The use of imprecise processing to improve accuracy in weather & climate prediction

Peter D. Düben[a], Hugh McNamara[b], T. N. Palmer[a]

[a] *University of Oxford, Atmospheric, Oceanic and Planetary Physics*
[b] *University of Oxford, Mathematical Institute*

Abstract

The use of stochastic processing hardware and low precision arithmetic in atmospheric models is investigated. Stochastic processors allow hardware-induced faults in calculations, sacrificing bit-reproducibility in exchange for improvements in performance and potentially accuracy and a reduction in power consumption. A similar trade-off is achieved using low precision arithmetic, with improvements in computation and communication speed and savings in storage and memory requirements. As high-performance computing becomes more massively parallel and power intensive, these two approaches may be important stepping stones in the pursuit of global cloud resolving atmospheric modelling.

The impact of both hardware induced faults and low precision arithmetic is tested using the Lorenz '96 model and the dynamical core of a global atmosphere model. In the Lorenz '96 model there is a natural scale separation, the spectral discretisation used in the dynamical core also allows large and small scale dynamics to be treated separately within the code. Such scale separation allows the impact of lower-accuracy arithmetic to be restricted to components close to the truncation scales, and hence close to the necessarily inexact parametrised representations of unresolved processes. By contrast, the larger scales are calculated using exact arithmetic. Hardware faults from stochastic processors are emulated using a bit-flip model with different fault rates.

Our simulations show that both approaches to inexact calculations do not substantially affect the mean behaviour, provided they are restricted to act only on smaller scales. By contrast, results with inexact calculations can be superior to those where smaller scales are parametrised. This suggests that inexact calculations at the small scale could reduce computation and power costs without adversely affecting the quality of the simulations. This would allow higher resolution models to be run at the same computational cost.

Keywords: Stochastic processor, scale separation, atmospheric models, spectral discretisation, Lorenz '96, single precision

1. Introduction

Energy demands and error resilience are two of the major challenges to be overcome in the building of "exascale" high-performance computing (HPC) hardware, planned to be realized

Email address: dueben@atm.ox.ac.uk (Peter D. Düben)

in 2020 [1]. An exascale HPC system is able to perform 10^{18} floating-point operations per second. Power consumption is already one of the major cost factors with modern HPC systems. Traditional processor design uses rather large tolerances to prevent natural fluctuations from impacting on the results of calculations. This ensures that every run of a programme produces exactly the same results – termed bit-reproducibility. Guarding against such fluctuations, which can have causes as diverse as thermal noise and cosmic ray impacts, requires that hardware be run at a higher voltage than otherwise necessary.

Through a suitable redesign of the processing hardware, a number of groups have demonstrated the possibility of a trade-off between exactness of computations and power consumption [2, 3, 4]. By relaxing the requirement of bit-reproducibility, HPC systems with much lower energy requirements become possible, with reductions in the costs of manufacturing, verification and testing [5]. While the work on such approaches is at an early prototype stage using simplified architectures, results suggest that power consumption could be reduced, on average, by anything from around 12–20% [3, 6] at low fault rates (1–2%) up to about 90% (at a fault rate of 10%, [7]).

These reductions in power requirements are achieved through voltage over-scaling – reducing the voltage applied to the processor beyond that at which all computation paths proceed successfully at a given clock-speed. The requirements of such an approach are that calculations degrade "gracefully" as this over-scaling is applied: rather than the computation failing entirely or producing a meaningless result when voltage is reduced, at least some accuracy remains even when the result is incorrect [8]. A change in the processor architecture can reduce the fault rates for reduced voltages. This effort is currently an active area of research, without a clear design emerging for such imprecise, or stochastic processors.[1] Nevertheless, data from early investigations can be used to construct a fault model which is used to emulate the effects of running code on such processors.

In weather and climate science, numerical models are a very important ingredient for forecasts and predictions. The HPC systems that are used to run climate and weather predictions are among the fastest computers in the world, but current computing power is still not sufficient. Higher computing performance allows higher resolution, and the resolution in state-of-the-art atmospheric simulations is still far from being adequate [9].

One of the key justifications for the development of approximate computing techniques and low-precision arithmetic lies in the nature of the parametrisation problem for weather and climate models. It has been argued elsewhere [10, 11] that the parametrisation problem is fundamentally stochastic in nature. Forecast systems using stochastic parametrisation have been shown to lead to more reliable forecasts and to reduced systematic errors [12]. Stochasticity in the representation of sub-grid processes will necessarily induce stochasticity in the elements of the dynamical core. We can expect induced stochasticity to be relatively strong near the truncation scale of a dynamical core, but relatively weak at large scales. As such, the use of double precision bit-reproducible dynamics for scales near the truncation scale will introduce unwarranted precision into the dynamical-core computations. That is to say, current dynamical cores may be over-engineered, given the inherent inaccuracy of the parametrisation problem. If we can relax the exactness of the dynamical core in a scale-selective fashion, we may be able to develop much higher resolution models, for a given computational resource. Consistent with this, a recent

[1] It is the view of the authors that the best term to describe such hardware is "imprecise", rather than "stochastic". As it is the convention in the computer science community which is working on the design of this hardware however, "stochastic" will be used from now on.

paper showed that a decrease in precision on the software level can lead to a decrease of the computational cost without degrading the quality of the model simulations, using an inexact, fast Legendre Transform. The increase in performance allows simulations with the forecasting model of the European Centre for Medium-range Weather Forecasting (ECMWF) with higher resolution (up to T7999) than possible with a common Legendre Transform [13].

Against the background of a severe demand for computing power and the trend towards the use of stochastic methods in weather and climate models for physical reasons, stochastic processors seem to be a promising tool for atmospheric simulations. Stochastic processors not only have the potential to significantly decrease the energy cost, or increase the performance, it is furthermore possible that the "random noise" introduced by the faults of the processors could bring a benefit to the model simulations and allow for hardware-based ensembles. To date very little is known about the behaviour of numerical simulations, particularly those of atmospheric dynamics, when computed on stochastic hardware. Without further investigation, it is unclear whether such simulations can be run successfully without crashes or instabilities, or what could be done to make current code robust in the presence of hardware-induced faults.

This paper records the first attempt to apply emulated stochastic processors to an atmospheric simulation. The code of a "toy" model for atmospheric dynamics (Lorenz '96) and of a dynamical core of a spectral atmospheric model (the IGCM, see Section 4) is modified to emulate the effects of a stochastic processor. This work follows the approach proposed in [11] in that the small-scale (high-wavenumber) dynamics are affected by the stochastic hardware emulation, while the large-scale (low-wavenumber) components are calculated exactly. This respects the fact that small-scale dynamics close to the truncation scale are anyway inexactly computed, whereas the large-scale dynamics are crucially important. The Held-Suarez test-case for atmospheric simulations is evaluated [14].

The emulator for stochastic processors can also be used to emulate the use of scale-dependent low-precision arithmetic in model simulations. In modern HPC systems, communication and storage costs contribute more and more to the power and time costs of simulations. The use of low precision numbers to store small-scale components could reduce these costs substantially, while not significantly impacting on the accuracy of calculations. We examine the impact of using low-precision representations for small-scale components in the IGCM using the same test case.

Section 2 describes the fault model used to emulate stochastic processors and low precision calculations. Section 3 presents the investigation of the Lorenz '96 model, including a description of the model and the results with emulated stochastic processors. In section 4 a short description of the atmospheric model IGCM is given, the validity and feasibility of scale-separation is discussed, and the simulations and results for emulated stochastic processors and low precision arithmetic are presented. A discussion of the results and an outlook towards future investigations and research is given in section 5.

