Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2018MS001341

Key Points:
- Lowering precision could accelerate an ensemble Kalman filter
- The level of precision used should fit the level of model error
- We perform tests with a spectral dynamical core

Correspondence to:
S. Hatfield,
samuel.hatfield@physics.ox.ac.uk

Citation:
Hatfield, S. E., Düben, P. D., Chantry, M., Kondo, K., Miyoshi, T., & Palmer, T. N. (2018). Choosing the optimal numerical precision for data assimilation in the presence of model error. *Journal of Advances in Modeling Earth Systems*, *10*, 2177–2191. https://doi.org/10.1029/2018MS001341

Received 10 APR 2018
Accepted 7 AUG 2018
Accepted article online 13 AUG 2018
Published online 6 SEP 2018

Choosing the Optimal Numerical Precision for Data Assimilation in the Presence of Model Error

Sam Hatfield[1], Peter Düben[2], Matthew Chantry[1], Keiichi Kondo[3], Takemasa Miyoshi[4], and Tim Palmer[1]

[1] Atmospheric, Oceanic and Planetary Physics, University of Oxford, Oxford, UK, [2] European Centre for Medium Range Weather Forecasts, Reading, UK, [3] Japan Meteorological Agency, Meteorological Research Institute, Tsukuba, Japan, [4] RIKEN Center for Computational Science, Kobe, Japan

Abstract The use of reduced numerical precision within an atmospheric data assimilation system is investigated. An atmospheric model with a spectral dynamical core is used to generate synthetic observations, which are then assimilated back into the same model using an ensemble Kalman filter. The effect on the analysis error of reducing precision from 64 bits to only 22 bits is measured and found to depend strongly on the degree of model uncertainty within the system. When the model used to generate the observations is identical to the model used to assimilate observations, the reduced-precision results suffer substantially. However, when model error is introduced by changing the diffusion scheme in the assimilation model or by using a higher-resolution model to generate observations, the difference in analysis quality between the two levels of precision is almost eliminated. Lower-precision arithmetic has a lower computational cost, so lowering precision could free up computational resources in operational data assimilation and allow an increase in ensemble size or grid resolution.

Plain Language Summary In order to produce a weather forecast, we must have a good estimate of the current state of the atmosphere. We can observe the atmosphere using satellites and other instruments, but observations alone do not tell the whole picture. We must combine observational data with computational models in order to estimate the atmospheric state comprehensively. This process is known as data assimilation. Data assimilation is a very computationally expensive process as it requires the atmospheric model to be run many times over. This paper proposes a novel method to reduce the computational cost of data assimilation: reduced-precision calculations. Reducing the precision of the calculations inside the atmospheric model might be expected to degrade the data assimilation process. However, our atmospheric models are inherently imperfect because they do not capture all of the important scales of atmospheric motion. Therefore, the degradation from reducing precision is not actually significant when compared with this unavoidable error. The computational savings that we make by reducing precision could be reinvested to actually improve the data assimilation system and therefore the skill of weather forecasts. For example, we could run more simulations (i.e., use a larger *ensemble*) for no extra cost.

1. Introduction

Skillful forecasting of weather is highly dependent on the generation of accurate initial conditions. This is accomplished through *data assimilation*, a statistical procedure that combines observations with an atmospheric state estimate derived from a model to produce an *analysis* that can be used as the initial weather state. One example of a data assimilation scheme is the ensemble Kalman filter (EnKF) which uses an ensemble to represent the probability distribution of the weather state and an atmospheric model to propagate this distribution through time in a Monte Carlo fashion.

The quality of the analysis produced by an EnKF is directly related to the ensemble size. Sample distributions derived from an ensemble are less prone to sampling error when the ensemble size is large (Miyoshi et al., 2014) and are therefore less dependent on rather ad hoc covariance inflation (e.g., Whitaker & Hamill, 2012) and localization (e.g., Hamill et al., 2001) techniques. For example, among a range of possible operational upgrades to the EnKF at the Meteorological Service of Canada, the first operationally implemented EnKF (Houtekamer & Mitchell, 2005), an ensemble size increase delivered the largest analysis error reduction per computational cost increase, along with an increase in the horizontal resolution (Houtekamer, Deng, et al.,

2014; the operational system has 256 members as of November 2014; Houtekamer & Zhang, 2016). Further, Kondo and Miyoshi (2016) demonstrated that localization can be removed entirely for an ensemble of 10,240 members (using the same model considered in this article). Such a data assimilation system can be expected to have superior balance properties as well (Greybush et al., 2011), which will also contribute toward a better analysis and more skillful forecasts.

