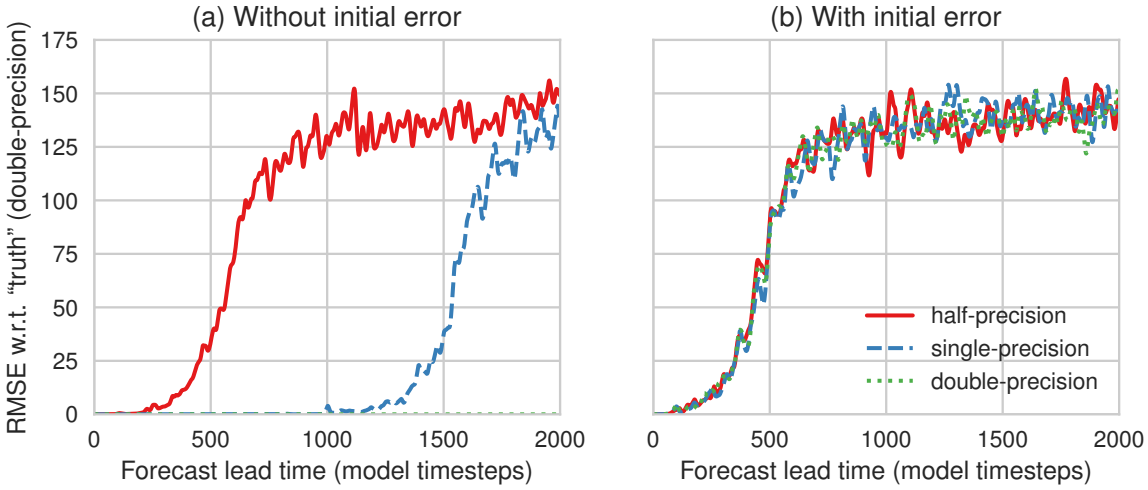


Figure 1.1: The forecast skill of half-, single- and double-precision Lorenz '63 models compared with a "truth", which is the same model at double-precision, (a) with no initial error and (b) with initial error.

we can forecast the weather with skill. Given this, using a high-precision will not necessarily give a better weather forecast than using a low-precision. This is illustrated in Figure 1.1. We performed many "forecasts" of the Lorenz '63 toy weather system (Lorenz, 1963) at half-, single-, and double-precision (we define these terms in Section 1.4, but to summarise, half-precision is lower than single-precision which is lower than double-precision) and compared with a "truth", in this case just a double-precision integration of the same model. When there is no initial error, the root-mean-square error (RMSE) of the half-precision model quickly saturates at the climatological level indicating a loss of skill, followed by the single-precision model (the error is zero for the double-precision model as it is identical to the truth). However, when we introduce just a small initial error, in this case a random and unbiased normally distributed perturbation of standard deviation 0.05 (a relative error of around 1%), the three models produce forecasts of the same skill. A similar argument applies to the case of model error, though with the difference that model error acts throughout the forecast and not just at the initial time.

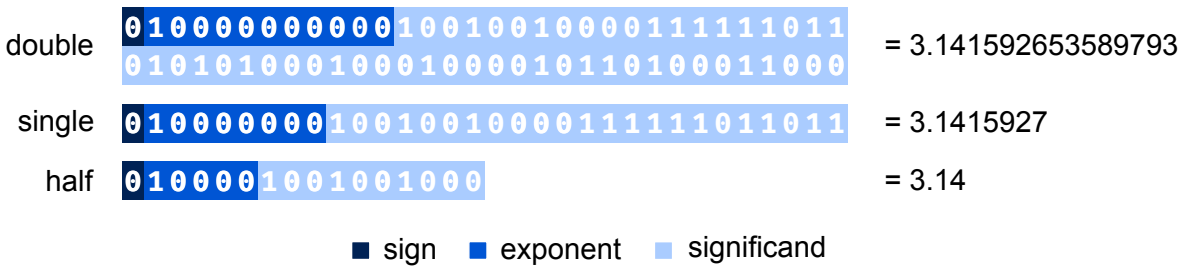


Figure 1.2: The value of π as represented by an IEEE 754 floating-point number at half-, single- and double-precision. The different components of the floating-point number, the sign, the exponent and the significand, are labelled.

1.4 Floating-point arithmetic

In this thesis we focus on IEEE 754 floating-point numbers (IEEE, 2008), also known simply as "floats". This number format is the standard for storing and computing real numbers on computers. Similar to scientific notation, a float, x, has three components: a sign, a significand and an exponent. The significand, also known as the mantissa or fraction, is a factor between 1 and 2 which stores the leading digits of the represented number. The exponent determines the represented number's order of magnitude. If s is the sign bit, $\{b_i\}$ are the n_s significand bits and $\{c_i\}$ are the n_e exponent bits, then the "normal" range of a float is defined by

$$\begin{aligned} x = \text{sign} \times \text{significand} \times \text{exponent} &= (-1)^s \left(1 + \sum_{i=1}^{n_s} b_i 2^{-i}\right) 2^E, \\ \text{where } E &= \sum_{i=1}^{n_e} c_i 2^{i-1} - 2^{n_e - 1} + 1. \end{aligned} \tag{1.1}$$

When all of the bits of the exponent are zero, the behaviour of a float changes. We call the range of numbers for which this is the case the "subnormal range". Essentially, subnormal numbers are defined by Equation 1.1 except without the "1" in the significand. The existence of subnormal numbers ensures that the difference between two numbers both with a zeroed exponent can be represented. They should not be used in normal conditions, however, as the relative precision drops off gradually as zero is approached. The IEEE 754 standard defines three standard levels of precision: half-, single- and double-precision defined by

$(n_s, n_e) = (10, 5)$, $(n_s, n_e) = (23, 8)$ and $(n_s, n_e) = (52, 11)$, respectively. The encoding of the value of π for half-, single- and double-precision floats is shown in Figure 1.2.

Floating-point numbers are so-called in contrast to fixed-point numbers as the number of digits after the decimal place can vary. Specifically, larger floats have fewer decimal places than smaller floats. The rounding error from storing a real number in a float is a relative error, which simplifies the analysis of rounding errors. For example, the float equivalent of the operation $x + y$ is to a good approximation

$$\text{fl}(x + y) = (x + y)(1 + \delta), \tag{1.2}$$

where δ is the rounding error bounded by $|\delta| \leq \epsilon$ and ϵ is the machine epsilon, given by $\epsilon = 2^{-n_s-1}$. Equation 1.2 uses the so-called "standard model" of floating-point rounding error analysis (Higham, 2002).

When reducing the precision of a floating-point operation, two effects can be anticipated. The first is an increase in the rounding error that occurs. For each significand bit that is removed, ϵ doubles and therefore larger rounding errors become possible. The second effect comes from a reduction in the exponent width, n_e. The exponent width determines the range of representable numbers. For example, the maximum double-precision number (11 exponent bits) is around 10^{308} whereas the maximum half-precision number (5 exponent bits) is only 65504. If a number larger than this maximum is stored in a floating-point variable, or results from a floating-point operation, an overflow occurs and the number is rounded to infinity.

Reducing the significand width, n_s, of the floats used in a model introduces a "graceful" error into each calculation, which can be compared with other sources of error, such as model error. However, reducing the exponent width raises the possibility of an overflow, from which a model is generally not recoverable. This procedure therefore requires careful treatment, as we detail in Chapter 6.

1.5 The Reduced Precision Emulator (rpe)

Current computers only natively support two levels of numerical precision: single- (32 bit) and double-precision (64 bit). Although there is hardware that also supports half-precision (16 bit) floats, namely GPUs, the vast majority of numerical models are written in standard Fortran which cannot run on this hardware without substantial code changes[2]. GPUs are therefore not suitable for our purposes, as we would like to assess the impact of reduced-precision arithmetic on existing models, written in Fortran, with as few code changes as possible. Furthermore, we would like to assess changes in model behaviour as the floating-point width is varied bit-by-bit, both in the exponent and the significand, and not just when changing between the three IEEE 754 floating-point formats. To this end, we employ a software emulator known as the Reduced Precision Emulator (rpe) (Dawson and Düben, 2017) which allows an arbitrary specification of both the significand and exponent widths.

```
real :: x

x = 1.0928
print *, x
! --> 1.0928
```

(a) The original code.

```
use rp_emulator
type(rpe_var) :: x
x%sbits = 6
x = 1.0928
print *, x
! --> 1.09375
```

(b) The code with the emulator.

Figure 1.3: An example of how the Reduced Precision Emulator (rpe) is used within Fortran.

The rpe library defines a Fortran derived-type, rpe_var, which can be used as a drop-in replacement for a Fortran real variable, as illustrated in Figure 1.3. This derived-type is essentially a double-precision float bundled with an integer, called "sbits", specifying the significand width. The rpe library works by overloading the assignment operator. Whenever a value is assigned to an rpe_var variable, a subroutine is called which manually truncates the significand of the value before it is placed in the variable, according to the value

[2]Note, however, that the Japanese Fugaku supercomputer, coming online in 2020, will allow half-precision floating-point arithmetic within Fortran codes through an extension of the Fortran standard (Satoshi Matsuoka, personal communication, April 2019).

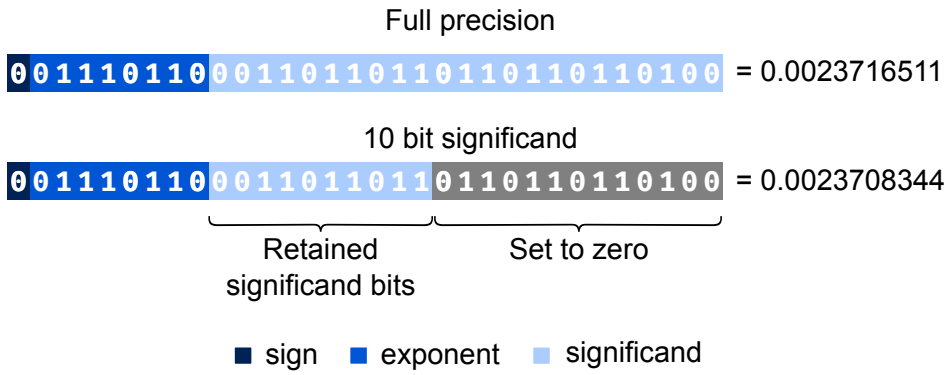


Figure 1.4: An example of the significand truncation carried out by the rpe library on a particular value in the case where 10 significand bits are retained. Note that rpe actually uses 64 bit floats as the underlying storage format, not 32 bit floats as shown here. Figure adapted from (Dawson and Düben, 2017).

of sbits. This truncation procedure is illustrated in Figure 1.4. The library also allows the exponent width to be reduced, though currently only the half-precision exponent (5 bits) is supported. If the relevant flag is activated, then a check is performed on the output of the truncation procedure. If the value is outside the range of a half-precision float then an overflow or underflow floating-point exception is raised.

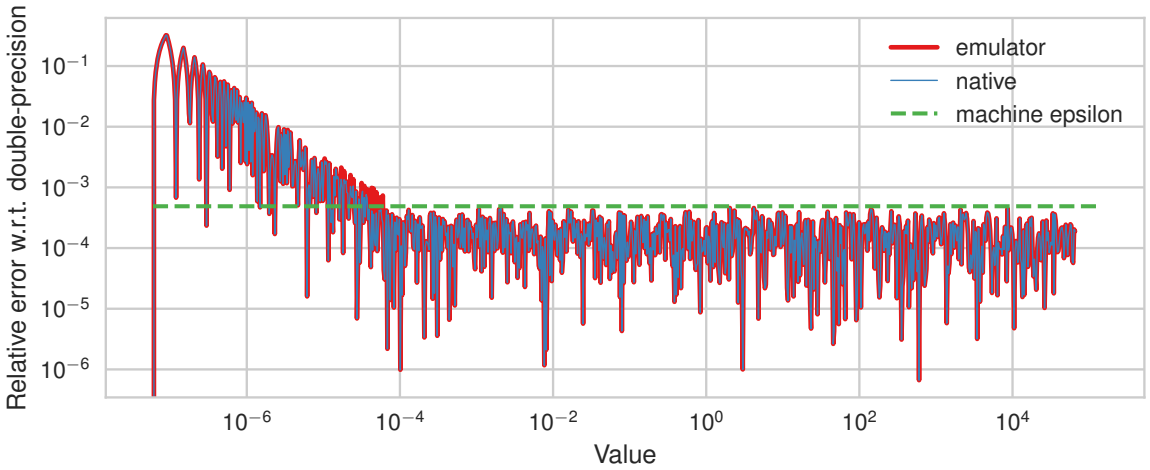


Figure 1.5: A verification of the emulated half-precision floating-point format provided by the rpe library.

The rpe library is, in principle, fully capable of emulating the half-precision floating-point format. To verify this we compared the truncation procedure of rpe with that of hardware that natively supports half-precision arithmetic, namely an NVIDIA P100 GPU.

We generated 1024 double-precision floats logarithmically and equally spaced between the smallest and largest representable half-precision values. We then placed these numbers into half-precision containers, triggering a rounding, and computed the relative error of these numbers with respect to the original numbers. This is shown in Figure 1.5. The rounding error is bounded by the machine epsilon within the "normal" range, as predicted by the standard model, equation 1.2. The relative error increases in the "subnormal" range, as expected. For almost all numbers considered, the emulator gave an identical rounding error to the GPU. The only difference occurred for some subnormal numbers around 10^{-5} where the rounding behaviour is not entirely correct (Andrew Dawson, personal communication, September 2018). However, this only affects these subnormal numbers by a small amount, which are in any case not often reached during normal arithmetic operations. We may therefore be confident that the emulator is working correctly.

1.6 Outline of thesis

This study aims to assess the potential of reduced-precision arithmetic to accelerate atmospheric models, specifically those used in data assimilation. In Chapter 2 we summarise the theory behind the two pillars of modern data assimilation, four dimensional variational assimilation (4D-Var) and the ensemble Kalman filter (and its variants). We then consider the use of reduced-precision arithmetic within a toy data assimilation system based on the Lorenz '96 system in Chapter 3. We continue the investigation of the ensemble Kalman filter with an intermediate complexity general circulation model, SPEEDY, in Chapter 4. In Chapter 5, we present an investigation of the effects of single-precision arithmetic on the convergence of a 4D-Var minimisation scheme. In Chapter 6, we demonstrate how half-precision arithmetic could be used to accelerate an expensive kernel within the ECMWF model, the Legendre transform, and assess the impact of this change on the forecasting skill. Finally, we present overall conclusions in Chapter 7 along with some suggestions for future studies and a discussion of upcoming techniques by which NWP and data assimilation systems might be improved even further.

1.7 Statement of originality

In Chapter 6, the Integrated Forecasting System (IFS) experiments were prepared and submitted in cooperation with Matthew Chantry. Matthew provided a modified version of OpenIFS from which the Legendre transform experiments were developed, and also ported changes to the OpenIFS code to the IFS. Apart from these exceptions, all of the work in this thesis is entirely my own.

1.8 Publications

The chapters in this thesis are derived from the following publications.

- Chapter 3 is based on Sam Hatfield, Aneesh Subramanian, Tim Palmer, and Peter Düben. Improving weather forecast skill through reduced precision data assimilation. *Monthly Weather Review*, 146:49–62, 2018b. doi: 10.1175/MWR-D-17-0132.1.
- Chapter 4 is based on Sam Hatfield, Peter Düben, Matthew Chantry, Keiichi Kondo, Takemasa Miyoshi, and Tim Palmer. Choosing the optimal numerical precision for data assimilation in the presence of model error. *Journal of Advances in Modeling Earth Systems*, 2018a. doi: 10.1029/2018MS001341.
- Chapter 5 is based on a manuscript in preparation for publication.
- Chapter 6 is based on Sam Hatfield, Matthew Chantry, Peter Düben, and Tim Palmer. Accelerating High-Resolution Weather Models with Deep-Learning Hardware. In *Proceedings of the Platform for Advanced Scientific Computing Conference - PASC '19*, pages 1–11. ACM Press, 2019. doi: 10.1145/3324989.3325711. The author was the recipient of the PASC Best Paper Prize for this submission. This paper will be developed for further publication as a journal article.

Chapter 2

Data assimilation theory

2.1 Introduction

Numerical weather prediction (NWP) is an initial value problem. This means that, even if we use a perfect model, the time in which our forecasting system is skillful will be limited if we do not specify an accurate initial condition. Data assimilation, the process by which this initial condition is constructed, is therefore of central importance. This is supported by the fact that assimilation and forecasting are allotted approximately equal time in the European Centre for Medium-Range Weather Forecasts's (ECMWF's) operational "time critical path" (one hour each).

Early data assimilation algorithms simply attempted to automate on a computer the so-called "subjective analysis" procedure, in which synoptic scale weather charts were drawn by hand based on observations (Daley, 1991; Panofsky, 1949). The label "analysis" is still used today to refer to the data assimilation product. When developing these techniques it was quickly appreciated that a prior guess of the atmospheric state before considering new observations, known as the "background", could improve the analysis accuracy (Gilchrist and Cressman, 1954; Bergthórsson and Döös, 1955). Nowadays, many data assimilation algorithms follow this basic paradigm — taking a first-guess, or background, subtracting this from the observation vector (to produce the so-called innovation) and then, taking account of the error covariance matrices of the background and observations, constructing a correction to be added to the background to produce the analysis.

In this thesis we focus on the two currently most popular data assimilation algorithms: 4D-Var and the ensemble Kalman filter (EnKF). Both algorithms take observations across a window of time and a prior estimate of the physical state and fuse these to produce an improved state estimate. The algorithms differ substantially in formulation, however. For example, 4D-Var employs linearised models to minimise a cost function that explicitly measures the error of a proposed analysis. On the other hand, the EnKF analysis procedure consists of finding the linear combination of an ensemble of background states that best matches the observations. In Section 2.2 we introduce the basic problem that data assimilation tackles, in Section 2.3 we describe the ensemble Kalman filter and two of its derivatives, in Section 2.4 we describe 4D-Var, both the standard form and the incremental form, and finally we conclude in Section 2.5.

2.2 The data assimilation problem

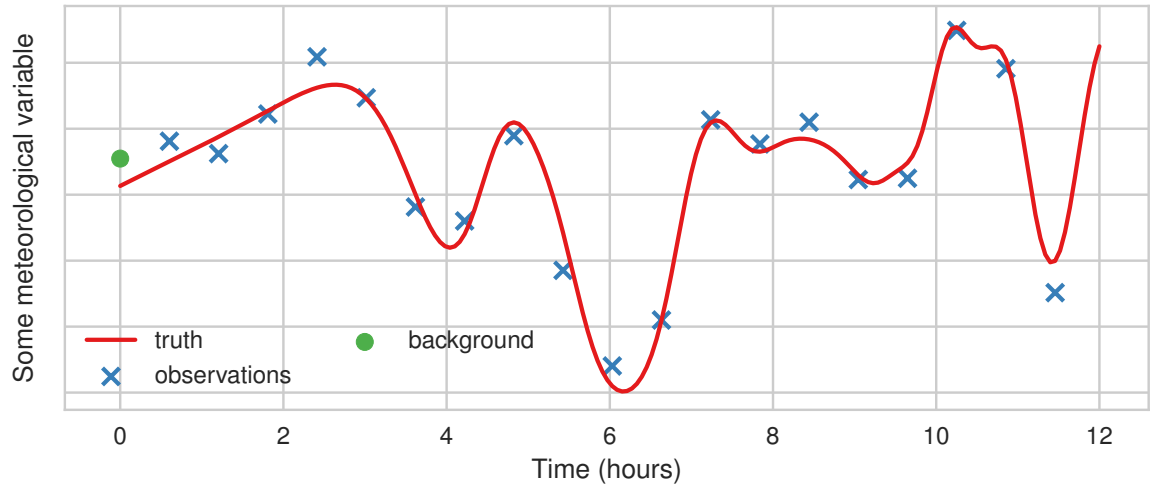


Figure 2.1: An illustration of the basic data assimilation problem: finding the model trajectory that minimises the distance to the observations across the assimilation window, as well as the distance to the background.

Before introducing the data assimilation algorithms, we first define the data assimilation problem. Over a window of time leading up to the beginning of the forecast we receive observations from various instruments distributed around the globe and in space. This window is typically around 12 hours for atmospheric data assimilation but several days for

ocean data assimilation (for the ocean, errors grow more slowly and observational coverage is poorer than in the atmosphere, so a longer window is appropriate) (Fowler and Lawless, 2016). We denote the observations by $\{\boldsymbol{y}_i, \boldsymbol{y}_i \in \mathbb{R}^p\}$ where i is the time index and p is the number of scalar observations available at each observation time. These observations are spatiotemporally sparse, so we do not have full coverage of the model grid. Furthermore, they suffer from both instrument errors and representativity errors (namely that a point observation does not represent an entire grid box potentially tens of kilometres wide). The random component of these errors can be represented by a matrix $\mathbf{R} \in \mathbb{R}^{p \times p}$, known as the observation error covariance matrix. If we assume that the error covariances do not change in time (an assumption we may have to relax if observation errors display, for example, a diurnal cycle), then $\mathbf{R}$ can be defined from any $\boldsymbol{y}_i$ as

$$\mathbf{R} = E[(\boldsymbol{y}_i - E[\boldsymbol{y}_i])\,(\boldsymbol{y}_i - E[\boldsymbol{y}_i])^\intercal] \tag{2.1}$$

where $E[\boldsymbol{x}]$ denotes the expectation value of $\boldsymbol{x}$. Here, we assume that observations are regularly distributed in time and that there are m model time steps between observation times.

In addition to the observations, we also have a background estimate of the atmospheric state, $\boldsymbol{x}_0^b \in \mathbb{R}^n$ where n is the dimension of the model state vector, valid at the beginning of the assimilation window. This is a prior estimate of the atmospheric state, before taking the observations into account. Some observation types, such as satellite radiances, cannot uniquely identify the atmospheric state — there can be infinitely many atmospheric states consistent with a particular satellite image. The background term essentially suggests one atmospheric state as more plausible as others and therefore allows more information to be extracted from observations. We denote the background error covariance matrix by $\mathbf{P}_0^b \in \mathbb{R}^{n \times n}$. Note that, for the EnKF, we actually have a background *ensemble* $\{\boldsymbol{x}_{k,0}^b, k = 1 \ldots K\}$ where K is the number of ensemble members. A sample of the background error covariance matrix can be constructed from the ensemble.

The goal of the data assimilation algorithm is to produce an atmospheric state estimate valid at the end of this assimilation window which makes best use of the observations and the background. This analysis can then be used as the initial condition for a weather forecast. The problem we describe is illustrated in Figure 2.1 for some fictional meteorological variable at a model grid point. In addition to the observations and the background, we have two so-called "forward operators". The first is the observation operator, $\boldsymbol{H} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ which maps a model state vector to the observation that would be obtained were the atmosphere to be described by that state vector (and there was no observation error). For example, for satellite observations $\boldsymbol{H}$ would contain a radiative transfer model that produces a simulated satellite image from an atmospheric model state vector. The second is the model operator, $\boldsymbol{M}_m : \mathbb{R}^n \rightarrow \mathbb{R}^n$ which advances a model state vector forward by m model time steps, i.e.

$$\boldsymbol{x}_m = \boldsymbol{M}_m \left(\boldsymbol{x}_0 \right). \tag{2.2}$$

For simplicity, we assume in this chapter that there is no model error, in other words that the evolution of both nature and the model can be described by the operator $\boldsymbol{M}_m$. Model error will in fact be a crucial factor in investigating a reduction of precision but, since the means for dealing with model error differ substantially between the EnKF and 4D-Var, we defer a discussion of this factor to the relevant chapters. We now proceed to describe the EnKF approach to data assimilation.

2.3 Ensemble Kalman filter

The EnKF, first proposed by Evensen (1994) and developed further by Houtekamer and Mitchell (1998), consists of two steps: an assimilation step and a forecast step. An assimilation step is performed whenever observations become available and consists of constructing an analysis from a background and these observations. A forecast step is performed in between observation times, in which the analysis is integrated forward in time to become the background for the next assimilation step. As mentioned previously, the EnKF assimilation

step only gives an analysis valid at one instant of time. However, by performing analysis and forecast steps repeatedly in a cycle, we are able to assimilate observations over a period of time. The analysis at the end of the window will therefore implicitly contain information from all of the previous observations.

At the time when observations $\boldsymbol{y}_i$ are valid, we perform an assimilation step by first calculating the sample background error covariance matrix from the background ensemble using

$$\widetilde{\mathbf{P}}_i^b = \frac{1}{K-1} \sum_{k=1}^{K} \left(\boldsymbol{x}_{k,i}^b - \overline{\boldsymbol{x}}_i^b\right) \left(\boldsymbol{x}_{k,i}^b - \overline{\boldsymbol{x}}_i^b\right)^\mathsf{T} \tag{2.3}$$

where $\overline{\boldsymbol{x}}_i^b$ is the background ensemble mean. Then, we apply the following update to each member of the background ensemble to produce the analysis ensemble $\{\boldsymbol{x}_{k,i}^a, k = 1 \dots K\}$:

$$\boldsymbol{x}_{k,i}^a = \boldsymbol{x}_{k,i}^b + \mathbf{K}\left(\boldsymbol{y}_i + \boldsymbol{\epsilon}_k - \boldsymbol{H}(\boldsymbol{x}_{k,i}^b)\right), \tag{2.4}$$

where $\boldsymbol{\epsilon}_k$ is a random vector drawn from the same distribution as the observation error and $\mathbf{K}$ is the Kalman gain matrix, given by

$$\mathbf{K} = \widetilde{\mathbf{P}}_i^b \mathbf{H}^\mathsf{T} \left(\mathbf{H} \widetilde{\mathbf{P}}_i^b \mathbf{H}^\mathsf{T} + \mathbf{R}\right)^{-1} \tag{2.5}$$

and $\mathbf{H}$ is the observation operator linearised about $\overline{\boldsymbol{x}}_i^b$. The Kalman gain matrix determines how much the background ensemble is corrected by the observations. Consider the case where $\boldsymbol{H} = \mathbf{I}$ and $\mathbf{R}$ contains only zeros. Then, $\boldsymbol{x}_{k,i}^a = \boldsymbol{y}_i$, as would be desired if we had perfect observations. On the other hand, if $\mathbf{R}$ contains infinitely large elements then $\boldsymbol{x}_{k,i}^a = \boldsymbol{x}_{k,i}^b$ (in other words, the observations have been ignored), as would be desired if the observations were infinitely bad.

Early studies on the EnKF applied the above scheme without the noise term $\boldsymbol{\epsilon}_k$ (Evensen and van Leeuwen, 1996; Evensen, 1997). However, it was quickly recognised that the scheme generates underdispersive ensembles if all ensemble members assimilate the same observation. It is necessary to add a different random vector drawn from the same distribu-

tion as the observation error to the observation for each ensemble member (Burgers et al., 1998), so that each member assimilates a slightly different observation.

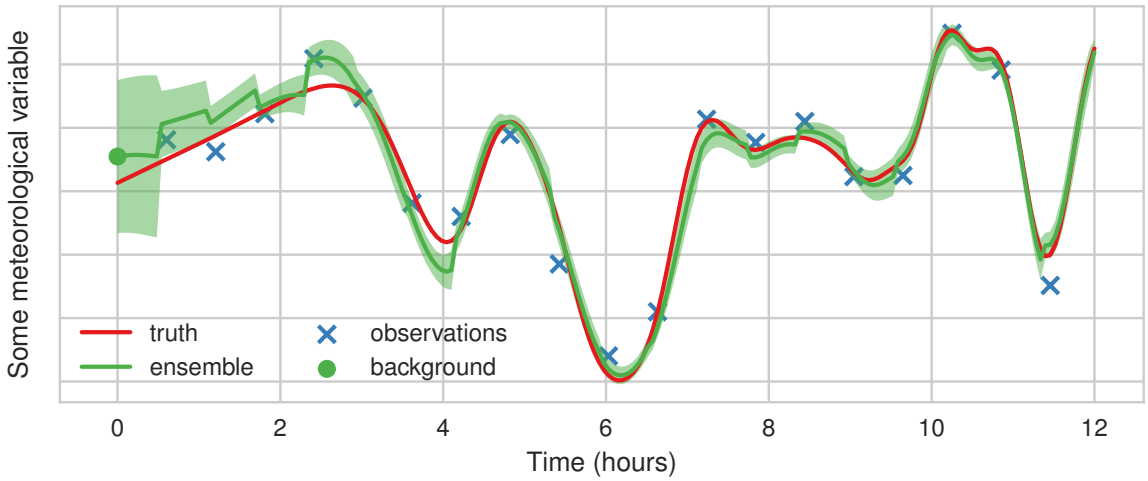


Figure 2.2: The ensemble Kalman filter applied to the example problem given in Figure 2.1. The shaded area shows the ensemble mean plus and minus one standard deviation.

Once the analysis ensemble is formed through Equation 2.4 it can be advanced forward in time to form the background ensemble for the next set of observations. We apply the following operation to each analysis ensemble member

$$\boldsymbol{x}^b_{k,i+1} = \boldsymbol{M}_m(\boldsymbol{x}^a_{k,i}). \tag{2.6}$$

This is also how the background ensemble at time $i = 1$ would be formed from the background ensemble at time $i = 0$, in the specific case of Figure 2.1.

Equations 2.4 and 2.6 form a closed loop analysis cycle. An example of such an analysis cycle is shown in Figure 2.2. At the beginning of the window, the ensemble spread is large, reflecting the high uncertainty about the true state. Then, as more and more observations are assimilated the ensemble spread decreases and the ensemble mean converges on the truth. By continuously assimilating observations we are able to synchronise the ensemble with the truth.

A key concern with the EnKF is sampling error in the ensemble estimate of the background error covariance matrix given in Equation 2.3. This matrix must be accurately

specified so that observational information is assimilated into the background ensemble correctly. If the diagonal elements of $\widetilde{\mathbf{P}}_i^b$ are too small, for example, then the error of the background will be underestimated leading to an underdispersive ensemble. This underdispersion will then be exacerbated at the next analysis step in the cycle, and so on for following analysis steps. Eventually the ensemble becomes so confident of its own background estimates that it begins to ignore the observations entirely rendering the analyses useless as estimates of the true state, a condition known as "filter divergence" (Hamill et al., 2001). We leave a detailed discussion of the procedures for avoiding filter divergence to later chapters.

The term "ensemble Kalman filter" refers both to a class of filter algorithms but also a particular member of that class, which we presented above. This particular EnKF is also sometimes called the "stochastic EnKF" (Houtekamer and Zhang, 2016) due to the use of perturbed observations. Below we discuss two alternative "deterministic EnKF" algorithms which avoid the use of perturbed observations, the ensemble square-root filter and the local ensemble transform Kalman filter. As a result, these algorithms generally give a better analysis for the same ensemble size than the stochastic EnKF, as they avoid the issue of sampling error in the perturbed observations which is especially prevalent for small ensemble sizes. These two algorithms differ from the stochastic EnKF presented above only in the assimilation step.

2.3.1 Ensemble square-root filter

The ensemble square-root filter (EnSRF) (Whitaker and Hamill, 2002; Tippett et al., 2003) defines a separate assimilation procedure for the background ensemble mean and the perturbations about that mean (the ensemble with the mean subtracted from each member) which are stored in the columns of a matrix $\mathbf{X}_i^b$. The ensemble mean is updated by Equation 2.4 with $\boldsymbol{\epsilon}_k = 0$. The perturbation matrix, however, is updated with a modified Kalman gain $\widetilde{\mathbf{K}}_i$ as

$$\mathbf{X}_i^a = \left(\mathbf{I} - \widetilde{\mathbf{K}}_i \mathbf{H}\right) \mathbf{X}_i^b \tag{2.7}$$

where

$$\widetilde{\mathbf{K}}_i = \widetilde{\mathbf{P}}_i^b \mathbf{H}^\intercal \left[\left(\sqrt{\mathbf{H}\widetilde{\mathbf{P}}_i^b \mathbf{H}^\intercal + \mathbf{R}} \right)^{-1} \right]^\intercal \left[\sqrt{\left(\mathbf{H}\widetilde{\mathbf{P}}_i^b \mathbf{H}^\intercal + \mathbf{R} \right)} + \sqrt{\mathbf{R}} \right]^{-1}. \tag{2.8}$$

The analysis ensemble can then be constructed simply by adding $\overline{\boldsymbol{x}}_i^a$ to each column of $\mathbf{X}_i^a$.

In Chapter 3 we assume that observation errors are uncorrelated, i.e. that $\mathbf{R}$ is diagonal. This permits us to treat the observation vector $\boldsymbol{y}_i$ *serially*, by assimilating each element of $\boldsymbol{y}_i$, $y_{i,l}$, one-by-one as a scalar observation. For a scalar observation, Equation 2.8 simplifies to

$$\widetilde{\mathbf{K}}_i = \left(1 + \sqrt{\frac{R_{ll}}{\mathbf{H}\widetilde{\mathbf{P}}_i^b \mathbf{H} + R_{ll}}} \right)^{-1} \mathbf{K}_i, \tag{2.9}$$

where R_{ll} is the (l,l)th element of $\mathbf{R}$. The $\mathbf{X}_i^a$ matrix for one scalar observation becomes the $\mathbf{X}_i^b$ matrix for the next scalar observation. We continue this procedure until all scalar observations have been processed.

2.3.2 Local ensemble transform Kalman filter

The ensemble transform Kalman filter similarly updates the ensemble mean and the perturbations about that mean with different expressions (Bishop et al., 2001). The perturbation matrix is updated by

$$\mathbf{X}_i^a = \mathbf{X}_i^b \mathbf{W}_i, \tag{2.10}$$

where

$$\mathbf{W}_i = \mathbf{C}_i (\mathbf{\Gamma}_i + \mathbf{I})^{-\frac{1}{2}} \tag{2.11}$$

and $\mathbf{C}_i \mathbf{\Gamma}_i \mathbf{C}_i^\intercal$ is the eigendecomposition of $\mathbf{X}_i^{b\intercal} \mathbf{H}^\intercal \mathbf{R}^{-1} \mathbf{H} \mathbf{X}_i^b$ (Tippett et al., 2003). The *local* ensemble transform Kalman filter (LETKF) computes an independent ensemble transform Kalman filter analysis at each gridpoint of a model (Hunt et al., 2007). Around each gridpoint, only observations within a radius of influence are considered for the analysis at that point. This breaks a global high-dimensional problem into a series of parallel low-dimensional problems (Ott et al., 2004).

2.4 4D-Var

Variational assimilation, in contrast to the EnKF, consists of first defining a cost function that measures the fit of a proposed analysis to the observations and the background, and then finding the minimum of that cost function to obtain the optimal analysis. In the case of 4D-Var, the following cost function is used:

$$J(\boldsymbol{x}_0) = \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}_0^b\|^2_{\mathbf{P}_0^b} + \frac{1}{2}\sum_{i=1}^{q}\|\boldsymbol{y}_i - \boldsymbol{H}(\boldsymbol{M}_{im}(\boldsymbol{x}_0))\|^2_{\mathbf{R}}, \tag{2.12}$$

where $\boldsymbol{x}_0$ is the model state at the beginning of the window, $\|\boldsymbol{z}\|^2_{\mathbf{A}}$ is defined as $\boldsymbol{z}^\intercal\mathbf{A}^{-1}\boldsymbol{z}$, q is the number of time instances in the assimilation window when observations are available and $\boldsymbol{M}_{im}$ denotes an integration of $i \times m$ model time steps. The first term of Equation 2.12 measures the distance from the proposed analysis $\boldsymbol{x}_0$ to the background $\boldsymbol{x}_0^b$, weighted by the inverse of the background error covariance matrix $\mathbf{P}_0^b$. The second term measures the distance between the observations and the corresponding points along the model trajectory resulting from an integration initialised at $\boldsymbol{x}_0$ (after having applied the observation operator), weighted by the inverse of the observation error covariance matrix. We denote the optimal value of $\boldsymbol{x}_0$ by $\boldsymbol{x}_0^*$. Of course, even though the control variable is $\boldsymbol{x}_0$, we actually desire the model state at the end of the window as this is the initial condition for the forecast. This can be easily obtained by integrating the model across the window starting from $\boldsymbol{x}_0^*$.

We typically use a gradient-based algorithm to minimise 2.12. However, in order to compute the gradient of the observation terms of 2.12 we need to linearise the model operator. To see why consider just one of the terms in the sum of Equation 2.12 which we denote by J_i^o:

$$J_i^o(\boldsymbol{x}_0) = \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{H}(\boldsymbol{M}_{im}(\boldsymbol{x}_0))\|^2_{\mathbf{R}}. \tag{2.13}$$

The gradient of this term with respect to the model state at observation time i, $\boldsymbol{x}_i$ =

$\boldsymbol{M}_{im}(\boldsymbol{x}_0)$, is

$$\frac{\partial J_i^o}{\partial \boldsymbol{x}_i} = \mathbf{H}^\intercal \mathbf{R}^{-1} \left(\boldsymbol{y}_i - \boldsymbol{H}(\boldsymbol{M}_{im}(\boldsymbol{x}_0)) \right), \tag{2.14}$$

where $\partial J / \partial \boldsymbol{z}$ is taken to mean the column vector where each element is the derivative of the scalar J with respect to the corresponding element in the vector $\boldsymbol{z}$ (i.e. $\partial J / \partial \boldsymbol{z} = \nabla J$). Equation 2.14 can be readily computed. However, we desire the gradient with respect to the model state at the *beginning* of the window, not at time i. For this we must use the chain rule, so

$$\frac{\partial J_i^o}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{x}_0} \frac{\partial J_i^o}{\partial \boldsymbol{x}_i}. \tag{2.15}$$

where $\partial \boldsymbol{x}_i / \partial \boldsymbol{x}_0$ is a matrix whose elements follow the pattern

$$\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{x}_0} = \begin{pmatrix} \frac{\partial x_{i,0}}{\partial x_{0,0}} & \cdots & \frac{\partial x_{i,n}}{\partial x_{0,0}} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_{i,0}}{\partial x_{0,n}} & \cdots & \frac{\partial x_{i,n}}{\partial x_{0,n}} \end{pmatrix} \tag{2.16}$$

and $x_{i,l}$ is the lth element of $\boldsymbol{x}_i$. It can be shown that this matrix is the transpose of the Jacobian of the model operator, evaluated about $\boldsymbol{x}_0$. We call the latter the "tangent-linear model" and the former the "adjoint model". Symbolically, if $\boldsymbol{M}_m$ integrates the model state forward by m model time steps, then $\mathbf{M}_m(\boldsymbol{x}_0)$ is the tangent-linear model and $\mathbf{M}_m^\intercal(\boldsymbol{x}_0)$ is the adjoint model. The latter two are usually implemented as separate models by differentiating the code underlying $\boldsymbol{M}_m$ line-by-line, either by hand or by automatic code differentiation (Lawless, 2013)

Returning to the gradient of $J_i^o(\boldsymbol{x}_0)$ with respect to $\boldsymbol{x}_0$, we may now write this as

$$\frac{\partial J_i^o}{\partial \boldsymbol{x}_0} = \mathbf{M}_{im}^\intercal(\boldsymbol{x}_0) \mathbf{H}^\intercal \mathbf{R}^{-1} \left(\boldsymbol{y}_i - \boldsymbol{H}(\boldsymbol{M}_{im}(\boldsymbol{x}_0)) \right). \tag{2.17}$$

This allows us to now write down the gradient of 2.12:

$$\frac{\partial J}{\partial \boldsymbol{x}_0} = \left(\mathbf{P}_0^b \right)^{-1} \left(\boldsymbol{x}_0 - \boldsymbol{x}_i^b \right) + \sum_{i=1}^{q} \mathbf{M}_{im}^\intercal(\boldsymbol{x}_0) \mathbf{H}^\intercal \mathbf{R}^{-1} \left(\boldsymbol{y}_i - \boldsymbol{H}(\boldsymbol{M}_{im}(\boldsymbol{x}_0)) \right). \tag{2.18}$$

This expression for the gradient would be given to a minimisation algorithm, which may require several hundred iterations in order to convergence. Each evaluation of the gradient requires one integration of the model operator $\boldsymbol{M}$ forwards across the window and one evaluation of the adjoint model $\mathbf{M}^{\mathsf{T}}$ "backwards" across the window. This is prohibitively expensive for a NWP model and so an alternative algorithm must be used that either requires fewer iterations and/or allows a lower resolution linearised model to be used. We present such an algorithm in the next subsection.