2. A fault model for stochastic processors and low precision floating-points

Numbers stored by computers and used for calculations must be represented by a finite sequence of bits, each either 1 or 0. Two main types of representation are used, *integer* and *floating point*. According to the IEEE754 standard, a double precision floating-point number x is represented by a sequence of 64 bits. The first of these is the *sign bit*, denoted s, which is followed by 11 bits which comprise the *exponent*, denoted $c_{10}, c_9, \ldots, c_0$. The remaining 52 bits are the

mantissa or *significand*, denoted $b_{-1}, b_{-2}, \ldots, b_{-52}$. The relationship between x and its bit representation as s, c_i and b_{-i} is given by

$$x = (-1)^s \left(1 + \sum_{i=1}^{52} b_{-i} 2^{-i}\right) 2^E, \qquad \text{where} \qquad E = \left(\sum_{i=0}^{10} c_i 2^i\right) - 1023.$$

This work focuses on the effects of transient faults which alter the results of floating-point computations. Other manifestations of faults, such as memory corruption or control flow deviations, may be overcome using simple, low-overhead techniques. Such techniques have long been an active area of research, [15, 16, 17, 18, 19].

In this paper, we adopt the following fault model: when a fault occurs in a calculation, the impact of the faulty hardware is modelled by randomly flipping one bit in the significand of the result, without any impact on the exponent or sign bits. This model follows from results in [20], and is also used in [21]. As bit flips in the sign or exponent bits tend to produce catastrophic errors and crashes, it is supposed that future designs will seek to preserve this behaviour. In [20] it is further observed that such bit-wise errors tend to be distributed among the most- and least-significant bits. To reduce the complexity of the fault model we allow faults to occur with equal likelihood at any position along the significand, so that a fault consists of flipping a randomly chosen bit from the 52 bits of the significand of the result of a calculation (the b_{-i} above).

A stochastic processor is emulated using the following model to inject faults into calculations:

1. An average fault rate, $0 \leq p \leq 1$, is specified.
2. After *every* floating-point operation (including basic algebraic operations such as addition and multiplication as well as standard library functions like sine and cosine) it is randomly decided whether a fault has occurred (a Bernoulli trial with probability p).
3. If a fault is indicated, a position for the fault is randomly chosen from a uniform distribution over the integers 1–52. The bit at this position in the significand of the result of the calculation is changed – a 1 becomes a 0 or vice versa.

It is assumed that initialisation, output, and testing will be performed on exact hardware, therefore no faults are introduced in these parts of the code. This supposes an approach with both exact and inexact hardware, with programming control over which portions of the code to execute on which hardware.

The emulator for stochastic processors can also be used to emulate the use of low precision floating-point arithmetic and storage. Numbers stored at lower precision will consist of shorter bit sequences. A crude representation of lower precision is to take a longer sequence and truncate the accuracy of this sequence by "flipping" a particular bit 50% (on average) of the results of floating-point operations. This destroys the precision of the floating-point number beyond the switched bit. Tests here truncate the significand to 6 usable bits, a very severe restriction of precision (compared even with single precision floating point representations which use 23 bits for the significand). We do not reduce the range of the exponent, but we expect that bits can be reduced for the exponent as well. For example, a 12-bit floating-point system with 1 sign bit, 5 exponent bits and 6 significand bits could store numbers between 10^{-4} and 10^4 approximately, albeit at very low precision.

3. The Lorenz '96 model

Two models are referred to as the Lorenz '96 model (or sometimes the Lorenz '95 model), both introduced by Lorenz in a talk and associated paper (originally a technical report, eventually published as [22]). The two models can be seen as coarse discretisations of atmospheric flow on a line of latitude, supporting complicated wave-like and chaotic behaviour [23, 24]. Both models have been used widely as test-beds for data assimilation methods [25, 26] and for closure or parametrisation schemes [27, 28, 29, 30]. The second model, called the two-level Lorenz '96 model, schematically describes the interaction between small-scale (eddy) waves with larger scale motions. Large scale motions are described by variables $X_k, k = 1, \ldots, K$ and are coupled to small-scale variables Y_j, $j = 1, \ldots, KJ$.

$$\frac{\mathrm{d}X_k}{\mathrm{d}t} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b} \sum_{j=J(k-1)+1}^{kJ} Y_j \tag{1}$$

$$\frac{\mathrm{d}Y_j}{\mathrm{d}t} = -cbY_{j+1}(Y_{j+2} - Y_{j-1}) - cY_j + \frac{hc}{b} X_{\mathrm{int}[(j-1)/J]+1} \tag{2}$$

A schematic showing the coupling of the large- and small-scale variables is shown in Figure 1. The variables are coupled together periodically, so that waves may propagate around a circle both in the Xs and the Ys. The parameters specify the coupling strength (h) and time- and space-scale separations (c and b respectively) between the X and Y variables. The large-scale forcing, F, could be an arbitrary function of time, here we use a constant forcing. The parameters are chosen as in [31], which corresponds to relatively large separation of length-scales ($b = 10$), straightforward coupling ($h = 1$) and strong forcing ($F = 10$). Two values of the time-scale separation are investigated: $c = 10$, which is a large separation, and a more moderate $c = 4$. Each large-scale variable is coupled to 32 small-scale variables ($J = 32$) and there are 8 of these ($K = 8$) leading to a total system size of $K + KJ = 264$ variables. These two combinations of parameters both produce chaotic behaviour with irregular aperiodic waves and sensitivity to initial conditions.

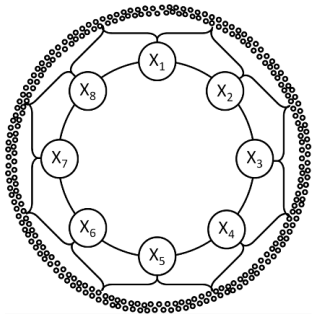

Figure 1: Schematic of the Lorenz '96 model, after [30].

In this study, the two-level Lorenz '96 model (referred to from now on as L96) will be used to investigate the effects of hardware faults. The faults will be allowed to affect only the smaller, faster scales (the Y_j), and the impact on the simulation of the larger scales (the X_k) is considered. Here we focus on "climatic" effects: Does the introduction of faults at smaller scales impact on

the long-term statistics of the larger scales. Emulated low precision arithmetic applied to the Lorenz '96 model is not presented.

3.1. Simulations and results

The simplicity of the L96 system allows large ensembles of simulations to be performed over a very long time in order to build up reliable climate statistics. A single initial condition was obtained by "spinning-up" the unperturbed system, and this was used for all ensemble members. Each ensemble member evolves the L96 system through 20000 model time units (according to the original paper, [22], one model time unit corresponds to approximately 5 atmospheric days), with 50 sample points per time unit (one sample every 20 numerical time-steps). The evolution uses the common 4th order Runge-Kutta scheme. This scheme involves 4 evaluations of the right-hand-side functions every time-step, each of which consists of six multiply-add operations (i.e. operations of the form $y \leftarrow a \times x + b$). The emulated fault model is applied to each of these operations in the Y portion of the RHS calculation.[2]

For each parameter case ($c = 10$ and $c = 4$) and fault rate an ensemble of 100 simulations was run. Ensemble results are compared with both a fault-free simulation and the behaviour of a stochastically parametrised simulation. The stochastic parametrisation does away with the Y variables completely, replacing the coupling term in the X equations with a formula which models the missing contributions. Such parametrisations are presented in [31], and the simple AR(1) additive version from that work is used for comparison.