Increasing the ensemble size is beneficial, but it comes at a cost. The cost of the background ensemble forecast in the EnKF and the application of the observation operator to this ensemble should both scale linearly with the ensemble size, as each member is independent. The cost scaling of the EnKF update algorithm with ensemble size depends on the details of the algorithm; the Canadian EnKF algorithm, for example, scales linearly (Houtekamer, He, & Mitchell, 2014), whereas the local ensemble transform Kalman filter (LETKF) scales between quadratically and cubically, depending on the number of observations and grid points(Hunt et al., 2007). Nevertheless, given that the cost of the ensemble forecast dominates over other costs, we can assume that the cost of an operational EnKF will scale approximately linearly (Houtekamer & Zhang, 2016).

Recently, several articles have proposed the idea of lowering the numerical precision of the model to reduce the cost. Lower-precision floating-point numbers have a smaller footprint in memory and therefore can be transferred from memory to the CPU at a greater rate (an operation which is the main bottleneck in most weather and climate simulations). Lower-precision numbers can also take advantage of vectorized operations, which promise, for example, a doubling of floating-point operation rate for single-precision arithmetic, with respect to double-precision. Additionally, the lower the precision, the more data can be fit into cache or a process-to-process communication (e.g, via Message Passing Interface). This raises the possibility of lowering precision to reduce the computational cost of the EnKF and reinvesting any saved computational resources to increase the ensemble size, model resolution, or complexity. Váňa et al. (2017) ran the European Centre for Medium-Range Weather Forecasts model at single-precision and demonstrated a 40% computational cost saving. They then compared a single-precision ensemble forecast using 50 members with an equivalent cost double-precision forecast using 34 members and found that the single-precision forecast with a larger ensemble had a higher skill. Similarly, Nakano et al. (2018) considered a reduction of precision in an atmospheric model double-precision to single-precision though only in the dynamical core. They found that the error introduced by reducing precision was much smaller than the differences between independent dynamical cores, as determined through a baroclinic wave test case. Düben et al. (2014) and Düben and Palmer (2014) both reduced-precision in an intermediate complexity atmospheric model using a manual floating-point truncation procedure similar to ours and found that the model could produce accurate forecasts and climatologies even when using a precision substantially lower than single-precision. They did not consider data assimilation, however.

For data assimilation, Hatfield et al. (2018) proposed to reinvest the saved computational resources from lowering precision into the ensemble size and demonstrated improved analyses and forecasts for an EnKF with a toy model. Here we will extend this latter study to consider an intermediate complexity general circulation model and discuss aspects relating to model error. We will test this idea using a so-called observing system simulation experiment, in which a *nature run* is used to generate synthetic observations. These observations are then assimilated into the model using the EnKF, and the resulting analyses can be verified with respect to the perfectly known truth. Often a perfect model assumption is adopted, whereby the nature run and assimilation are performed with the same model. We will demonstrate that this is inappropriate for estimating the impact of reducing precision, as it unrealistically penalizes the reduced-precision model, and consider ways to introduce model error into an observing system simulation experiment. Note that in this study, we do not consider the computational cost saving from reducing precision. Estimating this saving is not easy as it depends on many factors such as the computing architecture, the compiler, and the *vectorizability* of the code. However, for a memory-bound application such as a weather or climate model we expect a speedup linearly proportional to the number of bits eliminated when reducing precision. Reducing precision may even allow a greater than linear speedup if more data can be fit into cache.

Our setup is similar to some earlier studies. Hamill and Whitaker (2005) also used a high-resolution and low-resolution configuration of the same model, to compare different covariance inflation schemes (additive and multiplicative) in the presence of model error due to unresolved scales. Miyoshi (2011) considered the Simplified Parametrizations, primitivE-Equation DYnamics (SPEEDY) model for imperfect model studies, though, unlike us, they modified both the diffusion and the convection scheme. Hatfield et al. (2018) con-

sidered a reduction of precision in a toy data assimilation system, and Miyoshi et al. (2016) briefly compared double-precision and single-precision for the dynamical model in a convective scale data assimilation system. However, to the best of our knowledge, our study is the first to provide a detailed analysis of a precision reduction in an atmospheric data assimilation system.