2.4.1 Incremental 4D-Var

Incremental 4D-Var, first presented in Courtier et al. (1994), consists of the following iterative procedure. First, we choose an initial guess for the model state at time $i = 0$, $\boldsymbol{x}_0^{(j)}$. For the first step of the iterative procedure, $j = 0$, we choose this to be the same as the background $\boldsymbol{x}_0^b$ (Lawless et al., 2005). We then evolve $\boldsymbol{x}_0^{(j)}$ across the window using the nonlinear model $\boldsymbol{M}$ and subtract the observations at the observation times to form the so-called innovations, $\boldsymbol{d}_i^{(j)} = \boldsymbol{y}_i - \boldsymbol{H}(\boldsymbol{M}_{im}(\boldsymbol{x}_0^{(j)}))$. Then, we define a cost function in terms of a perturbation to be added to $\boldsymbol{x}_0^{(j)}$, denoted by $\boldsymbol{\delta x}_0^{(j)}$:

$$\widetilde{J}(\boldsymbol{\delta x}_0^{(j)}) = \frac{1}{2}\|\boldsymbol{\delta x}_0^{(j)} - (\boldsymbol{x}_0^b - \boldsymbol{x}_0^{(j)})\|^2_{\mathbf{P}_0^b} + \frac{1}{2}\sum_{i=1}^{q}\|\mathbf{H}\mathbf{M}_{im}(\boldsymbol{x}_0^{(j)})\boldsymbol{\delta x}_0^{(j)} - \boldsymbol{d}_i^{(j)}\|^2_{\mathbf{R}}. \tag{2.19}$$

Here we have used the relation

$$\boldsymbol{\delta x}_i^{(j)} = \boldsymbol{M}_{im}\left(\boldsymbol{x}_0^{(j)} + \boldsymbol{\delta x}_0^{(j)}\right) - \boldsymbol{M}_{im}\left(\boldsymbol{x}_0^{(j)}\right) \approx \mathbf{M}_{im}(\boldsymbol{x}_0^{(j)})\boldsymbol{\delta x}_0^{(j)}. \tag{2.20}$$

The minimum of 2.19, $\boldsymbol{\delta x}_0^{(j),*}$, is found by a minimisation algorithm and then added to $\boldsymbol{x}_0^{(j)}$ to give the next guess state:

$$\boldsymbol{x}_0^{(j+1)} = \boldsymbol{x}_0^{(j)} + \boldsymbol{\delta x}_0^{(j),*}. \tag{2.21}$$

The procedure above, known as an "outer loop iteration", is then repeated a number of times. The number of iterations in the outer loop is normally fixed in advance at around 3

or 4 (we use 3 in Chapter 5).

Equation 2.19, unlike Equation 2.12, defines a convex quadratic cost function. This has two advantages over the "plain" form of 4D-Var presented earlier, namely that there is a single global minimum and efficient minimisation algorithms that assume quadraticity may be used (Bonavita et al., 2018). Typically, a Krylov subspace method such as conjugate gradients is used to minimise 2.19. This does not require an expression for the gradient, but instead an operator that evaluates the matrix-vector multiplication $\mathbf{S}\boldsymbol{w}$, where $\mathbf{S}$ is the Hessian of 2.19 and $\boldsymbol{w} \in \mathbb{R}^n$ is an arbitrary vector. This can be shown to be (Haben, 2011)

$$\mathbf{S}\boldsymbol{w} = \left(\mathbf{P}_0^b\right)^{-1}\boldsymbol{w} + \sum_{i=1}^{q} \mathbf{M}_{im}^{\intercal}(\boldsymbol{x}_0^{(j)})\mathbf{H}^{\intercal}\mathbf{R}^{-1}\mathbf{H}\mathbf{M}_{im}(\boldsymbol{x}_0^{(j)})\boldsymbol{w}, \tag{2.22}$$

As before, the adjoint model is present in Equation 2.22, though the tangent-linear model now also appears. Unlike the plain 4D-Var scheme, incremental 4D-Var allows simplified linear models to be used in computing the gradient. For example, we can use simplified physics or a lower resolution with respect to the model that is used to compute the innovations. This reduces the computational cost substantially, rendering it feasible for operational NWP. In Chapter 5, we investigate how reducing precision in the linearised models might also reduce the computational cost.

2.5 Conclusion

In this Chapter we have introduced the two standard data assimilation algorithms, the ensemble Kalman filter and 4D-Var. Of course, we have omitted many details and the algorithms as presented are not optimal for operational use. For the EnKF, the effect of a limited (computationally affordable) ensemble size on the quality of the sampled background covariance matrix has already been mentioned. Another important consideration is bias correction. Both observations and models can be biased, and the need for automatic correction of this bias has inspired a whole sub-field of data assimilation (Dee, 2005). Additionally, we did not mention how the background covariance matrix $\mathbf{P}_0^b$ of 4D-Var (incremental or not)

is created. In the EnKF, $\widetilde{\mathbf{P}}_i^b$ is derived from the ensemble and is therefore "flow dependent" (can evolve in time). However, in 4D-Var, we must prescribe a background covariance matrix, static or otherwise.

Many studies have been conducted comparing the EnKF with 4D-Var (e.g. Lorenc (2003); Kalnay et al. (2007); Miyoshi et al. (2010); Buehner et al. (2010)). Given the difference in formulation between the two algorithms, it is difficult to make a direct comparison. Any comparison should be careful to compare implementations of an equivalent computational cost. For example, it would be unfair to conclude that the EnKF is inherently inferior to 4D-Var due to the limited ensemble size effects if, for the same cost as the 4D-Var implementation, an adequate ensemble size to mostly eliminate these effects could be used. Nonetheless, some commentators have identified several advantage of 4D-Var over the EnKF. For example, Bonavita et al. (2015) compared the LETKF and 4D-Var within the ECMWF forecasting system and found that 4D-Var made better use of satellite radiance observations than the EnKF, especially in the stratosphere (though the exact reason why is the subject of ongoing research). On the other hand, the development and maintenance of the tangent-linear and adjoint models required by 4D-Var is a formidable task for an operational NWP model, requiring two full-time staff at ECMWF. The physical parametrization schemes in particular require careful treatment (Janisková and Lopez, 2012), though the UK Met Office avoid such issues by eliminating most of the physics from their "perturbation forecast model" (used instead of a tangent-linear model), at the cost of a larger analysis error (Rawlins et al., 2007). The choice of which data assimilation algorithm to use is therefore not just scientific but also economic, and this is reflected by the wide variety of algorithms used by operational centres around the world.

Instead of viewing these algorithms as mutually exclusive, it is more productive to view them as two ends of a spectrum, within which many different hybridisations are possible. Many attempts have been made to bridge the two algorithms, taking the best features from both (Bannister, 2016). There is, for example, a four-dimensional EnKF (Hunt et al., 2004) which is designed to assimilate observations across a window in a single analysis step. Ad-

ditionally, ECMWF make use of an ensemble of 4D-Var assimilations (the Ensemble of Data Assimilations), each member with differently perturbed observations, sea surface temperatures and model physics, in order to construct initial perturbations for use in the ensemble prediction system (Bonavita et al., 2016). Such hybridisations are the future of data assimilation and this is reflected by their increasing adoption in operational centres.

Chapter 3

Experiments with the Lorenz '96 system

3.1 Introduction

In this chapter we introduce some of the key concepts that we will be building on in later chapters, namely the interplay between numerical precision and uncertainty. Reducing precision naturally degrades the individual arithmetic operations comprising a model. However, given that these models are formulated in an inherently imperfect way, a reduction of precision will not necessarily affect the forecast skill measured with respect to observations. Here, we investigate these questions within an ensemble data assimilation framework, the serial ensemble square-root filter (Serial-EnSRF), introduced in Chapter 2. We also investigate the potential for a precision reduction to allow an increase in the ensemble size of the Serial-EnSRF, by reinvesting the computational savings from reducing precision. If a doubling, or even quadrupling, of the ensemble size could be delivered in an operational system, then this would substantially reduce the analysis error and therefore also the forecast error (Houtekamer et al., 2014).

To investigate these topics, we begin with a simplified model of atmospheric dynamics, the Lorenz '96 system (Lorenz and Emanuel, 1998; Lorenz, 2006). This system includes terms analogous to advection, damping, external forcing and coupling to small-scale motion. It is a simple and efficient system for generating spatiotemporally chaotic time series analogous to those displayed by prognostic variables in the atmosphere, and as a result it is a common test-bed for new data assimilation algorithms and ideas (e.g. Whitaker and Hamill (2002);

Ott et al. (2004); Nerger (2014)).

This chapter introduces the first instance of an Observing System Simulation Experiment (OSSE). An OSSE allows different assimilation algorithms to be compared and tested with complete freedom over the quantity and quality of observations. A "nature run" is first generated through a single integration of the best possible model available (the "truth model"), and observations of this nature run are then generated by adding random noise with prescribed statistics. These observations are fed into the assimilation system. The output from the assimilation system can then be verified against the (perfectly characterized) nature run.

In Sections 3.2 and 3.3 we introduce the Lorenz '96 system and assimilation setup in detail. In Section 3.4 we quantify the impact of reducing precision on the quality of the EnSRF analyses. In Section 3.5 we demonstrate how our results are sensitive to the length of the assimilation window. In Section 3.6 we show how precision can be traded for more ensemble members in the Serial-EnSRF. Finally, we conclude in Section 3.7.

3.2 The Lorenz '96 system

In our OSSE, we use the Lorenz '96 system for both the truth and forecast models. For the truth model, we use the three-tier variant of the Lorenz '96 system, introduced in Thornes et al. (2017). The tiers are labelled X, Y and Z and are analogous to large-, medium- and small-scale dynamics, respectively, in the atmosphere. Each tier is composed of a line of grid points with periodic boundary conditions. Each X grid point is connected to a number of Y grid points and each Y grid point is connected to a number of Z grid points. The forecast model used in the Serial-EnSRF is a two-tier variant of the Lorenz '96 system with a parametrization of the tendency of the missing Z tier. In this way, the difference between the forecasting and truth models is intended to mimic the difference between a real forecasting model and the atmosphere itself. The truth and forecast models are illustrated in Figure 3.1.

The three-tier system is described by the equations

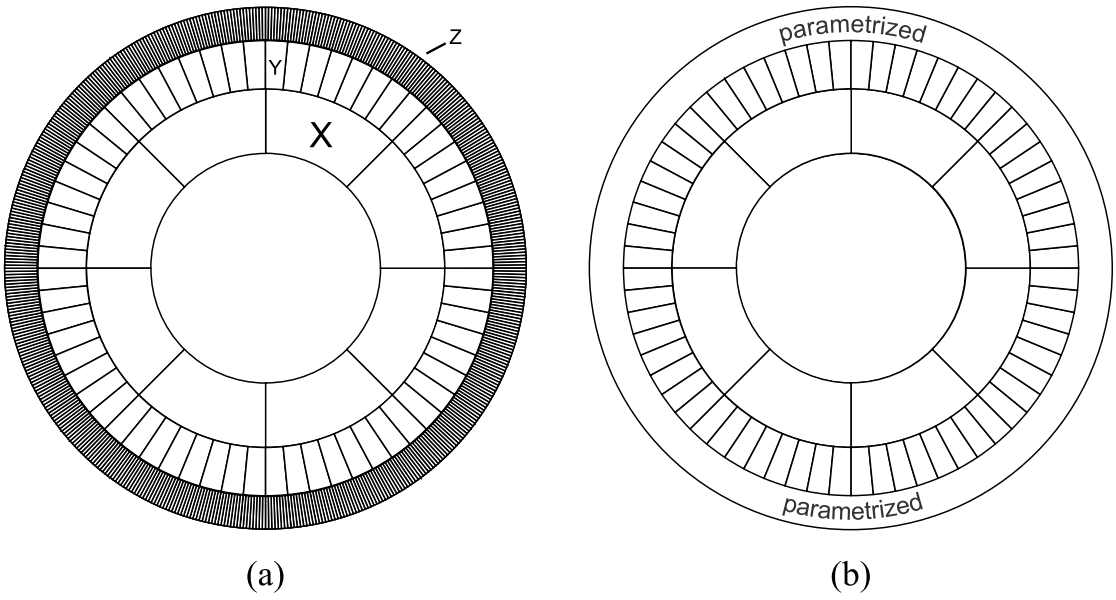


Figure 3.1: (a) The three-tier Lorenz '96 system, used to generate the nature run (b) The two-tier system, used as the forecast model. The inner tier is labelled "X", the central tier is labelled "Y" and the outer tier (which isn't present in the forecast model) is labelled "Z".

$$\frac{dX_k}{dt} = X_{k-1}\left(X_{k+1} - X_{k-2}\right) - X_k + F - \frac{hc}{b}\sum_{j=1}^{J} Y_{j,k}, \tag{3.1}$$

$$\frac{dY_{j,k}}{dt} = -cbY_{j+1,k}\left(Y_{j+2,k} - Y_{j-1,k}\right) - cY_{j,k} + \frac{hc}{b}X_k - \frac{he}{d}\sum_{i=1}^{I} Z_{i,j,k}, \tag{3.2}$$

$$\frac{dZ_{i,j,k}}{dt} = edZ_{i-1,j,k}\left(Z_{i+1,j,k} - Z_{i-2,j,k}\right) - g_Z eZ_{i,j,k} + \frac{he}{d}Y_{j,k}. \tag{3.3}$$

The indices i, j and k range from 1 up to I, J, and K for the Z, Y and X tiers, respectively. The number of X variables is given by I, the number of Y variables is given by IJ and the number of Z variables is given by IJK. In this case, I, J and K are all equal to 8. There are therefore 584 grid points in total, as shown in Figure 3.1a. The parameter F is a large-scale forcing term which determines the chaoticity of the model and is set to 20 in this case, placing the model firmly within the chaotic regime. The remaining parameters, h, c, b, e, d and g_Z allow the frequency and amplitude of oscillation of the three tiers, as well as the coupling between tiers, to be tuned. These parameters are tuned such that the

X tier oscillates slowly with a large amplitude, while the two tiers below vary more quickly at a lower amplitude. The parameter values are $h = 1$, $c = b = e = d = 10$ and $g_Z = 1$.

If Equations 3.1, 3.2 and 3.3 are defined as the "truth" then a complementary simplified forecast model can be constructed. We use a two-tier variant of the Lorenz '96 system similar to Equations 3.1 and 3.2, only the Z tendency is replaced with a parametrization term, U_p:

$$\frac{dY_{j,k}}{dt} = -cbY_{j+1,k}\left(Y_{j+2,k} - Y_{j-1,k}\right) - cY_{j,k} + \frac{hc}{b}X_k - U_p. \tag{3.4}$$

The term U_p is tuned to match the average tendency of the missing Z tier, for which we use "additive noise", given by

$$U_p = U_p^{\text{det}}(Y_{j,k}) + s_{j,k}, \tag{3.5}$$

where $U_p^{\text{det}}(Y_{j,k})$ is a deterministic polynomial function of the resolved variable $Y_{j,k}$ and $s_{j,k}$ is a random number generated from an autoregressive first-order stochastic process (i.e. red noise). For U_p^{det} we use a fourth-order polynomial:

$$U_p^{\text{det}}(Y_{j,k}) = c_4 Y_{j,k}^4 + c_3 Y_{j,k}^3 + c_2 Y_{j,k}^2 + c_1 Y_{j,k} + c_0. \tag{3.6}$$

The first-order stochastic process is generated by

$$s_{j,k} = \phi s_{j,k}^{\text{last}} + \sqrt{1-\phi^2}e_{j,k}, \tag{3.7}$$

where ϕ is the lag-1 autocorrelation (characterising the persistence of the random process), $s_{j,k}^{\text{last}}$ is the previous value of the process and $e_{j,k}$ is a random number normally distributed with zero mean and variance σ_e^2 (characterising the amplitude of the noise). The parameters ϕ and σ_e are tuned by matching the average behaviour of the two-tier system to the three-tier system, in a similar process to that described in Arnold et al. (2013) and Wilks (2005).

The values of the parameters used are shown in Table 3.1.

c_4	c_3	c_2	c_1	c_0	ϕ	σ_e^2
0.001892	-0.066811	0.131826	0.242742	0.039970	0.997	0.015876

Table 3.1: The parameters defining the parametrization of the missing Z tier in the forecast model.

Both the truth and forecast models are integrated with a 4th order Runge-Kutta scheme of timestep 0.005 nondimensional model time units (MTUs). For the red noise term we simply apply the same random forcing at each intermediate evaluation of Equation 3.4 in the Runge-Kutta scheme and only update the noise between full time steps.

3.3 Data assimilation setup

To perform data assimilation, we use the Serial-EnSRF. We derive observations of the Y variables from the nature run and assimilate these into the two-level model. As alluded to in Chapter 2, in order to use a practicable ensemble size (i.e. when the number of ensemble members is much less than the number of state variables) we must contend with the sampling errors that small ensembles introduce into the background covariance matrix $\widetilde{\mathbf{P}}^f$ and the resulting potential for filter divergence. In order to counter sampling errors, we use two covariance preprocessing techniques. The first technique is multiplicative covariance inflation, actually applied to the "scaled square root" of the background covariance matrix $\mathbf{X}^b$, where $\widetilde{\mathbf{P}}^b = \mathbf{X}^b \left(\mathbf{X}^b\right)^\intercal / (K-1)$ and K is the number of ensemble members, before the assimilation of observations:

$$\mathbf{X}^b_{\text{inf}} = r\mathbf{X}^b, \tag{3.8}$$

where r is a number slightly larger than 1 (for example, 1.01).

Inflation is used to counter the "inbreeding effect" in ensemble data assimilation that is present when the same ensemble is used to estimate the Kalman gain and the analysis ensemble (Houtekamer and Mitchell, 1998; Anderson and Anderson, 1999; Hamill et al., 2001). However, it also has a use when model error is present (Hamill, 2006). The treatment of the

EnKF in Chapter 2, as mentioned, did not consider model error. In reality, the background error covariance matrix estimated even from a very large ensemble will be deficient as it does not take into account the fact that the model is imperfect. The true covariance matrix will in fact be given by

$$\mathbf{P}^b_{\text{true}} = \widetilde{\mathbf{P}}^b + \mathbf{Q}, \tag{3.9}$$

where $\mathbf{Q}$, known as the model error covariance matrix, describes the statistics of the forecast error at the end of the forecast window (i.e. 6 hour lead-time forecast errors if data assimilation is performed every 6 hours), assuming perfect initial conditions. We attempt to compensate for the lack of an explicit $\mathbf{Q}$ matrix through a combination of stochastic parametrization (perturbations to the model integration throughout the forecast window) and covariance inflation. Although we did not attempt to quantify whether these techniques accurately match $\mathbf{Q}$, the fact that our data assimilation setup does not diverge suggests that we were successful.

The second technique for dealing with ensemble-derived error covariance matrices is covariance localization, which is used in order to filter out spurious long-range correlations in $\widetilde{\mathbf{P}}^b$ that tend to contaminate the analysis (Hamill et al., 2001; Houtekamer and Mitchell, 2001). The spurious correlations are a result of sampling error from an inadequate ensemble size, and so the required localization reduces as the ensemble size increases. Before observations are assimilated, a "localization vector", $\boldsymbol{\rho}$, is applied to the Kalman gain through an element-wise multiplication (a Schur product):

$$\mathbf{K}_{\text{loc}} = \boldsymbol{\rho} \circ \mathbf{K}. \tag{3.10}$$

The localization vector has a value of 1 at the grid point where the observation currently being assimilated is located. Moving away from this grid point, the localization vector reduces in magnitude and reaches 0 after a finite distance, known as the "localization scale". We use the "Gaspari-Cohn" function to achieve this, which is a polynomial approximation to a Gaussian with compact support (Gaspari and Cohn (1999), their Equation 4.19). This

localization technique is often referred to as "B localization" ("B" refers to another notation for the background covariance matrix, **B** (Greybush et al., 2011)). Note that B localization is usually applied to the full covariance matrix $\mathbf{P}^b$ instead of to **K**. In the case of sequential observation processing, the two techniques are equivalent.

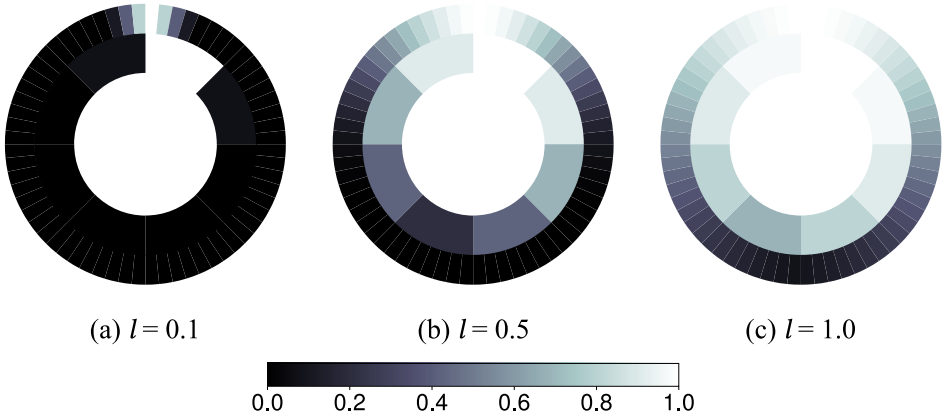


Figure 3.2: The localization vector, $\boldsymbol{\rho}$, applied to the Kalman gain when assimilating the first Y variable for three different length scales. The shading indicates the degree of localization.

Localization is applied not only to the covariances between Y grid points, but also to the covariances between Y grid points and X grid points. The localization between Y grid points depends simply on the distance (the number of Y grid points) between them. When localising between a Y grid point and an X grid point, first the X grid point associated with the Y grid point is determined. Then the localization is computed based on the distance separating the two X grid points. The localization scale is assumed to be the same function of ensemble size in both cases, and so both are tuned simultaneously through the parameter l, where the Y-Y localization scale is $l_{YY} = 26l$ and the X-Y localization scale is $l_{XY} = 8l$. Figure 3.2 shows several examples of the localization vector, $\boldsymbol{\rho}$, when assimilating the first Y variable.

Both the optimal inflation and the optimal localization are assumed to be functions of ensemble size, assimilation frequency and observation error, and so the inflation and localization are retuned with every change of one of these parameters. An example of the tuning process is illustrated in Figure 3.3. The gray region denotes parameter combinations that resulted in filter divergence. In this example, the optimal choice of inflation was 1.13

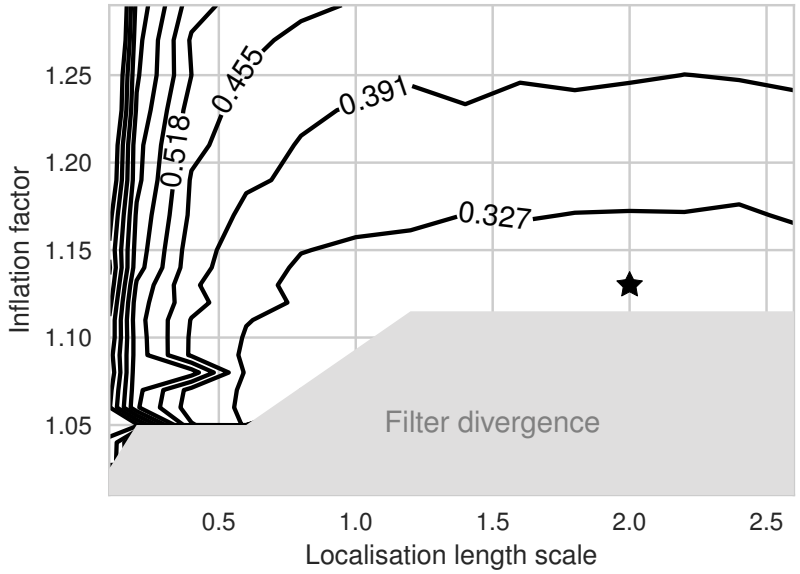


Figure 3.3: Ensemble mean root mean square error as a function of covariance inflation and localization parameters, for 160 member assimilation. The star denotes the optimal choice of parameters. The shaded area denotes parameter choices that lead to unstable filtering, with a high likelihood of filter divergence.

and the optimal localization length scale was 2.0, as the minimum error is located at this point, indicated by a star. On all occasions, we found that the optimal parameters for double, single and half-precision assimilation were roughly the same. Therefore, we only tune the system at double-precision, and use the same parameters for single and half-precision assimilation.

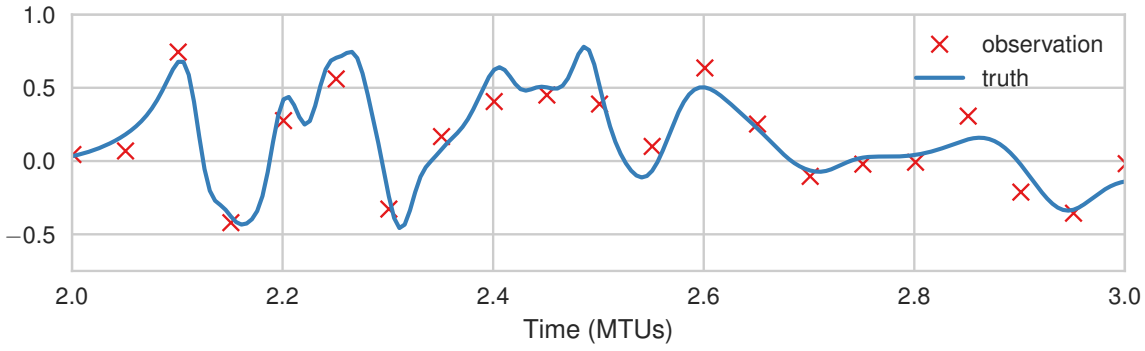


Figure 3.4: A time series comparison of the signal displayed by the $Y_{1,1}$ grid point and the corresponding observational signal "seen" by the assimilation system. The observation error is 30% of the natural root-mean-square variability.

In order to make a fair comparison between full and reduced precision results, it is im-

portant to choose a reference scenario that accurately represents an operational setup. For the operational EnKF at the Canadian Meteorological Centre, for example, assimilation is typically performed every 6 hours (Houtekamer et al., 2014). Equivalently, we assimilate every 0.05 MTUs. This time scale was determined by computing the error-doubling time of the large scale (X) tier, and comparing with the error-doubling time of the real atmosphere (approximately 1.5 days (Lorenz, 2006)). It should be noted, however, that this time scale is only approximate, and the true value could be anywhere between 0.025 and 0.075 MTUs. We use 40 ensemble members for our reference scenario and double-precision for all calculations. Additionally, observation errors for the Y variables are drawn from a zero-mean normal distribution with a standard deviation, σ, of 0.1, which is approximately 30% of the natural root-mean-square (RMS) variability of these variables. Figure 3.4 shows an excerpt of the evolution of the first Y variable from a nature run and the observations derived from this run. Limited results will be given for two other levels of observation error, $\sigma = 0.0035$ and $\sigma = 0.017$, which are approximately 1% and 5% of the natural RMS variability, respectively.

3.4 Reduced-precision analyses

We consider three levels of numerical precision in our investigation: double (64 bits), single (32 bits) and half (16 bits). As discussed in Chapter 1, we emulate half-precision variables using the Reduced Precision Emulator (rpe). We also perform some limited experiments with a fixed double-precision exponent size and flexible significand width, using the same emulator. In the experiments that follow, all calculations in the main loop (forecast/update cycle) are performed in the specified precision. The nature run is always generated using double-precision. Thornes et al. (2017) found error metrics particularly sensitive to the numerical precision of the X tier in the same Lorenz '96 system used here, reasoning that large scale dynamics are more important than small scale dynamics and therefore require more precise computations. Here, we do not consider such a scale-selective precision, and both tiers of the forecast model are performed at the same precision.

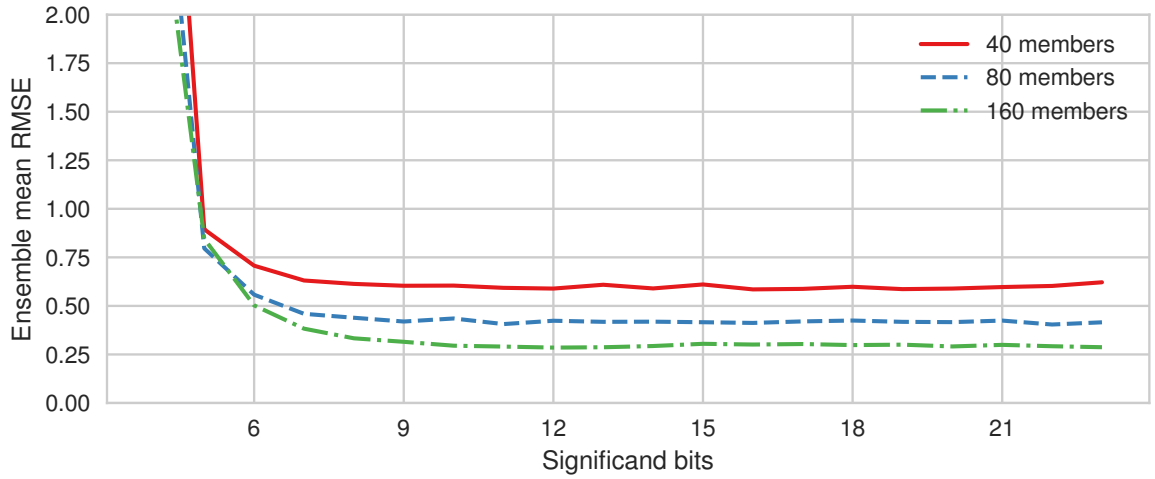


Figure 3.5: Ensemble mean RMSE averaged over 250 MTUs at various levels of numerical precision, determined by the number of bits in the significand. Three sizes of ensemble are considered. The exponent is kept at the double-precision value, 11 bits.

Figure 3.5 shows the analysis ensemble mean root mean square error (RMSE) averaged over 5,000 analyses of a data assimilation cycle for three different ensembles, 40, 80 and 160 members, as a function of n_s, the number of significand bits. Assimilation is performed every 0.05 MTUs (every 10 timesteps), so the total length of simulations is 250 MTUs or around 4 atmospheric years. The ensemble mean RMSE at time k is given by

$$\text{RMSE}_k = \sqrt{\frac{1}{n_x}\sum_{i}^{n_x}\left(\overline{x}^a_{i,k} - x^{\text{t}}_{i,k}\right)^2} \tag{3.11}$$

where n_x is the number of X variables (8 in this case) and $\overline{x}^a_{i,k}$ and $x^{\text{t}}_{i,k}$ are the ith X variables of the analysis and truth at analysis time k, respectively. The RMSE is proportional to the distance between the analysis ensemble mean and the truth, and therefore gives the simplest measure of the performance of an assimilation system. No half-precision range checking is performed in this case, so the exponent is still represented by 11 bits. The significand width can be reduced to around 8 bits with no discernible increase in the ensemble mean RMSE. Beyond this point, the error explodes and the system becomes unstable. This indicates that half-precision assimilation may be possible, as the half-precision significand width is 10 bits. In order to verify this, it is necessary to also emulate a 5 bit exponent. In

all of the following results of this chapter, we do this by performing half-precision range checking after all calculations. We can therefore be certain that we are correctly emulating the IEEE754 half-precision format.

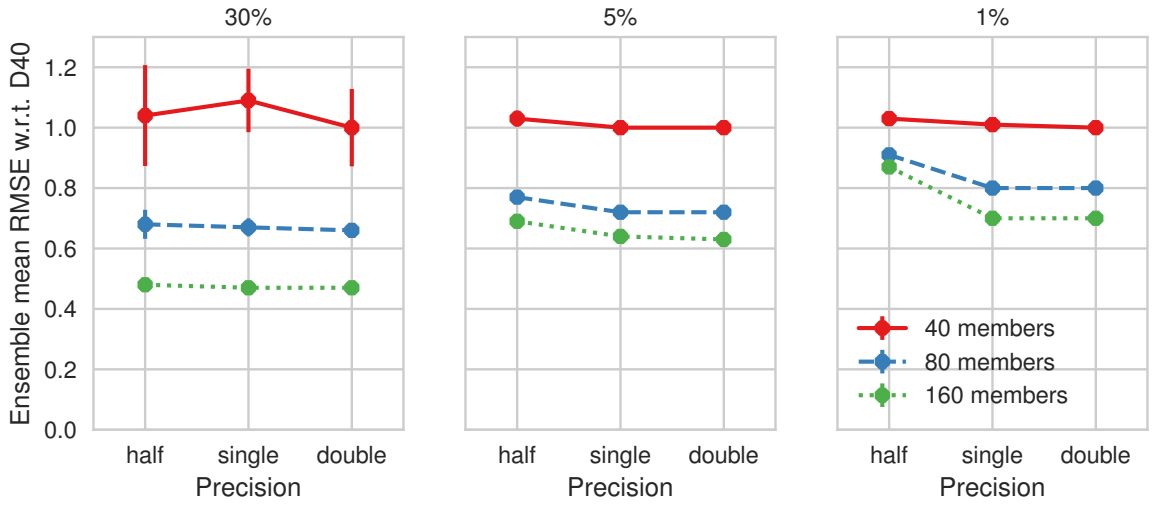


Figure 3.6: RMSE as a function of numerical precision, ensemble size and observation error. The observation error is stated as a percentage of the natural variability of the root-mean-square of the variable, 30%, 5% and 1%, from left to right. Assimilation is performed at the "normal rate": every 0.05 MTUs (every 10 timesteps).

Figure 3.6 shows the analysis ensemble mean RMSE with respect to that of the double-precision, 40 member reference, "D40", as a function of precision for three ensemble sizes and three levels of observation error. Values are stated as a fraction of the ensemble mean RMSE of the D40 reference, for each level of observation error. The values shown were obtained by averaging over 10 experiments of 250 MTUs (5,000 analyses) each, and the error bars were computed from the standard deviation of the same 10 experiments. When the observation error is 30% of the natural RMS variability, there is no noticeable degradation in the quality of the analyses as precision is reduced from double to half-precision, for either of the three ensemble sizes considered. When lowering the observation error to 5% of the natural RMS variability, the gains from increasing ensemble size diminish. Additionally, there is a slight, but significant, degradation in the analyses when using half-precision, which increases further as the observation error is lowered to 1% of the natural RMS variability (though we note that in both cases half-precision with 80 and 160 members

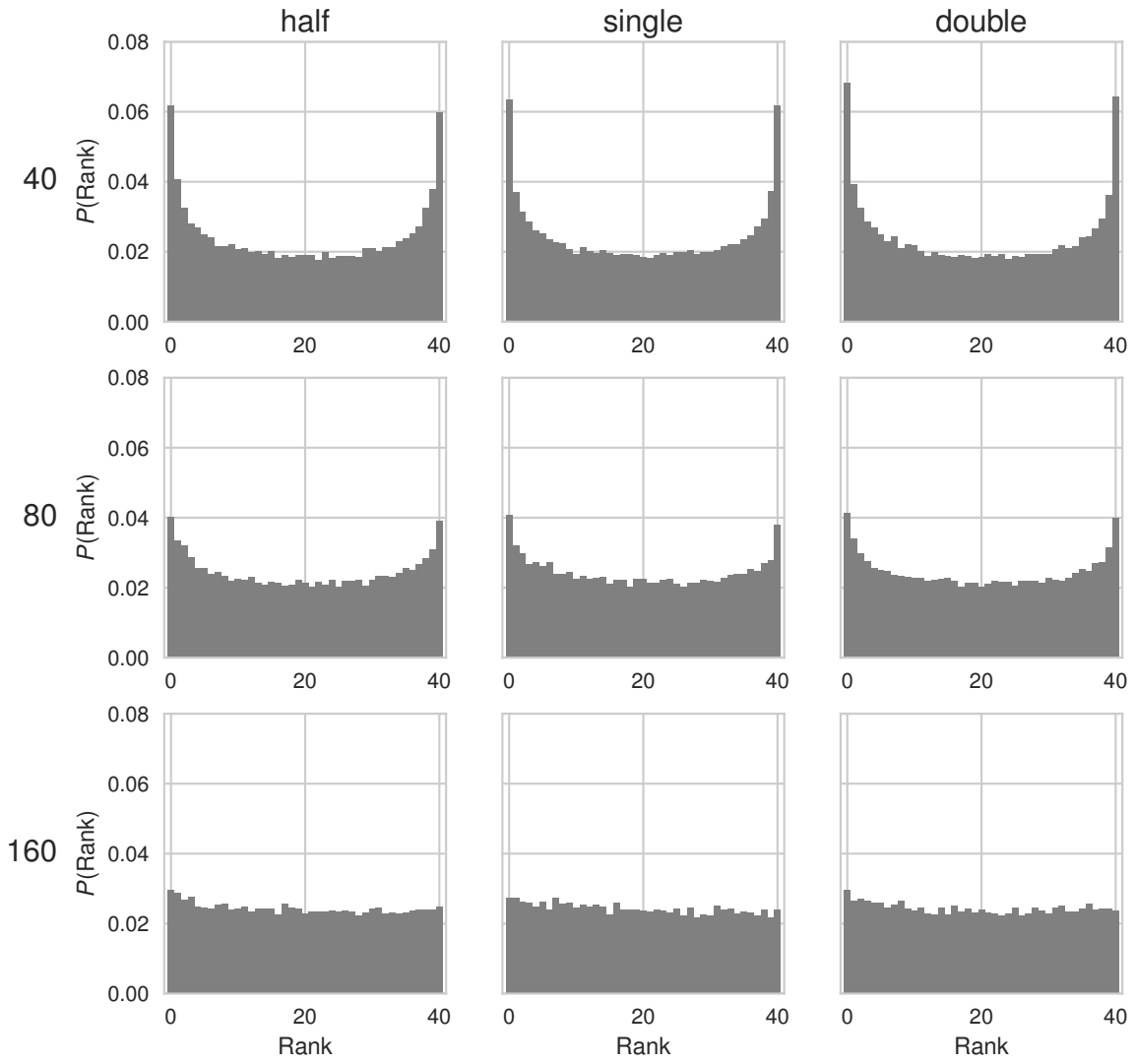


Figure 3.7: Rank histograms of analyses generated using different assimilation setups: half, single and double-precision, from left to right, with 40, 80 and 160 members, from top to bottom. The observation error is 30% of the natural RMS variability.

still gave a lower error than the reference, D40). Our results are therefore sensitive to the observation error chosen. We consider observation errors of 5% and 1% to be unrealistic, and only consider observation errors of 30% from now on, as is done in other studies (Ott et al., 2004; Anderson et al., 2009). This may seem high compared with the error of a typical atmospheric instrument, but it is important to remember that instrument error is not the only source of error in observations. One must also consider representativity error (that a point observation does not represent an entire grid box), which is far from negligible when

doing atmospheric data assimilation (Janjić and Cohn, 2006)[1].