Ensemble averages were taken for the various diagnostics and compared with the fault-free run and stochastically parametrised results. For both of the two parameter cases, fault rates of 20% are used. For each faulty run, a random integer $k \in \{1, \ldots, 8\}$ is chosen, and the statistics are calculated for X_k with this k. Power spectral density, autocorrelation and a kernel density estimate of the PDF of values taken by this X_k are calculated in Python.[3] For the cross-correlation and 2d PDF, the adjacent k is used.

Figure 2 (a–f) shows the results for the $c = 10$ case. Here, there are some clearly defined wave modes, seen in the spectral peaks and oscillating auto- and cross-correlations. The faulty simulations stay very close to the "truth", and the ensemble standard deviation of each statistic is small: about 1% for the power spectrum, correlations and 1d PDF, rising to around 5% for the 2d PDF estimate. For all results the faulty calculation significantly outperforms the stochastically parametrised version.

The $c = 4$ case is shown in Figure 3 (a–f). The dynamics here are more strongly chaotic, with no clear spectral peaks and rapidly decreasing auto- and cross-correlations. The correlation figures show some deviation from the fault-free case at moderate lags (2–3 model time units), but at these lag times the correlations have already decayed substantially, so little information is lost.

4. The Intermediate Global Climate Model (IGCM)

In this section, we will investigate the use of emulated stochastic processors and low-precision arithmetic in a dynamical core of an atmospheric model. The section starts with a short descrip-

[2]Note that this means there are 24 such operations per Y-variable per time-step. At the fault rates considered here there is a strong probability that every Y-tendency is affected by faults every time-step.

[3]The gaussian-kde routine from SciPy is used for the 1d and 2d kernel density estimates. The MatPlotLib psd routine calculated the power spectral density and the correlations are calculated using Fast Fourier Transforms.

tion of the model, the used test-case, and the setup of the simulations with imprecise processing. Afterwards, the numerical results are presented.

4.1. Model description and scale separation

The Intermediate Global Climate Model (IGCM), sometimes called the Reading Spectral Model, is a three dimensional model of the global atmosphere [32, 33, 34, 35]. The IGCM simulates the primitive equations in σ-coordinates on the sphere. The set of equations is outlined in Appendix A.

In IGCM the equations are discretised using a spectral discretisation scheme, which transforms between spherical harmonic and grid-space representations in every time step. The transformations are necessary since the tendencies of the non-linear terms of the equations of motion, such as $\left(U^2 + V^2\right)$, UT_A, $\mathfrak{F}_u$, and $\frac{\partial(VT_A)}{\partial\mu}$, are calculated in grid-point space. The calculated tendencies are than transformed back to the space of spherical harmonics, and used to calculate the right-hand side of the equations of motion, when time stepping schemes are performed. In full atmosphere models, most of the parametrisation schemes and the tracer dynamics are also calculated in grid point space. In order to compute the grid-space representation from the representation as spherical harmonics, first a Legendre transform (LT) and then a Fourier Transform (using an FFT) are applied in succession. These transforms are applied in reverse order to return to the space of spherical harmonics.

In the following, we consider the simulator as comprises of three portions. The first consists of all operations in spectral space, and the Legendre Transform operation, the second is the FFT, and the third is the non-linear calculations in grid-point space. The three portions will be denoted SS & LT, FFT and NL, respectively.

The two transforms form a large part of the computational workload. Table 1 shows the proportion of the time of the full simulation spend in each portion of the code at different resolutions. The resolutions are listed by a 'T' followed by the wavenumber at which the spectral series of the spherical harmonics is truncated. The 'T' represents the use of a triangular truncation [32]. The separation into large- and small-scale components is straightforward when dealing with spectral components (as in the SS & LT column in Table 1), and would also be possible if a crude discrete Fourier transform were used. The nature of the FFT algorithm makes scale-separation much more difficult and less worthwhile since wavenumber components are rearranged into pairs, each of one small and one large wavelength (see for example [36] for a description of the FFT). Thus no clear "small-scale" calculations can be distinguished in the NL and FFT portions of the simulation.

Table 1: Distributions of computation costs for different resolutions of IGCM. All simulations are performed with 20 vertical layers.

Resolution	SS & LT	FFT	NL
T21	41%	35%	23%
T31	45%	35%	20%
T42	48%	33%	19%
T84	64%	25%	11%

The proportional costs strongly depend on the horizontal resolution, and different parts of the computation show different scaling behaviour with the spectral wave number N at which the series is truncated. While the LT scales like $O(N^3)$, the FFT scales as $O(N \log N)$. It is therefore not surprising that the SS & LT portion of the code makes up an increasingly large proportion of the workload as the resolution is increased.

The distribution of the workload in a full, high resolution simulation of the atmosphere is quite different compared to that of the low resolution dynamical core. For the non-hydrostatic Integrated Forecasting System (IFS) developed at the ECMWF, the computational cost for the FFT compared to the LT is about 2:3 at a resolution of T799, and about 1:3 at a resolution of T3999. The relative workload in grid-point space is much higher for IFS than for IGCM, due to parametrisation schemes and tracer dynamics, and forms about 60% of the computational cost for simulations at T3999 (Nils Wedi personal communication). Another major difference between the IFS and the version of the IGCM used here is the use of a reduced Gaussian grid so that the IFS has fewer grid-points near the poles, [37].

4.2. Test-case and simulations

The Held-Suarez test is often used to validate the behaviour of dynamical cores of atmospheric models, [14]. The test involves relaxation to a prescribed, zonally symmetric temperature field. We simulated the Held-Suarez case at horizontal resolutions of T31 and T42 with 20 vertical levels. The first 1000 days of simulation time were discarded as spin-up time, with the results drawn from the following 10000 days.

Running the entire IGCM dynamical core with emulated fault injection or strongly reduced precision caused the code to crash almost immediately. Only when the emulation was applied only to certain portions of the code was it possible to obtain meaningful results at certain fault rates and truncation levels.

The following simulations were carried out:

Case 0: Control simulations at T31 and at T42 resolutions. We performed two T42 control simulations with different initial conditions and one simulation in which we used 4th order diffusion with six hour diffusion time scale instead of the 8th order diffusion with 2.4 hours diffusion time scale which was used for all other simulations.

Case 1: Two simulations at T42 in which the NL portion of the code uses the emulated stochastic processor or emulated 6-bit precision. The remainder of the code uses exact processing. In this case 18% of the floating-point operations are carried out through the emulated fault model.

Case 2: Two simulations at T42 in which the SS & LT portion for total wave-numbers between 32 and 42 and the NL portion use the emulated stochastic processor or emulated 6-bit precision. This increases the proportion of operations using the emulator to 31%.

Case 3: Two simulations at T42 in which the SS & LT portion for total wave-numbers between 32 and 42 and the NL and FFT portions use the emulated stochastic processor or emulated 6-bit precision. 84% of the floating-point operations are performed with the emulator.

In the following discussion we will refer to cases 1, 2, and 3 as given above. A fault rate of 10% was stable, while 30% caused the code to crash. Similarly, truncating floating-point precision to 6 bits in the significand worked, while truncating to 4 bits caused a crash.

4.3. Results with emulated stochastic processor

Figure 4 shows the resulting zonal- and time-mean zonal velocity for all of the above cases at a fault rate of 10%. The differences between cases 1, 2 and 3 and the T42 control run are hardly noticeable. Differences are plotted in Figure 5, where additionally we show results from a fault rate of 1%. The difference between cases 1, 2 and 3 at the 1%, and case 1 at the 10% fault rate have the same magnitude as the difference between the two T42 control runs. It is very clear that in case 3, where the FFT as well as the SS & LT and NL parts are faulty, the error is increased at 10 % fault rate. For all cases, the changes are smaller than the changes obtained when performing simulations with 4th order diffusion (instead of 8th order diffusion). The changes for case 3 with 10 % fault rate show a similar pattern to the changes we get when we use the different diffusion scheme, projecting strongly onto the major mode of annular variability.