In section 2 we describe the experimental setup, consisting of the forecast model and the assimilation system, in section 3 we give the results from the perfect model experiments, in section 4 we introduce model error, and in section 5 we present some discussion and conclusions.

2. Model and Assimilation Algorithm

2.1. The SPEEDY Model

2.1.1. Model Description

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 model with sigma levels in the vertical direction with vorticity and divergence as the prognostic variables for the wind. 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. Horizontal diffusion is performed in spectral space, for example, as the Laplacian (and higher powers of the Laplacian) have a particularly simple form there. This is ultimately because the spherical harmonics are eigenfunctions of the Laplacian in spherical coordinates (Krishnamurti et al., 1998). 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. These nonlinear products are instead performed in grid point space, and the result is transformed back to spectral space, where the time stepping occurs. Physical parametrization schemes are also difficult to formulate in spectral space, so the contribution to the field tendencies from physics are computed in grid point space as well.

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 T*N*, 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 time step of 40 min, and the T39 configuration used a 120×60 grid with a time step of 20 min. There are eight sigma levels for both the T30 and T39 configurations, situated at $\sigma = 0.95, 0.835, 0.685, 0.51, 0.34, 0.2, 0.095$, 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 test bed. 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 localization and reveals statistically significant correlations across the globe, even over only a 6-hr 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 EnKF. SPEEDY has even been used for simple coupled data assimilation experiments with the NEMO ocean model (Sluka et al., 2016).

2.1.2. Reducing Precision in the Forecast Model

For investigating precision, we developed a reduced-precision version of SPEEDY. In this model, all real variables in the main loop of the program are replaced with a FORTRAN derived type which allows one to perform floating-point operations at an arbitrary precision. For example, the emulator supports IEEE754 compliant *half-precision* numbers, also known as FP16 as they require 16 bits in memory. However, the two components of a floating-point number (excluding the 1 bit reserved for the sign), the significand and the exponent, can be specified arbitrarily. The significand of a floating-point number stores a number between 1 and 2 which represents the leading digits of the corresponding real number. This is what is usually referred to by *precision*. The exponent stores a multiplier to the significand which scales it up by a power of 2 and therefore represents the order of magnitude of the corresponding real number. Reducing the size of the significand increases the rounding error that occurs during arithmetic operations, as fewer numbers within the range of allowable

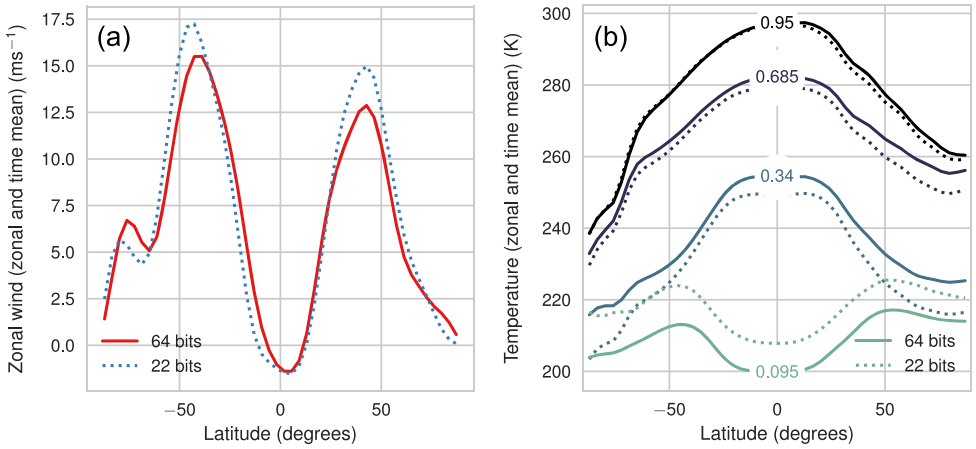

Figure 1. (a) The zonal mean time mean zonal wind at $\sigma = 0.51$ computed from climatologies of the 22-bit model and the 64-bit model. (b) The zonal mean time mean temperature at selected vertical levels of the model computed from climatologies of the 22-bit model and the 64-bit model. The sigma level is superimposed on each pair of lines.

numbers can be represented. Reducing the size of the exponent reduces the range of allowable numbers and therefore makes overflow and underflow errors more common. We found that SPEEDY crashed when using half-precision floating-point arithmetic, as many variables underflowed or overflowed. Therefore, we do not consider a reduction of the exponent below 11 bits (double-precision) in this study. The FORTRAN-derived type used for reducing precision came originally from the FORTRAN module rpe (Dawson & Düben, 2017), though we modified it to allow complex number arithmetic as well as real number arithmetic. The emulator was not activated during model initialization (in which, e.g., 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 6-hr integration used to generate the background ensemble within the LETKF.