An assimilation system should be judged not just by mean-error statistics, but also by the reliability of the ensembles that it produces. Rank histograms allow us to measure this (Hamill et al., 2001). At each analysis time, the values of a given X variable reported by each analysis ensemble member are used to define bin limits. Thus, for an ensemble of 40 members there will be 41 bins. The truth at that time is then placed in a bin according to its value. This is repeated, in this case, for 5,000 analyses (as before), and for all X variables, to produce a histogram. If this histogram is flat then the ensemble has an ideal spread — the truth is, on average, a plausible member of the ensemble. If the histogram slopes up at the edges, then the analyses are underdispersive, and if it slopes down at the edges then it is overdispersive. Figure 3.7 shows rank histograms of analyses generated from the same setups considered in Figure 3.6 with an observation error of 30% of the natural RMS variability. For the 80 and 160 member ensembles, 40 members are randomly sampled in order to determine the bin limits. Figure 3.7 indicates some under-dispersiveness for all setups, with the degree of underdispersiveness decreasing as the ensemble size is increased, but crucially there is no apparent increase or decrease in dispersiveness as the numerical precision is altered.

3.5 Sensitivity to length of the assimilation window

The results presented so far indicate that half-precision data assimilation is feasible, at least for a toy model. However, if the level of uncertainty in the system is reduced, the errors introduced by half-precision arithmetic become more noticeable. One way to reduce uncertainty is to reduce the length of the assimilation window or, in other words, to perform assimilation more frequently. In this regime, the model will be corrected by assimilation more frequently and the average analysis error will therefore be reduced. Figure 3.8 shows

[1]We note that representativity error should in fact be considered a deficiency in the model, as this error decreases as the model resolution increases. However, it is easier to treat representativity error as another component of instrument error, arising from the interpolation required to compare the model output with observations.

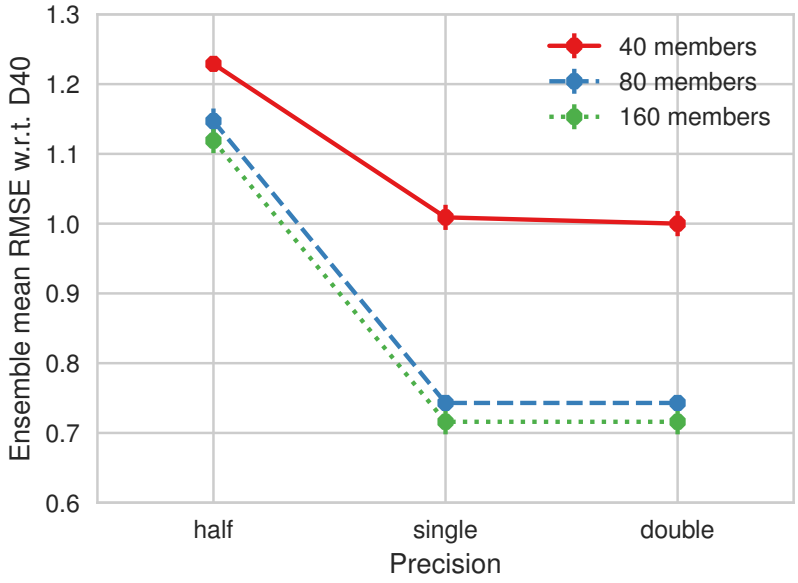


Figure 3.8: Analysis ensemble mean RMSE as a function of IEEE 754 numerical precision for frequent assimilation.

the analysis ensemble mean RMSE when the assimilation window is made ten times shorter. Clearly, the analyses generated using half-precision arithmetic are substantially worse than those generated using single or double-precision. In fact, even with 160 ensemble members, half-precision is worse than single-precision with only 40 members.

As previous studies showed, some operations are affected more than others when reducing precision (Düben et al., 2017; Dawson and Düben, 2017). Therefore, when a degradation occurs, it is worthwhile to look closer and try to find the most sensitive operations. For example, a precision reduction can have a large impact on additions involving numbers of considerably different magnitude. Consider the operation $a := a + b$, where a and b are half-precision floating point numbers with values 8 and 1×10^{-3}, respectively. The next representable floating point number above 8 is 8.0078125, which is larger than the true value of the sum, 8.001. The increment b will therefore be rounded to zero, and the addition will have no affect.

Düben et al. (2017) identified a similar problem within the time-stepping scheme of a cloud resolving model. Consider the Euler time-stepping scheme, given by

$$\psi^{t+1} = \psi^t + \Delta t \frac{\partial \psi}{\partial t}, \tag{3.12}$$

where ψ^t is a field at time t and Δt is the timestep and all variables are at half-precision. If the increment $\Delta t \frac{\partial \psi}{\partial t}$ is smaller than the floating-point spacing then it will be rounded to zero and ψ^{t+1} will be equal to ψ^t. The field will therefore not advance in time, and significant errors will accrue.

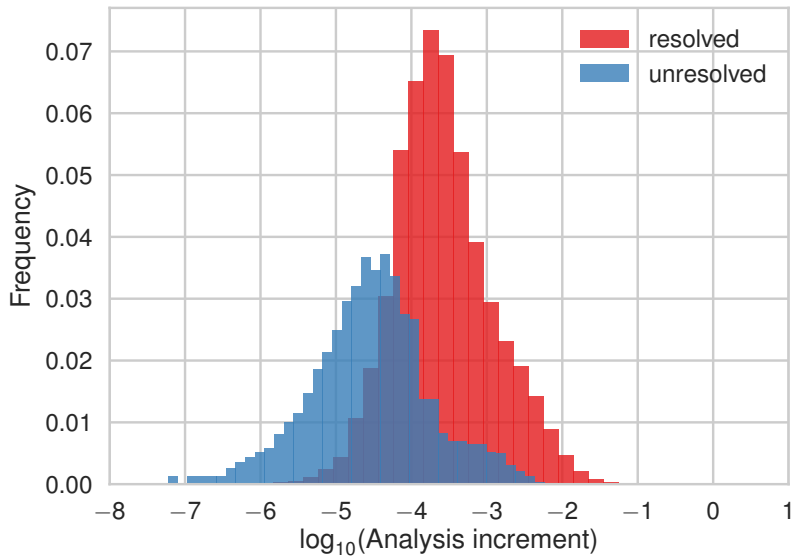


Figure 3.9: Histograms of analysis increments for half-precision, frequent assimilation, both resolved and (due to rounding errors) unresolved. The histogram bars are translucent.

One candidate for a precision-sensitive calculation in this particular setup is the Kalman filter update equation for the ensemble mean,

$$\overline{\boldsymbol{x}}^a = \overline{\boldsymbol{x}}^b + \mathbf{K}\left(y_l - \mathbf{H}\overline{\boldsymbol{x}}^b\right), \tag{3.13}$$

where $\overline{\boldsymbol{x}}^a$ and $\overline{\boldsymbol{x}}^b$ refer to the analysis and background ensemble mean state vectors, respectively, y_l is the lth scalar observation, $\mathbf{H}$ is the observation operator (in our case perfectly representable as a linear operator) and $\mathbf{K}$ is the Kalman gain. In a low-error assimilation regime (for example, when assimilation is performed frequently) the analysis increments will be very small. If an increment is too small, it will be rounded to zero, and that observa-

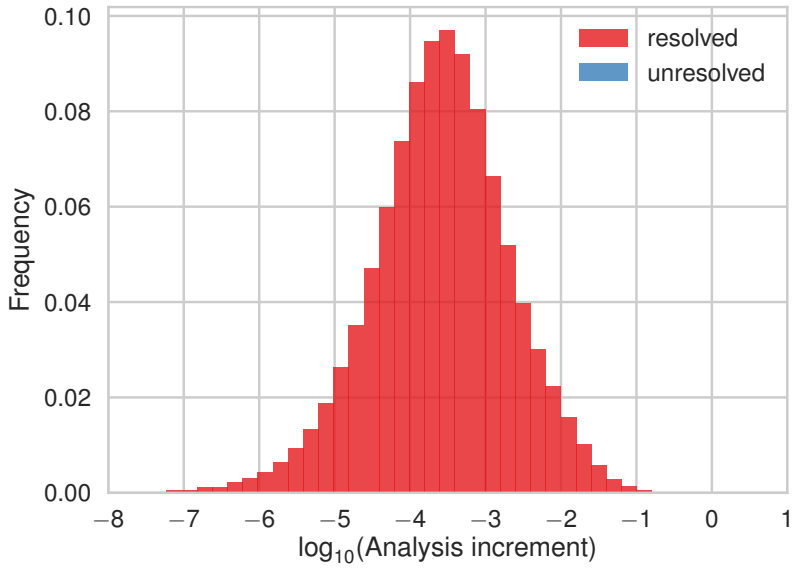


Figure 3.10: The same as Figure 3.9, only the analysis and background ensemble means are kept at double-precision. All increments are resolved.

tion will therefore have no effect. Figure 3.9 illustrates this. Over 100 forecast/assimilation cycles, the analysis increments (the individual elements of $\mathbf{K}\left(y_l - \mathbf{H}\overline{\boldsymbol{x}}^b\right)$) were divided based on whether they were resolved or unresolved (rounded to zero), and then plotted on a histogram (which is normalised by the total number of increments considered). Clearly a significant proportion of increments are rounded to zero. Note that some analysis increments were exactly zero already, due to the localization scheme, and are not shown on the histogram.

If the ensemble means, $\overline{\boldsymbol{x}}^b$ and $\overline{\boldsymbol{x}}^a$, are kept at higher precision, however, the addition of the analysis increment to the ensemble mean can be performed with no rounding error. When doing this, the analysis increments are all resolved, as shown in Figure 3.10. Only one variable, representing the background and analysis ensemble means, must be changed from half-precision to double-precision. Düben et al. (2017) found a similar solution for the cloud resolving model; the contribution of advection to the $\frac{\partial \psi}{\partial t}$ term could be performed in half-precision, as long as the field ψ itself was kept at a higher precision.

Figure 3.11 gives the analysis error for frequent assimilation with the correction just mentioned. The error is considerably lower at half-precision compared to the error in Fig-

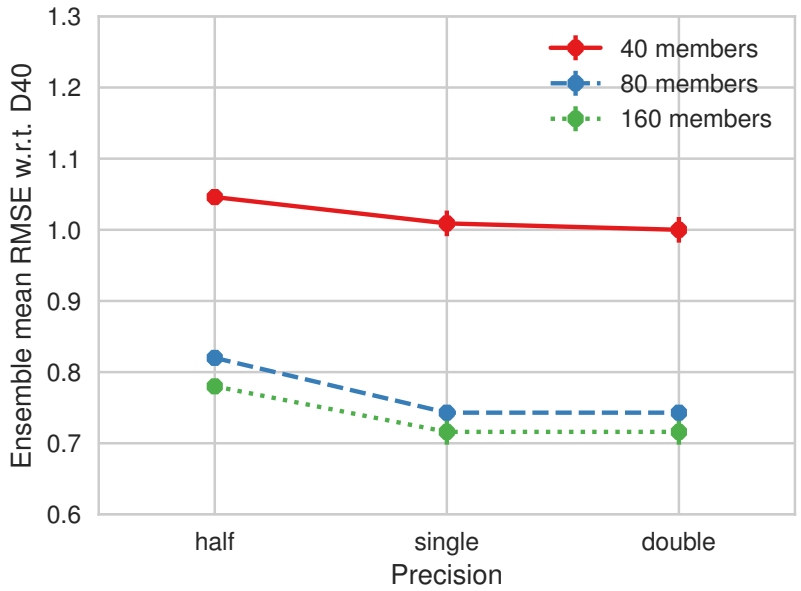


Figure 3.11: Analysis ensemble mean RMSE as a function of IEEE 754 numerical precision for frequent assimilation. The ensemble mean is kept at double-precision.

ure 3.8. However, the error is still slightly higher than at double and single-precision. It may be that, in this low-error assimilation regime, sensitivities in the forecast model to numerical precision become more noticeable, and that the "scale-selective" approach advocated by Thornes et al. (2017) is warranted. This would require keeping the X variables at a higher precision. In any case, the frequent assimilation case corresponds to assimilation every 30 minutes, approximately, which is far more frequent than the global assimilation performed in operational centres. Therefore, the results for frequent assimilation do not undermine those presented before in Section 3.6, as the scenario presented there is more realistic.

3.6 Trading precision for ensemble size

The scaling of computational cost with numerical precision is difficult to predict. The speedup obtained by moving from double to single-precision, for example, will depend on the degree to which the system is memory bound, as discussed in Chapter 1. To acknowledge the uncertainty in the cost estimate when reducing precision, we compare reduced precision data assimilation setups assuming a range of cost reductions. For single-precision,

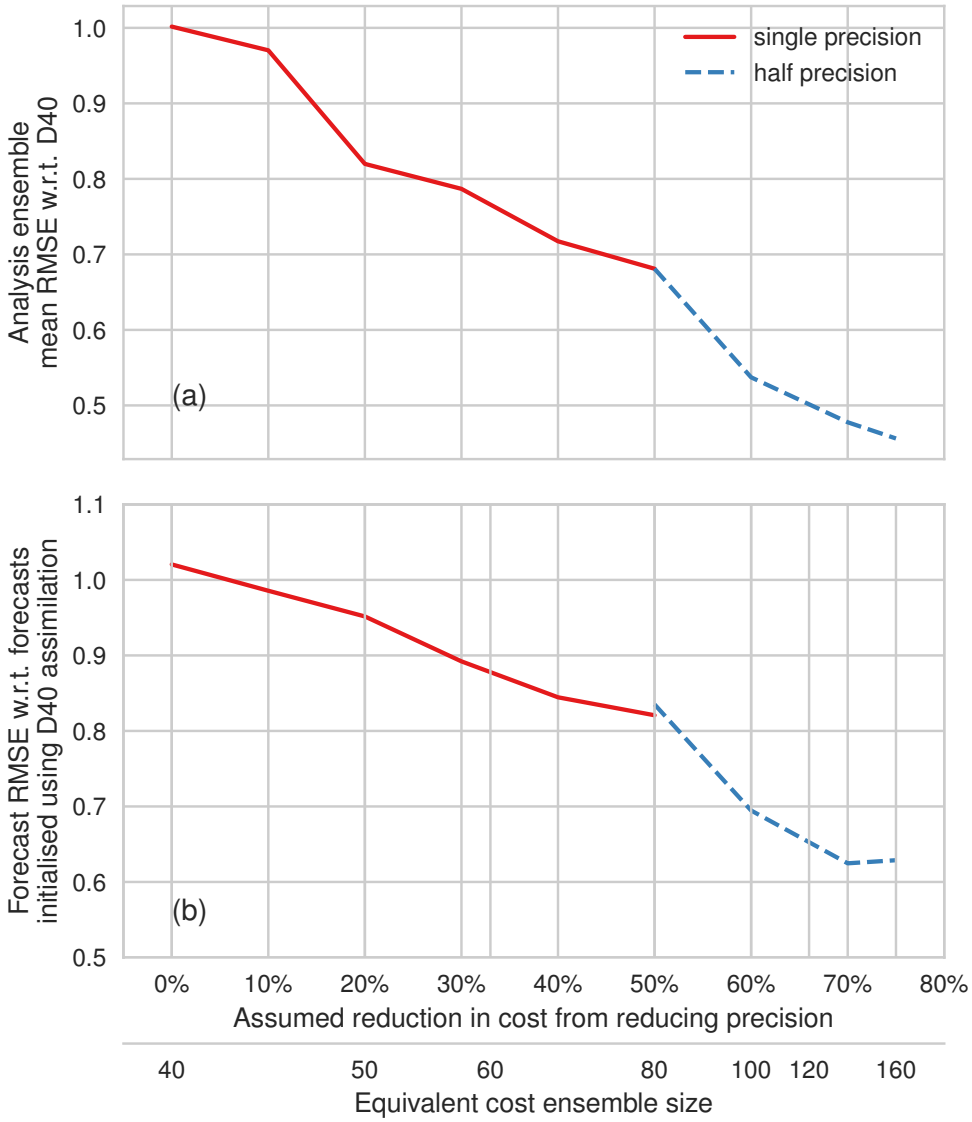


Figure 3.12: (a) The analysis mean RMSE of single and half-precision assimilation, with respect to double-precision with 40 members ("D40"), after reinvesting precision into the ensemble size. A range of possible speedups due to reducing precision are assumed and the equivalent cost ensemble size after reducing precision is given by the lower x-axis. For example, if single-precision computations deliver a 20% reduction in the cost, then 50 members can be run at single-precision for the same cost as 40 double-precision members. This would deliver a 20% reduction in the analysis error. (b) The same as (a), but the forecast RMSE at day 3 of forecasts initialised using each reduced precision assimilation setup is shown, with respect to forecasts initialised using double-precision with 40 members.

we assume that the reduction in computational cost will lie somewhere between 0% and 50%, and for half-precision, we consider reductions between 50% and 75%.

Figure 3.12a shows the average analysis error, with respect to a double-precision, 40 member reference experiment ("D40"), for a variety of assumptions about how the computational cost will fall when reducing precision. For each assumption, the ensemble size is increased in proportion to the reduction in computational cost, so that each reduced precision setup has the same computational cost as the reference, D40. For all ensemble sizes, we again use an observation error standard deviation of 0.1 and perform assimilation every 0.05 MTUs. Figure 3.12b shows the forecast RMSE at day 3 of single-member forecasts initialised using the same assimilation setups as in Figure 3.12a. These forecasts were started from the ensemble mean of each analysis in an analysis cycle of 1000 MTUs (around 200,000 forecasts in total), excluding an assimilation spin-up period of 10 MTUs. Note that the forecast error reaches its minimum at around 140 ensemble members, and therefore the slight increase in forecast errors for 160 members is simply a fluctuation from this saturation error.

Figure 3.12 allows one to estimate the gains from reinvesting precision into the ensemble size, for a given speedup that is measured for the model on a given hardware architecture. For example, if we assume that single-precision computations would allow a reduction in wall-clock time of 20%, then we would be able to afford to use an additional 10 members, and the analysis error would fall by around 20% as a result. This would deliver a 5% reduction in day 3 RMSE of deterministic forecasts. Note that Figure 3.12 presumes a linear scaling in the cost of the data assimilation system as ensemble size is increased. This is true for the ensemble forecast, as each member is independent, and it is also true for the EnSRF algorithm, as computational complexity arguments predict (Tippett et al., 2003). The cost of the full data assimilation system should therefore also scale linearly with the number of ensemble members, and we measured the speed of our configuration to confirm that this is the case.

What degree of cost reduction should we expect for a real atmospheric data assimila-

tion system? It can be argued that weather and climate models, being, in general, memory-bound, will display a linear scaling of compute time with the width of the floating point number. This would give roughly a 50% reduction in wall-clock time when reducing precision from double to single, and another 50% reduction when moving from single to half-precision. Experiments with a single-precision version of the Integrated Forecasting System (IFS) at the European Centre for Medium-Range Weather Forecasts (ECMWF) found a reduction in execution time of 40%, which is close to a 50% cost saving (Váňa et al., 2017). Nakano et al. (2018) also found a similar reduction in execution time of 45% for the global nonhydrostatic model NICAM (at 4 km resolution). Furthermore, recent developments in the graphics processing unit (GPU) market suggest that another reduction of 50% in compute time from single to half-precision can be achieved (NVIDIA, 2017b)[2]. However, to the best of our knowledge no fluid dynamical model has been tested on this hardware at this level of precision, so naturally any results that follow from the linear scaling assumption should be considered hypothetical[3]. Although some groups experimenting with single-precision data assimilation report a computation time reduction of as little as 10-20% (Jeffrey Anderson, personal communication, July 2017), our results show that even a very small speedup would be beneficial, if savings are reinvested. Also, note that, in our case, we do not realise any computational savings at half-precision, because we use a software emulator to emulate half-precision arithmetic. A final caveat concerns I/O. If I/O is already performed at single-precision and constitutes a significant fraction of the total cost of the system, compared with the model integration or assimilation algorithm, then running data assimilation at single-precision may not deliver a significant speedup.

If a 50% reduction in wall-clock time could be realised by using single-precision instead of double-precision (in other words, if computations can be made two times faster), the analysis and forecast errors would fall by around 30% and 25%, respectively, if the savings were reinvested into the ensemble size. Furthermore, if half-precision computations could

[2] An even greater speedup can be achieved on the latest series of GPUs for matrix-matrix multiplications. We discuss an application for this hardware in Chapter 6.

[3] Although the COSMO NWP model was successfully ported to 5,000 GPUs, no tests were performed at half-precision (Fuhrer et al., 2018).

deliver a 75% reduction in wall-clock time, with respect to double-precision, the analysis and forecast errors could be reduced by 55% and 40%, respectively, through another ensemble size reinvestment. Though we cannot predict the exact reduced-precision speedup of an operational data assimilation system, we feel that these results should encourage operational centres to consider this trade-off seriously.

3.7 Conclusion

Using a toy model of atmospheric dynamics we have carried out an initial investigation of the effects of reducing numerical precision from double-precision to half-precision on the performance of an ensemble data assimilation routine. We found that the analysis error was largely insensitive to a significant reduction in the width of the floating-point significand, with a tolerance to as few as 8 bits. We then considered three IEEE 754 floating-point formats: half-, single- and double-precision. We again found no increase in the analysis error as the precision was reduced from double to half, for a regime designed to approximate an operational data assimilation scenario (for example, using an observation error of 30% of the natural RMS variability of the prognostic variables). These results were sensitive to the level of observation error — when reducing the noise in the observations, the analysis error increased for half-precision assimilation.

When the length of the assimilation window was reduced, half-precision assimilation gave a higher analysis error than single and double-precision. We conjectured that this was because half-precision variables were unable to resolve the small analysis increments that result from such a low-error assimilation regime. However, the positive results from half-precision observed in the previous experiment can be partly regained by keeping just one key operation in double-precision: the Kalman filter update equation.

Finally, we considered a trade-off between numerical precision and ensemble size. By reducing precision and reinvesting the resulting cost saving into more ensemble members, the analysis and forecast errors can be reduced. The exact error reduction depends on one's assumption about how much the cost will fall when reducing precision. For example,

if single-precision computations are 20% cheaper, the analysis error can be reduced by 20%, but if they are 50% cheaper, the error can be reduced by 30%. Additionally, if half-precision computations are 75% cheaper, then the analysis error can be reduced by 55%. It remains to be seen, however, whether the cost of a real atmospheric model will scale so favourably when reducing to half-precision.

These results demonstrate that forecast models are subject to different constraints when used for data assimilation instead of for, say, medium-range forecasting. Thornes et al. (2017) reported "deleterious results" when computing the evolution of the X variables in the two-tier Lorenz' '96 system at half-precision, so that they must be promoted to single-precision (even though the Y variables could be kept at half-precision). We experienced no such degradation, finding half-precision to be sufficient for both scales of the Lorenz '96 system. This suggests that model error can be higher within an assimilation context, perhaps because it is partially masked by the other inherent error sources, such as observation error.

The cost of the Serial-EnSRF update algorithm scales linearly with the ensemble size (Tippett et al., 2003). Therefore, the precision-ensemble size trade-off described in Section 3.6 can deliver a significant increase in the ensemble size. However, not all algorithms scale so favourably. For example, the scaling of the cost of the local ensemble transform Kalman filter with ensemble size is between quadratic and cubic (Hunt et al., 2007), and therefore a reduction in precision may not deliver as significant gains as with the EnSRF. Nonetheless, if the algorithm can be made faster through a precision reduction, without impacting the quality of analyses, this would be beneficial for operational data assimilation, where the analyses must be computed within an alloted time.

Chapter 4

Experiments with the SPEEDY general circulation model

4.1 Introduction

In this chapter we extend the previous study to an intermediate complexity atmospheric general circulation model, the SPEEDY (Simplified Parametrizations, primitivE-Equation DYnamics) model. Again, we consider an Observing System Simulation Experiment (OSSE), where observations are derived from a nature run and assimilated into an assimilation model. We compare a double-precision assimilation model with a reduced-precision assimilation model for assimilating observations in both a perfect model and an imperfect model setup. Our study builds on the earlier work of Düben et al. (2014), who found that only 6 significand bits could be used for 84% of the floating-point calculations in a low-resolution general circulation model's dynamical core. However, their model did not include physical parametrizations, nor did they consider the data assimilation procedure.

In Sections 4.2 we introduce the atmospheric model used for both the nature run and the assimilation model. In Section 4.3 we introduce the assimilation setup based on the local ensemble transform Kalman filter. In Sections 4.4 and 4.5 we present results assuming a perfect nature and imperfect nature run, respectively. Finally, we conclude in Section 4.6.

4.2 The SPEEDY model

We use the SPEEDY model both for generating the nature run and for performing data assimilation. SPEEDY is an intermediate complexity atmospheric general circulation model

with a spectral dynamical core and a suite of simple physics schemes (Molteni, 2003). It is a hydrostatic, sigma level model with prognostic variables for vorticity, divergence, temperature, specific humidity and surface pressure. In a spectral model, such as SPEEDY, a three-dimensional scalar meteorological field is discretised vertically into a series of spherical shells extending from the Earth's surface to the upper atmosphere. Each spherical shell is then formulated as a weighted sum of spherical harmonic basis functions. The applied discretisation is similar to a Fourier decomposition on the plane and we refer to the space spanned by these basis functions as "spectral space". We discuss issues of the spectral dynamical core in Chapter 6.

In spectral space, SPEEDY's prognostic fields are advanced forward in time using a leapfrog scheme corrected by the Roberts-Asselin-Williams filter (Amezcua et al., 2011). In grid point space, fields are represented on a Gaussian grid. The spectral resolution of the model is denoted by TN, where the "T" refers to the fact that spectral fields are triangularly truncated. Two different resolution configurations of SPEEDY were considered, T30 and T39. The T30 configuration used a 96×48 grid with a timestep of 40 minutes and the T39 configuration used a 120×60 grid with a timestep of 20 minutes. Note that, for both resolutions, the mapping of spectral resolution to grid point resolution is unusual. The T30 resolution grid point fields can be coupled with a spectral space of up to T47 resolution (half the number of latitudes minus one) and the T39 resolution grid point fields can be coupled with a spectral space of up to T59 resolution. We do not know why the model developers chose this combination of spectral and grid point resolutions but use it nevertheless. There are 8 sigma levels for both the T30 and T39 configurations, with values of 0.95, 0.835, 0.685, 0.51, 0.34, 0.2, 0.095 and 0.025. SPEEDY represents several physical processes, including convection, radiation and fluxes of momentum and energy from the land and sea and there is also a realistic land-sea mask and orography map. There is no diurnal cycle, but there is a seasonal cycle.

SPEEDY continues to be used extensively within the data assimilation community as a testbed. Kondo and Miyoshi (2016) and Miyoshi et al. (2014) used SPEEDY to perform large

ensemble data assimilation experiments, up to 10,240 members. They found that a large ensemble size allows one to eliminate ad hoc covariance localisation, and reveals statistically significant correlations across the globe, even over only a 6 hour window. Miyoshi (2011) used SPEEDY to test an adaptive covariance inflation scheme. Ruiz and Pulido (2015) and Ruiz et al. (2013) used SPEEDY to perform parameter estimation studies, where physical parameters are updated dynamically from observations along with prognostic fields. Lien et al. (2013) used SPEEDY for testing a new rainfall assimilation scheme, which transforms the rainfall variable such that it is distributed normally and is therefore amenable to assimilation with an ensemble Kalman filter. SPEEDY has even been used for simple coupled data assimilation experiments with the NEMO ocean model (Sluka et al., 2016).

For investigating precision, we developed a reduced-precision version of SPEEDY. As in Chapter 3, all real variables in the main loop of the program were replaced with the Fortran derived type provided by the Reduced Precision Emulator (rpe). We modified the emulator to allow complex number arithmetic as well as real number arithmetic, so that it could be used for computations involving spectral coefficients. Unlike with the Lorenz '96 system, we found that SPEEDY crashed when using half-precision floating-point arithmetic, as many variables underflowed or overflowed. This is a significant limitation of half-precision numbers — the smallest (excluding subnormal numbers) and largest representable numbers are 0.000061035 and 65504, respectively. Nonetheless, there is likely a use for half-precision arithmetic within specific algorithmically simple but expensive portions of the model code. We present such a potential application in Chapter 6. We also discuss how rescaling model variables may allow the use of half-precision in spite of the limited range in Chapter 7. For this chapter, we eliminate the possibility of underflows and overflows by keeping the exponent at 11 bits wide (as with double-precision floating-point numbers). Additionally, the emulator was not activated during model initialisation (in which, for example, boundary files are read in) and during the output of fields. We assumed that these two procedures have a negligible computational cost compared with the forward integration used to generate the background ensemble within the assimilation

system.

We began by running the SPEEDY model with 64 bit arithmetic (double-precision). Then, we lowered the significand width and studied how this affected its performance. We found that the model can run with as few as 6 significand bits (for all operations), though the rounding error at this level was so severe that strong artefacts are clearly visible in the prognostic fields. We found that 10 significand bits were required to mostly eliminate this artefact and therefore chose a 22 bit model (10 significand bits, 11 exponent bits and 1 sign bit) for comparison with the 64 bit model.

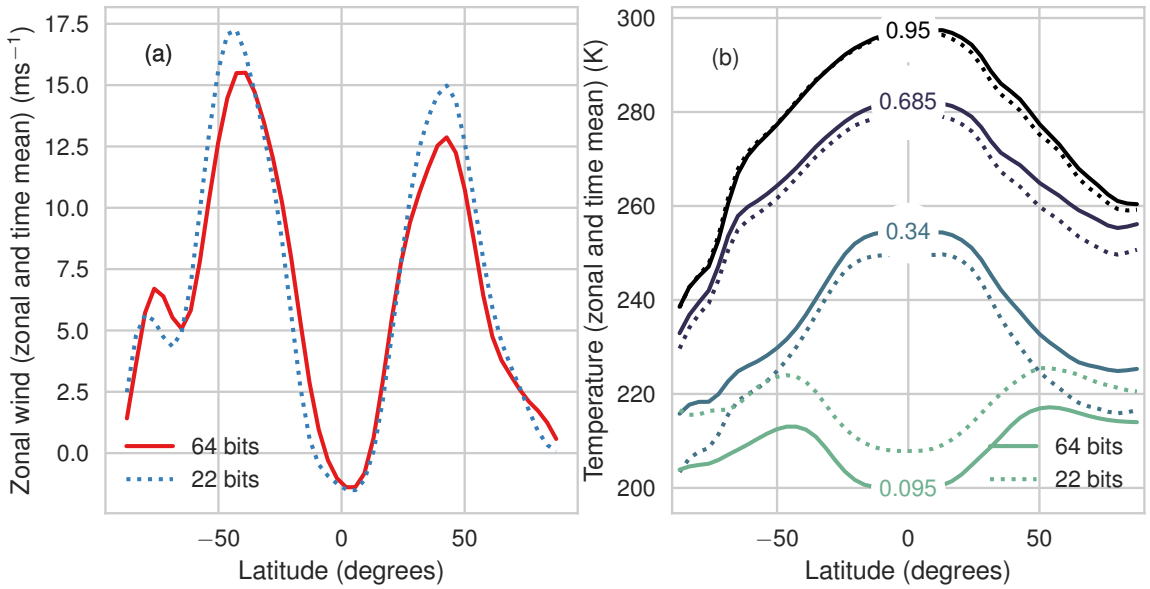


Figure 4.1: (a) The zonal-mean time-mean zonal wind at $\sigma = 0.51$ computed from climatologies of the 22 bit model and the 64 bit model. (b) The zonal-mean time-mean temperature at all vertical levels of the model computed from climatologies of the 22 bit model and the 64 bit model. The sigma level is superimposed on each pair of lines.

Figure 4.1 illustrates some differences in the zonal-mean time-mean climatologies of the 22 bit and 64 bit models. Figure 4.1a shows the zonal wind at the $\sigma = 0.51$ vertical level (approximately 500 hPa). Jets are clearly visible in both hemispheres though the 22 bit model jets are around 20% stronger. Figure 4.1b shows the temperature at every other model level. The temperature with 22 bits is systematically lower than for 64 bits at the bottom 6 levels, with the bias increasing for higher model levels. The sign of the bias reverses for the top two stratospheric levels. The spatial pattern of the temperature bias at $\sigma = 0.95$ is

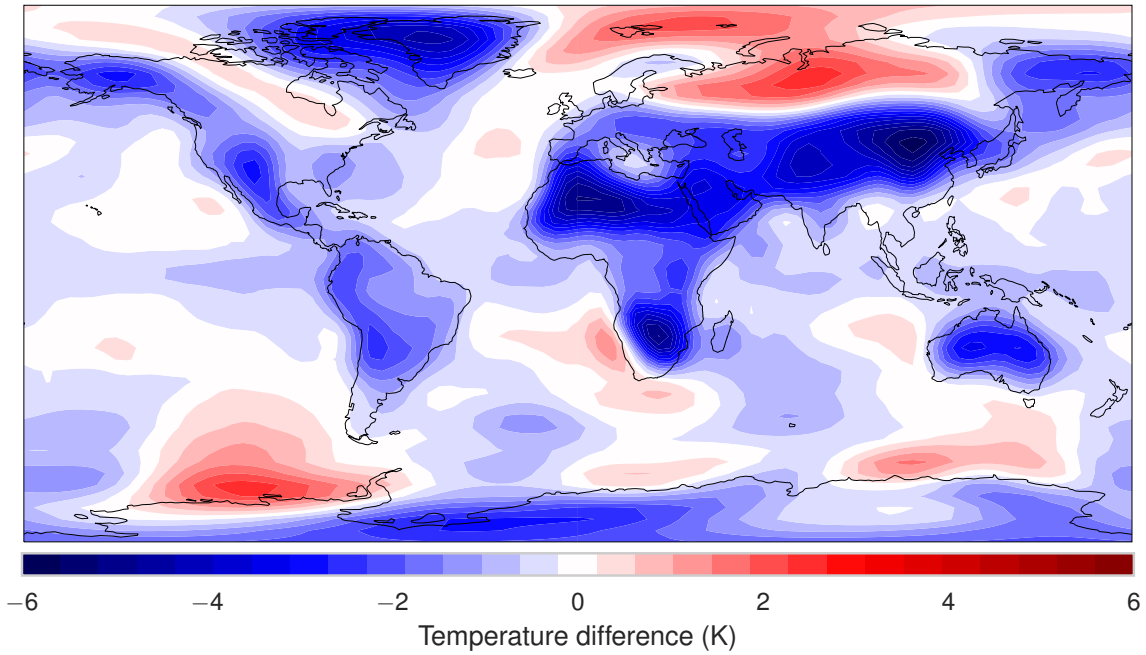


Figure 4.2: The time-mean difference (bias) between the lowest level ($\sigma = 0.95$) temperature of SPEEDY with 22 bits and 64 bits. The temperature of the 64 bit model was subtracted from the temperature of the 22 bit model.

illustrated in Figure 4.2 to be concentrated over the continents. It seems that, for SPEEDY, the land surface scheme is particularly sensitive to the numerical precision. Additionally, the significant temperature errors in the stratosphere may originate in the radiation scheme. With some effort, we could likely isolate the origin of these errors and devise a "bespoke" fix to the rounding-errors, much like in the example in Chapter 3. However, we instead choose to proceed with the model as it is and hope to demonstrate that it is usable within data assimilation in spite of the perturbations that we have introduced to its climatology.

4.3 Data assimilation setup

To perform assimilation, we used the local ensemble transform Kalman filter (LETKF) (Hunt et al., 2007), introduced in Chapter 2. The LETKF computes an independent analysis at each grid point. We used the LETKF to assimilate synthetic observations every 6 hours, using the system developed by Miyoshi (2005). We extracted from a nature run values of surface pressure as well as temperature, zonal wind and meridional wind at each level and spe-

Table 4.1: Properties of the synthetic observations derived from the nature run.

Variable	Levels	Observation error	Number of observations
Zonal wind	All	1ms^{-1}	3328
Meridional wind	All	1ms^{-1}	3328
Temperature	All	1K	3328
Specific humidity	Bottom four	0.001 kgkg^{-1}	1664
Surface pressure	N/A	1hPa	416
			Total: 12064

cific humidity in the bottom four levels at 416 locations around the globe. We distributed these observations in a way that mimics the real conventional observing network, with observations concentrated over the Northern Hemisphere continents (the red dots in Figure 4.3). The observing network did not change with time. Instrument error was simulated by adding Gaussian noise to each extracted value, and the instrument error for each variable is given in Table 4.1. In all experiments, the model was spun-up from a global rest state over one year, and was then integrated for a further 14 months to generate the nature run. Data assimilation was performed over the latter 14 month period, and time mean quantities were computed after discarding the first 2 months, as the data assimilation system is still spinning up over this time.

To try and compensate for small ensemble effects, we employed both covariance inflation and covariance localisation. For localisation we employed a standard Gaspari-Cohn Gaussian-esque correlation function of finite support (Gaspari and Cohn, 1999), as in Chapter 3, with a horizontal correlation length scale of 1000 km and a vertical length scale of 0.1 sigma levels. We keep these two constant throughout the chapter, reasoning that the optimal localisation length scale mainly depends on the ensemble size and the observing network, which are not changed.