To evaluate the impact of the stochastic processing on the representation of eddies, the transient eddy-momentum was calculated as $\overline{[u'^* v'^*]}$, where u and v are the zonal and meridional wind, the overbar and square brackets denote time-averaging and zonal-averaging, and the prime and asterix denote deviations from the time and zonal averages, respectively. The same diagnostic was used in the Aqua-Planet Experiments (APE) for model intercomparison [38]. Figure 6 shows the results of this diagnostic for the various cases. Figure 7 shows the differences between the transient eddy-momentum of the simulations in Figure 6 and a reference T42 simulation with different initial conditions. Although changes can be seen for case 2 and 3, especially for the simulation in which the FFT is also performed on the stochastic processor, the changes do not exceed the impact of changing the diffusion scheme.

Figure 8 shows the daily mean of the horizontal kinetic energy spectra for the different cases at the tenth vertical level (at a standard height around five kilometers) and a fault rate of 10%. Again it can be seen that the errors in the calculations with a stochastic processor produce small changes in the results for cases 2 and 3, mostly for the simulation in which the FFT was performed on the emulated stochastic processor, but they do not lead to a large change in the spectra, and the results are significantly improved compared to the exact simulation with a spectral resolution of T31.

4.4. Results with emulated low precision

The same test cases were simulated (case 1-3), this time the emulator reduces the precision of floating-point calculations. The emulation affected the significand of floating-point numbers, polluting the representation beyond the 6th bit of the significand.

The mean zonal velocity shows little difference caused by low precision at small scales. Figure 9 shows the differences between a T42 control simulation and each low precision test case, and compares them with the difference between two T42 control runs. All cases show a very similar magnitude effect as the difference between the two T42 runs. The differences in the transient eddy-momentum are also very small, as can be seen in Figure 10. Figure 11 shows that the effects on the energy spectra are also negligible.

5. Conclusion and Outlook

These results suggest that the use of imprecise computing strategies, particularly focused on the small-scale dynamics, would be of use in atmospheric simulations and should be further investigated. Of coarse, a number of criticisms could be made of this work for example with respect to the low resolution, idealised configuration, fairly crude diagnostics, and the reduction of the

tests to the dynamical core and a toy model. We still believe that the results indicate the potential usefulness of such approaches, and advocate for further investigation and experimentation.

The most severe cases presented here (case 3 described above) represent a simulation of the dynamical core in which 84% of the floating-point calculations are performed through an emulated stochastic processor or with emulated low precision of only 6 bits in the significand and would cost far less in terms of energy consumption. The lack of severe penalties were found, at least for a 1% fault rate, suggest that this is a worthwhile effort. Given a budget for computer resources, the use of imprecise hardware would allow for higher resolution, with the small scales imprecisely simulated.

Early efforts with the dynamical core, without scale separation, show that a crude implementation of imprecise strategies will not pay off. Interesting questions should be looked into regarding the level of scale separation required and how to efficiently implement numerical algorithms using a mix of exact and imprecise hardware. It does appear that separation into exact and imprecise scales is a necessary exercise, however a more robust implementation may allow a variable fault rate across the scales of the simulation. Examining the power consumption of different parts of the code would also be of benefit in targeting imprecise strategies to where they would have most impact.

The emulation of low precision floating-point storage and arithmetic employed here is still very crude, but it shows remarkable results. The impact on all diagnostics was minimal, despite a rather severe truncation of the data. Since communication and storage are very expensive components of large HPC systems, especially for weather and climate simulations, the reduction of bits that need to be stored and communicated seems to have a very high potential.

In future studies we will investigate more sophisticated test cases, perform a more detailed cost/benefit evaluation and perform similar tests with grid point models. We will apply stochastic processors in a state-of-the-art atmospheric model (the IFS developed at ECMWF) to test the possibility of using inexact stochastic processors for "hardware based ensembles" and stochastic parametrisation.

The results presented here cannot answer the question if it will be possible to use stochastic processors or heavily reduced precision arithmetic in weather and climate modelling, but they do show that these methods offer huge potential.

Acknowledgements

We thank Hannah Arnold for helpful discussions and her dedicated support for the L96 simulations. The help of Fenwick Cooper and Mike Blackburn was crucially important for the simulations with IGCM. While Hugh McNamara is supported by the Oxford Martin School (grant number LC0910-017), the position of Peter Düben is funded by an ERC grant (Towards the Prototype Probabilistic Earth-System Model for Climate Prediction, project number DCLALJE1).

Appendix A. The model equations

In this section we present the primitive equations which the dynamical core of the IGCM approximates. A detailed description of the discretisation approach and on the IGCM itself can be found in [32, 33, 34, 35]. The following set of equations are simulated:

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= \frac{1}{1-\mu^2}\frac{\partial \mathfrak{F}_v}{\partial \lambda} - \frac{\partial \mathfrak{F}_u}{\partial \mu} - \frac{\zeta-\mu}{\tau_F} + K(-1)^{p_d/2}\nabla^{p_d}(\zeta-\mu) \\
\frac{\partial D}{\partial t} &= \frac{1}{1-\mu^2}\frac{\partial \mathfrak{F}_u}{\partial \lambda} + \frac{\partial \mathfrak{F}_v}{\partial \mu} - \nabla^2\left(\frac{U^2+V^2}{2(1-\mu^2)} + \Phi + T_R\ln(p_s)\right) - \frac{D}{\tau_F} + K(-1)^{p_d/2}\nabla^{p_d}D \\
\mathfrak{F}_u &= V\zeta - \dot{\sigma}\frac{\partial U}{\partial \sigma} - T_A\frac{\partial \ln p_s}{\partial \lambda}, \qquad \text{and} \qquad \mathfrak{F}_v = -U\zeta - \dot{\sigma}\frac{\partial V}{\partial \sigma} - T_A\left(1-\mu^2\right)\frac{\partial \ln p_s}{\partial \mu} \\
\frac{\partial T_A}{\partial t} &= \frac{1}{1-\mu^2}\frac{\partial (UT_A)}{\partial \lambda} - \frac{\partial (VT_A)}{\partial \mu} + DT_A - \dot{\sigma}\frac{\partial T}{\partial \sigma} + \frac{\kappa T\omega}{p} + \frac{T_E - T}{\tau_E} + K(-1)^{p_d/2}\nabla^{p_d}T_A \\
\frac{\partial(\ln p_s)}{\partial t} &= -\frac{U}{1-\mu^2}\frac{\partial(\ln p_s)}{\partial \lambda} - V\frac{\partial(\ln p_s)}{\partial \mu} - D - \frac{\partial \dot{\sigma}}{\partial \sigma} \\
\frac{\partial \Phi}{\partial(\ln \sigma)} &= -T \\
U &= -\left(1-\mu^2\right)\frac{\partial \Psi}{\partial \mu} + \frac{\partial \alpha}{\partial \lambda}, \qquad \text{and} \qquad V = \frac{\partial \Psi}{\partial \lambda} + \left(1-\mu^2\right)\frac{\partial \alpha}{\partial \mu} \\
\zeta &= 2\mu + \nabla^2\Psi, \qquad \text{and} \qquad D = \nabla^2\alpha.
\end{aligned} \tag{A.1}
$$

Here, ζ is the absolute vorticity, D is the horizontal divergence, λ is the longitude, ϕ is the latitude, μ is $\sin\phi$, Φ is the geopotential, τ_F is the time scale for Rayleigh friction, K is the coefficient for diffusion which is dependent on the diffusion time scale, p_d is an even number that fixes the order of diffusion, U and V is the velocity along the longitude and latitude times $\cos(\phi)$, the temperature is given by $T = T_R(\sigma) + T_A$, where T_R is a reference temperature and T_A is the temperature anomaly, p is pressure, p_s is the surface pressure, σ is equal to $\frac{p}{p_s}$, ω is the vertical velocity, T_E is the temperature pattern used for Newtonian cooling, τ_E is the time scale of Newtonian cooling, Ψ is the streamfunction, and α is the velocity potential.