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 artifacts are clearly visible in the prognostic fields. We found that 10 significand bits were required to mostly eliminate

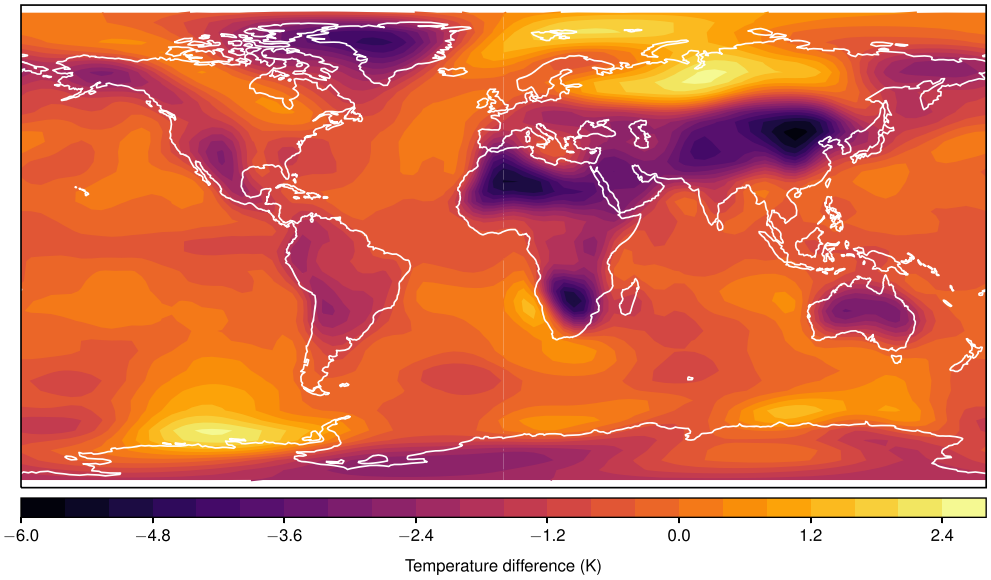

Figure 2. The time mean difference (bias) between the lowest level ($\sigma = 0.95$) temperature of SPEEDY with 22 and 64 bits. The temperature of the 64-bit model was subtracted from the temperature of the 22-bit model. SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

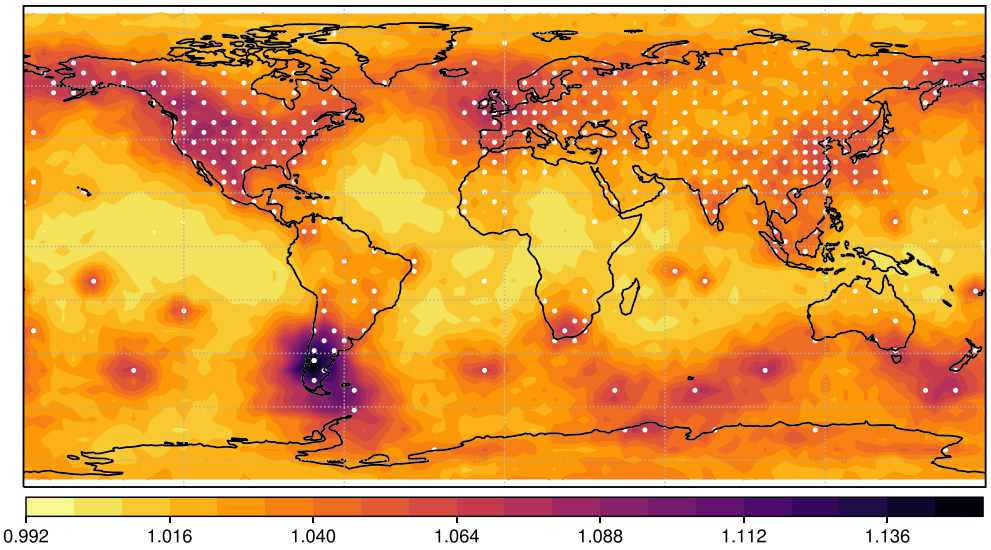

Figure 3. The spatial distribution of the RTPP inflation factor, f, for temperature at the lowest model level. The locations of observations are superimposed as white dots. RTPP = relaxation-to-prior perturbations.