For inflation we used the relaxation-to-prior perturbations (RTPP) technique, introduced in Zhang et al. (2004) and expanded upon in Whitaker and Hamill (2012). After assimilating observations, the difference between analysis ensemble member i and the analysis ensemble mean, $\boldsymbol{x}_i'^a$, is replaced with a weighted sum of itself and the difference between background ensemble member i and the background ensemble mean, $\boldsymbol{x}_i'^b$:

$$\boldsymbol{x}_i'^{a,\mathrm{RTPP}} = (1-\alpha)\boldsymbol{x}_i'^{a} + \alpha\boldsymbol{x}_i'^{b} \tag{4.1}$$

where α determines the relative contributions from the background and analysis perturbations. The factor α must be tuned to produce a low analysis ensemble mean root-mean-square error and an error-to-spread ratio close to one. The posterior spread is therefore *relaxed* towards the prior spread. Here, in fact, the RTPP scheme is applied to the weight matrix, used in the LETKF to transform the prior perturbations to the posterior perturbations:

$$\mathbf{W}^{\mathrm{RTPP}} = (1-\alpha)\mathbf{W} + \alpha\mathbf{I} \tag{4.2}$$

where $\mathbf{W}$ is the original weight matrix, $\mathbf{W}^{\mathrm{RTPP}}$ is the corrected weight matrix and $\mathbf{I}$ is the identity matrix. This formulation is identical to that presented in Equation 4.1 (Guo-Yuan Lien, personal communication, November 2017).

The RTPP scheme is similar to other adaptive inflation schemes, such as those presented in Miyoshi (2011), in that the degree of inflation can vary in both space and time. It was therefore suitable for our data assimilation system, as we used a spatially non-uniform observation network. For a given component of the state vector, j, the degree of inflation, f, can be written as

$$f = \frac{(1-\alpha)x_{i,j}'^{a} + \alpha x_{i,j}'^{b}}{x_{i,j}'^{a}} = 1 - \alpha + \alpha\frac{x_{i,j}'^{b}}{x_{i,j}'^{a}}. \tag{4.3}$$

It can be seen that the degree of inflation is proportional to the factor by which the ensemble perturbation about the mean would be reduced by the assimilation of observations. This factor will be larger over observation dense areas, and therefore the inflation will be higher over those areas. If the ratio of background to analysis perturbations is unity, in other words, if the state variable j of ensemble member i is not changed by the assimilation of observations, then the inflation factor is also unity. This can be compared with traditional constant multiplicative inflation, such as that used in Chapter 3, where f is a constant with

a value slightly greater than one.

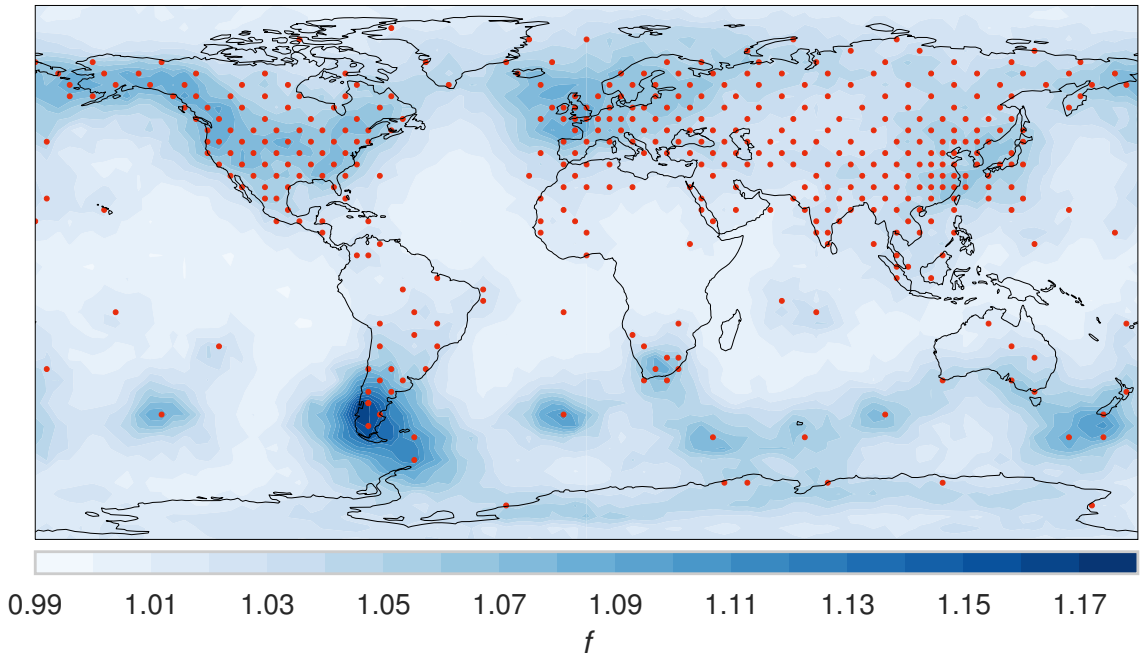


Figure 4.3: The locations of observations taken from the nature run (red dots). The spatial distribution of the relaxation-to-prior perturbations inflation factor, f, for temperature at the lowest model level is also shown.

The value of f can be diagnosed retrospectively from $\boldsymbol{x}'^{b}_{i,j}$ and $\boldsymbol{x}'^{a,\mathrm{RTPP}}_{i,j}$ using

$$f = \frac{(1-\alpha)\boldsymbol{x}'^{a,\mathrm{RTPP}}_{i,j}}{\boldsymbol{x}'^{a,\mathrm{RTPP}}_{i,j} - \alpha\boldsymbol{x}'^{b}_{i,j}}. \tag{4.4}$$

Using Equation 4.4, we computed f for temperature at the lowest level and averaged it over 5 months of assimilation. We found that around 2% of the values of f lie outside the range of $[0.5, 2.0]$. We excluded these values when performing the time mean, reasoning that they were likely to be sampling errors from an insufficient ensemble size. The resulting map of f is shown in Figure 4.3. The inflation is generally concentrated over areas with observations, as expected, with notable peaks directly over the sparsely distributed observations in the Southern Hemisphere.

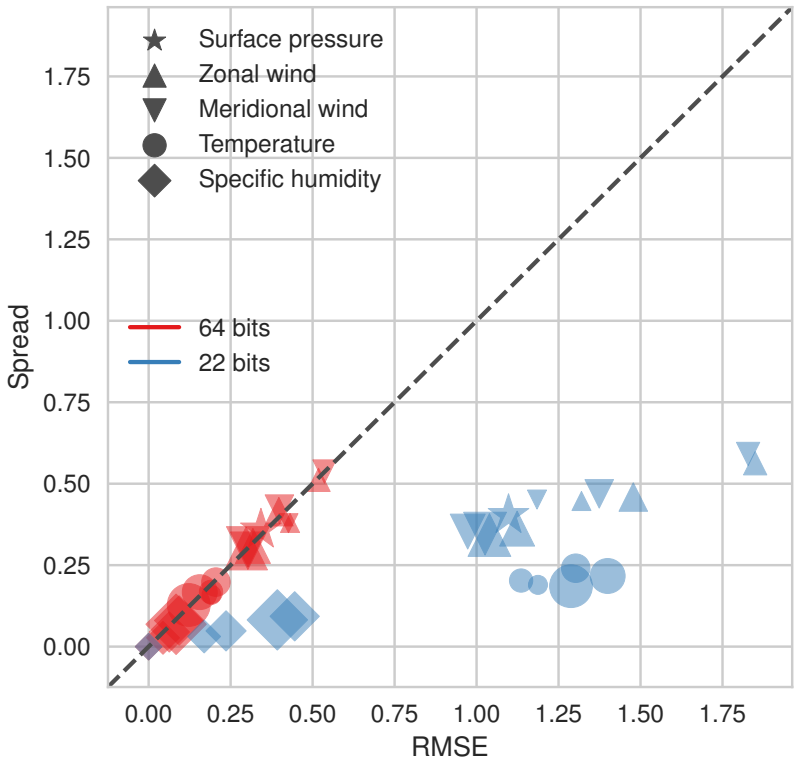


Figure 4.4: The analysis ensemble mean root-mean-square error (abscissa) and spread (ordinate) of all prognostic fields at levels 0.95, 0.835, 0.51, 0.34 and 0.095, normalised by the respective observation error, for both the 64 bit and 22 bit configurations of SPEEDY. The larger markers correspond to the lower sigma levels and vice-versa. The nature run is generated using the 64 bit configuration.

4.4 Perfect model results

Firstly, we considered a perfect model experiment. The nature run was generated using the T30, 64 bit configuration of SPEEDY, and the synthetic observations were assimilated using the same model in both the 64 bit and 22 bit configurations. We tuned the RTPP α parameter and found a value of 0.4 to produce an optimal error-to-spread ratio when assimilating with the 64 bit model. Figure 4.4 shows the analysis ensemble mean root-mean-square error (RMSE) and analysis ensemble spread for all prognostic fields at sigma levels 0.95, 0.835, 0.51, 0.34 and 0.095, and for both of the model configurations. The error and spread for each field are normalised by the respective observation errors given in Table 4.1 so that all fields can be compared. All values were obtained by averaging over latitude, longitude and time. The RMSE values shown include both the contribution from

the bias and the random component. Removing the bias by subtracting the time mean from both fields before computing the RMSE did not significantly alter these results or any of the results in Section 4.5 either. However, given that the reduced-precision model certainly has spatially varying biases with respect to the double-precision reference (see Figures 4.1 and 4.2), this may simply be because the biases balance each other in the global mean metric shown in Figure 4.4.

We also computed mean innovation (i.e. observation minus background) statistics by applying the observation operator to the background ensemble mean and subtracting it from the observation vector. Because we prescribe the observation error to be unbiased, a statistically significant non-zero time-averaged innovation will indicate a model bias (Dee, 2005). We define a statistically significant innovation bias as occurring whenever the mean innovation is greater than two standard errors from zero where the standard error is computed from the sample standard deviation of the innovation time series. Figure 4.5 shows the statistically significant mean temperature innovation biases at $\sigma = 0.51$ (approximately 500 hPa) for the same experiment as in Figure 4.4.

For the 22 bit model a substantial degradation in the assimilation system is observed. The analysis error averaged over all fields and levels when using the 22 bit configuration of SPEEDY was around three times higher than when using the 64 bit configuration. Figure 4.4 also indicates that, while the 64 bit configuration has a good match of spread to RMSE, the 22 bit configuration suffers from substantial underdispersiveness. When performing optimally, the EnKF ensemble-mean RMSE with respect to the verifying truth matches the spread of the ensemble members around the mean (the standard deviation of the ensemble). This is the case for the 64 bit configuration, as almost all of the fields fall on the diagonal line perfectly, but it is not the case for the 22 bit configuration. Furthermore, Figure 4.5 indicates significant model biases across the globe for the 22 bit configuration, with a magnitude of up to around 75% of the temperature observation error. Other fields show a similar result (not shown here).

Figure 4.6 shows the same result as in Figure 4.4, only a 26 bit configuration of SPEEDY

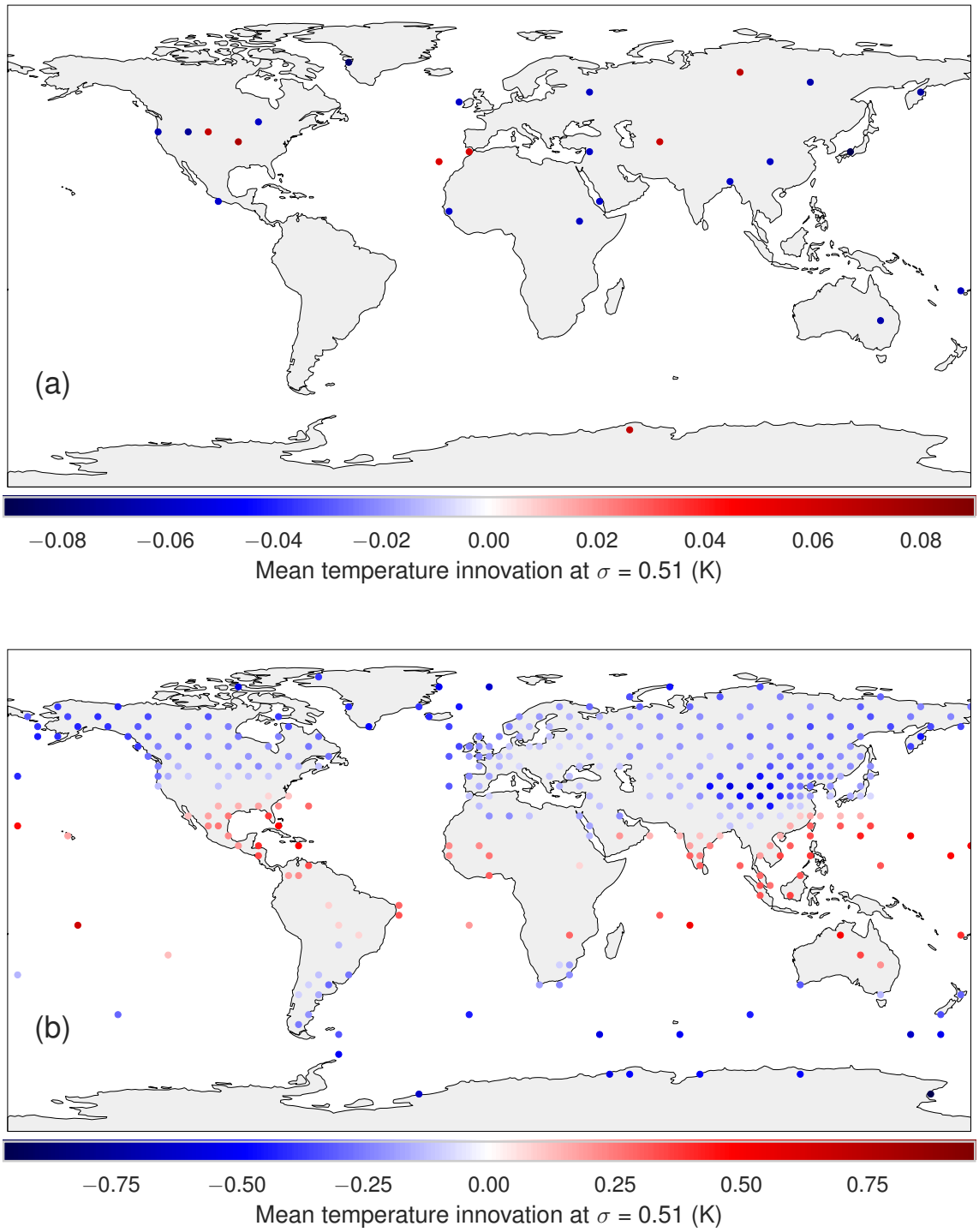


Figure 4.5: The temperature innovations at $\sigma = 0.51$ averaged over all assimilation cycles of the assimilation experiment shown in Figure 4.4 with (a) a 64 bit assimilation model and (b) a 22 bit assimilation model. Note the difference in the color scale between (a) and (b).

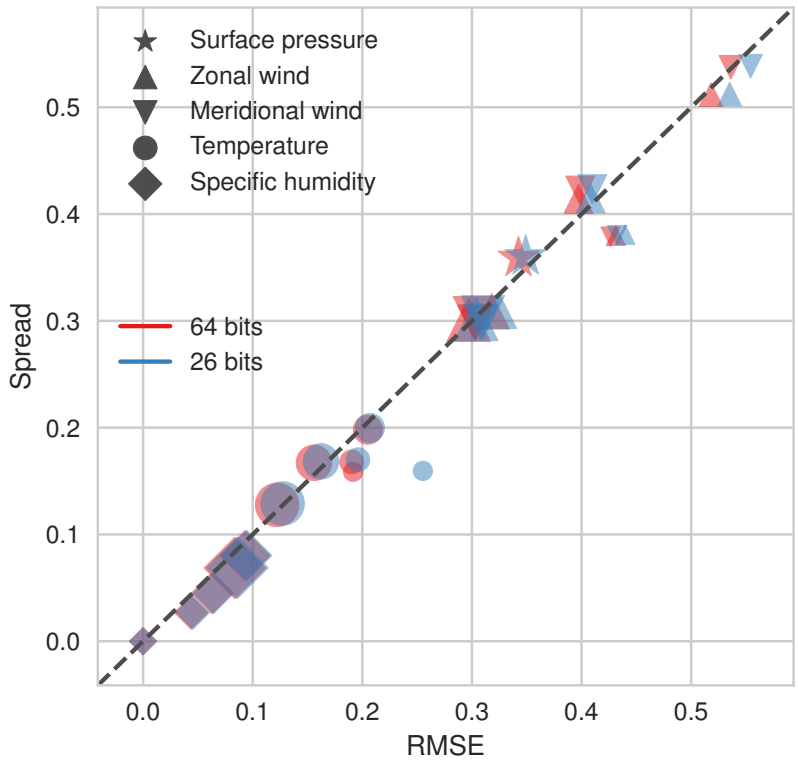


Figure 4.6: The analysis ensemble mean root-mean-square error and spread of all prognostic fields, normalised by the respective observation error, for both the 64 bit and 26 bit configurations of SPEEDY, as in Figure 4.4. The nature run is generated using the 64 bit configuration.

(1 sign bit, 11 exponent bits and 14 significand bits) was compared with the 64 bit model. The two models perform almost identically in this setup, though the stratospheric temperature is still slightly worse for the 26 bit model. Even though only four additional significand bits are required to match the performance of the 64 bit model, at least in the troposphere, we will still consider the 22 bit model from here on. We are particularly interested in using 10 bits for the significand because the IEEE-standardised half-precision (16 bit) floating-point type supported by existing hardware has a significand of the same length. The rounding errors incurred within the 22 bit model are then broadly the same as those that would be incurred for a 16 bit model, though the exponent must also be reduced for a proper emulation of 16 bit arithmetic (we defer this discussion to the conclusion).

The substantial degradation of assimilation quality when using 22 bit arithmetic might be expected given that the rounding error has been substantially increased by reducing the

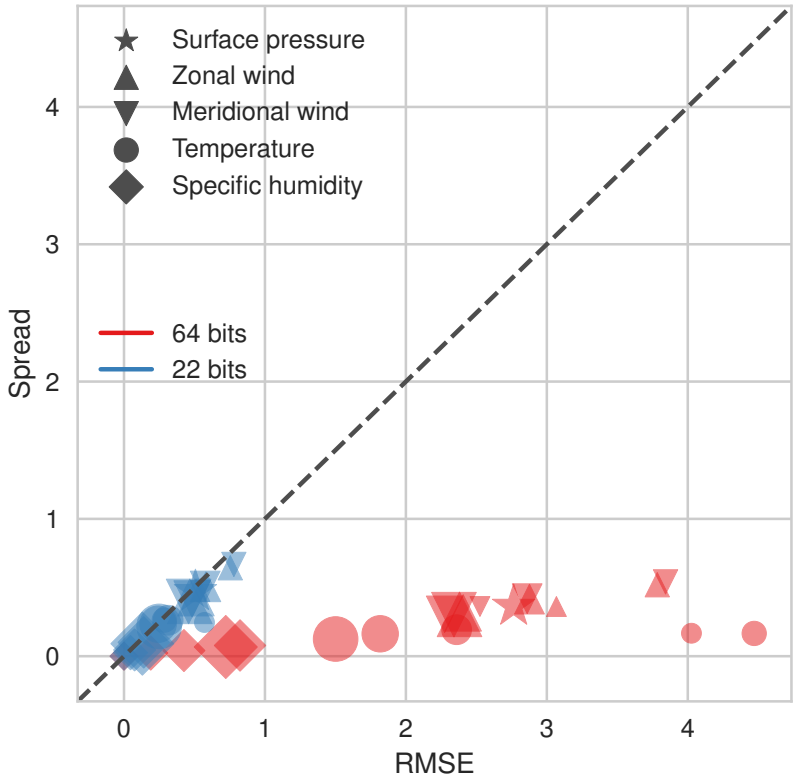


Figure 4.7: The analysis ensemble mean root-mean-square error and spread of all prognostic fields, normalised by the respective observation error, for both the 64 bit and 22 bit configurations of SPEEDY, as in Figure 4.4. The nature run is generated using the 22 bit configuration.

significand width from 52 bits to only 10 bits. In fact, the gap between two consecutive floating point numbers is increased by 12 orders of magnitude, when reducing the significand from 52 to 10 bits ($2^{-10}/2^{-52} \approx 10^{12}$). We might therefore be tempted to disregard the 22 bit model. However, we found that the 64 bit and 22 bit configurations of SPEEDY could not be fairly compared without some quantification of model error. This is because the nature run is derived from the 64 bit model, and therefore the 22 bit model, when used in the LETKF, has a source of model error not present when the 64 bit model is used for assimilation: the rounding error. To demonstrate this, we repeated the previous experiment but instead used the 22 bit configuration of SPEEDY to produce the nature run. The results of this experiment are given in Figure 4.7. Compared with Figure 4.4, the 64 and 22 bit results are swapped, and the 22 bit assimilation model actually gave a lower error than the 64 bit model.

We argue that, given the inherent, unavoidable model errors present in operational data assimilation, we should be able to reduce precision in our assimilation models substantially. Naturally, in order to test this hypothesis, we must include a "simulation" of typical model error. We will consider two ways to relax the perfect model assumption: by changing the diffusion scheme of the assimilation model and by using a higher resolution model to derive the nature run.

4.5 Imperfect model results

4.5.1 Introducing model error: decreased diffusion

One of the simplest ways to introduce model error into a perfect model experiment is to modify the diffusion scheme of the assimilating model. For this experiment, we generated the nature run exactly as in the previous experiment, but when assimilating observations into the 64 and 22 bit models we doubled the maximum damping time for horizontal diffusion of temperature, vorticity and divergence from 2.4 hours to 4.8 hours. We also doubled the maximum damping time for extra diffusion in the stratosphere from 12 hours to 24 hours. By increasing the damping time, diffusion is relaxed at the higher wavenumbers. In other words, by increasing the damping time, we are allowing more activity in the smaller scales of all fields.

Figure 4.8 shows the assimilation error and spread when using this modified diffusion scheme in the assimilation model. Apart from this modified scheme and the RTPP factor, which is increased from 0.4 to 0.9 to account for the increased tendency for ensemble collapse in the presence of model error, the experiment is identical to the perfect model experiment. Comparing with Figure 4.4, there is a clear reduction in the error gap between the two configurations (the 22 bit and 64 bit models). Furthermore, according to Figure 4.9, the innovation biases are reduced for the 22 bit model and increased for the 64 bit model, though they still differ significantly in spatial distribution. This indicates that model error can, at least partially, mask the errors introduced by reducing precision. However, to

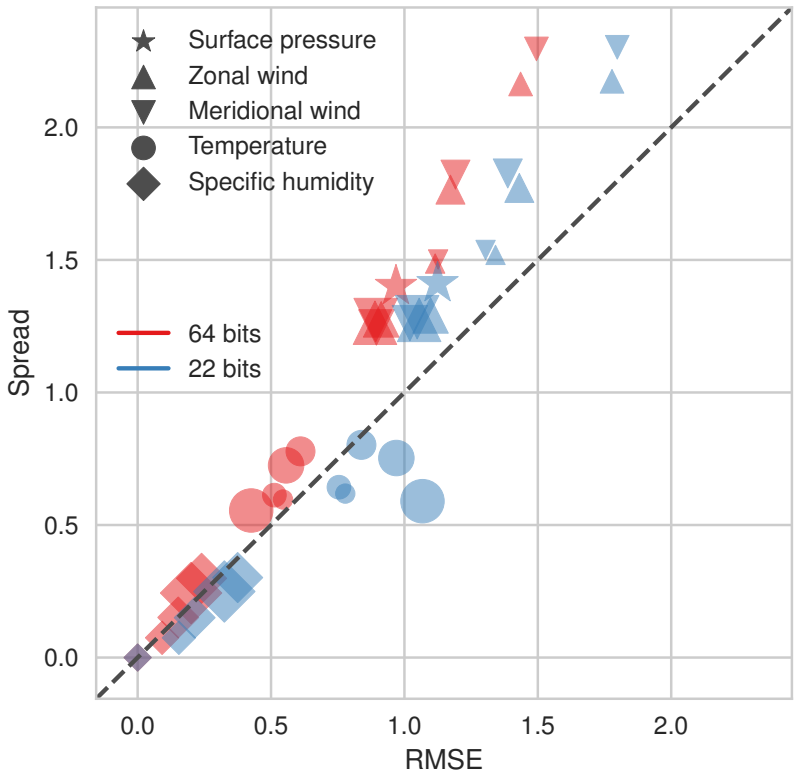


Figure 4.8: The analysis ensemble mean root-mean-square error and spread of all prognostic fields, normalised by the respective observation error, for both the 64 bit and 22 bit configurations of SPEEDY, as in Figure 4.4. The nature run is generated as in Figure 4.4 but the assimilation model has a modified diffusion scheme.

strengthen this conclusion, another form of model error should be considered. In the next section, we consider changing the resolution of the nature run.

4.5.2 Introducing model error: high resolution nature run

Another way to "simulate" model error is to use a higher resolution model to perform the nature run, compared with the assimilation model. For this experiment, we performed the nature run with a T39 configuration of SPEEDY and assimilated observations into a T30 configuration. We took the T39 nature run, transformed all fields into spectral space, truncated those fields to T30, and transformed back to the same Gaussian grid used by the T30 model. We used the resulting fields both for generating the synthetic observations and for verifying the analyses. We could have derived the observations directly from the higher resolution grid, but an interpolation would still be required in order to compare the back-

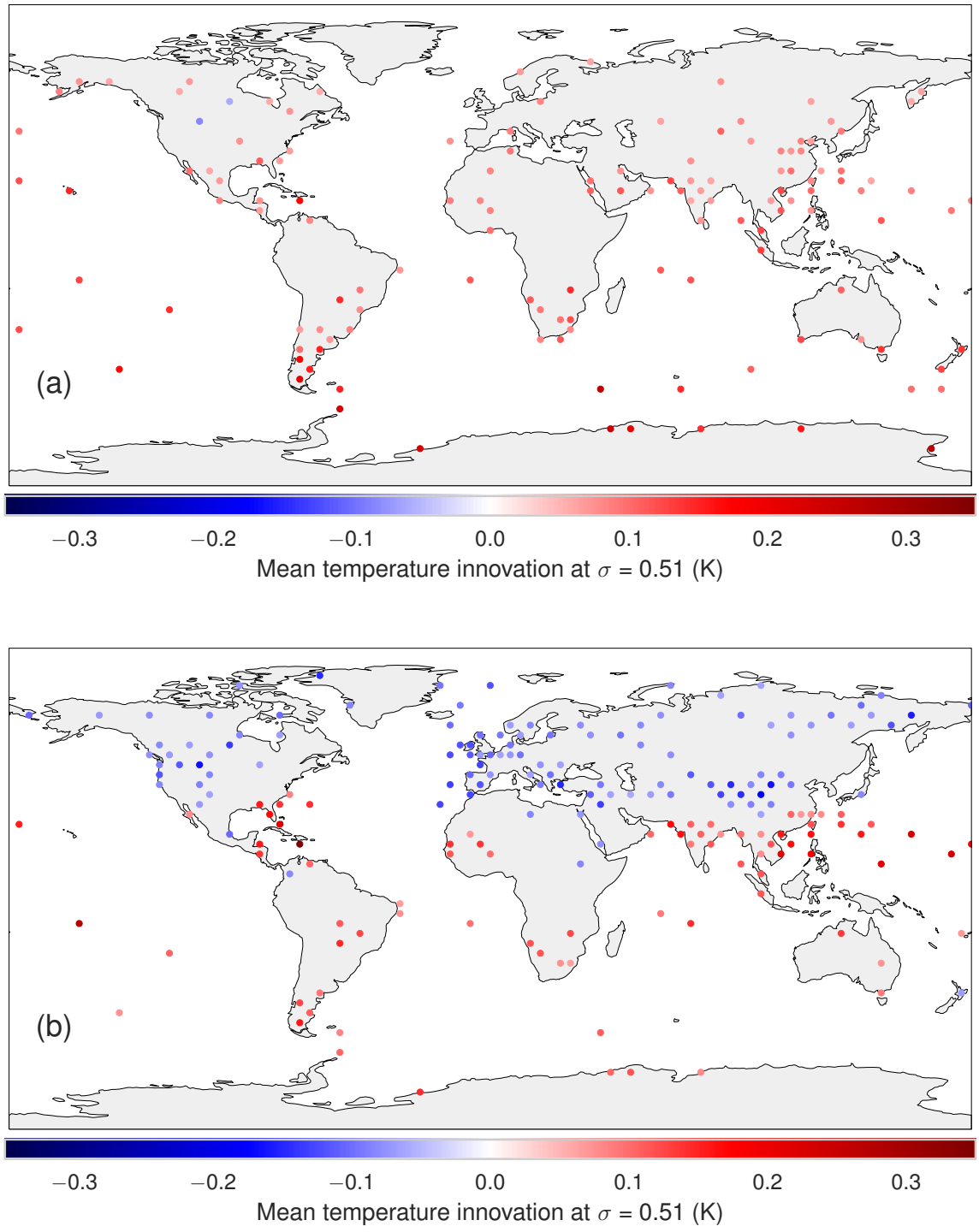


Figure 4.9: The temperature innovations at $\sigma = 0.51$ averaged over all assimilation cycles of the assimilation experiment shown in Figure 4.8 with (a) a 64 bit assimilation model and (b) a 22 bit assimilation model. The nature run is generated as in Figure 4.5 but the assimilation model has a modified diffusion scheme. Note that the colour scales of (a) and (b) are the same.

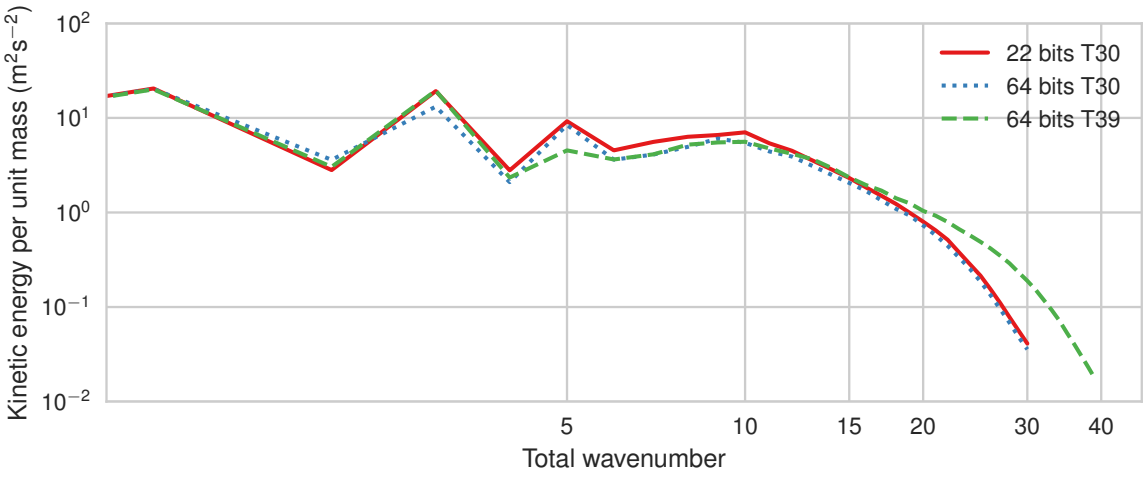


Figure 4.10: The kinetic energy spectra for SPEEDY at T30 with 64 bit and 22 bit arithmetic and at T39 resolution with 64 bit arithmetic.

ground with the observations and for verifying the analyses. A comparison of the kinetic energy spectra of the T39 and T30 configurations is given in Figure 4.10. Even though the truncated nature run is at the same grid resolution as the T30 configuration of SPEEDY, the smaller scales are considerably less damped and their evolution is determined by interactions with scales not present in the T30 model (i.e. scales with total wave number greater than 30). Note that the spectrum for the 22 bit T30 model is not significantly different from that of the 64 bit T30 model.

Figure 4.11 shows the error and spread of the analysis prognostic variables for this experiment. In this case, it was necessary to increase the RTPP inflation factor even further, to 0.95. The difference in analysis quality between the 64 bit and 22 bit configurations is now very small, and the overall error difference is less than 1%, on average. Furthermore, according to Figure 4.12, the biases exhibited by each model are very similar in magnitude in the presence of this form of model error. We note that some of the bias will be caused by the assimilation of observations derived from a model with a higher resolution orography (Baek et al., 2009). This is evident in the "positive-negative-positive" pattern over the Tibetan Plateau in Figure 4.12. In other parts of the globe, such as Europe, the 64 bit and 22 bit models have differing patterns of bias, but the magnitude of the bias is broadly similar for both models. The model error introduced by upgrading the resolution of the nature

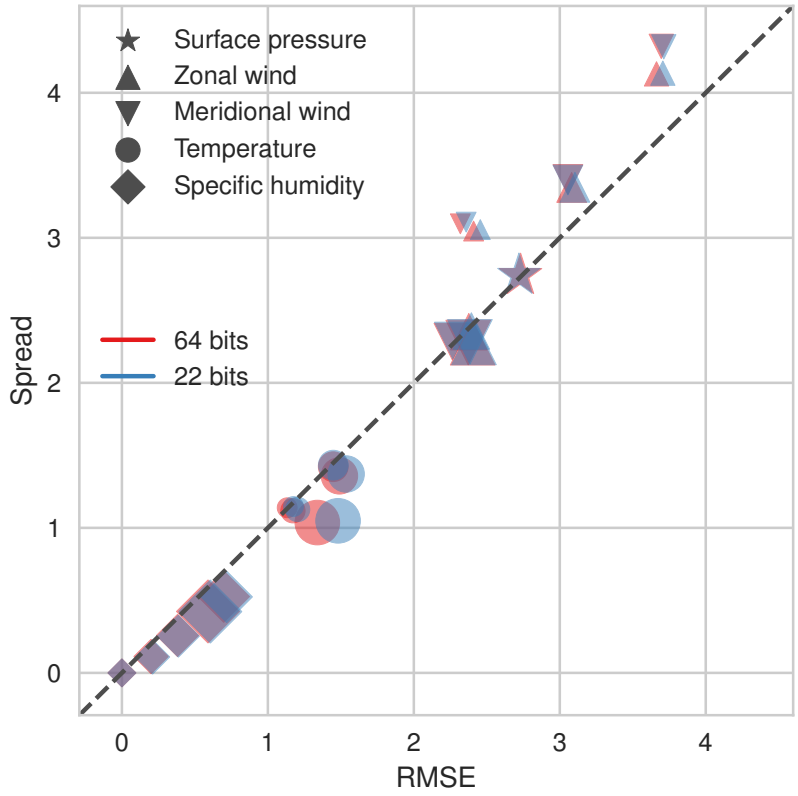


Figure 4.11: The analysis ensemble mean root-mean-square error and spread of all prognostic fields, normalised by the respective observation error, for both the 64 bit and 22 bit configurations of SPEEDY, as in Figure 4.4. The nature run is T39 resolution (64 bit) but the assimilation model is T30 resolution.

run from T30 to T39 is modest. Indeed, the true nature has in theory an infinite number of scales of motion. However, even this modest model error is almost sufficient to mask the rounding errors entirely. Naturally, the strongest test of a proposed modification to a forecasting system (here, the generation of the initial conditions) is to verify the actual forecasts produced against real observations. We consider this beyond the scope of this investigation. However, given that even a modest model error can hide the errors from a substantial precision reduction, we do not expect the results of the experiments to change.

4.6 Conclusion

In this chapter we have demonstrated the impact of a significant reduction of precision in the numerical model used for background generation in a LETKF. We reduced the width of

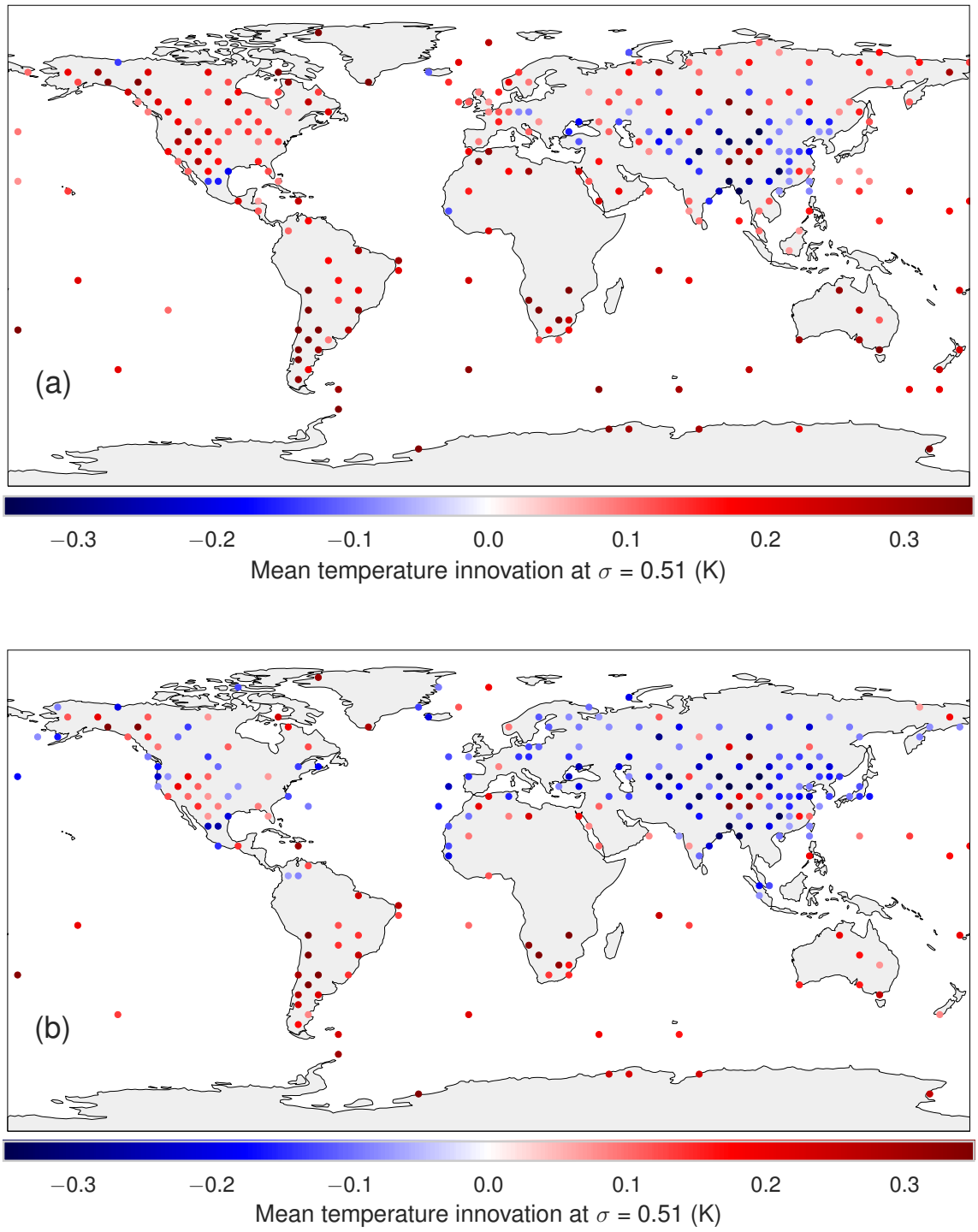


Figure 4.12: The temperature innovations at $\sigma = 0.51$ averaged over all assimilation cycles of the assimilation experiment shown in Figure 4.11 with (a) a 64 bit assimilation model and (b) a 22 bit assimilation model. The nature run is T39 resolution (64 bit) but the assimilation model is T30 resolution. Note that the colour scales of (a) and (b) are the same.

the floating-point significand in the SPEEDY model from 52 bits to 10 bits, though we did not change the width of the exponent. We then compared the performance of the LETKF when using the 22 bit model and the original 64 bit model. We found that the quality of the analyses produced from the 22 bit model is only degraded in an unrealistic perfect model setup. When a modest degree of model error is introduced the difference in performance between the 22 bit and 64 bit models is reduced almost entirely.