[1] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, K. Yelick, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Keckler, D. Klein, P. Kogge, R. S. Williams, K. Yelick, Exascale computing study: Technology challenges in achieving exascale systems Peter Kogge, Editor & Study Lead (2008).

[2] K. Palem, Energy aware computing through probabilistic switching: a study of limits, Computers, IEEE Transactions on 54 (9) (2005) 1123 – 1137.

[3] A. Kahng, S. Kang, R. Kumar, J. Sartori, Slack redistribution for graceful degradation under voltage overscaling, in: Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific, 2010, pp. 825 –831.

[4] A. B. Kahng, S. Kang, R. Kumar, J. Sartori, Recovery-driven design: A power minimization methodology for error-tolerant processor modules (2010).

[5] ITRS, International technology roadmap for semiconductors - design, 2007.

[6] J. Sartori, J. Sloan, R. Kumar, Stochastic computing: Embracing errors in architecture and design of processors and applications, in: Compilers, Architectures and Synthesis for Embedded Systems (CASES), 2011 Proceedings of the 14th International Conference on, 2011, pp. 135 –144.

[7] A. Lingamneni, K. K. Muntimadugu, C. Enz, R. M. Karp, K. V. Palem, C. Piguet, Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling, in: Proceedings of the 9th conference on Computing Frontiers, CF '12, ACM, New York, NY, USA, 2012, pp. 3–12.

[8] J. Sartori, R. Kumar, Architecting processors to allow voltage/reliability tradeoffs., in: R. K. Gupta, V. J. Mooney (Eds.), CASES, ACM, 2011, pp. 115–124.

[9] J. Shukla, T. Palmer, R. Hagedorn, B. Hoskins, J. Kinter, J. Marotzke, M. Miller, J. Slingo, Toward a new generation of world climate research and computing facilities, Bulletin of the American Meteorological Society 91 (10) (2010) 1407–1412.

[10] T. N. Palmer, A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic

parametrization in weather and climate prediction models, Quarterly Journal of the Royal Meteorological Society 127 (572) (2001) 279–304.
[11] T. N. Palmer, Towards the probabilistic earth-system simulator: a vision for the future of climate and weather prediction, Quarterly Journal of the Royal Meteorological Society 138 (665) (2012) 841–861.
[12] T. N. Palmer, R. Buizza, F. Doblas-Reyes, T. Jung, M. Leutbecher, G. Shutts, M. Steinheimer, M. Weisheimer, Stochastic parametrization and model uncertainty, ECMWF Technical Memoranda (598).
[13] P. W. Wedi, M. Hamrud, G. Mozdzynski, A fast spherical harmonics transform for global NWP and climate models, Mon. Wea. Rev. accepted.
[14] I. M. Held, M. J. Suarez, A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models, Bull. Amer. Meteor. Soc. 75 (1994) 1825–1830.
[15] K.-H. Huang, J. Abraham, Algorithm-based fault tolerance for matrix operations, Computers, IEEE Transactions on C-33 (6) (1984) 518–528. doi:10.1109/TC.1984.1676475.
[16] N. Oh, P. Shirvani, E. McCluskey, Controlflow checking by software signatures, Reliability, IEEE Transactions on 51 (1) (2002) 111 – 122.
[17] N. Nakka, Z. Kalbarczyk, R. Iyer, J. Xu, An architectural framework for providing reliability and security support, in: Dependable Systems and Networks, 2004 International Conference on, 2004, pp. 585–594. doi:10.1109/DSN.2004.1311929.
[18] K. S. Yim, C. Pham, M. Saleheen, Z. Kalbarczyk, R. Iyer, Hauberk: Lightweight silent data corruption error detector for gpgpu, Parallel and Distributed Processing Symposium, International 0 (2011) 287–300. doi:http://doi.ieeecomputersociety.org/10.1109/IPDPS.2011.36.
[19] J. Sloan, R. Kumar, G. Bronevetsky, Algorithmic approaches to low overhead fault detection for sparse linear algebra, in: Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on, 2012, pp. 1–12. doi:10.1109/DSN.2012.6263938.
[20] C. T. Kong, Study of voltage and process variations impact on the path delays of arithmetic units, Master's thesis, University of Illinois at Urbana-Champaign (2008).
[21] J. Sloan, D. Kesler, R. Kumar, A. Rahimi, A numerical optimization-based methodology for application robustification: Transforming applications for error tolerance, in: Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on, 2010, pp. 161 –170.
[22] E. N. Lorenz, Predictability—a problem partly solved, in: T. N. Palmer, R. Hagedorn (Eds.), Predictability of Weather and Climate, Cambridge University Press, 2006, Ch. 3, pp. 40–58.
[23] S. Herrera, J. Fernández, M. A. Rodríguez, J. M. Gutiérrez, Spatio-temporal error growth in the multi-scale Lorenz 96 model, Nonlinear Processes in Geophysics 17 (2010) 329–337.
[24] A. Karimi, M. R. Paul, Extensive chaos in the Lorenz-96 model, Chaos 20 (4).
[25] M. Leutbecher, A data assimilation tutorial based on the lorenz-95 system, Tech. rep., European Centre for Medium-Range Weather Forecasting (2009).
[26] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. J. Patil, J. A. Yorke, A local ensemble kalman filter for atmospheric data assimilation, Tellus 56 (5) (2004) 415–428.
[27] I. Fatkullin, E. Vanden-Eijnden, A computational strategy for multiscale systems with applications to Lorenz 96 model, Journal of Computational Physics 200 (2) (2004) 605–638.
[28] F. Kwasniok, Data-based stochastic subgrid-scale parametrization: an approach using cluster-weighted modelling, Philosophical Transactions of The Royal Society A 370 (2012) 1061–1086.
[29] T. P. Sapsis, A. J. Majda, A statistically accurate modified quasilinear gaussian closure for uncertainty quantification in turbulent dynamical systems, Physica D: Nonlinear Phenomena 252 (2013) 34–45.
[30] D. S. Wilks, Effects of stochastic parameterizations in the Lorenz '96 system, Quarterly Journal of the Royal Meteorological Society 131 (606) (2005) 389–407.
[31] H. M. Arnold, I. M. Moroz, T. N. Palmer, Stochastic parametrizations and model uncertainty in the Lorenz '96 system, Philosophical Transactions of The Royal Society A 371.
[32] B. J. Hoskins, A. J. Simmons, A multi-layer spectral model and the semi-implicit method, Quarterly Journal of the Royal Meteorological Society 101 (429) (1975) 637–655.
[33] A. J. Simmons, D. M. Burridge, An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates, Mon. Wea. Rev. (109) (1981) 758 – 766.
[34] M. Blackburn, Program description for the multi-level global spectral model.
[35] I. N. James, J. P. Dodd, A simplified global circulation model.
[36] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex fourier series, Mathematics of Computation 19 (90) (1965) pp. 297–301.
[37] M. Hortal, A. J. Simmons, Use of reduced gaussian grids in spectral models, Mon. Wea. Rev. 119 (1991) 1057–1074.
[38] D. L. Williamson, M. Blackburn, K. Nakajima, W. Ohfuchi, Y. O. Takahashi, Y. Y. Hayashi, H. Nakamura, M. Ishiwatari, J. McGregor, H. Borth, V. Wirth, H. Frank, P. Bechtold, N. P. Wedi, H. Tomita, M. Satoh, M. Zhao, I. M.