We note that our results obtained from reducing the precision of the assimilation model are not necessarily transferable to models used to produce actual weather forecasts. The assimilation model is restarted every 6 hours by the assimilation update step, and so it may be that this model is more tolerant to a precision reduction than if it were running freely. Additionally, the update step (here, an LETKF) may require a different precision to the assimilation model as the computational profile is quite different, consisting of mostly matrix operations.

As discussed in Section 4.2, we did not reduce the floating-point exponent width below 11 bits. We do not expect a strong impact on the results when reducing the exponent to 8 bits as this alone provides around 80 orders of magnitude, which is more than adequate for an atmospheric model (see, for example Váňa et al. (2017) and Nakano et al. (2018), who both ran atmospheric models at a much higher resolution almost entirely in single-precision with no overflow or underflow problems). However, as discussed, the 5 bit exponent of half-precision floating-point numbers gives a limited allowable range of values. Nevertheless, there is a use for half-precision arithmetic in certain parts of the model code. In Chapter 6 we show how half-precision arithmetic can be used within the Legendre transforms of a spectral model in spite of the limited range.

We also did not discuss the computational cost saving which can be expected from reducing precision. As mentioned before, estimating this cost saving is very difficult. We measured a 40% reduction in the wallclock time of the SPEEDY model when reducing precision from double- to single- precision (and a similar speed-up for the LETKF algorithm). This alone could allow roughly a doubling of the ensemble size which can be expected to

significantly improve the quality of the analysis.

Chapter 5

Reducing precision in 4D-Var

5.1 Introduction

Chapters 3 and 4 studied the effects of reducing precision in the generation of the background ensemble within an ensemble Kalman filter (EnKF). However, in an operational context, the EnKF is not typically used by itself but in conjuction with a variational technique to construct initial conditions. As an example, the Meteorological Service of Canada uses an EnKF for their ensemble prediction system but 4D-Var for their deterministic forecast (Houtekamer et al., 2014). Other centres that employ both ensemble and variational techniques include Deutscher Wetterdienst (Schraff et al., 2016) and the UK Met Office (Bowler et al., 2009). At the European Centre for Medium-Range Weather Forecasts the EnKF is not used at all beyond research (Bonavita et al., 2015; Hamrud et al., 2015). Therefore, given its continuing prevalence any study aimed at improving operational data assimilation would be incomplete without considering 4D-Var.

In both the EnKF and 4D-Var we use numerical models to perform data assimilation. However, the nature of these schemes are sufficiently different that it is not immediately clear how the results from the previous chapters could be extended to 4D-Var. As discussed in Chapter 2, incremental 4D-Var, the most popular variant of 4D-Var for operational purposes, requires the use of the tangent-linear and adjoint models. The tangent-linear model is used to evolve perturbations valid at the beginning of the assimilation window through time. The adjoint model is used to compute the gradient of terms in the cost function at later

times with respect to those initial perturbations. These two linear models when written as matrices are the transpose of each other, and so one can be constructed exactly from the other simply by rearranging the underlying data in computer memory. In reality, however, the tangent-linear and adjoint models are programmed as matrix-vector multiplication subroutines — the individual matrix elements are not made available (for an operational numerical weather prediction model, this matrix would have $\mathcal{O}(10^{18})$ elements and so would be prohibitively large). When the tangent-linear or adjoint model acts on an input vector, rounding errors are therefore incurred. These rounding errors will be different for each model and so this effectively introduces an asymmetry between the two. This asymmetry will increase as we reduce precision and could affect the 4D-Var minimization convergence process. In this chapter we present a preliminary investigation of these rounding error effects.

In Section 5.2 we recap the incremental 4D-Var algorithm introduced in Chapter 2 and in Section 5.3 we describe our experimental setup. In Section 5.4 we discuss the impacts of reducing precision on the tangent-linear and adjoint models, in Section 5.5 we discuss how this procedure affects the convergence of 4D-Var and in Section 5.6 we describe the benefits of reorthogonalisation. Finally, we conclude in Section 5.7.

5.2 Reducing precision in 4D-Var

Given a set of observations distributed over a window of time, 4D-Var finds the trajectory that minimises both the total distance between the trajectory and the observations and the trajectory and an initial guess (the background), while remaining physically consistent with the known dynamical laws of the system. In this section, we focus on the incremental form of 4D-Var (Courtier et al., 1994), whereby a global, potentially non-convex minimisation problem is split into a series of simpler, quadratic minimisation problems known as "inner loops".

As mentioned in Chapter 2, a Krylov subspace method such as conjugate gradients (CG) is typically used to minimise the cost function of incremental 4D-Var. These methods

require an expression for the matrix-vector multiplication $\mathbf{S}\boldsymbol{w}$ where $\mathbf{S}$ is the Hessian of the cost function and $\boldsymbol{w}$ is an arbitrary vector. This expression (dropping the time index on the background error covariance matrix $\mathbf{P}^b$ for brevity) is given by

$$\mathbf{S}\boldsymbol{w} = \left(\mathbf{P}^b\right)^{-1}\boldsymbol{w} + \sum_{i=1}^{q} \mathbf{M}_{im}^{\intercal}(\boldsymbol{x}_0)\mathbf{H}^{\intercal}\mathbf{R}^{-1}\mathbf{H}\mathbf{M}_{im}(\boldsymbol{x}_0)\boldsymbol{w}. \tag{5.1}$$

Evaluating 5.1 requires evaluating the tangent-linear and adjoint models, $\mathbf{M}_{im}(\boldsymbol{x}_0)$ and $\mathbf{M}_{im}^{\intercal}(\boldsymbol{x}_0)$, respectively. Again, the tangent-linear model $\mathbf{M}_{im}(\boldsymbol{x}_0)$ is the Jacobian of the non-linear model operator, $\boldsymbol{M}_{im}$, evaluated at the point $\boldsymbol{x}_0$. The adjoint is simply the transpose of this linear operator. These linear operators are formed through a chain-rule decomposition in time. The tangent-linear model is given by the expression

$$\mathbf{M}_{im}(\boldsymbol{x}_0) = \mathbf{M}_1(\boldsymbol{x}_{im-1})\mathbf{M}_{im-1}(\boldsymbol{x}_0) = \mathbf{M}_1(\boldsymbol{x}_{im-1})\mathbf{M}_1(\boldsymbol{x}_{im-2})\ldots\mathbf{M}_1(\boldsymbol{x}_1)\mathbf{M}_1(\boldsymbol{x}_0). \tag{5.2}$$

In other words, between timesteps 0 and im we must evaluate the Jacobian at each model timestep. We use this chain-rule decomposition because, in reality, a numerical model only provides the model operator for a single time step: $\boldsymbol{M}_1$. The Jacobian $\mathbf{M}_{im}(\boldsymbol{x})$ is therefore not "directly" available, only $\mathbf{M}_1(\boldsymbol{x})$. The adjoint model is formed simply by taking the transpose of Equation 5.2. Considering that the transpose of the product of two linear operators is the reversed product of the individually transposed operators, the adjoint model is given by

$$\mathbf{M}_{im}^{\intercal}(\boldsymbol{x}_0) = \mathbf{M}_1^{\intercal}(\boldsymbol{x}_0)\mathbf{M}_1^{\intercal}(\boldsymbol{x}_1)\ldots\mathbf{M}_1^{\intercal}(\boldsymbol{x}_{im-2})\mathbf{M}_1^{\intercal}(\boldsymbol{x}_{im-1}). \tag{5.3}$$

For CG the Hessian must be applied to the current search direction on every iteration of the minimisation (the algorithm will be discussed further in Section 5.6). This operation typically dominates the total computational cost for incremental 4D-Var. To save computational resources, the tangent-linear and adjoint models of the operational 4D-Var system of the European Centre for Medium-Range Weather Forecasts (ECMWF) are performed at a lower resolution than the full nonlinear model used to compute the innovations, with the

resolution increasing as convergence is approached. We argue that using a lower precision in the tangent-linear and adjoint models could provide another way to lower the cost of the 4D-Var minimisation. However, we might expect the increased rounding errors from low precision arithmetic to impact the convergence or stability of the minimisation algorithm.

5.3 Experimental setup

In order to explore issues of reducing precision within 4D-Var, we use the Object Oriented Prediction System (OOPS) of ECMWF. This is a data assimilation framework written in C++ which is entirely model-agnostic — it can be applied both to idealised models and to full-complexity weather models such as the Integrated Forecasting System of ECMWF with relatively few code changes. We perform data assimilation experiments using a simple quasigeostrophic system as described below.

5.3.1 Data assimilation setup

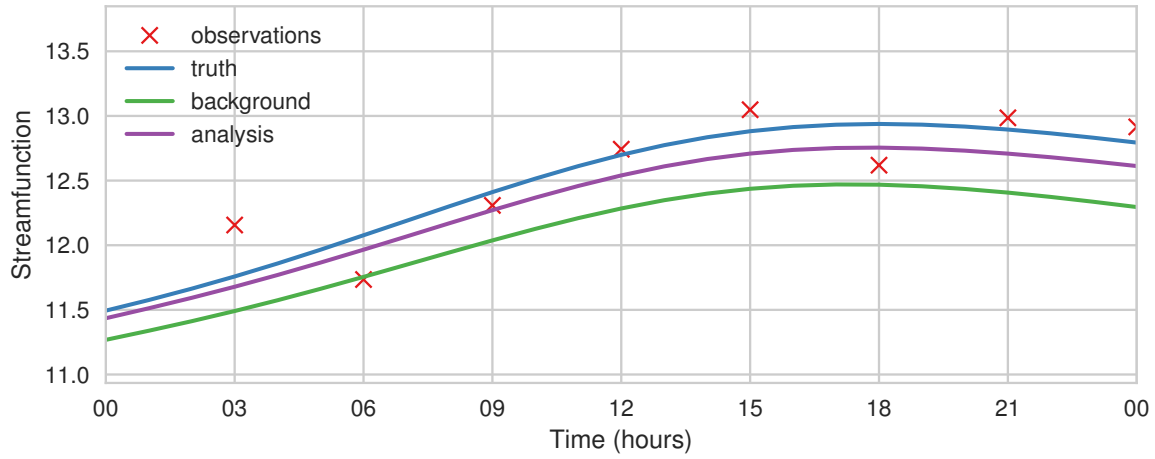


Figure 5.1: The 4D-Var reference experiment. The streamfunction at a randomly chosen observation location is plotted.

Our data assimilation setup consists of a strong-constraint incremental 4D-Var cost function minimised by the CG method. For the background error covariance matrix $\mathbf{P}^b$ we choose the default matrix provided by OOPS — a standard correlation matrix with a horizontal length scale of 1,000 km and a vertical correlation between the two layers of

the model of 0.2. We employ preconditioning in order to reduce the condition number of the Hessian, thereby reducing the number of iterations required for convergence of the CG algorithm (Golub and Van Loan, 1986; Trefethen and Bau III, 1997). In this technique, the original incremental 4D-Var problem, which is essentially to solve

$$\mathbf{S}\boldsymbol{w} = \boldsymbol{b}, \tag{5.4}$$

for $\boldsymbol{w}$, where $\boldsymbol{b} \in \mathbb{R}^n$ is a constant vector which depends on the innovations whose full form we omit for brevity (Haben, 2011), is replaced with the equivalent problem

$$\hat{\mathbf{S}}\boldsymbol{v} = \hat{\boldsymbol{b}}, \tag{5.5}$$

where $\hat{\mathbf{S}} = \mathbf{P}^{\frac{1}{2}}\mathbf{S}\mathbf{P}^{\frac{1}{2}}$, $\boldsymbol{v} = \mathbf{P}^{-\frac{1}{2}}\boldsymbol{w}$, $\hat{\boldsymbol{b}} = \mathbf{P}^{\frac{1}{2}}\boldsymbol{b}$ and $\mathbf{P}$ is the preconditioner matrix. The solution to the original problem can then be recovered by applying $\mathbf{P}^{\frac{1}{2}}$ to the solution of the modified problem. In order for preconditioning to be effective, the matrix $\mathbf{P}$ must be a good approximation to $\mathbf{S}^{-1}$. A common choice for $\mathbf{P}$ is the background error covariance matrix $\mathbf{P}^b$ and we use this in our experiments as well.

We run the forecast model for 18 days to generate a nature run and perform assimilation over the final day. We assimilate observations of streamfunction, horizontal wind components and wind speed which are generated at selected grid points by adding perturbations to the nature run drawn from a zero mean normal distribution with a diagonal observational error covariance matrix, $\mathbf{R}$. The observation error standard deviations for streamfunction, horizontal wind and wind speed are 0.4, 0.6 and 1.2, respectively (these fields are nondimensionalised). The background is generated by drawing perturbations from a zero mean normal distribution with error covariance matrix $\mathbf{P}^b$. Figure 5.1 shows the assimilation window for the reference experiment, for which we use entirely double-precision. We perform three outer loop iterations and the inner loops terminate either when the norm of the cost function gradient has reduced by a factor of 100 or when 50 iterations have been performed.

5.3.2 The two-layer quasigeostrophic system

We use the two-layer quasigeostrophic (QG) model provided by OOPS to generate the nature run and assimilate observations (Pedlosky, 1979; Fandry and Leslie, 1984). This model describes the advection of quasigeostrophic potential vorticity on two model layers q_1 and q_2 over a β-plane, i.e.

$$\frac{\mathrm{D}q_1}{\mathrm{D}t} = \frac{\mathrm{D}q_2}{\mathrm{D}t} = 0, \tag{5.6}$$

where

$$q_1 = \nabla^2\psi_1 - F_1(\psi_1 - \psi_2) + \beta y, \text{ and} \tag{5.7}$$

$$q_2 = \nabla^2\psi_2 - F_2(\psi_2 - \psi_1) + \beta y + R_s. \tag{5.8}$$

Here ψ_i is the streamfunction on layer i, β is the β-plane parameter, y is the y-coordinate, R_s is a bottom forcing term representing e.g. orography and

$$F_i = \frac{f_0^2 L^2}{D_i g \Delta\theta/\overline{\theta}} \tag{5.9}$$

is the coupling term between the two layers defined from the Coriolis parameter at the centre of the domain f_0, the length scale L, the thickness of layer i D_i, the gravitational acceleration g, the difference in potential temperature across the layer interface $\Delta\theta$ and the mean potential temperature $\overline{\theta}$. All terms in Equations 5.7 and 5.8 are nondimensionalised using typical scales of $L = 10^6$ m for length and $U = 10$ ms^{-1} for velocity. The orographic forcing term R_s is therefore also nondimensional and is related to the dimensional orography by

$$R_s = \frac{S}{\mathrm{Ro}D_2}, \tag{5.10}$$

where S is the height of the orography and $\mathrm{Ro} = U/f_0 L$, the Rossby number. In our case S takes the form of a Gaussian function of horizontal distance from a point at the bottom left of the model domain with a height of 2000 m and a standard deviation of 10^6 m.

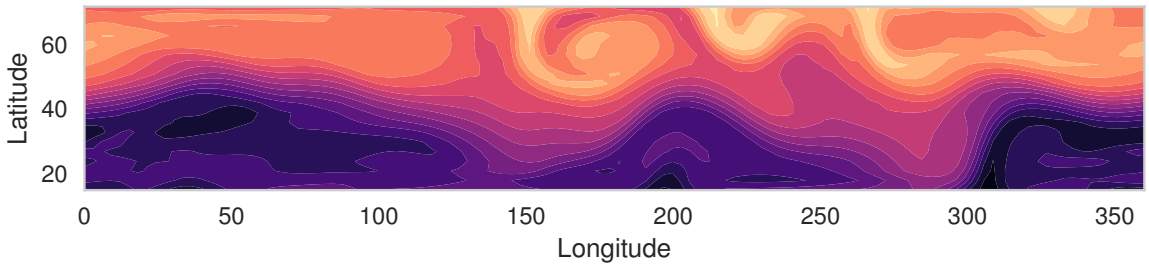


Figure 5.2: A snapshot of top-layer quasigeostrophic potential vorticity q_1.

The spatial domain of one layer of the QG model is 120 by 20 gridpoints and the top and bottom layers have thicknesses of 5500 m and 4500 m, respectively (so that the top of the top layer has an elevation of 10 km). A snapshot of the top-layer quasigeostrophic vorticity is illustrated in Figure 5.2. The domain of the model is equivalent to a zonal band on the sphere between approximately 15° and 72° north. The nondimensional parameter $\beta L/f_0$ takes a value of around 0.16, and therefore the model formulation is pushing the quasigeostrophic assumptions towards their limit of applicability (the quasigeostrophic equations 5.7 and 5.8 are derived under the assumption that $|\beta L| \ll |f_0|$ (Vallis, 2006)). In any case, we are not attempting to assimilate real observations or accurately capture midlatitude dynamics. We simply require a toy fluid dynamical model for performing assimilation experiments.

The potential vorticity is advected by a semi-Lagrangian technique with bi-cubic interpolation to the departure point. Values of the streamfunction one grid point to the north and south of the model domain are required to calculate the zonal velocity during the semi-Lagrangian step. These are constants and determine the mean zonal velocity for each layer. The streamfunction at the southern boundary is zero for both layers whereas the streamfunction at the northern boundary takes values of $\psi_i = -l\overline{u}_i$, where l is the meridional extent of the model domain (6.3×10^6 m) and $\overline{u}_i$ is the mean zonal velocity for layer i. We use $\overline{u}_1 = 40$ ms^{-1} and $\overline{u}_2 = 10$ ms^{-1}.

In order to investigate the impact of a precision reduction on the data assimilation procedure we, as before, use the Reduced Precision Emulator (rpe) to emulate reduced-precision floating-point arithmetic. Although the goal of this study is to investigate whether single-precision 4D-Var is feasible, we expect any effect from reducing precision to be evident and

more easily identifiable if we consider levels of precision other than just double and single. Here we only investigate the impact of reducing the significand width, which determines the precision of a calculation in terms of the number of significant digits of the output. We do not consider a reduction of the exponent below the double-precision width of 11 bits.

5.4 Tangent-linear/adjoint model asymmetry

Before beginning the data assimilation experiments, we first investigated how a precision reduction would affect the standard tangent-linear/adjoint model test. By definition, the relation

$$(\mathbf{M}_{im}(\boldsymbol{x}_0)\boldsymbol{\delta x})^\intercal\, \boldsymbol{\delta y} = \boldsymbol{\delta x}^\intercal\, (\mathbf{M}_{im}^\intercal(\boldsymbol{x}_0)\boldsymbol{\delta y})\,, \tag{5.11}$$

where $\boldsymbol{\delta x}$ and $\boldsymbol{\delta y}$ are arbitrary vectors, relates the tangent-linear and adjoint models. The operators $\mathbf{M}_{im}(\boldsymbol{x}_0)$ and $\mathbf{M}_{im}^\intercal(\boldsymbol{x}_0)$ are in reality subroutines using floating-point arithmetic and are subject to different rounding errors when they operate on the input vector. The identity in equation 5.11 will therefore not hold exactly but will instead be violated slightly. At ECMWF, as a rule of thumb, the identity should be satisfied to at least the 10th decimal place to ensure convergence of 4D-Var (Filip Váňa, personal communication, June 2017). Certainly, when testing the validity of the code of the adjoint model with respect to the tangent-linear model the identity should be satisfied to as many digits as possible, and the use of double-precision would ensure around 15 decimal places. In the following experiments we will investigate how a violation of this identity beyond the 7th decimal place (i.e. single-precision and below) could prevent or slow down the convergence process of 4D-Var.

Figure 5.3 shows the relative error between the left-hand side and right-hand side of equation 5.11 for the QG model as a function of numerical precision, along with the machine epsilon, $\epsilon = 2^{-p}$, where p is the number of significand bits. We define the relative error as

$$\frac{|\text{LHS} - \text{RHS}|}{|\text{RHS}|} = \frac{|\,(\mathbf{M}_{im}(\boldsymbol{x}_0)\boldsymbol{\delta x})^\intercal\, \boldsymbol{\delta y} - \boldsymbol{\delta x}^\intercal\, (\mathbf{M}_{im}^\intercal(\boldsymbol{x}_0)\boldsymbol{\delta y})\,|}{|\boldsymbol{\delta x}^\intercal\, (\mathbf{M}_{im}^\intercal(\boldsymbol{x}_0)\boldsymbol{\delta y})\,|}. \tag{5.12}$$

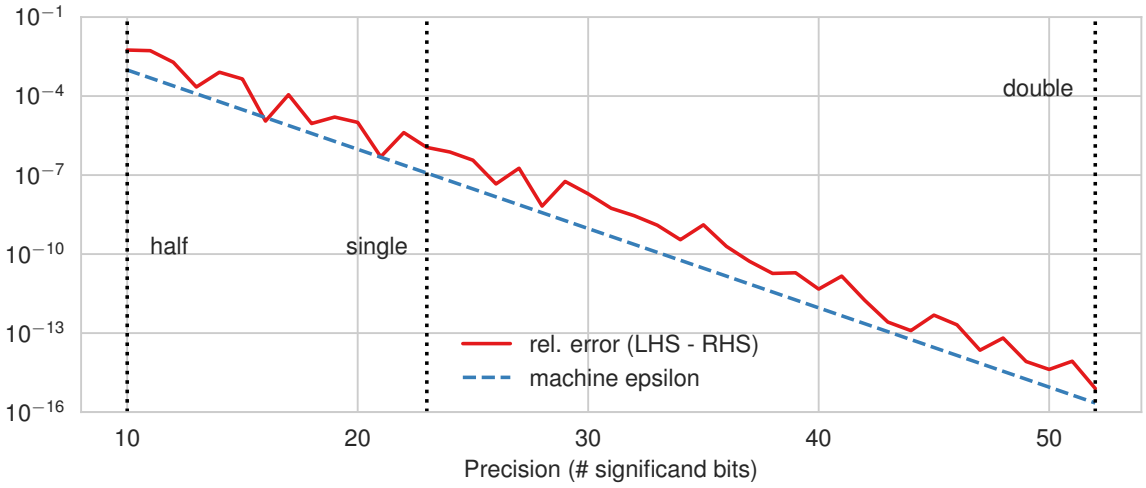


Figure 5.3: The relative error between the left-hand side and right-hand side of the tangent-linear/adjoint model test, equation 5.11, as a function of numerical precision. The machine epsilon is also plotted as a function of precision for comparison.

The relative error in a single floating-point operation is proportional to ϵ (Higham, 2002), i.e. when one significand bit is removed the floating-point truncation error doubles. However, Figure 5.3 shows that this scaling is also valid for the tangent-linear and adjoint models overall. This is perhaps not surprising considering that they are, by definition, linear in the input. Essentially, even though the tangent-linear and adjoint models are formulated as computer subroutines, they behave like a matrix-vector multiplication. To see this, consider Figure 5.4, which shows the same result as in Figure 5.3 only the tangent-linear model has been replaced with a square matrix whose elements are random numbers and whose dimension is the same as the state vectors $\boldsymbol{\delta x}$ and $\boldsymbol{\delta y}$ (and the adjoint was replaced with the transpose of that matrix). The scaling of the relative error is identical to that in Figure 5.3. It is reassuring that although the impacts of reducing precision may be deleterious they are at least relatively predictable. We now proceed to investigate the impacts of reducing precision in the tangent-linear and adjoint models in data assimilation experiments.

5.5 Effects of reducing precision on convergence

We begin by dividing particular incremental 4D-Var problems into those that are well-conditioned and those that are ill-conditioned. The condition number of the Hessian, $\mathbf{S}$,

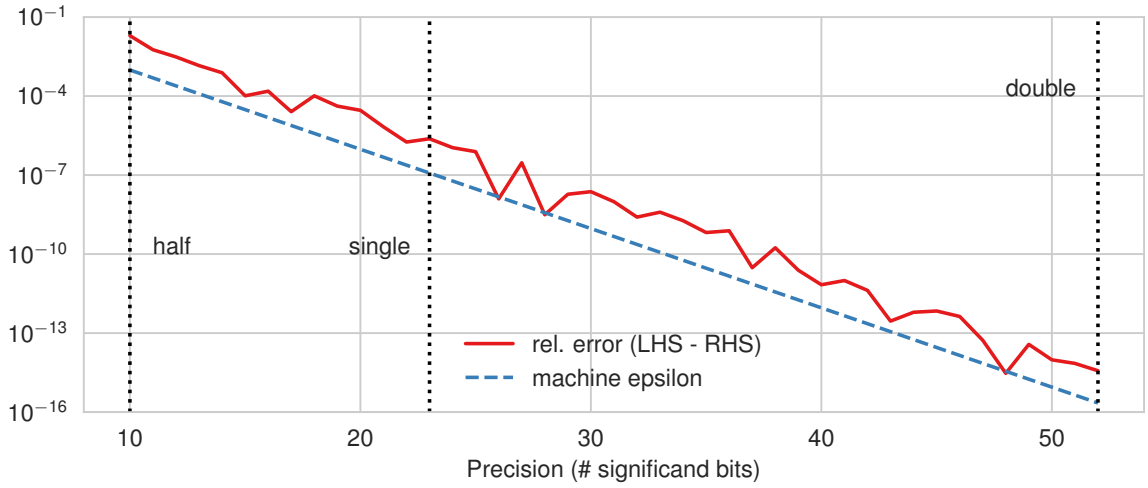


Figure 5.4: The same as Figure 5.3, only the tangent-linear and adjoint models have been replaced with matrices of random numbers.

determines the number of iterations required to reach convergence — the higher the condition number, the longer a minimisation algorithm will take to converge. We focus on the so-called spectral condition number, defined by

$$\kappa(\mathbf{S}) = \frac{\lambda_{\max}(\mathbf{S})}{\lambda_{\min}(\mathbf{S})}, \tag{5.13}$$

where $\lambda_{\max}(\mathbf{S})$ and $\lambda_{\min}(\mathbf{S})$ are the maximum and minimum eigenvalues of the Hessian, respectively. In fact, the relevant condition number is actually that of the Hessian of the preconditioned system, which by construction is simply equal to the largest eigenvalue of this Hessian, as preconditioning by $\mathbf{P}^b$ renders the smallest eigenvalue equal to unity (Haben, 2011). The condition number of the Hessian contains information about the shape of the corresponding quadratic cost function — a condition number of unity indicates a perfectly symmetric quadratic "bowl" whereas a higher condition number indicates ellipticity.

Figure 5.5 illustrates the behaviour of the convergence of incremental 4D-Var as precision is reduced in the tangent-linear and adjoint models, for well-conditioned and ill-conditioned problems, with condition numbers of around 15 and 1500, respectively. The color axis indicates the logarithm of the value of the cost function evaluated at the current iterate. We estimated the condition number of each problem through the Lanczos

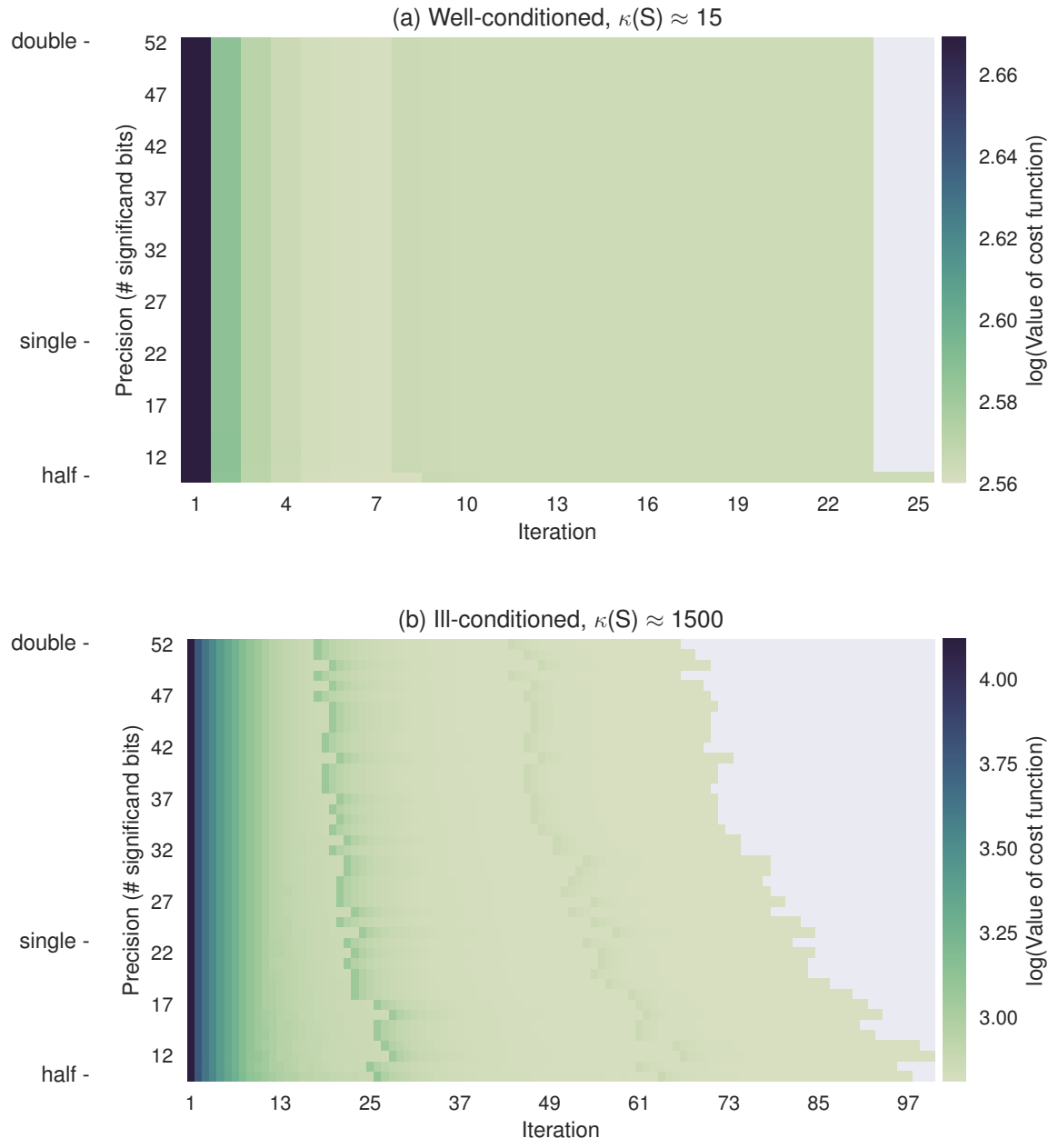


Figure 5.5: An illustration of the behaviour of convergence of the preconditioned conjugate gradient minimiser as precision is reduced in the tangent-linear and adjoint models, for (a) well-conditioned and (b) ill-conditioned problems. The number of iterations required to meet the convergence criteria is shown on the x-axis. The color indicates the value of the cost function evaluated at the current iterate as the minimisation proceeds from left to right. The condition number of the problem is given by κ(S).

algorithm (Fisher, 1998). The conditioning was changed by changing the observation error standard deviation: for the well-conditioned problem we used the baseline values mentioned in section 5.3.1, whereas for the ill-conditioned problem we used values 10 times smaller than this. The ratio of the two condition numbers, $1500/15$, is consistent with theory, which states that the condition number of the Hessian increases with the inverse of the square of the observation error standard deviation (Haben et al., 2011). Essentially, more accurate observations place a tighter restriction on the minimization of the cost function so that more iterations are required to reach the minimum within a given tolerance. Note that the temporary increases in the value of the cost function as the iteration proceeds, most clearly visible for Figure 5.5b as dark vertical lines, demarcate the beginning and end of subsequent inner loops. It should also be noted that our ill-conditioned problem is only defined with respect to the well-conditioned case. A condition number of 1500 is in fact well below the typical value encountered in operational NWP centres. At ECMWF, for example, a typical assimilation problem may have a condition number of as much as 20000 (Marcin Chrust, personal communication July 2019). This limits the conclusions we draw in the following as we make no attempt to probe these much more ill-conditioned problems.

Figure 5.5 illustrates how, for the well-conditioned problem, precision can be reduced significantly without impacting the convergence process. Going from double-precision (52 significant bits) to single-precision (23 significand bits) and beyond results in essentially the same solution. On the other hand, for the ill-conditioned problem, although the reduction of the cost function is more or less equivalent for the first 15 iterations, regardless of precision, more and more iterations are required to reach convergence as significand bits are removed. Specifically, using single-precision means that around 25% more iterations must be performed in order to reach the same solution as when using double-precision. The trend continues as precision is reduced further, with half-precision tangent-linear and adjoint models requiring around twice as many iterations as double-precision. Of course, if using single-precision results in a longer convergence process then this may negate any computational speed-up from using reduced-precision arithmetic. Nevertheless, we show

in the next section how we might still be able to use such a low precision in 4D-Var successfully.

5.6 Re-orthogonalising the residuals

$\boldsymbol{w}_0 = \mathbf{0}, \boldsymbol{r}_0 = \boldsymbol{b}, \boldsymbol{p}_0 = \boldsymbol{r}_0$
for $n = 1, 2, 3, \ldots$ **do**

$$\alpha_n = (\boldsymbol{r}_{n-1}^\mathsf{T}\boldsymbol{r}_{n-1})/\boldsymbol{p}_{n-1}^\mathsf{T}\mathbf{A}\boldsymbol{p}_{n-1} \qquad \text{step length}$$
$$\boldsymbol{w}_n = \boldsymbol{w}_{n-1} + \alpha_n\boldsymbol{p}_{n-1} \qquad \text{approximate solution}$$
$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \alpha_n\mathbf{A}\boldsymbol{p}_{n-1} \qquad \text{residual}$$
$$\beta_n = (\boldsymbol{r}_n^\mathsf{T}\boldsymbol{r}_n)/(\boldsymbol{r}_{n-1}^\mathsf{T}\boldsymbol{r}_{n-1}) \qquad \text{improvement this step}$$
$$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_n\boldsymbol{p}_{n-1} \qquad \text{search direction}$$

end

Algorithm 5.1: The conjugate gradient (CG) algorithm.

We believe that the standard re-orthogonalisation technique used ubiquitously for the CG algorithm, which we omitted in the previous experiments for the sake of argument, may allow the use of reduced-precision arithmetic in the tangent-linear and adjoint models. As is well known, rounding errors can cause problems for the CG algorithm. To see why, consider again the incremental 4D-Var problem, to solve

$$\mathbf{S}\boldsymbol{w} = \boldsymbol{b} \tag{5.4}$$

for $\boldsymbol{w}$. The CG algorithm for solving Equation 5.4 is outlined in Algorithm 5.1 (Trefethen and Bau III, 1997). For simplicity, we only show the un-preconditioned CG algorithm. At each step of the iteration we compute the residual $\boldsymbol{r}_n$, which can be shown to be the negative of the gradient of the cost function evaluated at $\boldsymbol{w}_n$. The CG algorithm therefore bears a resemblance to the simple gradient descent method. However, by taking advantage of the fact that the incremental 4D-Var Hessian, $\mathbf{S}$, is real, symmetric and positive definite, we are able to dramatically reduce the number of iterations required for convergence.

For fast convergence, the CG algorithm requires that the residuals computed at successive iterations be orthogonal to each other, and if we could use perfect arithmetic then this

$\boldsymbol{q}_0 = \boldsymbol{b}/||\boldsymbol{b}||, \mathbf{Q}_0 = \boldsymbol{q}_0$
for $n = 1, 2, 3, \ldots$ **do**

$\tilde{\boldsymbol{q}}_n = \mathbf{S}\boldsymbol{q}_{n-1}$	next Krylov vector
$h_i = \tilde{\boldsymbol{q}}_n^{\mathsf{T}}\boldsymbol{q}_i, i = 0, 1 \ldots n-1$	get projection on previous Krylov vectors
$\hat{\boldsymbol{q}}_n = \tilde{\boldsymbol{q}}_n - \sum_{i=0}^{n-1} h_i \boldsymbol{q}_i$	reorthogonalise next Krylov vector
$\boldsymbol{q}_n = \hat{\boldsymbol{q}}_n/\|\|\hat{\boldsymbol{q}}_n\|\|$	normalise next Krylov vector
$\mathbf{Q}_n = [\mathbf{Q}_{n-1}\|\boldsymbol{q}_n]$	append Krylov vector to Krylov matrix
$\boldsymbol{y} = \arg\min \|\|\mathbf{S}\mathbf{Q}_n\boldsymbol{y} - \boldsymbol{b}\|\| \quad (= \|\|\boldsymbol{r}_n\|\|)$	minimise residual norm in Krylov subspace
$\boldsymbol{w}_n = \mathbf{Q}_n\boldsymbol{y}$	approximate solution

end

Algorithm 5.2: The generalised minimal residual (GMRES) algorithm.

would indeed be the case. However, the use of finite precision floating-point arithmetic causes a loss of orthogonality among these residual vectors, more so for low-precision floats than for high-precision floats. We hypothesize that this loss of orthogonality is responsible for the retarded convergence process when reducing precision. To support this hypothesis, we repeated the experiment of Figure 5.5b but with the generalised minimal residual (GMRES) algorithm. This algorithm, outlined in Algorithm 5.2 (again, without preconditioning), minimises the 2-norm of the residual $\boldsymbol{b} - \mathbf{S}\boldsymbol{w}_n$ in a succession of so-called Krylov subspaces, spanned by vectors resulting from repeated application of the Hessian $\mathbf{S}$ to $\boldsymbol{b}$ (Trefethen and Bau III, 1997). The vectors spanning these Krylov subspaces must be orthogonal for efficient convergence, analogous to the orthogonality requirement of the CG residuals. This requires an explicit reorthogonalisation step (see step "reorthogonalise next Krylov vector"). Accordingly, GMRES does not suffer as much in the presence of rounding errors and, as shown in Figure 5.6, we were able to use as few as 15 significand bits without affecting the rate of convergence of the 4D-Var minimization when using this algorithm.