Held, M. J. Suarez, M. I. Lee, M. Watanabe, M. Kimoto, Y. Liu, Z. Wang, A. Molod, K. Rajendran, A. Kitoh, R. A. Stratton, The Aqua-Planet Experiment (APE): Response to changed meridional SST profile, Journal Of The Meteorological Society Of Japan.

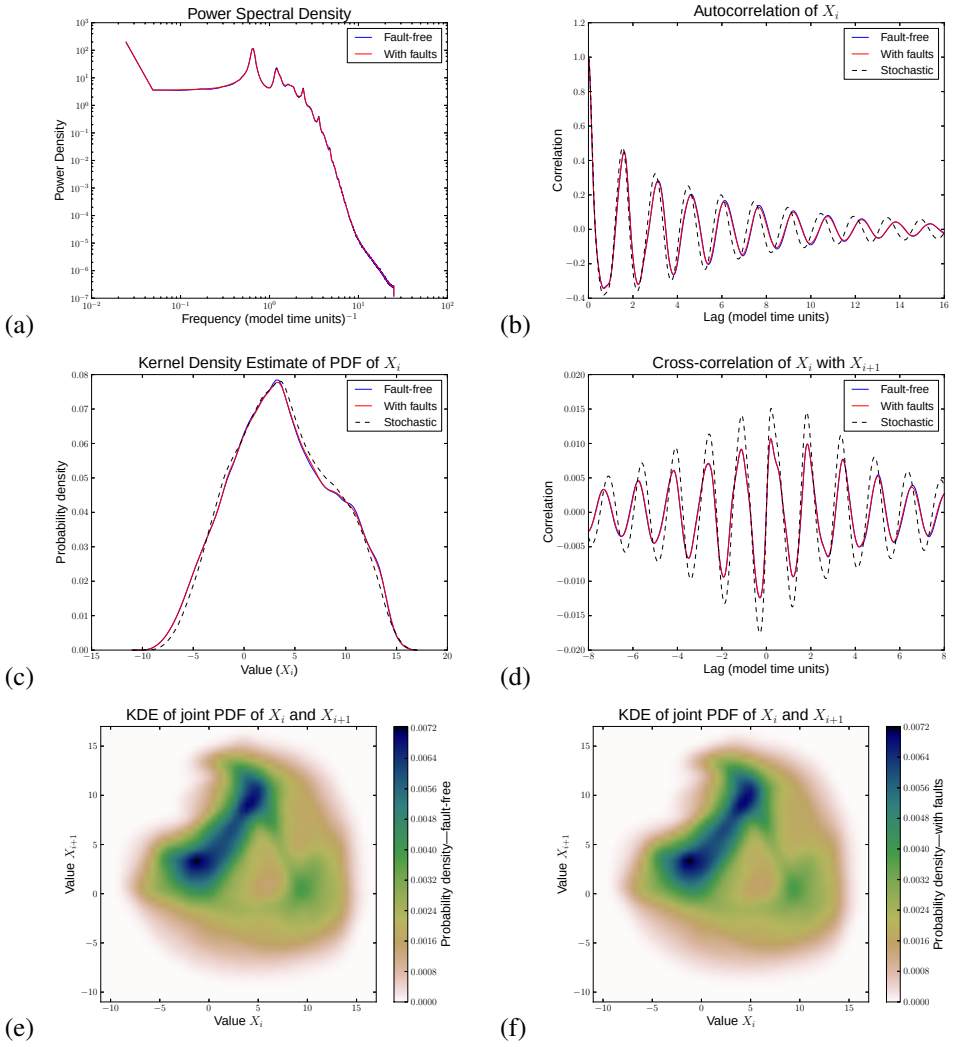

Figure 2: Faulty results from the $c = 10$ case with 20% fault rate. Fault-free lines are in blue, ensemble average faulty results in red, reference stochastic results are the black dashed lines. Fault-free and faulty lines nearly coincide for (a)–(d). Some slight de-correlation is seen in the autocorrelation (b) and cross-correlation (d), but this is still small after 5 model time units (25 atmospheric days). The joint probability density estimates of neighbouring X variables are shown for fault-free (e) and faulty (f) runs at the end.

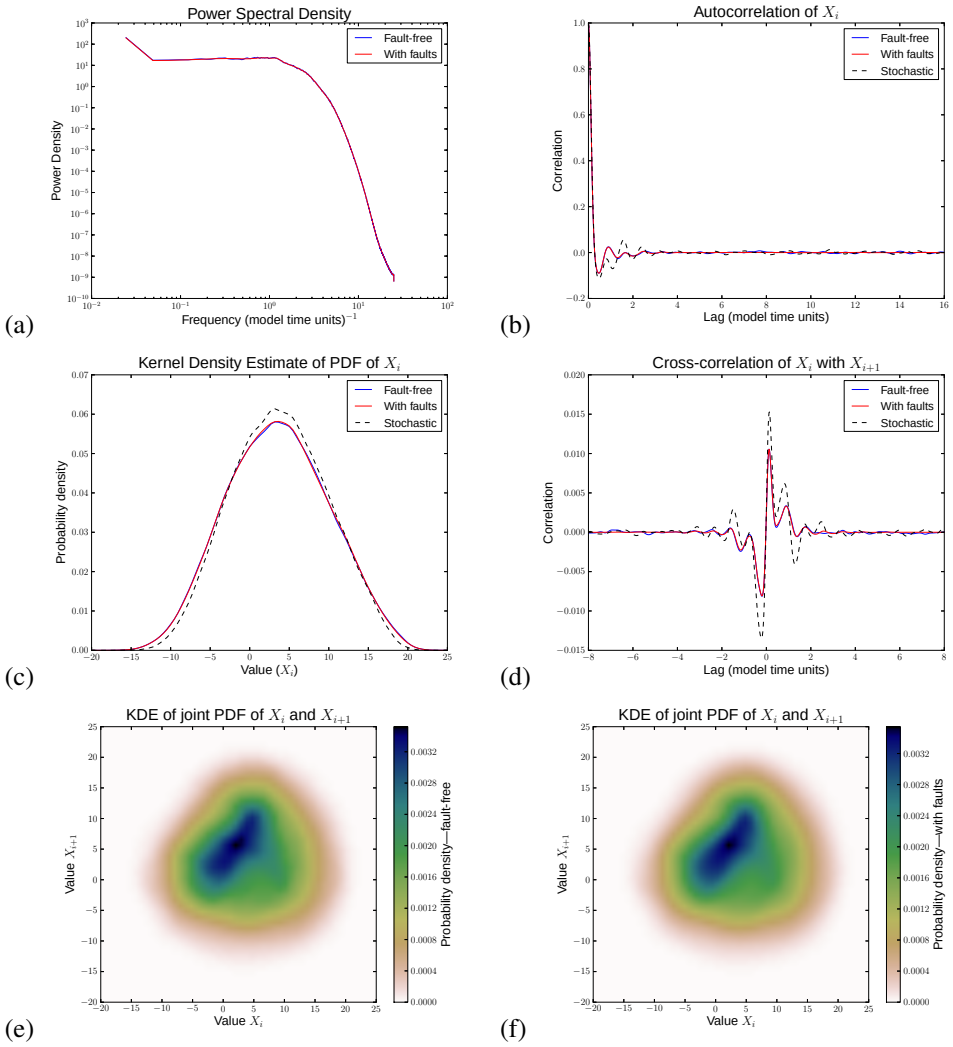

Figure 3: Faulty results from the $c = 4$ case with 20% fault rate. The lower time-scale separation changes the dynamics significantly from Figure 2, but the faulty simulations still remain remarkably consistent with the fault-free run, and significantly better than the stochastic run.