At this point we could simply reject the CG algorithm in favour of GMRES, given how well it handles a reduction in precision. However, CG has several advantages over GMRES in the solution of physical problems. Whereas GMRES minimises the 2-norm of the residual, CG minimises the $\mathbf{S}$-norm of the error vector $(\boldsymbol{w}^* - \boldsymbol{w}_n)^{\mathsf{T}}\,\mathbf{S}\,(\boldsymbol{w}^* - \boldsymbol{w}_n)$, where $\boldsymbol{w}^*$

is the solution. The Hessian $\mathbf{S}$ encodes physical information about the uncertainty of the background and observations (it is related to the inverse of the respective error covariance matrices in Equation 5.1) and so when computing the $\mathbf{S}$-norm of a vector, the elements of the vector are weighted according to their physical uncertainty. In the case of the 2-norm of GMRES, all elements of the residual are treated equally.

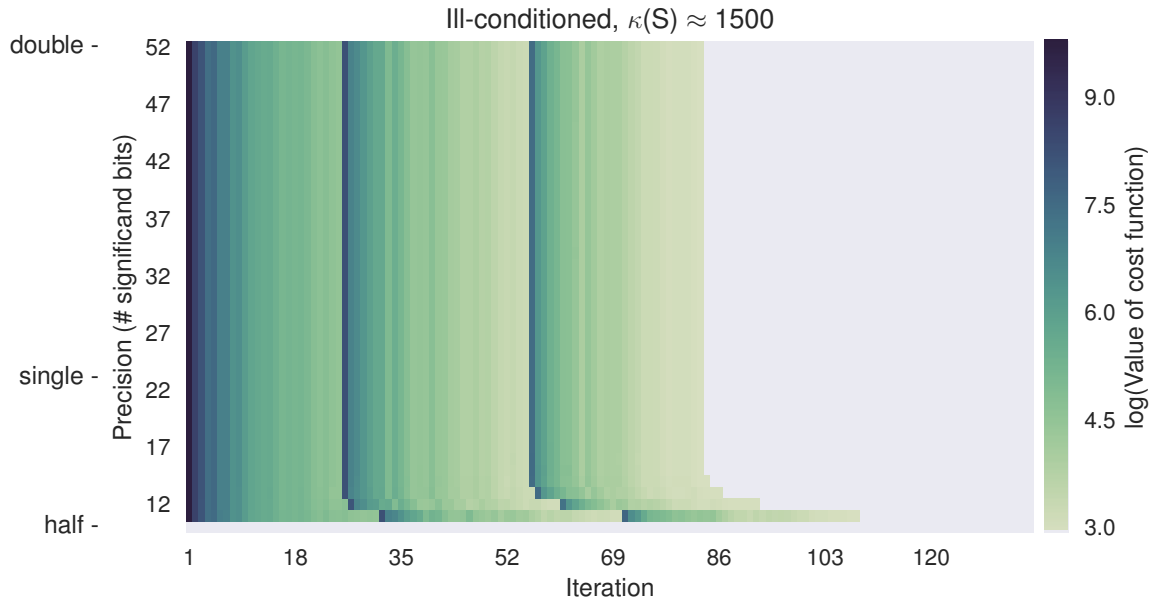


Figure 5.6: The same as Figure 5.5b except that the generalized minimal residual (GMRES) minimization technique is used.

The standard solution to the problem of losing orthogonality, in the case of CG, is to re-orthogonalise at each iteration the current residual against all of the previous residuals using the Gram-Schmidt process. In our case the situation is complicated by the presence of the preconditioner matrix, but the solution is very similar. Essentially, the residuals must be made conjugate with respect to the inverse of the preconditioner matrix. Using re-orthogonalisation is beneficial even with double-precision and is used operationally at ECMWF (Fisher, 1998) and, by default, is used within the PCG minimiser of OOPS.

We repeated the experiment shown in Figure 5.5 but with re-orthogonalisation and the results are shown in Figure 5.7. Clearly, re-orthogonalisation allows one to use a far lower precision in the tangent-linear and adjoint models than otherwise. Even with only 11

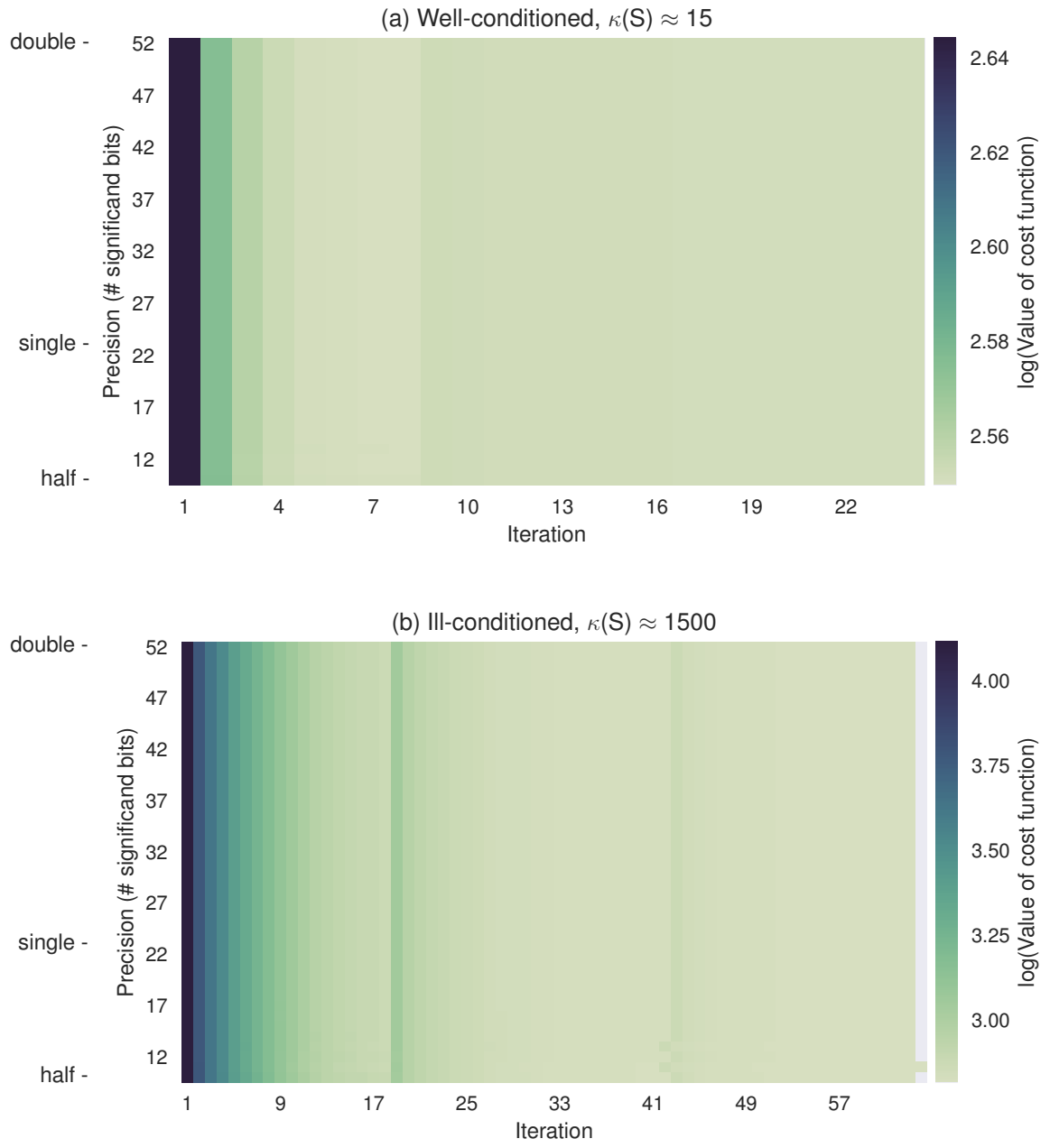


Figure 5.7: The same as Figure 5.5 except that re-orthogonalisation is used in the preconditioned conjugate gradient minimiser

significand bits, which is almost half-precision, the convergence behaviour of PCG remains the same. Figure 5.7 indicates that the asymmetry introduced between the tangent-linear and adjoint models (Figure 5.3) when reducing precision does not preclude a successful minimisation of the 4D-Var cost function.

5.7 Conclusion

The tangent-linear and adjoint models are an obvious target for using reduced-precision arithmetic as they dominate the computational cost of 4D-Var. For this reason they are typically run at a lower resolution than the full nonlinear model. Reducing precision in these models could provide another means to reduce their computational overhead. However, reducing precision in these models introduces a stronger asymmetry between them, compared with double-precision, which we can expect to disrupt the convergence process. We have demonstrated that this is indeed the case, for ill-conditioned problems, where lowering precision results in a deleterious loss of orthogonality in the conjugate gradient residual vectors. However, by using the standard re-orthogonalisation technique, a technique already used operationally, we are able to minimise the 4D-Var cost function using single-precision linear models in the same number of iterations as with double-precision.

Our results are promising and indicate that the use of single-precision (or even lower precision) linear models in 4D-Var for operational numerical weather prediction may be viable. However, as mentioned earlier, the 4D-Var problem as encountered in operational systems is somewhat more poorly conditioned than the one presented here. It remains to be seen to what extent reorthogonalization can cope with much higher condition numbers. We also did not include any model error in our data assimilation experiments. Chapter 4 demonstrated that the presence of model error can allow precision to be reduced to a much lower level than otherwise and therefore we must include model error to make a truly fair test of reduced-precision models in 4D-Var. This would open up the possibility of investigating weak-constraint 4D-Var, in which the model acts only as a weak-constraint to the trajectory over the window, compared with the strong-constraint it provided in our ex-

periments. Trémolet (2006) describes several different ways to relax the strong-constraint assumption and suggests how this might affect the conditioning of the problem. For example, the 4D-Var$_x$ method breaks down the assimilation window into a number of "sub-windows", within each of which the model provides a strong-constraint. Although the conditioning of this problem appears to be poorer than for strong-constraint 4D-Var (with the condition number increasing as the number of sub-windows increases, at least under some simplifying assumptions), it remains to be seen whether more sophisticated preconditioning techniques couldn't bring the condition number down to a similar level as for the strong-constraint problem.

A natural next test would be to consider a full-complexity model such as the IFS of ECMWF. Reducing precision throughout the tangent-linear and adjoint models would not be straightforward owing to the complexity of the model code. However, as a simple test one could choose a computationally expensive but algorithmically simple "kernel" in the tangent-linear and adjoint models and only reduce precision in that kernel. For example, one could use a single-precision matrix-matrix multiply in the Legendre transform, as in the half-precision experiments of Chapter 6. This would increase the asymmetry between the tangent-linear and adjoint models and would therefore allow the reorthogonalization technique to be tested without making significant code changes to the IFS.

Chapter 6

Accelerating Legendre transforms through reduced-precision arithmetic

6.1 Introduction

The use of single-precision represents a low-hanging fruit for atmospheric modelling, in the sense that single-precision provides an adequate range for atmospheric variables and is also widely available on existing hardware. It is feasible to run virtually an entire model at single-precision, and such a procedure should deliver around a 50% reduction in computational cost (Váňa et al., 2017; Nakano et al., 2018). Half-precision, however, is not as simple. In Chapter 4 we demonstrated that the use of half-precision floating-point arithmetic throughout an atmospheric model is non-trivial, mainly because half-precision floats have a limited range. Nevertheless, half-precision arithmetic does have a role to play in accelerating computationally expensive "kernels" — certain algorithms that reoccur throughout a model and often dominate in terms of computational cost (Müller et al., 2019). In this chapter we focus on the use of half-precision arithmetic in the Legendre transform, a key component of spectral dynamical cores such as that used in the Integrated Forecasting System (IFS) of the European Centre for Medium-Range Weather Forecasts (ECMWF).

The spectral dynamical core has been the method of choice for discretising the equations of motion of the atmosphere at ECMWF almost since its inception (Simmons et al., 1989). The use of a spectral model opens opportunities for the use of mixed-precision. Chantry

et al. (2019), for example, developed a "scale-selective" approach to reducing-precision in the spectral space calculations of the IFS of ECMWF: they used high precision for large-scale atmospheric variables and low precision for small-scale atmospheric variables. Overall this allowed them to use far fewer bits than the baseline, global double-precision, without affecting the weather forecasting skill.

Spectral models perform only some operations in spectral space, (hence why they are sometimes called *pseudo*-spectral), in which fields are represented using a basis of spherical harmonics. Time-stepping and computation of derivatives, among other things, are efficiently computed in spectral space. Computation of the Laplacian of a field, for example, is trivial in spectral space (Krishnamurti et al., 2014). Other operations are harder to perform or even formulate in spectral space however. Products of fields and tendencies, in the case of advection, are particularly troublesome as they involve higher wavenumbers that must also be representable in the discretised spectral space. If the products are evaluated in spectral space then the number of required computations rapidly becomes infeasible as resolution is increased (Randall, 2013). These nonlinear products are instead performed in grid point space, and the result is transformed back to spectral space, where the timestepping occurs. Physical parametrization schemes are also difficult to formulate in spectral space, so the contributions to the field tendencies from physics are computed in grid point space as well.

This splitting of the computations between spectral space and grid point space requires a transform of several fields from one space to the other and back every timestep. Each transform requires the use of a fast Fourier transform and a Legendre transform. In the past, the transforms did not constitute a significant fraction of the total computational cost of the model and so this seemingly unwieldy process paid off. However, the Legendre transform, which algorithmically is a batch of matrix-matrix multiplications, scales poorly and the cost of this transform tends to dominate the total cost of the system as horizontal resolution is increased (Wedi, 2014). At the highest operational resolution (9 km), the transforms back and forth between spectral and grid point space already constitute around 40% of the

total computational cost (atmosphere only and without I/O (Düben et al., 2019 (submitted)). This problem has motivated the development of more efficient Legendre transform algorithms than the standard matrix-matrix multiplication, which scales as $\mathcal{O}(n^3)$, where n is proportional to the number of latitudes and longitudes and therefore denotes the model's horizontal resolution. There is, for example, the "fast Legendre transform" which was developed to allow spectral models to run up to 1 km resolution and higher (Wedi et al., 2013). Like the technique we will outline below, the fast Legendre transform trades accuracy for speed (but through a compression of the Legendre polynomials, not through reduced-precision floating-point arithmetic). However, although the algorithm scales well, as $\mathcal{O}(n^2 \log n)$, there is a large minimum cost to this algorithm such that it only becomes competitive at very high resolutions. For this reason the fast Legendre transform is still not used in operational forecasts at ECMWF.

Given its high cost and relatively low complexity, the Legendre transform is a suitable candidate for reduced-precision arithmetic. A treatment of the Legendre transforms similar to that of the SPEEDY model in Chapter 4 would no doubt prove fruitful. However, the use of the 22 bit floating-point format would require specialised hardware and, unfortunately, operational meteorology is simply not a large enough market to influence the design of computers. A more productive alternative is to find uses for reduced-precision hardware designed for deep-learning. This is another area where high-precision is not necessary, given the probabilistic nature of the training procedure (e.g. Micikevicius et al. (2017)). The development of high-performance computers is increasingly influenced by the field of deep-learning, backed by key players such as Google, Intel and NVIDIA, and this is evident in the profusion of reduced-precision processors that have recently become available. As an example, the NVIDIA V100 graphics processing unit (GPU) features a separate matrix-matrix multiplier known as a "Tensor Core" for which they claim a speed-up factor of 16 times compared with a double-precision matrix-matrix multiply (NVIDIA, 2017a). This device operates on half-precision floating-point numbers (the output can be half-precision or single-precision) and is designed for accelerating common operations encountered in

deep-learning[1]. We believe that, in the future, numerical weather model developers will increasingly have to "piggyback" on these developments in deep-learning-oriented hardware in order to make optimal use of future supercomputers.

In this chapter, we use the software emulation technique outlined in Chapter 1 to investigate the use of half-precision arithmetic and the aforementioned Tensor Core in the Legendre transform. We assess the impact on the meteorological skill of the IFS and find that, with a few simple treatments, both half-precision arithmetic and the Tensor Core can be used successfully. In Section 6.2 we introduce the Legendre transforms mathematically and algorithmically, in Section 6.3 we discuss the implications for reducing precision in the Legendre transforms, in Section 6.4 we evaluate the impact of this procedure on the skill of weather forecasts, both deterministic and probabilistic, and finally we conclude in Section 6.6.

6.2 The Legendre transform

Any real horizontal scalar field on the sphere $f^p(\lambda, \theta)$, where λ is longitude, θ is latitude and p indexes the field (i.e. which model level and meteorological variable), can be written as a sum of spherical harmonics, $Y_{m,n}(\lambda, \theta)$:

$$f^p(\lambda, \theta) = \sum_{m=-\infty}^{\infty} \sum_{n=|m|}^{\infty} \psi^p_{m,n} Y_{m,n}(\lambda, \theta) \tag{6.1}$$

where $\psi^p_{m,n}$ is the complex amplitude of the (m, n) spectral mode for field p, m is the zonal (or "Fourier") wavenumber and n is the total wavenumber. Spherical harmonics are the eigenfunctions of the Laplacian in spherical coordinates and can be written as a product of a function of latitude and a function of longitude:

$$Y_{m,n}(\lambda, \theta) = P_{m,n}(\mu) e^{im\lambda}, \tag{6.2}$$

[1]Note that, as of June 2019, the top supercomputer in the world is comprised mostly of V100 GPUs. This is Summit at the Oak Ridge National Laboratory.

where $P_{m,n}(\mu)$ is an associated Legendre polynomial of the first kind and $\mu = \cos(\theta)$.

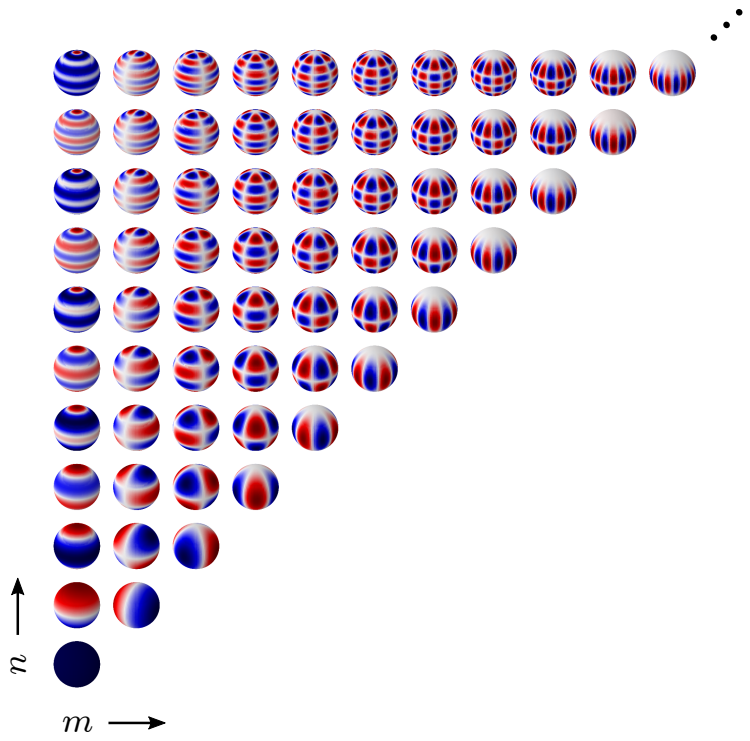


Figure 6.1: The spherical harmonics for a triangular truncation of $N = 10$.

In practice, the sums in Equation 6.1 are truncated at some total wavenumber N and zonal wavenumber M. Almost all modern numerical models use "triangular truncation", where $N = M$. Some early models used a "rhomboidal truncation", in which $N = M + |m|$, but the triangular approach eventually came to dominate for a number of reasons. For example, although models with rhomboidal truncation keep more spherical harmonic modes and therefore have a higher spectral resolution, the observed kinetic energy spectrum of the atmosphere conforms more closely with a triangular truncated spectrum (Randall, 2013). The spherical harmonics remaining after a triangular truncation of $N = M = 10$ are visualized in Figure 6.1. Note that only the spherical harmonics with positive zonal wavenumbers m need to be considered because all of the meteorological fields are real in grid point space (the negative m spectral coefficients must be the exact complex conjugate of the corresponding positive m coefficients). The truncation wavenumber N determines the model

resolution which is referred to by the TXN labelling convention, "T" meaning "triangular" and "X" referring to the type of grid used in grid point space. We consider both the linear reduced Gaussian grid (TLN) and the cubic octahedral reduced Gaussian grid (TCoN), the latter now being used operationally at ECMWF. A spectral resolution of TCo639 corresponds to a grid point resolution of approximately 18 km at the Equator, whereas TCo3999 corresponds to a resolution of approximately 2.5 km. To transform from grid point space to spectral space, a one-dimensional Fourier transform is applied to generate a Fourier representation of the fields along each longitude. In a second step, a Legendre transform is applied to obtain the spectral coefficients $\psi^p_{m,n}$ for the global representation in spectral space. To transform from spectral space to grid point space, an inverse Legendre transform is applied followed by an inverse Fourier transform.

Applying the Fourier transform to a grid point field produces the Fourier amplitudes at each latitude, $F^p_m(\mu)$. The direct Legendre transform then completes the transform into spectral space:

$$\psi^p_{m,n} = \int_{-1}^{1} F^p_m(\mu) P_{m,n}(\mu) d\mu. \tag{6.3}$$

Equation 6.3 is evaluated numerically using Gaussian quadrature:

$$\int_{-1}^{1} F^p_m(\mu) P_{m,n}(\mu) d\mu = \sum_{i=1}^{N+1} w_i F^p_m(\mu_i) P_{m,n}(\mu_i), \tag{6.4}$$

where the index i iterates over the Gaussian $N+1$ latitudes, w_i are the Gaussian weights and μ_i is the location of the ith Gaussian latitude. The inverse Legendre transform is defined by

$$F^p_m(\mu) = \sum_{n=|m|}^{N} \psi^p_{m,n} P_{m,n}(\mu). \tag{6.5}$$

In the IFS, equations 6.4 and 6.5 are computed as a series of matrix-matrix multiplications parallelised over each m. In practice, the exact operations differ from those presented so far due to several complications. For example, the spectral amplitudes are complex and so the above transforms must be performed separately for the real and imaginary compo-

nents. Additionally, the cost of both the direct and inverse transforms can be reduced by a factor of two by taking advantage of the symmetry properties of the associated Legendre polynomials.

6.3 Reducing precision in the Legendre transforms

Naively using half-precision arithmetic to perform the Legendre transforms will lead almost instantly to a crash of the model. This is because numbers larger than the maximum half-precision value occur, are rounded to infinity and then infect every calculation that they occur in eventually causing a floating-point exception. However, a simple procedure can secure the Legendre transforms from overflow errors. Given that equations 6.4 and 6.5 are linear operations, by multiplying the field-to-be-transformed by a scalar and dividing the transformed-field by the same scalar we recover the correct result had we applied no rescaling. This technique was used by Micikevicius et al. (2017) in order to allow mixed-precision training of a neural net. Higham et al. (2019) also investigated this technique for accelerating a linear solver. We propose to use this simple procedure to enable half-precision computations for the Legendre transforms. Note that, although underflow errors can also occur, these are not fatal for the matrix-matrix multiplications underlying the Legendre transform. An underflow error will only crash the model if the number that underflows is used as a divisor (leading to a divide-by-zero error) and this cannot occur in a matrix-matrix multiplication.

For both the forward and inverse transforms, we first compute the maximum of the incoming field. We then multiply the entire field by a rescaling factor such that this maximum is equal to a prescribed value a. We then apply the transform using either a half-precision matrix-matrix multiplication (half_trans) or the Tensor Core (tensor_core) and divide the result by the rescaling factor (in both cases using software emulation instead of real hardware, as discussed below).

We performed tuning experiments to determine the optimal value for a by computing the day 5 root-mean-square error of 500 hPa geopotential height with respect to a fully

double-precision simulation for different values of a between 0.1 and 1000 (values outside this range gave significantly larger errors or crashed entirely, likely due to overflows and underflows). We found no change in the error within this range. This is not surprising given that floating-point numbers have a constant relative precision — within the range of representable numbers there is no "optimal rescaling" that will minimise the error. We decided to use $a = 100$ as this happened to give a slightly lower error than other values, but we do not consider this choice to be significant. Note that it is technically possible (but unlikely) for overflows to occur when $a = 100$ for high resolutions. This is because the longest sum for a resolution of spectral truncation N is $N + 1$ elements long, and therefore if all elements of this sum happen to be close to a then the result of the sum will be larger than 65504.

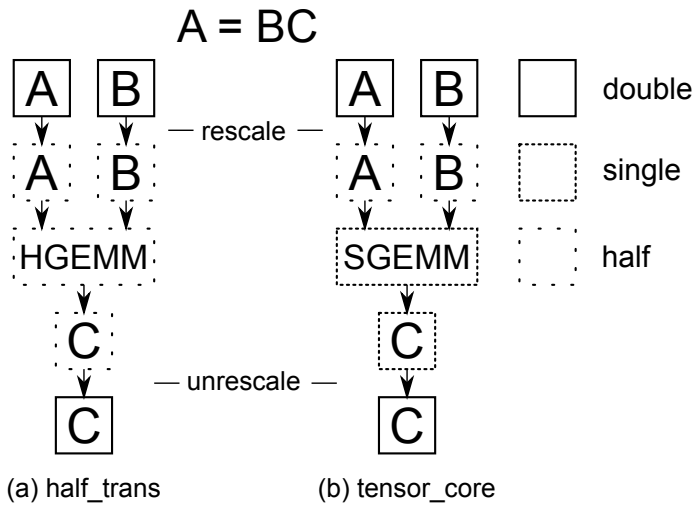


Figure 6.2: A visualisation of the two core experimental setups: (a) half_trans and (b) tensor_core.

To test the use of mixed-precision with IFS without porting the model code to GPUs we again use the Reduced Precision Emulator (rpe). We investigate the difference between the matrix-matrix multiplication in two reduced-precision setups, half_trans and tensor_core, that are illustrated in Figure 6.2. For both setups, the double-precision inputs A and B are first rescaled and then placed into half-precision containers. Then, the matrix-matrix multiplication is performed at half-precision for half_trans (HGEMM) and single-precision for

tensor_core (SGEMM). This is a realistic emulation of the Tensor Core on an NVIDIA V100 GPU that takes half-precision inputs but produces a single-precision output. We verified this emulation procedure against a real Tensor Core on an NVIDIA V100 GPU. We computed a matrix-matrix multiplication both using the emulator and using an actual Tensor Core and found that the difference between the two matrices was consistent with standard matrix-matrix multiplication error bounds (Higham, 2002). Finally, for both setups, the output is then "unrescaled" and placed into a double-precision output container.

When we applied the aforementioned scaling technique to the Legendre transforms, we encountered a further problem arising from the spectral modes with a zonal wavenumber (m) of zero. The spectral coefficient $\psi^p_{0,0}$ is by definition the global mean of the field p and so, for certain fields such as temperature, this has a much greater amplitude than the remaining coefficients. Therefore, when the $F_0(\mu)$ term is formed from the sum in Equation 6.5 (the inverse Legendre transform) rounding errors can lead to a loss of precision and spurious zonally-symmetric patterns in the transformed fields. A similar effect is observed for the direct Legendre transform, though the exact reason why is not clear to us. The direct transform consists of a sum over latitude and it is not obvious that one of the terms in this sum should dominate the others in magnitude in a similar way to the inverse transform. In any case, we can eliminate this problem simply by using double-precision arithmetic for the $m = 0$ direct and inverse transforms. The remaining zonal wavenumbers are still computed using half-precision and so this fix has a negligible impact on the total cost of the transforms. Additionally, the model initialisation was still performed with double-precision arithmetic, even if the Legendre transforms used half-precision arithmetic. Several transforms are performed before the main integration begins which we reason should be performed with a high-precision.

6.4 Forecast tests

In this section, we present results from deterministic and ensemble forecast experiments. Naturally, we expect reducing precision to change the outcome of the model integration.

However, weather and climate forecasts are perturbed by several sources of errors (imperfect initial conditions, the use of imperfect physical parametrization schemes due to a limited model resolution etc.). It is therefore possible that errors due to reducing precision will be insignificant in comparison to other errors. To test whether rounding errors are significant in our mixed-precision simulations, we compare the results with simulations that use a stochastic parametrisation scheme. These schemes are designed to generate spread between different simulations of an ensemble forecast by adding a stochastic forcing into the model configuration. Since this spread was adjusted to fit the average error of weather forecasts, the difference between model simulations with and without the stochastic parametrisation scheme enabled can serve as an estimate of the magnitude of model error. For our tests we used the Stochastically Perturbed Parametrisation Tendencies (SPPT) scheme which applies a multiplicative stochastic forcing to the total tendencies provided by the physics parametrisations (Palmer et al., 2009). As long as the impact from reducing precision is smaller than the impact of SPPT, reducing precision is not likely to noticeably affect the probabilistic skill of the model.

6.4.1 Deterministic forecast skill

As a first test we performed a double-precision 10 day simulation and compared it with several setups. For the experiments in this section we used the open-source variant of the IFS, OpenIFS, at cycle 38r1. This cycle used a linear grid. For each setup we computed the global mean root-mean-square error of the 500 hPa geopotential height field (Z500 RMSE) of the setup with respect to the double-precision control. We used the exact same initial conditions for all forecasts in this section so that only model error with respect to the double-precision control would cause two forecasts to diverge. For the first setup we used half-precision Legendre transforms (half_trans) and for the second setup we used the emulated Tensor Core (tensor_core), as discussed in Section 6.3. Our third setup consisted of a model with entirely single-precision Legendre transforms (single_trans). Our final setup consisted of an entirely double-precision model with SPPT.

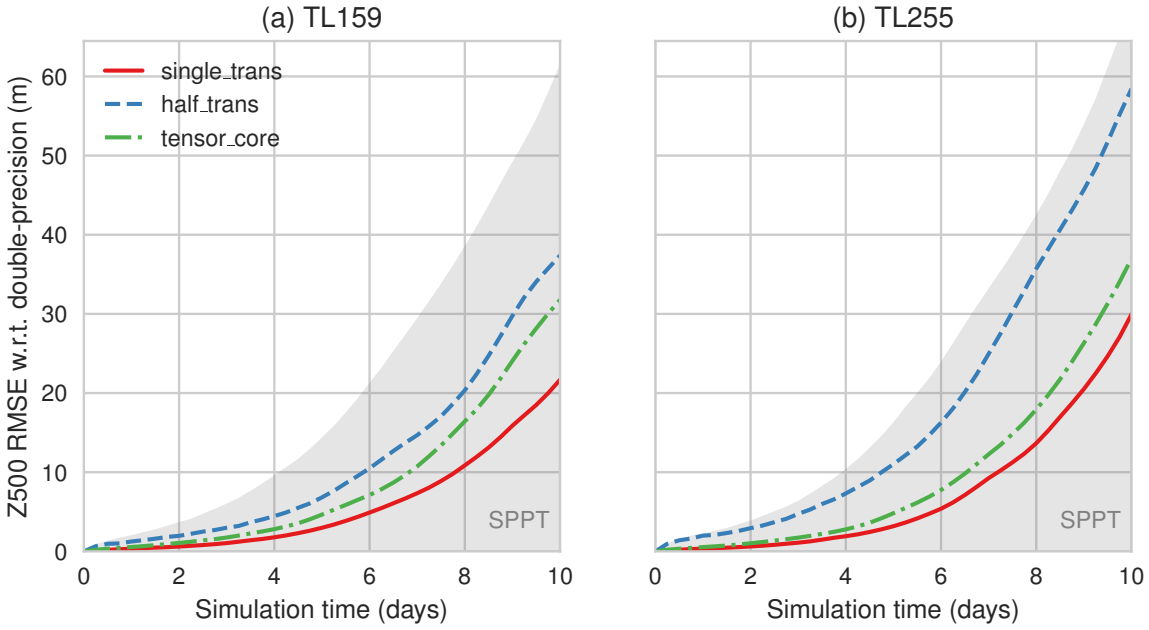


Figure 6.3: The root-mean-square error of the 500 hPa geopotential height field for various experimental setups with respect to a double-precision control experiment at (a) TL159 resolution and (b) TL255 resolution. The grey shaded area shows the error from a double-precision simulation with SPPT.

The results of this experiment for resolutions of TL159 and TL255 are shown in Figure 6.3. In this case we performed simulations for four different start dates and averaged the results. The lowest error occurred for single_trans, followed by tensor_core, half_trans and finally SPPT. The latter is displayed as a grey shaded area. For both resolutions, the error introduced by using half-precision Legendre transforms was lower than that introduced by using stochastic physics.

We also performed experiments with the same setups but at TL511 resolution and for only one date. However, we found that the error when using half-precision Legendre transforms was substantially higher even than the SPPT setup. To remedy this, it was necessary to keep the computation of several more zonal wavenumbers at double-precision, instead of just the first as in the previous half-precision experiments. Figure 6.4 shows the effect of computing the first c wavenumbers using double-precision, with c ranging from 1 (as in the previous experiments) up to 20. Here we only simulated one hour. Evidently, c is a parameter that can be tuned to reduce the error to any desired level. However, the larger

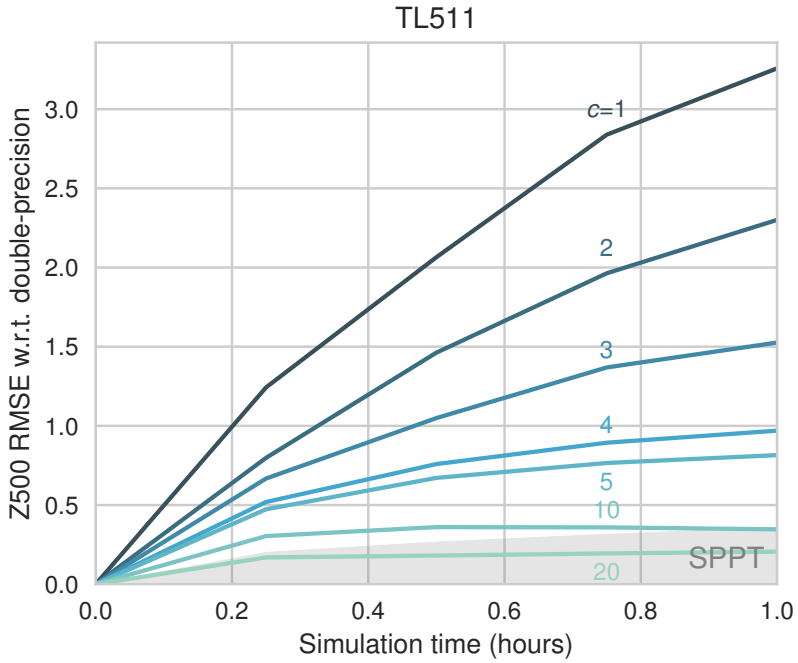


Figure 6.4: The root-mean-square error of the 500 hPa geopotential height field for different choices of the cutoff parameter c when using half-precision Legendre transforms and for a double-precision simulation with SPPT enabled.

the value of c, the fewer operations are performed at half-precision and therefore the lower the cost saving. We chose to keep $c = 10$ more zonal wavenumbers at double-precision as the error for this setup is of the same order as the error of SPPT (at least over the first hour, over which the impact of SPPT is known to be very small). When $c = 10$, only around 5% of the total number of floating-point operations in the Legendre transforms are computed at double-precision.

Figure 6.5 shows results from the 10 day TL511 simulations. The effect of performing the first 10 zonal wavenumber computations at double-precision (half_trans_10) instead of just the first (half_trans) is immediately noticeable. Note that, as of the time of writing, the resolution of the operational ensemble prediction system of ECMWF is TCo639 (our version of OpenIFS is not strictly comparable with the operational IFS as the latter uses a cubic grid whereas ours uses a linear grid). In the next section, we will evaluate the performance of the half_trans_10 and tensor_core setups in an operational-like context.

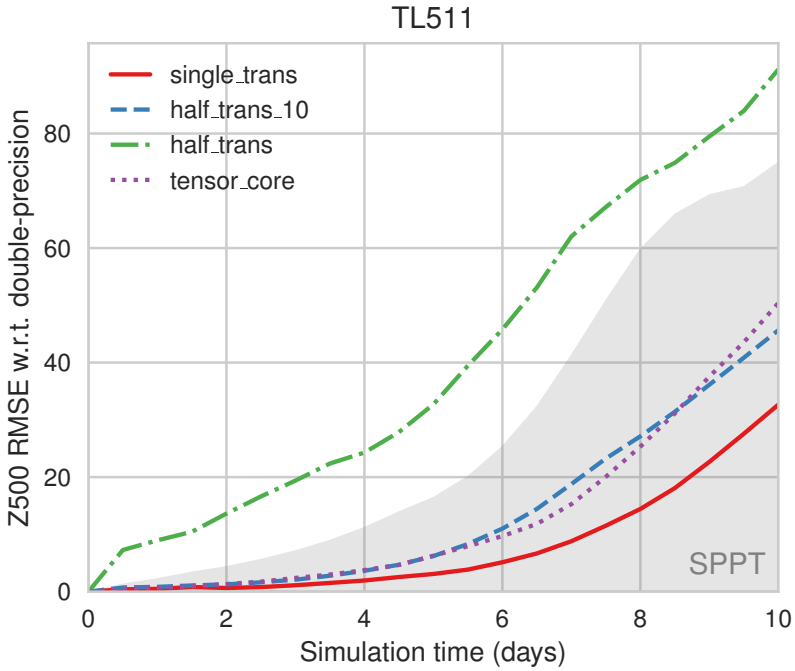


Figure 6.5: The root-mean-square error of the 500 hPa geopotential height field for various experimental setups with respect to a double-precision control experiment at TL511 resolution.

6.4.2 Probabilistic forecast skill

We performed a series of ensemble forecasts over 2017 with 12 start dates each separated by 30 days, beginning on January 1st. For the experiments in this section we used the IFS at cycle 43r3. We used 11 ensemble members for each forecast: 1 control member and 10 members with perturbed initial conditions and SPPT enabled. We performed the forecasts at TCo399 resolution which is roughly 50 km resolution at the Equator. To verify these forecasts, we used the ECMWF operational analysis as a reference. We verified the forecasts using the continuous ranked probability score (CRPS) corrected for the limited ensemble size (Leutbecher, 2018) and averaged this score over all forecasts. CRPS is a standard diagnostic to measure the quality of ensemble predictions. A low CRPS is desirable as it indicates an ensemble with a low spread and accurate mean. We considered three setups: double-precision (double), half-precision Legendre transforms for wavenumbers 10 up to 399 (half_trans_10) and Tensor Core Legendre transforms (tensor_core).