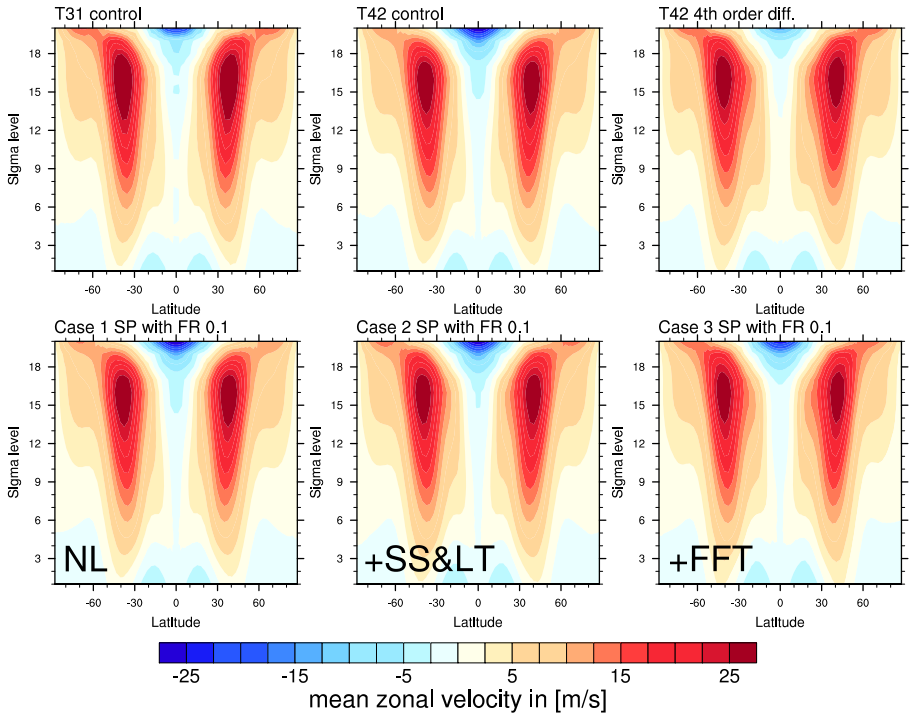

Figure 4: Mean zonal velocity for control runs (case 0) at T31 and T42 and with 4th order diffusion (top row, left–right) and for cases 1, 2 and 3 (bottom row, left–right) using an emulated stochastic processor and 10% fault rate.

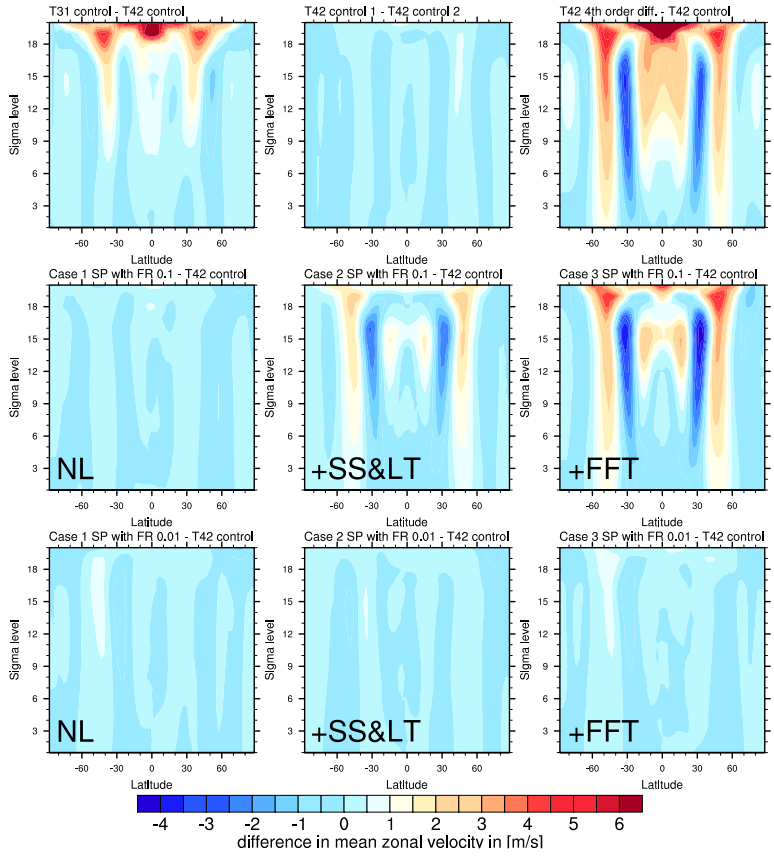

Figure 5: Differences in mean zonal velocity between the T42 control run and other simulations. The top row shows control runs at T31 and T42, and a T42 run with 4th order diffusion. The second row has differences with cases 1, 2 and 3 for a fault rate of 10%. The bottom row is the same, at a fault rate of 1%.

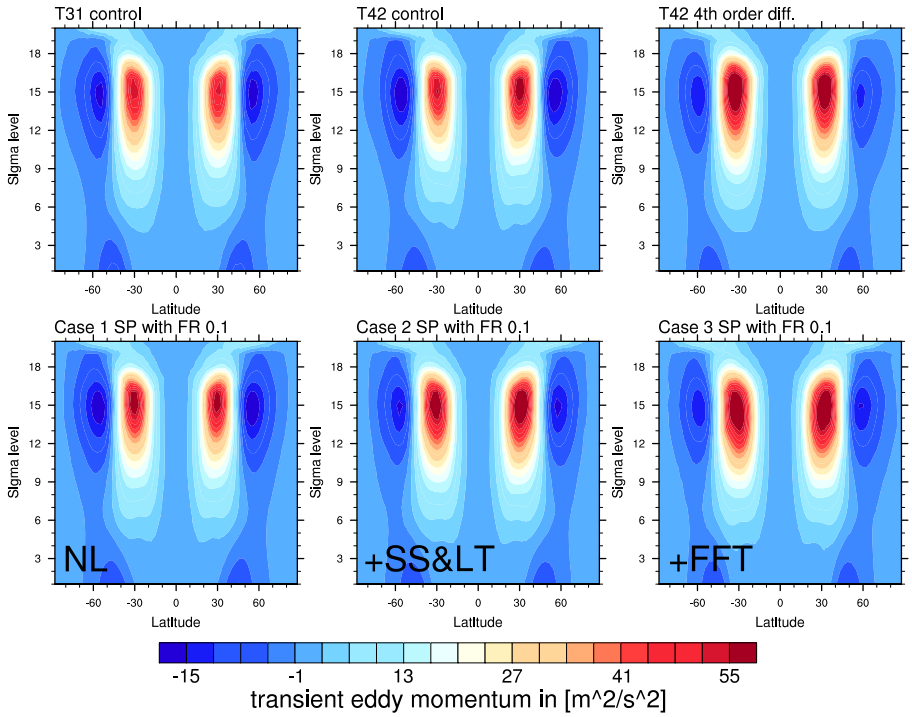

Figure 6: Transient eddy-momentum for control runs at T31 and T42 and with 4th order diffusion (top row, left–right) and for cases 1, 2 and 3 (bottom row, left–right) using an emulated stochastic processor and 10% fault rate.

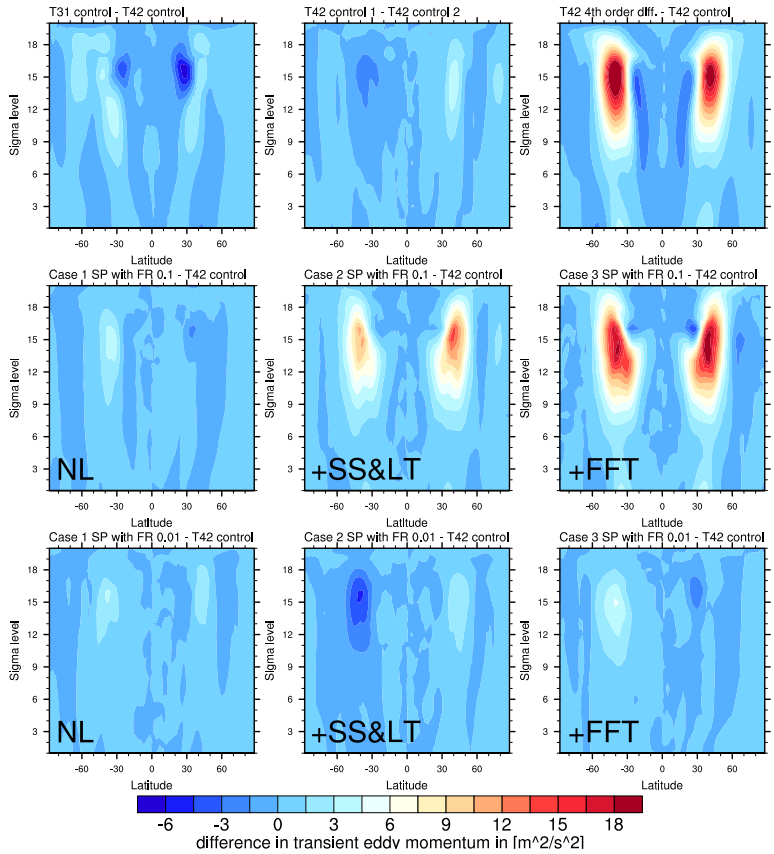

Figure 7: Differences in transient eddy-momentum between the T42 control run and other simulations. The top row shows control runs at T31 and T42, and a T42 run with 4th order diffusion. The second row has differences with cases 1, 2 and 3 for a fault rate of 10%. The bottom row is the same, at a fault rate of 1%.

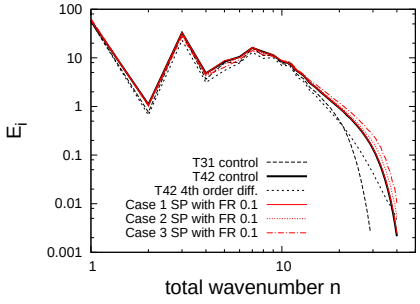

Figure 8: Daily mean of the energy spectra for cases 1, 2 and 3 with an emulated stochastic processor at a fault rate of 10% (red; solid, dashed and dotted) and control runs at T31 and T42 and with 4th order diffusion (black; dashed, solid and dotted resp.).

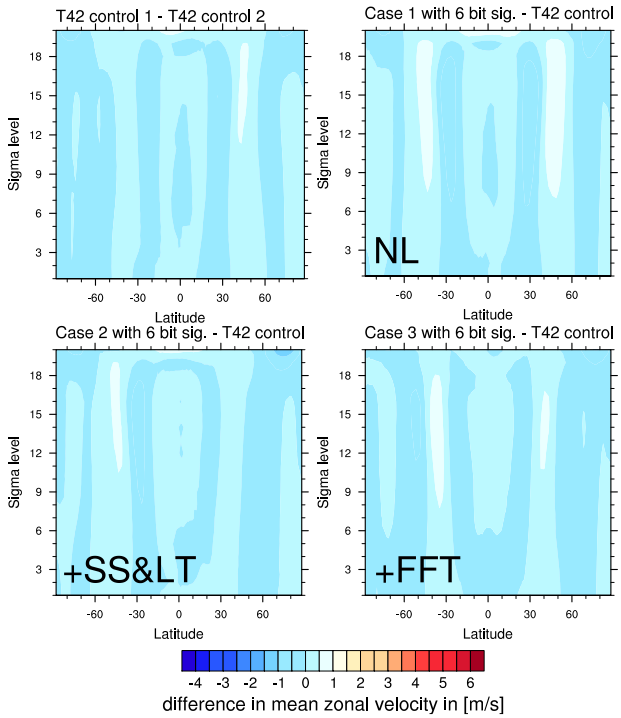

Figure 9: Differences in mean zonal velocity between the T42 control run and other simulations. Top left is the difference between two T42 control runs, top right is case 1. Case 2 and 3 are on the bottom row. All perturbed cases use the emulated 6-bit significand. To allow comparisons, the same colour scheme is used as in Figure 5.

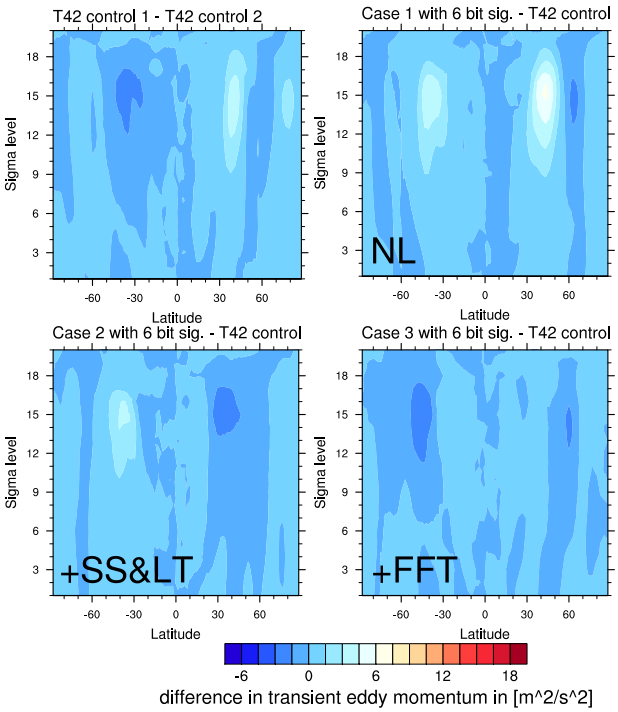

Figure 10: Differences in transient eddy-momentum between the T42 control run and other simulations. Top left is the difference between two T42 control runs, top right is case 1. Case 2 and 3 are on the bottom row. All perturbed cases use the emulated 6-bit significand. The colour scheme from Figure 7 is re-used to allow direct comparisons.

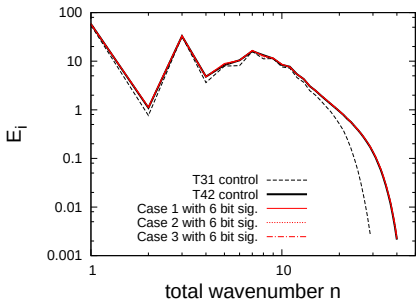

Figure 11: Daily mean of the energy spectra for cases 1, 2 and 3 for a 6 bit significand (red; solid, dashed and dotted) and control runs at T31 and T42 (black; dashed and solid resp.). The spectra of cases 1, 2 and 3 lie on-top of the spectra of the T42 control simulation.