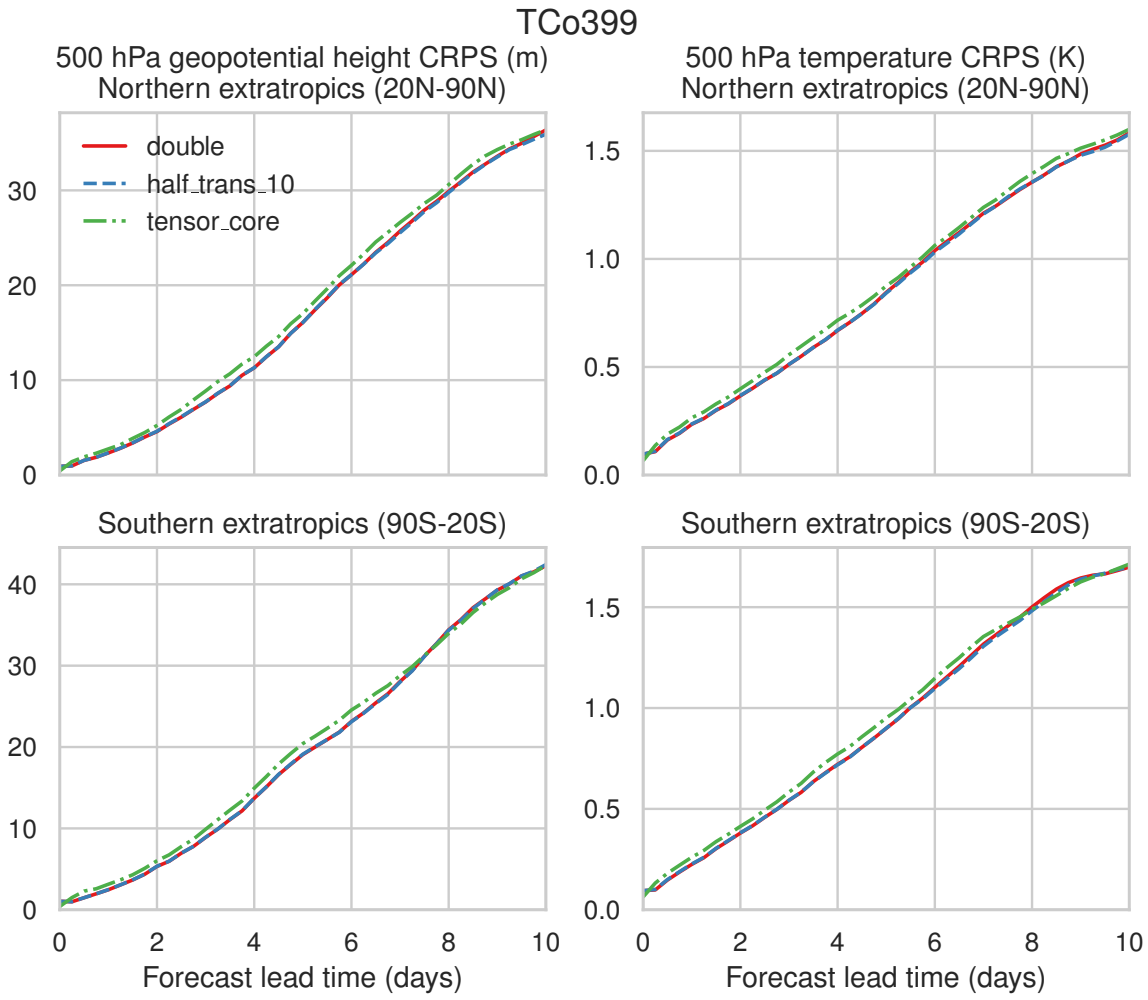


Figure 6.6: The continuous ranked probability score (CRPS) of the 500 hPa geopotential height and temperature fields for various experimental setups verified with respect to analysis at TCo399 resolution. The CRPS was averaged over 12 forecasts and there were 11 ensemble members for each forecast.

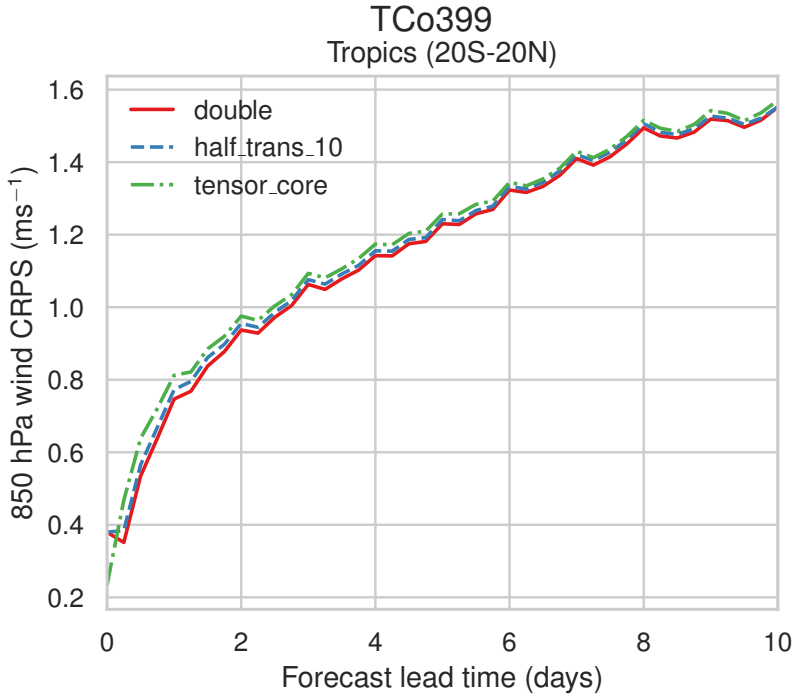


Figure 6.7: The continuous ranked probability score (CRPS) of the 850 hPa tropical wind fields (averaged from the zonal and meridional winds) for various experimental setups verified with respect to analysis at TCo399 resolution. The CRPS was averaged over 12 forecasts and there were 11 ensemble members for each forecast.

The results are illustrated in Figure 6.6 for the northern and southern extratropics, and for the 500 hPa geopotential height and temperature fields. We also show forecast scores for the tropical winds (averaging the CRPS for the zonal and meridional winds) in Figure 6.7. In all cases, the half-precision setup was competitive with the double-precision setup. The tensor_core setup, on the other hand, demonstrated a slight average decrease in the forecast skill (an increase in the CRPS). However, it is important to consider that we did not keep the first c zonal wavenumber computations at double-precision, as we did for half_trans_c. The skill of the tensor_core setup could be brought to the same level as that of double and half_trans_10 by employing this "scale-selective" precision.

As shown in Figures 6.3 and 6.5, reducing precision in the Legendre transforms has an impact on the model. However, this difference is hidden when considering the skill of probabilistic forecasts that are perturbed by both errors in the initial condition and model formulation. This demonstrates that, whenever assessing a model change, it is not sufficient

to consider only deterministic forecast verification metrics in a perfect model context. This is also consistent with the findings of Chapter 4.

6.4.3 Operational resolution deterministic forecast skill

We also performed high-resolution deterministic forecasts for the same three setups as in Figure 6.6. Again, we used the IFS at cycle 43r3. We used 6 dates for the double and tensor_core setups but only 3 dates for the half_trans setup, due to computational budget limitations. We ran these experiments at the same resolution as the operational deterministic forecast at ECMWF at the time of writing, TCo1279. This is approximately 9 km resolution at the Equator. As in Section 6.4.2 we verified with respect to analysis and therefore, unlike in Section 6.4.1, these forecasts did include initial condition and model error. The deterministic forecast tests in this section are therefore a more realistic assessment of our proposed model change. For the half_trans setup we decided to secure the first 25 zonal wavenumbers with double-precision arithmetic, instead of 10 for the TL511 and TCo399 experiments. We reasoned that the optimal ratio of wavenumbers computed at double-precision to those computed at half-precision (2:100 for TL511) is not likely to change as resolution is increased. Accordingly, we computed 25 out of 1280 wavenumbers ($c = 25$) using double-precision with half-precision for the remainder so this setup is labelled half_trans_25. However, we did not have sufficient computational resources to tune this value so we do not claim that it is optimal.

Figure 6.8 shows the results from these forecasts for the 500 hPa geopotential height and temperature fields, averaged across all available dates for each setup, in the northern and southern extratropics. Figure 6.9 shows the results for the tropical winds (again, the RMSE was averaged over the zonal and meridional winds). In all cases there were no problems with numerical stability for the tensor_core and half_trans_25 setups. In the extratropics there doesn't appear to be a significant difference between the three setups (the differences for half_trans_25 are probably due to using fewer start dates). However, there is a significant increase in the RMSE of the tropical winds for the half_trans_25 experiment, of

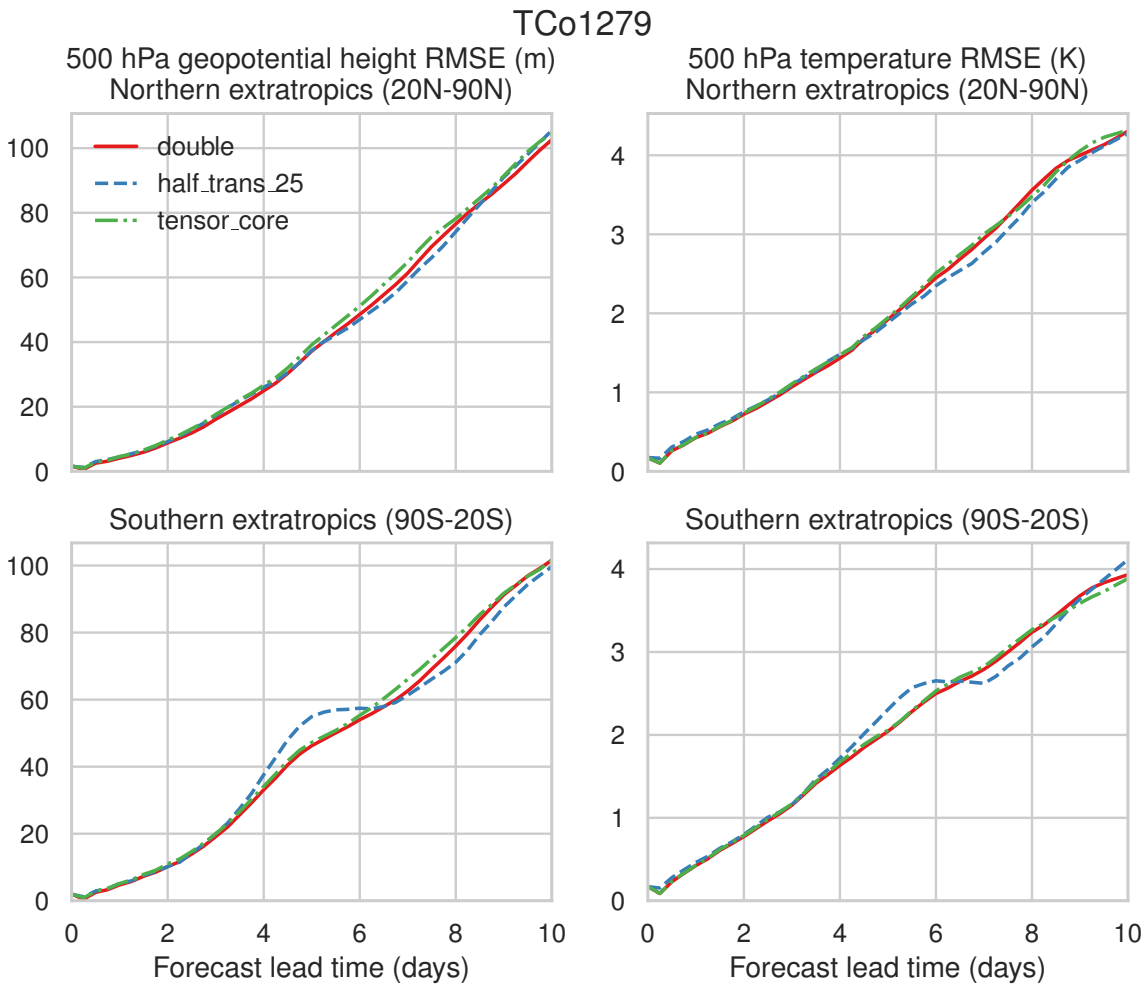


Figure 6.8: The root-mean-square error of the 500 hPa geopotential height and temperature fields for various experimental setups verified with respect to analysis at TCo1279 resolution. Error metrics are averaged across multiple dates. Six dates were used for double and tensor_core whereas three dates were used for half_trans_25.

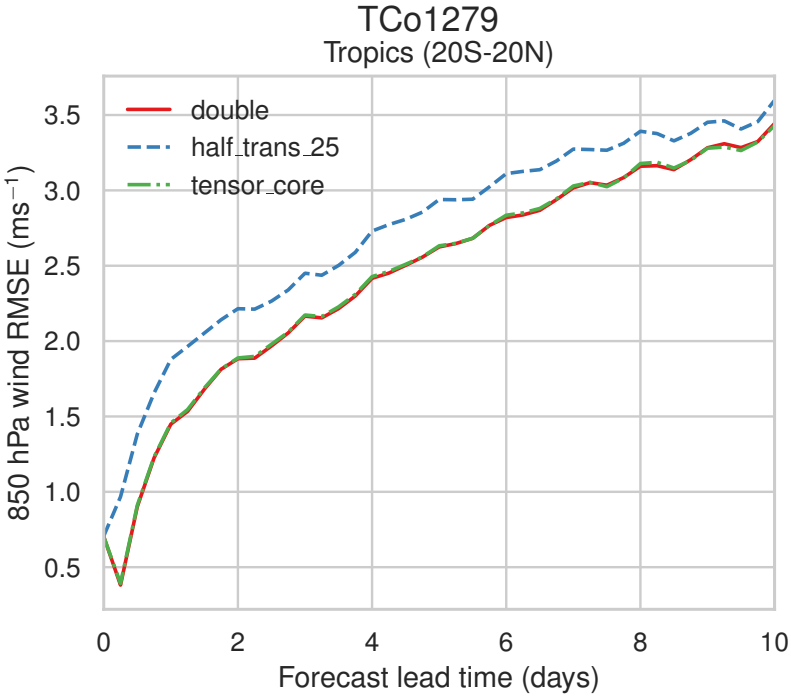


Figure 6.9: The root-mean-square error of the 850 hPa tropical wind fields (averaged from the zonal and meridional winds) for various experimental setups verified with respect to analysis at TCo1279 resolution. Error metrics are averaged across multiple dates. Six dates were used for double and tensor_core whereas three dates were used for half_trans_25.

around 0.3 ms^{-1} or 15%. This is present in all three forecast dates run for the half_trans_25 setup. We also show differences in model biases for each setup with respect to analysis in Figure 6.10. These plots show the difference between the analysis and each setup at day 5, averaged across each forecast date and across longitude, for four fields. We subtracted the value of each field from the analysis equivalent. Overall, the tensor_core setup shows a reduced or similar bias to the double-precision setup, whereas half_trans_25 shows an increased bias, particularly in the extratropics. The increase in the 850 hPa zonal wind bias just south of the Equator may explain the observed increase in forecast error in Figure 6.9. We don't rule out the use of half-precision at such a high-resolution, but we would need to run more than 3 forecast dates to truly determine its utility. In any case, the tensor_core setup performs comparably to the double-precision setup in all cases.

As a final test we performed a forecast of Hurricane Irma which caused significant damage in the Caribbean and the southeastern United States in September 2017. For computa-



Figure 6.10: The day 5 forecast bias of various experimental setups compared with analysis for 4 different fields at TCo1279 resolution. The bias was averaged across multiple dates and across longitude. Six dates were used for double and tensor_core whereas three dates were used for half_trans_25.

tional budget reasons we only compared the double and tensor_core setups. We performed this forecast at TCo1279 resolution beginning on the 5th September 2017 at 12:00 pm. Note that the hurricane is already present in the initial conditions, having begun forming 5 days earlier. We simply wish to assess the ability of the tensor_core setup to match the trajectory forecasted by the state-of-the-art double-precision model.

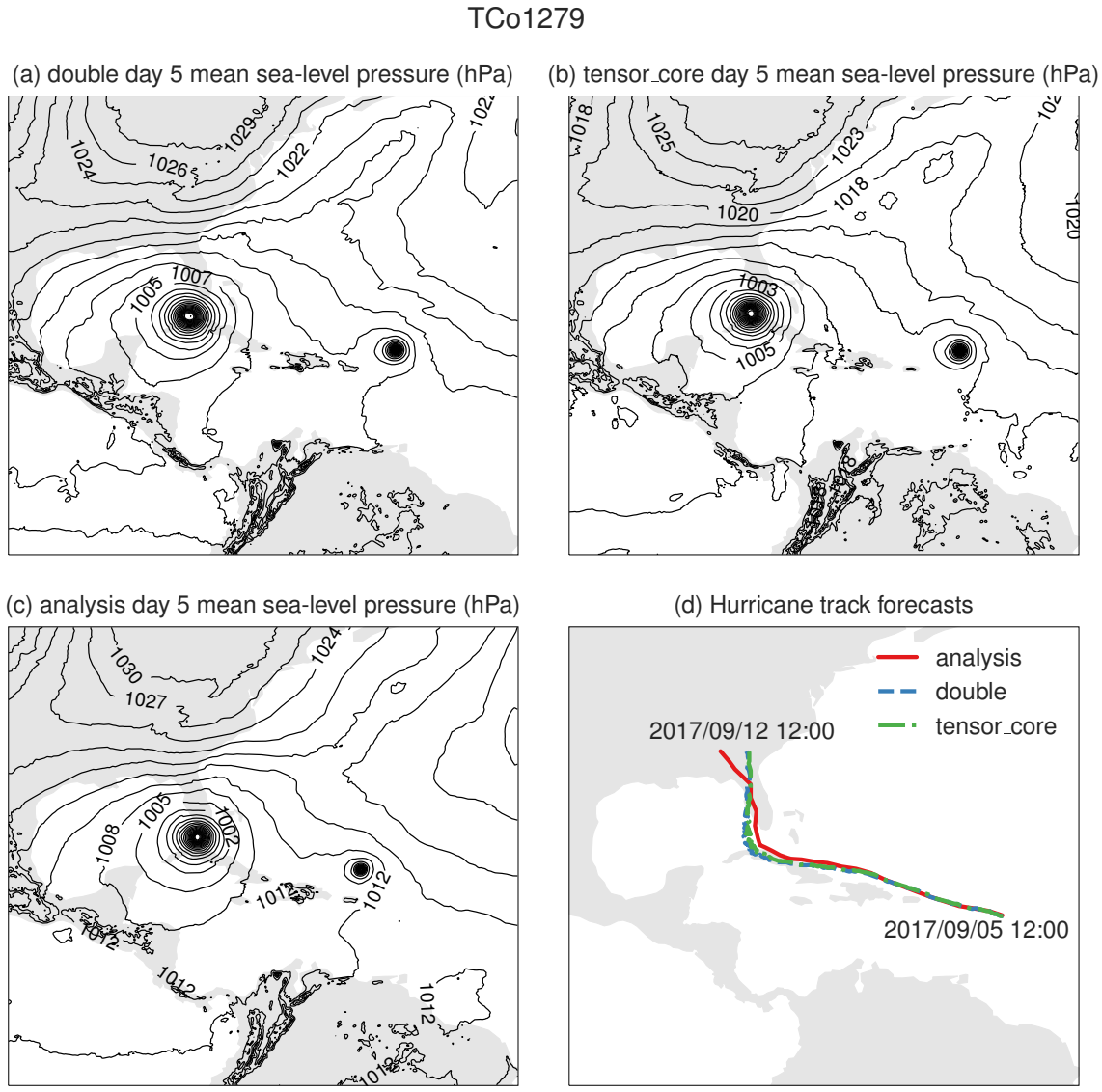


Figure 6.11: Forecasts of Hurricane Irma compared with analysis at TCo1279 resolution. Panes (a) and (b) show the mean sea-level pressure forecasts produced with the double and tensor_core setups, respectively, at 5 days lead time and pane (c) shows the corresponding analysis (actual conditions). Pane (d) compares the hurricane tracks up to day 7.

The results from this experiment are given in Figure 6.11. The mean sea-level pressure

at day 5 (12:00 pm on September 10th) is shown along with the simulated hurricane tracks for the analysis and the double and tensor_core setups. The tensor_core setup gives almost exactly the same forecast as the double setup. It is reassuring that there is minimal difference in the forecast skill of the double and tensor_core setups for specific extreme weather events as well as for large-scale spatially averaged verification metrics.

6.5 Computational cost estimates

The computational cost savings of the half_trans and tensor_core setups with respect to the double-precision reference are difficult to estimate. The exact cost saving will depend on how the target architecture is configured, including the host-GPU connection and the node-node interconnect. However, we can provide a simple cost saving estimate from a computational complexity argument, which should apply to a single-node system. We present results for the symmetric part of the inverse Legendre transform, since the complexity is almost identical for the direct and antisymmetric parts.

We first consider the half_trans setup. By computing the total number of floating-point operations (FLOPs) performed at half- and double-precision, weighting the half-precision FLOPs appropriately, then dividing by the total number of FLOPs, we can obtain the cost of half_trans with respect to the fully double-precision setup. The number of FLOPs performed for a particular wavenumber m, assuming the standard "schoolbook" matrix-matrix multiplication algorithm, is given by

$$N_{\text{FLOPs}}(m) \sim \mathcal{O}\left(N_{\text{fields}}(m) \times N_{\text{total wavenumbers}}(m) \times N_{\text{latitudes}}(m)\right), \tag{6.6}$$

where $N_{\text{fields}}(m)$, $N_{\text{total wavenumbers}}(m)$ and $N_{\text{latitudes}}(m)$ are the number of fields, total wavenumbers and latitudes that the transform is evaluated over for this m, respectively. The use of fused multiply-add operations, in which $a + b \times c$ may be performed in a single step, may change N_{FLOPs} by a constant factor but will not change the overall complexity. The number of fields is the same for all m and so can be factored out, apart from the factor of 2 that

it introduces for all $m \neq 0$ wavenumbers (which have real *and* imaginary components, unlike the $m = 0$ wavenumber). The number of total wavenumbers decreases linearly as we increase m according to the triangular truncation paradigm. Finally, the number of latitudes also decreases as we increase m, as the number of longitudes decreases as we move from equator to pole for the cubic octahedral grid. The high m waves therefore cannot be represented at these high latitudes.

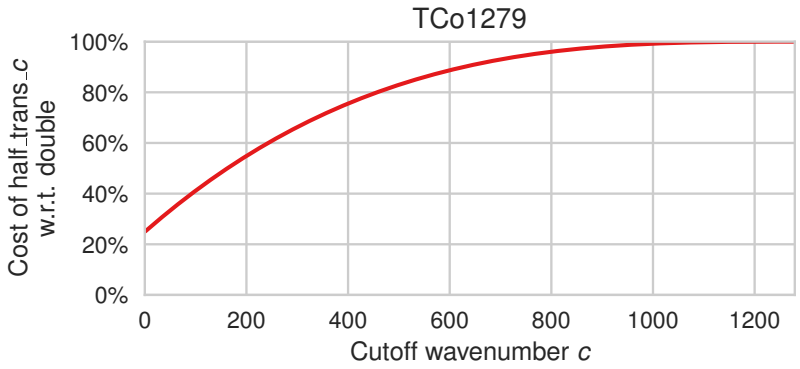


Figure 6.12: The cost of the half_trans setup with respect to the double-precision reference, as a function of the cutoff wavenumber c. Transforms use double-precision (half-precision) below (above) the cutoff wavenumber. Half-precision floating-point operations (FLOPs) are assumed to be four times cheaper than double-precision FLOPs.

To compute the cost of the half_trans setup with respect to the double-precision reference, for a particular cutoff wavenumber c, we use the following formula:

$$C_{\text{half_trans}}(c) = \frac{\sum_{m=0}^{c} N_{\text{FLOPs}}(m) + \sum_{m=c}^{N} S_{\text{half}} N_{\text{FLOPs}}(m)}{\sum_{m=0}^{N} N_{\text{FLOPs}}(m)}, \tag{6.7}$$

where S_{half} is the cost of a half-precision FLOP relative to a double-precision FLOP. We show this result as a function of the cutoff wavenumber c in Figure 6.12 for simulations with TCo1279 resolution. In this result, we assumed that a half-precision FLOP is four times cheaper than a double-precision FLOP, i.e. $S_{\text{half}} = 0.25$. If the cutoff wavenumber is N then all wavenumbers are computed at double-precision and there is no cost saving. If the cutoff wavenumber is 0 then all wavenumbers are computed at half-precision and the cost is 25% that of the double-precision reference. However, even when the cutoff wavenumber is 25 so that the first 25 wavenumbers are computed with double-precision (as in the previous

experiments) the cost only increases to 29%.

A similar argument can be applied to the tensor_core setup, although in this case the cutoff wavenumber is just 1. Taking the quoted speed-up factor of 16 times from NVIDIA, the tensor_core setup should cost around 7% that of the double-precision setup. We stress again that this result will likely change for a multi-node system, and that there may be issues with load balancing if the computation of different zonal wavenumbers is spread across multiple nodes.

6.6 Conclusion

Here we have assessed the impact of reduced-precision floating-point arithmetic within an expensive portion of an atmospheric model by emulating hardware designed for deep-learning. Specifically, we considered the use of half-precision matrix-matrix multiplication and the NVIDIA Tensor Core for accelerating the Legendre transforms of the operational global weather forecasting model of ECMWF. Firstly, we have found that half-precision arithmetic can be used without causing model instabilities or crashes by using a simple rescaling technique, as long as the most sensitive 5% of the calculations are kept at double-precision. Secondly we have shown that the use of half-precision or the Tensor Core does perturb the model. However, the perturbation is not significant when compared with other error sources that are present in real weather forecasts. We have shown that, in a probabilistic forecasting context with initial condition and model error, using half-precision or the Tensor Core gives a forecast competitive or equivalent to using double-precision. We have also demonstrated this result in a high-resolution, deterministic forecasting context, though we did find a poorer forecast score for the tropical winds when using half-precision. Finally, we have presented a simple cost estimate for the half-precision and Tensor Core Legendre transforms, though the exact numbers depend on the as-yet-unknown target architecture.

By 2025, ECMWF plan to increase the horizontal resolution of their ensemble forecasting system from TCo639 (roughly 18 km) to TCo1999 (roughly 5 km). To what extent can

the positive results shown here be extended to higher resolutions? As horizontal resolution increases, the summing dimension of the matrix-matrix multiplication in both the direct and inverse Legendre transforms also increases. Therefore, the accumulated rounding error in this operation should grow also, more so when using half-precision. However, it is extremely difficult to say *a priori* how this will impact a forecast product like the 10 day probabilistic forecast skill of 500 hPa geopotential height, given how errors feed back and interact in a nonlinear fashion within the model. Nevertheless, we are confident that a scale-selective approach, whereby larger scales (in our case represented by lower zonal wavenumbers) are kept at a higher precision than smaller scales, will enable the use of low-precision Legendre transforms at least up to TCo1999 and potentially beyond.

Chapter 7

Conclusion

We conclude with a summary of key findings and a discussion of future developments in Earth-System modelling.

7.1 Summary of key findings

Precision must be matched to system uncertainty

In the field of numerical modelling it is common to test new modelling approaches in a perfect model setting. We use a particular model configuration as a surrogate for the truth, for example, an expensive high-resolution model. We then evaluate the performance of our proposed model change against this perfectly-known "truth". We hope that our proposed model change gives a more accurate simulation of the truth than the reference model. In the case of a precision reduction, however, we are deliberately reducing the fidelity of the basic arithmetic within the model. If we use the system described above to test a proposed precision reduction, we will always conclude that the highest precision possible should be used. In Chapter 4 we demonstrated this explicitly for a perfect model data assimilation setup. We found that the reduced-precision model showed a significantly increased analysis error compared with the double-precision reference model. We also demonstrated this in Chapter 6, though not in a data assimilation context. We found that using reduced-precision arithmetic for the Legendre transforms always gave a non-zero error when the model was evaluated with respect to the double-precision reference model.

However, in both cases introducing model error was sufficient to hide the errors due to reducing precision. This indicates that the choice of which precision to use is not simply a computational matter, but also a physical one. As long as one is careful not to introduce a systematic error by reducing precision, the precision that we use should match the degree of uncertainty in the modelling formulation.

Different data assimilation systems respond differently when reducing precision

In this study we considered ensemble and variational data assimilation schemes — we studied the ensemble Kalman filter (EnKF) in Chapters 3 and 4 and 4D-Var in Chapter 5. Although both the EnKF and 4D-Var have basically the same external interface — both ingest observations and a background and from these produce an optimal analysis — they respond to a reduction in precision in very different ways. We attributed this to the different ways in which numerical models are used in both schemes. For the EnKF, we use only the nonlinear model operator to integrate forward an ensemble of analyses, giving the ensemble of backgrounds valid at the next observation time. The model is therefore used in a probabilistic context essentially as it would be used for an ensemble forecast. This probabilistic context, with a multitude of errors arising from, for example, imperfect observations and an imperfect model formulation, provides a margin of error within which we can lower precision without affecting the analysis quality, as argued above. With 4D-Var, however, the success of minimising the cost function when there are substantial rounding errors in the tangent-linear and adjoint models depends on how well the minimisation scheme can tolerate these errors. For example, we demonstrated in Chapter 5 that, unlike the generalised minimal residual method, the conjugate gradient minimiser is very sensitive to precision. Nevertheless, we may be able to use reduced-precision arithmetic in 4D-Var if re-orthogonalisation is used.

The small range of half-precision floating-point numbers limits their use within numerical modelling

Though we successfully made use of half-precision arithmetic within the toy model of Chapter 3, we were unable to use it throughout the atmospheric model of Chapter 4. This was due to the severely limited range of half-precision floating-point numbers, specifically the relatively small maximum of 65504. Though we were able to use half-precision floats to accelerate the Legendre transforms in Chapter 6, the feasibility of using such a low level of precision throughout a model remains an open question.

In order to use half-precision floats we would have to rescale model variables to fit within the half-precision range. This would not simply be a matter of rescaling the inputs and un-rescaling the output, as in Chapter 6, as full atmospheric model operators are non-linear. Instead, each line of code within the model would have to be carefully considered and made "overflow-proof". As an example, consider the Stefan-Boltzmann law which is used to compute blackbody emittance in the stratosphere for longwave radiation schemes, among other things. The law is given by $j = \sigma T^4$, where j is the emittance, σ is the Stefan-Boltzmann constant and T is temperature. If T is measured in Kelvin, then a typical value of $T = 300\text{K}$ when raised to the fourth power would lead instantly to an overflow. This is in spite of the fact that the computed value of j, around 460, is representable at half-precision. However, by reformulating the equation as $j = \left(\sigma^{\frac{1}{4}} T\right)^4$, overflows in intermediate calculations can be avoided. Using half-precision throughout a model will require careful consideration of each line of code, and not all cases may be as simple to fix as the one presented above. Nevertheless, the computational speed-up from using half-precision arithmetic motivates further investigation of this matter.

7.2 Avenues of future research in Earth-System modelling

We outline below a number of research avenues in the field of Earth-System modelling which we expect to be fruitful in the coming years. We discuss the use of number formats

not included in the IEEE 754 standard, the use of machine learning both within numerical weather prediction and data assimilation, and another possible use for low-precision floating-point numbers — to reduce communication costs within models.

7.2.1 Alternatives to IEEE 754 floating-point numbers

In this thesis we have focused mainly on the three IEEE 754 standardised floating-point formats, namely double, single and half-precision floats. However, these are only three possible ways to map bit strings to the real numbers. Several alternative number formats have been proposed, with varying levels of hardware support. These competing number formats attempt to give a more accurate answer than floats for a fixed number of bits or for a fixed computational cost. For example, the bfloat16 number format has been suggested as an appropriate low-precision number format to use for machine learning applications. A bfloat16 number has the same exponent as a single-precision float (8 bits), but only 7 significand bits (and 1 sign bit, totalling 16 bits). It therefore has the same range as single-precision, but a rounding error even higher than half-precision (around 8 times higher, in fact). Another alternative is "posits", which are like floats but have in addition to the sign, significand and exponent an additional bit field known as the regime. The regime and exponent fields can vary in size depending on the value being stored, allowing more precision closer to one than at the extreme ends of the representable range. They have a number of other advantages over floats of the same width, notably a wider range and more gentler overflow behaviour.

In order to compare these number formats, we introduce the notion of *decimal precision* (Klöwer et al., 2019). For a given real number x, we compare with the closest representable finite precision number $\widetilde{x}$ (e.g. closest float, posit etc.) by the formula

$$\text{decimal precision} = -\log_{10} |\log_{10} \left(\frac{\widetilde{x}}{x} \right)|. \tag{7.1}$$

This number should be as high as possible. To allow visualisation of decimal precision

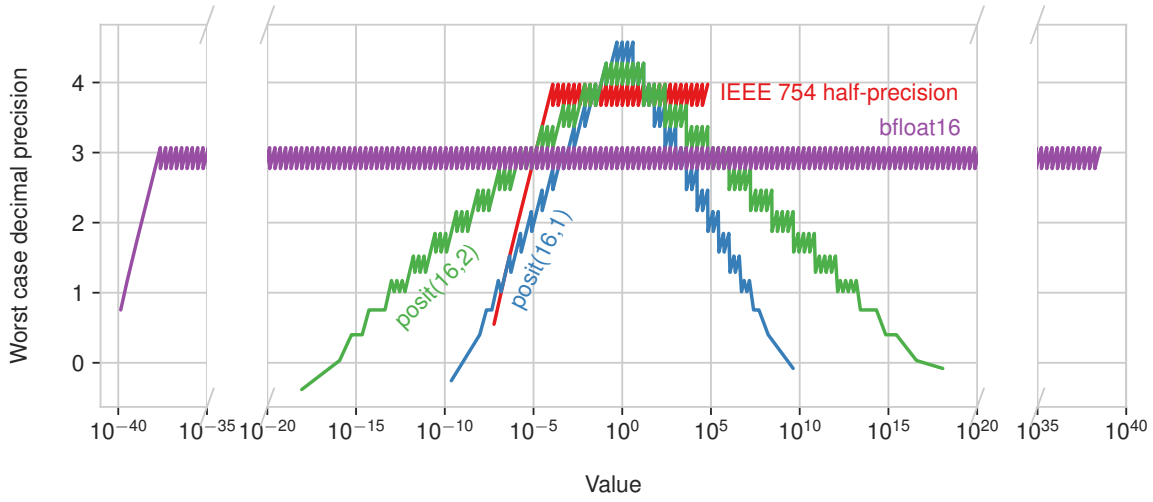


Figure 7.1: The worst case decimal precision for various number formats. Figure adapted from Klöwer et al. (2019) with permission.

across a wide range of values, we consider only the *worst case* decimal precision. For each exponent bit string, we compute the decimal precision of each possible significand bit string and store the lowest. We compare various number formats using this metric in Figure 7.1. For half-precision numbers, the limited range with a maximum of 65504 is immediately noticeable. The gradual decrease in accuracy for the subnormal numbers is also clearly visible. The advantage of the relatively wide exponent for bfloat16 numbers is indicated by the wide range of representable numbers — the largest is around 10^{38}. However, by trading 3 significand bits for 3 exponent bits we have paid the price of a reduction in decimal accuracy. Nevertheless, if bfloat16 becomes a standard number format supported natively on future hardware then possible uses for it within numerical modelling should be investigated.

Two 16 bit wide posit types are also displayed in Figure 7.1, with maximum exponent widths of 1 (posit(16,1)) and 2 bits (posit(16,2)). Unlike floats which have a fixed relative precision throughout the normal range, posits have a precision that depends on the value being stored. Because they have more precision close to one, this motivates a rescaling of model variables so that as many calculations as possible occur around this value. Additionally, both posit types have a wider range than the equivalent 16 bit floating-point type. The posit(16,1) type has a maximum representable value of around 3×10^6 whereas the

posit(16,2) type has a maximum representable value of around 7×10^{16}. These are both still lower than the value for a single-precision float — around 10^{38} — but both provide significantly more "breathing space" than a half-precision float. Furthermore, even if an overflow does occur, it is simply rounded down to the largest representable value instead of being rounded up to infinity, which is arguably a more preferable treatment. We therefore see great potential for the use of posits within numerical modelling.

7.2.2 Machine learning and data assimilation

Another avenue for future research concerns the hybridisation of traditional Earth-System modelling techniques and those from the field of machine learning. Neural nets can, in principle, emulate any mathematical function and so they demonstrate great potential for replacing or supplementing certain model components. The neural nets could be trained from observations or model output, or even both. Machine learning in the atmospheric sciences has a history of several decades, with early studies focusing on, for example, radiation schemes (Chevallier et al., 1998; Krasnopolsky et al., 2009). The recent development of easy-to-use machine learning software tools such as TensorFlow and the influx of investment from large software firms has catalysed research into machine learning in many areas of science, not the least Earth-System simulation science. Current research focuses on isolating those components of models where neural nets could bring the most benefit, and many approaches have been proposed. Dueben and Bauer (2018) attempted a wholesale replacement of a numerical model by a neural net, considering neural nets trained on both toy model output and reanalysis data. Their neural nets only provided tendency estimates, however, so a traditional time-stepping scheme was employed so that forecasts could be made from the neural nets. Watson (2019) proposed a more conservative approach of training a neural net to provide an error correction term to an existing model. This approach took advantage of existing knowledge about the evolution of the resolved flow (i.e. the equations of fluid dynamics), arguably simplifying the task of the neural net. Bolton and Zanna (2019) developed a similar system in order to correct the computed tendencies in

an ocean model to account for unresolved eddies. Brenowitz and Bretherton (2018) also employed an approach similar to Watson (2019) in emulating the effect of unresolved diabatic processes on the resolved flow of an aqua planet simulation. They demonstrated a key point, that for numerical stability the cost function of the neural net must be minimised over several time steps, not just a single time step.

To what extent can machine learning techniques augment data assimilation? In operational data assimilation there are many tuneable parameters that have a significant impact on the accuracy of the analysis. An example is the covariance inflation parameter of the ensemble Kalman filter which must always be employed for computationally affordable ensemble sizes. One could imagine training a neural net to produce ensemble perturbations (i.e. additive inflation) that account for sampling errors due to the small operational ensemble size, or even biases in the model. This could be trained offline with respect to a system with a very large ensemble size, which does not need inflation. De La Chevrotière and Harlim (2017) investigated a similar procedure for improving the off-diagonal terms of the sampled background covariance matrix. Large ensemble experiments have already been conducted for the SPEEDY model (Miyoshi et al., 2014), in which around 10,000 ensemble members were sufficient to render covariance manipulation techniques unnecessary, and this may prove a useful testbed.

As a preliminary investigation into this matter, we present here results from a data assimilation experiment in which we replaced the numerical model with one derived from a neural net. We used a simplified version of the setup in Chapter 3, in which we performed data assimilation with an ensemble square-root filter (EnSRF) and the Lorenz '96 system. We trained a neural net to reproduce the complete tendencies of the X variables from a two-level "high-resolution truth" simulation. We then gave these tendencies to a third-order Adams-Bashforth time-stepping scheme (a technique borrowed from Dueben and Bauer (2018)) which allowed us to perform forecasts of the X variables. Figure 7.2 compares the forecast performance of this system with a traditional numerical model that only simulates the X variables, both verified with respect to the two-level truth simulation. The neural net-

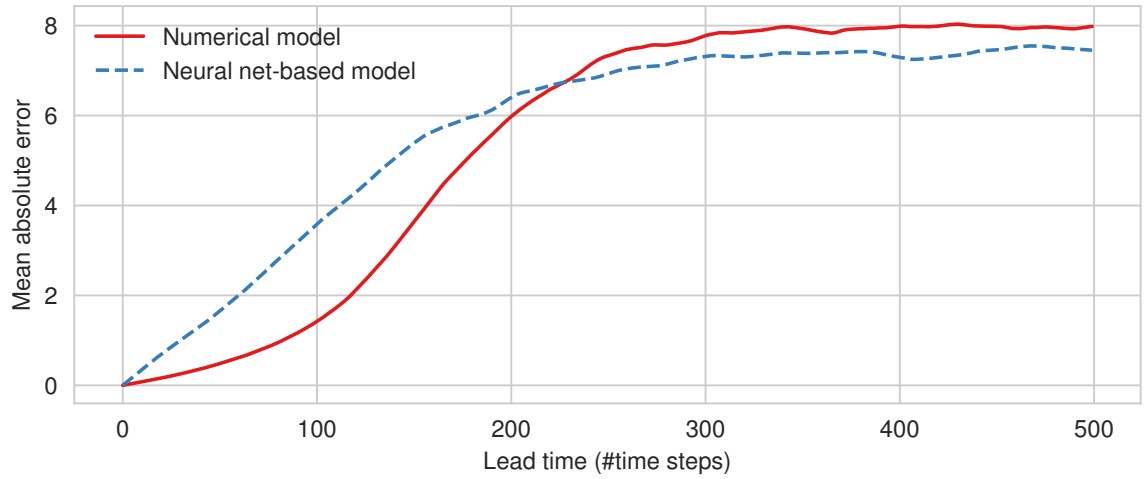


Figure 7.2: The forecast error of two forecasting systems, one based on a traditional numerical modelling approach and one based on a neural net, with respect to a high-resolution truth.

based model is clearly competitive with the numerical model, though the error is somewhat higher. However, we are confident that we could match the performance of the numerical model by better tuning the hyperparameters of the neural net (such as the number of hidden layers) and by employing more sophisticated activation functions.

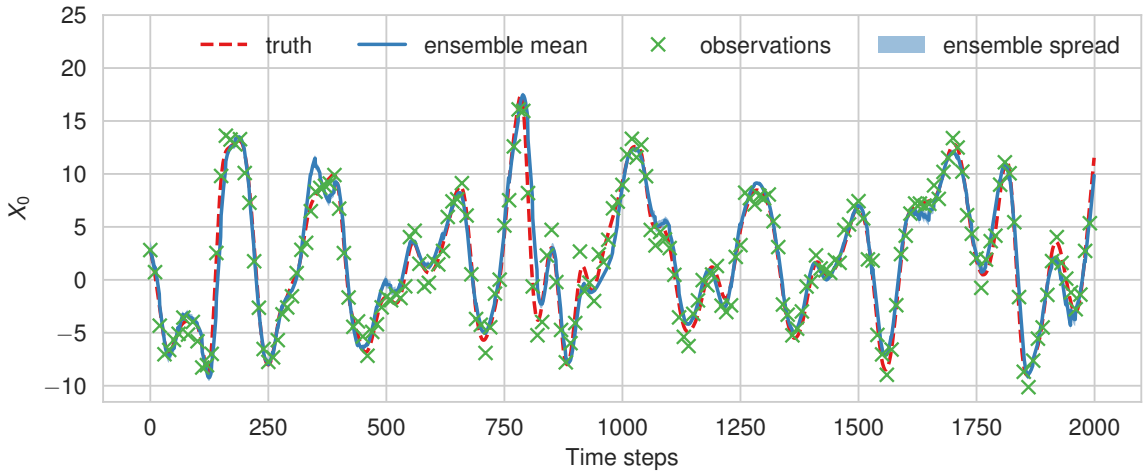


Figure 7.3: An assimilation cycle using the neural net-based model to produce the background forecasts.

We then used the neural net-based model to produce the background forecasts in an EnSRF data assimilation system. We assimilated observations derived from the X variables of the nature run every 10 timesteps using an ensemble of 200 members (we used a large

ensemble so that we didn't require covariance localisation). Figure 7.3 shows the results from this experiment, focusing on just one of the X variables. Clearly the neural net-based model is able to accurately track the evolution of the truth even though it has no a priori knowledge about the dynamics (that is, no knowledge other than what it obtained from the training process).

This technique is promising, though as pointed out by Watson (2019), it will probably be more fruitful to combine neural nets with existing models rather than trying to replace the models entirely. One area in which neural nets may prove especially useful to data assimilation is in emulating physical parametrizations. One could, for example, train a neural net to replicate the results from an existing cumulus cloud parametrization scheme. Then, an automatic differentiation of the neural net could be obtained trivially, because of the highly structured and relatively simple nature of the code underlying the net, comprised of differentiable activation functions. This would allow the construction of the tangent-linear and adjoint models with arguably much less effort.

7.2.3 Reduced-precision communication

Finally, we present another possible use for half-precision similar to that presented in Chapter 6. As mentioned in Chapter 1, communication between computing nodes in a cluster can dominate the total runtime for a weather or climate model. In this case, reduced-precision floating-point numbers could also be used as a kind of data compression to reduce the cost of communication, even if the actual computations are performed at a higher precision.

The idea is illustrated in Figure 7.4. Four double-precision numbers are first rescaled to fit inside the half-precision range, using the same procedure as in Chapter 6. This is necessary as double-precision and half-precision floats have different exponent biases, so we cannot simply truncate the exponents as we can for the significand. Then, the rescaled numbers are cast into half-precision containers. Then they are communicated as an 8 byte message (four half-precision numbers now occupying the same space as just one double-precision number). Finally, on the receiving end of the message the numbers are cast back

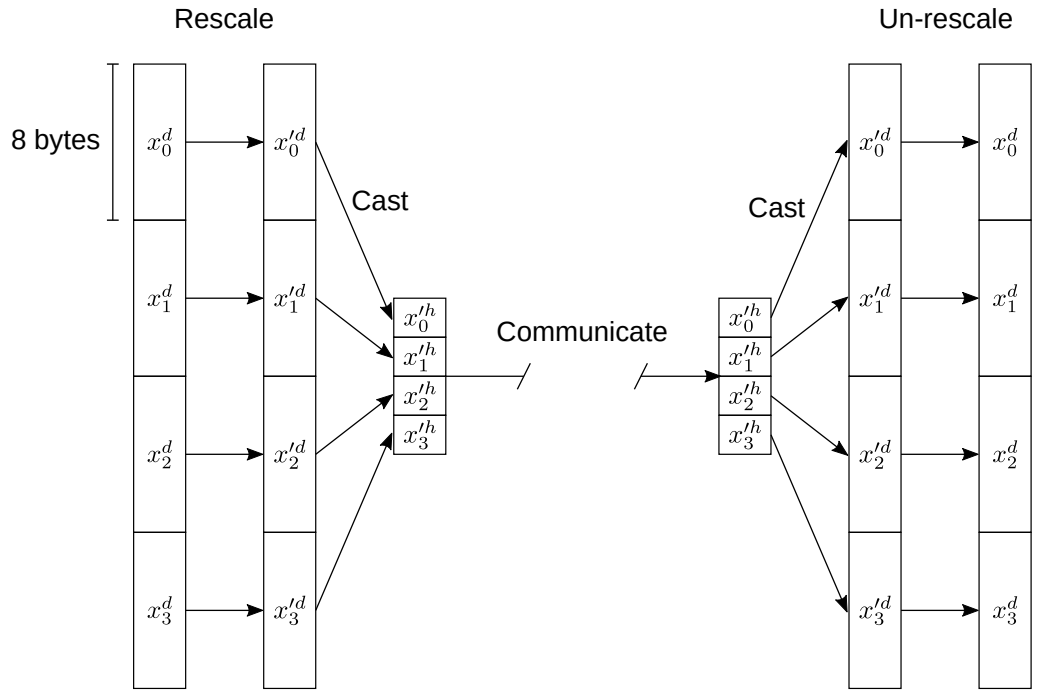


Figure 7.4: An illustration of reduced-precision communication, packing double-precision variables into half-precision containers to quarter the cost of communication (assuming that the cost of rescaling and casting is neglible).

to double-precision and unrescaled completing the communication.

The success of this technique will depend on the relative costs of the message preparation (rescaling and casting) and communication. If an efficient operation for performing these can be implemented in hardware then it may prove to be worthwhile. Additionally, we must contend with the rounding errors incurred from the casting procedure. This is a similar consideration to what we encountered in Chapter 6. As always, this must be compared with inherent unavoidable sources of uncertainty, like the uncertainty due to an imperfect model formulation, for example. If these rounding errors can be tolerated then reduced-precision communication could provide a significant acceleration of a high-resolution Earth-System simulation.

Reducing the precision of the calculations in a model may at first seem counter-intuitive.

Lowering the number of bits used to store and compute variables increases the floating-point spacing and therefore also the rounding error which will naturally degrade the result of any arithmetic operation. One may be tempted to conclude that the quality of a simulation will also be degraded as, ultimately, all models are composed of simple arithmetic operations. However, the modelling of the atmosphere is inherently uncertain and hence the accuracy of any simulation is constrained not only by the precision with which arithmetic operations are carried out, but also initial condition uncertainty and model uncertainty (not to mention scenario uncertainty for climate modelling). These latter sources of uncertainty provide a margin of error within which to "fit" the rounding error incurred by lowering precision. In this thesis we have demonstrated this result with a wide variety of models used for both data assimilation and numerical weather prediction. We believe that reduced-precision arithmetic will play an increasingly important role when preparing Earth-System models for use on future supercomputers with highly parallel and heterogeneous architectures.

Bibliography

Javier Amezcua, Eugenia Kalnay, and Paul D Williams. The Effects of the RAW Filter on the Climatology and Forecast Skill of the SPEEDY Model. *Monthly Weather Review*, 139 (2):608–619, 2011. doi: 10.1175/2010MWR3530.1.

Jeffrey L. Anderson and Stephen L. Anderson. A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts. *Monthly Weather Review*, 127(12):2741–2758, dec 1999. doi: 10.1175/1520-0493(1999)127<2741:AMCIOT>2.0.CO;2.

Jeffrey L. Anderson, Tim Hoar, Kevin Raeder, Hui Liu, Nancy Collins, Ryan Torn, and Avelino Avellano. The data assimilation research testbed a community facility. *Bulletin of the American Meteorological Society*, 90(9):1283–1296, 2009. doi: 10.1175/2009BAMS2618.1.

Akio Arakawa. The Cumulus Parameterization Problem: Past, Present, and Future. *Journal of Climate*, 17(13):2493–2525, jul 2004. doi: 10.1175/1520-0442(2004)017<2493:RATCPP>2.0.CO;2.

H. M. Arnold, I. M. Moroz, and T. N. Palmer. Stochastic parametrizations and model uncertainty in the Lorenz '96 system. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1991), apr 2013. doi: 10.1098/rsta.2011.0479.

Seung-Jong Baek, Istvan Szunyogh, Brian R. Hunt, and Edward Ott. Correcting for Surface

Pressure Background Bias in Ensemble-Based Analyses. *Monthly Weather Review*, 137 (7):2349–2364, 2009. doi: 10.1175/2008MWR2787.1.

R N Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 29(January):1–29, 2016. doi: 10.1002/QJ.2982.

Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015. doi: 10.1038/nature14956.

Páll Bergthórsson and Bo R Döös. Numerical Weather Map Analysis. *Tellus*, 7(3):329–340, 1955. doi: 10.1111/j.2153-3490.1955.tb01170.x.

Craig H. Bishop, Brian J Etherton, and Sharanya J Majumdar. Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects. *Monthly Weather Review*, 129(3):420–436, 2001. doi: 10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2.

Vilhelm Bjerknes. Das Problem der Wettervorhersage, betrachtet vom Standpunkte der Mechanik und der Physik. *Meteorologische Zeitschrift*, 21:1–7, 1904.

Thomas Bolton and Laure Zanna. Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization. *Journal of Advances in Modeling Earth Systems*, 11(1): 376–399, 2019. doi: 10.1029/2018MS001472.

Massimo Bonavita, Mats Hamrud, and Lars Isaksen. EnKF and Hybrid Gain Ensemble Data Assimilation. Part II: EnKF and Hybrid Gain Results. *Monthly Weather Review*, 143(12): 4865–4882, 2015. doi: 10.1175/MWR-D-15-0071.1.

Massimo Bonavita, Elias Hólm, Lars Isaksen, and Mike Fisher. The evolution of the ECMWF hybrid data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 142 (694):287–303, jan 2016. doi: 10.1002/qj.2652.

Massimo Bonavita, Peter Lean, and Elias Holm. Nonlinear effects in 4D-Var. *Nonlinear Processes in Geophysics*, 25(3):713–729, 2018. doi: 10.5194/npg-25-713-2018.

Neill E. Bowler, Alberto Arribas, Sarah E. Beare, Kenneth R. Mylne, and Glenn J. Shutts. The local ETKF and SKEB: Upgrades to the MOGREPS short-range ensemble prediction system. *Quarterly Journal of the Royal Meteorological Society*, 135(640):767–776, apr 2009. doi: 10.1002/qj.394.

N. D. Brenowitz and C. S. Bretherton. Prognostic Validation of a Neural Network Unified Physics Parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018. doi: 10.1029/2018GL078510.

Mark Buehner, P. L. Houtekamer, Cecilien Charette, Herschel L. Mitchell, and Bin He. Intercomparison of Variational Data Assimilation and the Ensemble Kalman Filter for Global Deterministic NWP. Part I: Description and Single-Observation Experiments. *Monthly Weather Review*, 138(5):1550–1566, may 2010. doi: 10.1175/2009MWR3157.1.

Gerrit Burgers, Peter Jan Van Leeuwen, and Geir Evensen. Analysis Scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, pages 1719–1724, 1998.

Matthew Chantry, Tobias Thornes, Tim Palmer, and Peter Düben. Scale-Selective Precision for Weather and Climate Forecasting. *Monthly Weather Review*, 147(2):645–655, feb 2019. doi: 10.1175/MWR-D-18-0308.1.

F. Chevallier, F. Chéruy, N. A. Scott, and A. Chédin. A Neural Network Approach for a Fast and Accurate Computation of a Longwave Radiative Budget. *Journal of Applied Meteorology*, 37(11):1385–1397, nov 1998. doi: 10.1175/1520-0450(1998)037<1385:ANNAFA>2.0.CO;2.

P Courtier, J.-N. Thépaut, and A Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quart. J. Roy. Meteor. Soc.*, 120(519):1367–1387, 1994. doi: 10.1002/qj.49712051912.

Roger Daley. *Atmospheric Data Analysis*. Cambridge University Press, 1991. ISBN 0521458250. doi: 10.4267/2042/51948.

Andrew Dawson and Peter D. Düben. Rpe v5: An emulator for reduced floating-point precision in large numerical simulations. *Geoscientific Model Development*, 10(6):2221–2230, 2017. doi: 10.5194/gmd-10-2221-2017.

Michèle De La Chevrotière and John Harlim. A Data-Driven Method for Improving the Correlation Estimation in Serial Ensemble Kalman Filters. *Monthly Weather Review*, 145 (3):985–1001, 2017. doi: 10.1175/MWR-D-16-0109.1.

D. P. Dee. Bias and data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131(November):3323–3343, 2005. doi: 10.1256/qj.05.137.

Peter D. Düben, Hugh McNamara, and T. N. Palmer. The use of imprecise processing to improve accuracy in weather & climate prediction. *Journal of Computational Physics*, 271:2–18, 2014. doi: 10.1016/j.jcp.2013.10.042.

Peter D. Düben, Francis P Russell, Xinyu Niu, Wayne Luk, and T. N. Palmer. On the use of programmable hardware and reduced numerical precision in earth-system modeling. *Journal of Advances in Modeling Earth Systems*, 7(3):1393–1408, sep 2015. doi: 10.1002/2015MS000494.

Peter D. Düben, Aneesh Subramanian, Andrew Dawson, and T. N. Palmer. A study of reduced numerical precision to make superparameterization more competitive using a hardware emulator in the OpenIFS model. *Journal of Advances in Modeling Earth Systems*, 9(1):566–584, mar 2017. doi: 10.1002/2016MS000862.

Peter D. Düben, Nils P. Wedi, and Sami Saarinen. Global simulations of the atmosphere at 1.45 km resolution with the Integrated Forecast System. *J. Meteor. Soc. Japan*, 2019 (submitted).

Peter D Dueben and Peter Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Model Dev*, 11:3999–4009, 2018. doi: 10.5194/gmd-2018-148.

ECMWF. Strategy 2016 - 2025, 2016.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5): 10143, 1994. doi: 10.1029/94JC00572.

Geir Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly Weather Review*, 125(6):1342–1354, 1997. doi: Doi10.1175/1520-0493(1997)125<1342: Adafsn>2.0.Co;2.

Geir Evensen and Peter Jan van Leeuwen. Assimilation of Geosat Altimeter Data for the Agulhas Current Using the Ensemble Kalman Filter with a Quasigeostrophic Model. *Monthly Weather Review*, 124(1):85–96, jan 1996. doi: 10.1175/1520-0493(1996)124<0085: AOGADF>2.0.CO;2.

C. B. Fandry and L. M. Leslie. A Two-Layer Quasi-Geostrophic Model of Summer Trough Formation in the Australian Subtropical Easterlies. *Journal of the Atmospheric Sciences*, 41(5):807–818, mar 1984. doi: 10.1175/1520-0469(1984)041<0807:ATLQGM>2.0.CO;2.

M Fisher. Minimization algorithms for variational data assimilation. In *Seminar on Recent Developments in Numerical Methods for Atmospheric Modelling*, pages 364–385. ECMWF, 1998.

Alison M. Fowler and Amos S. Lawless. An Idealized Study of Coupled Atmosphere Ocean 4D-Var in the Presence of Model Error. *Monthly Weather Review*, 144(10):4007–4030, 2016. doi: 10.1175/mwr-d-15-0420.1.

Oliver Fuhrer, Tarun Chadha, Torsten Hoefler, Grzegorz Kwasniewski, Xavier Lapillonne, David Leutwyler, Daniel Lüthi, Carlos Osuna, Christoph Schär, Thomas C. Schulthess, and Hannes Vogt. Near-global climate simulation at 1km resolution: Establishing a performance baseline on 4888 GPUs with COSMO 5.0. *Geoscientific Model Development*, 11 (4):1665–1681, 2018. doi: 10.5194/gmd-11-1665-2018.

Gregory Gaspari and Stephen E Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, jan 1999. doi: 10.1002/qj.49712555417.

B. Gilchrist and G. P. Cressman. An Experiment in Objective Analysis. *Tellus*, 6(4):309–318, nov 1954. doi: 10.1111/j.2153-3490.1954.tb01126.x.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. North Oxford Academic, 1986. ISBN 0-946536-05-8.

Steven J. Greybush, Eugenia Kalnay, Takemasa Miyoshi, Kayo Ide, and Brian R. Hunt. Balance and Ensemble Kalman Filter Localization Techniques. *Monthly Weather Review*, 139 (2):511–522, 2011. doi: 10.1175/2010MWR3328.1.

S. A. Haben, A. S. Lawless, and N. K. Nichols. Conditioning of incremental variational data assimilation, with application to the Met Office system. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 63(4):782–792, 2011. doi: 10.1111/j.1600-0870.2011.00527.x.

Stephen A. Haben. *Conditioning and Preconditioning of the Minimisation Problem in Variational Data Assimilation*. PhD thesis, 2011.

T Haiden, M Janousek, J. R. Bidlot, R Buizza, L Ferranti, F Prates, and F Vitart. Evaluation of ECMWF forecasts, including the 2018 upgrade. Technical Report 831, 2018.

Thomas M. Hamill. Ensemble-based atmospheric data assimilation. In Tim Palmer and Renate Hagedorn, editors, *Predictability of Weather and Climate*, chapter 6, pages 124–156. Cambridge University Press, 2006. ISBN 0521848822.

Thomas M. Hamill, Jeffrey S. Whitaker, and Chris Snyder. Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter. *Monthly Weather Review*, 129(11):2776–2790, 2001. doi: 10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2.

Mats Hamrud, Massimo Bonavita, and Lars Isaksen. EnKF and Hybrid Gain Ensemble Data Assimilation. Part I: EnKF Implementation. *Monthly Weather Review*, 143(12):4847–4864, 2015. doi: 10.1175/MWR-D-15-0071.1.

Sam Hatfield, Peter Düben, Matthew Chantry, Keiichi Kondo, Takemasa Miyoshi, and Tim Palmer. Choosing the optimal numerical precision for data assimilation in the presence of model error. *Journal of Advances in Modeling Earth Systems*, 2018a. doi: 10.1029/2018MS001341.

Sam Hatfield, Aneesh Subramanian, Tim Palmer, and Peter Düben. Improving weather forecast skill through reduced precision data assimilation. *Monthly Weather Review*, 146: 49–62, 2018b. doi: 10.1175/MWR-D-17-0132.1.

Sam Hatfield, Matthew Chantry, Peter Düben, and Tim Palmer. Accelerating High-Resolution Weather Models with Deep-Learning Hardware. In *Proceedings of the Platform for Advanced Scientific Computing Conference - PASC '19*, pages 1–11. ACM Press, 2019. doi: 10.1145/3324989.3325711.

Nicholas J Higham. *Accuracy and Stability of Numerical Algorithms*. 2002. ISBN 9780898715217. doi: 10.2307/2669725.

Nicholas J Higham, Srikara Pranesh, and Mawussi Zounon. Squeezing a Matrix into Half Precision, with an Application to Solving Linear Systems (preprint). 2019.

P. L. Houtekamer and Herschel L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126(3):796–811, mar 1998. doi: 10.1175/1520-0493(1998)126<0796:DAUAEK>2.0.CO;2.

P. L. Houtekamer and Herschel L. Mitchell. A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation. *Monthly Weather Review*, 129(1):123–137, 2001. doi: 10.1175/1520-0493(2001)129<0123:ASEKFF>2.0.CO;2.

P. L. Houtekamer and Fuqing Zhang. Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation. *Monthly Weather Review*, 144:MWR–D–15–0440.1, 2016. doi: 10.1175/MWR-D-15-0440.1.

P. L. Houtekamer, Xingxiu Deng, Herschel L. Mitchell, Seung-Jong Baek, and Normand Gagnon. Higher Resolution in an Operational Ensemble Kalman Filter. *Monthly Weather Review*, 142(3):1143–1162, 2014. doi: 10.1175/MWR-D-13-00138.1.

B. R. Hunt, E. Kalnay, E. J. Kostelich, E. Ott, D. J. Patil, T. Sauer, I. Szunyogh, J. A. Yorke, and A. V. Zimin. Four-dimensional ensemble Kalman filtering. *Tellus A*, 56(4):273–277, aug 2004. doi: 10.1111/j.1600-0870.2004.00066.x.

Brian R. Hunt, Eric J. Kostelich, and Istvan Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1-2):112–126, jun 2007. doi: 10.1016/j.physd.2006.11.008.

IEEE. IEEE Standard for Floating-Point Arithmetic. pages 1–70. 2008. ISBN 9780738157528. doi: 10.1109/IEEESTD.2008.4610935.

M. Janisková and P. Lopez. Linearized physics for data assimilation at ECMWF. Technical Report 666, 2012.

Tijana Janjić and Stephen E. Cohn. Treatment of Observation Error due to Unresolved Scales in Atmospheric Data Assimilation. *Monthly Weather Review*, 134(10):2900–2915, 2006. doi: 10.1175/MWR3229.1.

Eugenia Kalnay, Hong Li, Takemasa Miyoshi, Shu-Chih Yang, and Joaquim Ballabrera-Poy. 4-D-Var or Ensemble Kalman Filter? *Tellus*, 59A(5):758–773, 2007. doi: 10.1111/j.1600-0870.2007.00261.x.

Milan Klöwer, Peter D. Düben, and Tim N. Palmer. Posits as an alternative to floats for weather and climate models. *Proceedings of the Conference for Next Generation Arithmetic 2019 on - CoNGA'19*, (May):1–8, 2019. doi: 10.1145/3316279.3316281.

Keiichi Kondo and Takemasa Miyoshi. Impact of Removing Covariance Localization in an Ensemble Kalman Filter: Experiments with 10 240 Members Using an Intermediate AGCM. *Monthly Weather Review*, 144(12):4849–4865, 2016. doi: 10.1175/MWR-D-15-0388.1.

V. M. Krasnopolsky, M. S. Fox-Rabinovitz, Y. T. Hou, S. J. Lord, and A. A. Belochitski. Accurate and Fast Neural Network Emulations of Model Radiation for the NCEP Coupled Climate Forecast System: Climate Simulations and Seasonal Predictions*. *Monthly Weather Review*, 138(5):1822–1842, 2009. doi: 10.1175/2009mwr3149.1.

T. N. Krishnamurti, H. S. Bedi, and V. M. Hardiker. Mathematical Aspects of Spectral Models. In *An Introduction to Global Spectral Modeling*, pages 71–103. Springer-Verlag New York, 2014. ISBN 9780195094732.

Amos S. Lawless. Variational data assimilation for very large environmental problems. In *Large Scale Inverse Problems: Computational Methods and Applications in the Earth Sciences*. 2013. ISBN 9783110282269.

A.S. Lawless, S. Gratton, and N.K. Nichols. An investigation of incremental 4D-Var using non-tangent linear models. *Quarterly Journal of the Royal Meteorological Society*, 131 (606):459–476, jan 2005. doi: 10.1256/qj.04.20.

Bryan N. Lawrence, Michael Rezny, Reinhard Budich, Peter Bauer, Jörg Behrens, Mick Carter, Willem Deconinck, Rupert Ford, Christopher Maynard, Steven Mullerworth, Carlos Osuna, Andrew Porter, Kim Serradell, Sophie Valcke, Nils Wedi, and Simon Wilson. Crossing the Chasm: How to develop weather and climate models for next generation computers? *Geoscientific Model Development Discussions*, (September):1–36, 2017. doi: 10.5194/gmd-2017-186.

Martin Leutbecher. Ensemble size: How suboptimal is less than infinity? *Quarterly Journal of the Royal Meteorological Society*, (June):1–22, 2018. doi: 10.1002/qj.3387.

Guo-Yuan Lien, Eugenia Kalnay, and Takemasa Miyoshi. Effective assimilation of global precipitation: Simulation experiments. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 65(1):1–16, 2013. doi: 10.3402/tellusa.v65i0.19915.

Andrew C. Lorenc. The potential of the ensemble Kalman filter for NWP — a comparison with 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 129(595):3183–3203, 2003. doi: 10.1256/qj.02.132.

Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, mar 1963. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

Edward N. Lorenz. Predictability - a problem partly solved. In Tim Palmer and Renate Hagedorn, editors, *Predictability of Weather and Climate*, pages 40–58. Cambridge University Press, Cambridge, 2006. doi: 10.1017/CBO9780511617652.004.

Edward N. Lorenz and Kerry a. Emanuel. Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model. *Journal of the Atmospheric Sciences*, 55(3): 399–414, 1998. doi: 10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2.

Tony McNally, Massimo Bonavita, and Jean-Noël Thépaut. The Role of Satellite Data in the Forecasting of Hurricane Sandy. *Monthly Weather Review*, 142(2):634–646, 2014. doi: 10.1175/MWR-D-13-00170.1.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed Precision Training. *CoRR*, abs/1710.0, 2017.

Takemasa Miyoshi. *Ensemble Kalman Filter Experiments With a Primitive-Equation Global Model*. PhD thesis, 2005.

Takemasa Miyoshi. The Gaussian Approach to Adaptive Covariance Inflation and Its Implementation with the Local Ensemble Transform Kalman Filter. *Monthly Weather Review*, 139:1519–1535, 2011. doi: 10.1175/2010MWR3570.1.

Takemasa Miyoshi, Yoshiaki Sato, and Takashi Kadowaki. Ensemble Kalman Filter and 4D-Var Intercomparison with the Japanese Operational Global Analysis and Prediction System. *Monthly Weather Review*, 138(7):2846–2866, 2010. doi: 10.1175/2010MWR3209.1.

Takemasa Miyoshi, Keiichi Kondo, and Toshiyuki Imamura. The 10,240-member ensemble Kalman filtering with an intermediate AGCM. *Geophysical Research Letters*, 41:5264–5271, 2014. doi: 10.1002/2014GL060863.

F Molteni. Atmospheric simulations using a GCM with simplified physical parametrizations. I: Model climatology and variability in multi-decadal experiments. *Climate Dynamics*, pages 175–191, 2003. doi: 10.1007/s00382-002-0268-2.

Andreas Müller, Willem Deconinck, Christian Kühnlein, Gianmarco Mengaldo, Michael Lange, Nils Wedi, Peter Bauer, Piotr K. Smolarkiewicz, Michail Diamantakis, Sarah-Jane Lock, Mats Hamrud, Sami Saarinen, George Mozdzynski, Daniel Thiemert, Michael Glinton, Pierre Bénard, Fabrice Voitus, Charles Colavolpe, Philippe Marguinaud, Yongjun Zheng, Joris Van Bever, Daan Degrauwe, Geert Smet, Piet Termonia, Kristian P. Nielsen, Bent H. Sass, Jacob W. Poulsen, Per Berg, Carlos Osuna, Oliver Fuhrer, Valentin Clement, Michael Baldauf, Mike Gillard, Joanna Szmelter, Enda O'Brien, Alastair McKinstry, Oisín Robinson, Parijat Shukla, Michael Lysaght, Michał Kulczewski, Milosz Ciznicki, Wojciech Piątek, Sebastian Ciesielski, Marek Błażewicz, Krzysztof Kurowski, Marcin Procyk, Pawel Spychala, Bartosz Bosak, Zbigniew Piotrowski, Andrzej Wyszogrodzki, Erwan Raffin, Cyril Mazauric, David Guibert, Louis Douriez, Xavier Vigouroux, Alan Gray, Peter Messmer, Alexander J. Macfaden, and Nick New. The ESCAPE project: Energy-efficient Scalable Algorithms for Weather Prediction at Exascale. *Geoscientific Model Development Discussions*, (January):1–50, 2019. doi: 10.5194/gmd-2018-304.

Masuo Nakano, Hisashi Yashiro, Chihiro Kodama, Hirofumi Tomita, Masuo Nakano, Hisashi Yashiro, Chihiro Kodama, and Hirofumi Tomita. Single Precision in the Dynamical Core of a Nonhydrostatic Global Atmospheric Model: Evaluation Using a Baro-

clinic Wave Test Case. *Monthly Weather Review*, pages MWR–D–17–0257.1, 2018. doi: 10.1175/MWR-D-17-0257.1.

Lars Nerger. On sequential observation processing in localized ensemble Kalman filters. *Monthly Weather Review*, 143:1554, 2014. doi: 10.1175/MWR-D-14-00182.1.

NVIDIA. NVIDIA Tesla V100 GPU Architecture. Technical report, 2017a.

NVIDIA. NVIDIA TESLA P100 Technical Overview, 2017b.

Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D.J. Patil, and James A. Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 56 (5):415–428, jan 2004. doi: 10.3402/tellusa.v56i5.14462.

T. N. Palmer. Stochastic weather and climate models. *Nature Reviews Physics*, 2019. doi: 10.1038/s42254-019-0062-2.

T. N. Palmer, R. Buizza, F. Doblas-Reyes, T. Jung, Martin Leutbecher, G. Shutts, M. Steinheimer, and Antje Weisheimer. Stochastic Parametrization and Model Uncertainty. *ECMWF Tech. Memo. 598*, page 42, 2009.

Tim Palmer. The primacy of doubt: Evolution of numerical weather prediction from determinism to probability. *Journal of Advances in Modeling Earth Systems*, 9(2):730–734, jun 2017. doi: 10.1002/2017MS000999.

Tim Palmer, Roberto Buizza, Renate Hagedorn, Andy Lawrence, Martin Leutbecher, and Lenny Smith. Ensemble prediction: a pedagogical perspective. *ECMWF Newsletter*, 106 (106):10–17, 2006. doi: 10.21957/ab129056ew.

R. A. Panofsky. Objective weather-map analysis. *Journal of Meteorology*, 6(6):386–392, dec 1949. doi: 10.1175/1520-0469(1949)006<0386:OWMA>2.0.CO;2.

Joseph Pedlosky. *Geophysical Fluid Dynamics*. Springer-Verlag New York, 1 edition, 1979.

David Randall. *Numerical modeling of the atmosphere*. 2013.

F. Rawlins, S. P. Ballard, K. J. Bovis, A. M. Clayton, D. Li, G. W. Inverarity, A. C. Lorenc, and T. J. Payne. The Met Office global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133(623):347–362, jan 2007. doi: 10.1002/qj.32.

Lewis Fry Richardson. *Weather Prediction by Numerical Process*. 1922. doi: 10.2307/3603284.

Juan Ruiz and Manuel Pulido. Parameter Estimation Using Ensemble-Based Data Assimilation in the Presence of Model Error. *Monthly Weather Review*, 143(5):1568–1582, may 2015. doi: 10.1175/MWR-D-14-00017.1.

Juan Ruiz, Manuel PULIDO, and Takemasa MIYOSHI. Estimating Model Parameters with Ensemble-Based Data Assimilation: A Review. *Journal of the Meteorological Society of Japan. Ser. II*, 91(2):79–99, 2013. doi: 10.2151/jmsj.2013-201.

C. Schraff, H. Reich, A. Rhodin, A. Schomburg, K. Stephan, A. PeriÃ¡Ã±ez, and R. Potthast. Kilometre-scale ensemble data assimilation for the COSMO model (KENDA). *Quarterly Journal of the Royal Meteorological Society*, 142(696):1453–1472, 2016. doi: 10.1002/qj.2748.

Thomas C Schulthess. Programming revisited. *Nature Publishing Group*, 11(5):369–373, 2015. doi: 10.1038/nphys3294.

Thomas C. Schulthess, Peter Bauer, Oliver Fuhrer, Torsten Hoefler, Christoph Schar, and Nils Wedi. Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simu-lations. *Computing in Science and Engineering*, pages 30–41, 2018. doi: 10.1109/MCSE.2018.2888788.

A. J. Simmons, D. M. Burridge, M. Jarraud, C. Girard, and W. Wergen. The ECMWF medium-range prediction models development of the numerical formulations and the impact of

increased resolution. *Meteorology and Atmospheric Physics*, 40(1-3):28–60, 1989. doi: 10.1007/BF01027467.

Travis C. Sluka, Stephen G. Penny, Eugenia Kalnay, and Takemasa Miyoshi. Assimilating atmospheric observations into the ocean using strongly coupled ensemble data assimilation. *Geophysical Research Letters*, 43(2):752–759, 2016. doi: 10.1002/2015GL067238.

Tobias Thornes, Peter Düben, and Tim Palmer. On the use of scale-dependent precision in Earth System modelling. *Quarterly Journal of the Royal Meteorological Society*, 143(703): 897–908, 2017. doi: 10.1002/qj.2974.

Michael K. Tippett, Jeffrey L. Anderson, Craig H. Bishop, Thomas M. Hamill, and Jeffrey S. Whitaker. Ensemble Square Root Filters*. *Monthly Weather Review*, 131(7):1485–1490, jul 2003. doi: 10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2.

Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM, 1997.

Yannick Trémolet. Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006. doi: 10.1256/qj.05.224.

G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics*. 2006. ISBN 9780521849692. doi: 10.1002/qj.186.

Filip Váňa, Peter Düben, Simon Lang, T. N. Palmer, Martin Leutbecher, Deborah Salmond, and Glenn Carver. Single Precision in Weather Forecasting Models: An Evaluation with the IFS. *Monthly Weather Review*, 145(2):495–502, feb 2017. doi: 10.1175/MWR-D-16-0228.1.

Peter A. G. Watson. Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction. *Journal of Advances in Modeling Earth Systems*, page 2018MS001597, 2019. doi: 10.1029/2018MS001597.

N. P. Wedi. Increasing horizontal resolution in numerical weather prediction and climate simulations: illusion or panacea? *Philosophical Transactions of the Royal Society A: Math-*

ematical, Physical and Engineering Sciences, 372(2018):20130289–20130289, 2014. doi: 10.1098/rsta.2013.0289.

Nils P. Wedi, Mats Hamrud, and George Mozdzynski. A Fast Spherical Harmonics Transform for Global NWP and Climate Models. *Monthly Weather Review*, 141(10):3450–3461, 2013. doi: 10.1175/MWR-D-13-00016.1.

Jeffrey S. Whitaker and Thomas M. Hamill. Ensemble Data Assimilation without Perturbed Observations. *Monthly Weather Review*, 130(7):1913–1924, jul 2002. doi: 10.1175/1520-0493(2002)130<1913:EDAWPO>2.0.CO;2.

Jeffrey S. Whitaker and Thomas M. Hamill. Evaluating Methods to Account for System Errors in Ensemble Data Assimilation. *Monthly Weather Review*, 140(9):3078–3089, 2012. doi: 10.1175/MWR-D-11-00276.1.

Daniel S. Wilks. Effects of stochastic parametrizations in the Lorenz '96 system. *Quarterly Journal of the Royal Meteorological Society*, 131(606):389–407, jan 2005. doi: 10.1256/qj.04.03.

F. Zhang, Chris Snyder, and Juanzhen Sun. Impacts of Initial Estimate and Observation Availability on Convective-Scale Data Assimilation with an Ensemble Kalman Filter. *Monthly Weather Review*, 132(5):1238–1253, 2004. doi: 10.1175/1520-0493(2004)132<1238:IOIEAO>2.0.CO;2.