these artifacts 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 1 illustrates some differences in the zonal mean time mean climatologies of the 22- and 64-bit models. Figure 1a shows the zonal wind at $\sigma = 0.51$ (approximately 500 hPa). Jets are clearly visible in both hemispheres, though the 22-bit model jets are around 20% stronger. Figure 1b shows the temperature at every other model level. The temperature with 22 bits is systematically lower than for 64 bits at the bottom six 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 illustrated in Figure 2 to be concentrated over the continents. Evidently, the 22-bit model has its own climatology and significant biases with respect to the 64-bit model. However, it remains to be seen whether the differences between the 22-bit model and 64-bit model are outside of the envelope of uncertainty provided by, for example, model error and observation error in a data assimilation context.

2.2. Data Assimilation Setup

To perform assimilation, we used the LETKF (Hunt et al., 2007). This belongs to the class of *deterministic, square root* filters (Houtekamer & Zhang, 2016) and builds on the local ensemble Kalman filter (Ott et al., 2004) and the ensemble transform Kalman filter (Bishop et al., 2001). The LETKF computes an independent analysis at each grid point. Each grid point can be assigned to, for example, a different MPI process, and so in theory the algorithm is perfectly parallelizable. However, note that a reduction is required after every process has finished in order to produce the global analysis and the scalability may also be limited by the difficulty of balancing loads when observations are nonuniformly distributed (Hamrud et al., 2015). At each grid point, the following steps are performed:

- The ensemble perturbation matrix X^b is formed by subtracting the ensemble mean $\overline{\boldsymbol{x}}^b$ from each ensemble member $\{\boldsymbol{x}_i^b\}$ and using the resulting vectors as the columns of the matrix. This matrix has k columns (the number of ensemble members) and m rows (the number of prognostic variables). It can be considered as a transformation from a k-dimensional space $\tilde{S}$ to an m-dimensional space S. In $\tilde{S}$, the ith background ensemble member is represented by the k-dimensional column vector $\boldsymbol{w}_i^b$ with zeros in each element apart from the ith element which contains a 1. The corresponding ensemble member in S can be recovered through the operation $\boldsymbol{x}_i^b = \overline{\boldsymbol{x}}^b + \mathrm{X}^b \boldsymbol{w}_i^b$.
- A square root filter analysis is performed in $\tilde{S}$ instead of S. The cost function to be minimized in $\tilde{S}$ is

$$\tilde{J}(\boldsymbol{w}) = \|\boldsymbol{w}\|^2_{(k-1)^{-1}\mathrm{I}} + \|\boldsymbol{y}^o - H\left(\overline{\boldsymbol{x}}^b + \mathrm{X}^b\boldsymbol{w}\right)\|^2_{\mathrm{R}}, \tag{1}$$

where I is the identity matrix, $\boldsymbol{y}^o$ is the observation vector, H is the observation operator, R is the observation error covariance matrix, and $\|\boldsymbol{a}\|_{\mathrm{A}}$ denotes the norm of the vector $\boldsymbol{a}$ with respect to the covariance matrix A

Table 1
Properties of the Synthetic Observations Derived From the Nature Run

Variable	Levels	Observation error	Number of observations
Zonal wind	All	1 m/s	3,328
Meridional wind	All	1 m/s	3,328
Temperature	All	1 K	3,328
Specific humidity	Bottom four	0.001 kg/kg	1,664
Surface pressure	N/A	1 hPa	416
			Total: 12,064

(the Mahalanobis distance). Note that in $\tilde{S}$, the background covariance matrix is simply the identity divided by $k-1$.

- The outcome of the square root filter will be a set of vectors $\{\boldsymbol{w}_i^a\}$ representing the analysis ensemble in $\tilde{S}$. The analysis ensemble in S can then be generated by computing $\boldsymbol{x}_i^a = \overline{\boldsymbol{x}}^b + \mathrm{X}^b \boldsymbol{w}_i^a$ for each member i.

We used the LETKF to assimilate synthetic observations every 6 hr, 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 specific 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 white dots in Figure 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 1. In all experiments, the model was spun-up from a global rest state over 1 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 localization. For localization we employ a standard Gaspari-Cohn Gaussian-esque correlation function of finite support (Gaspari & Cohn, 1999), with a horizontal correlation length scale of 1,000 km and a vertical length

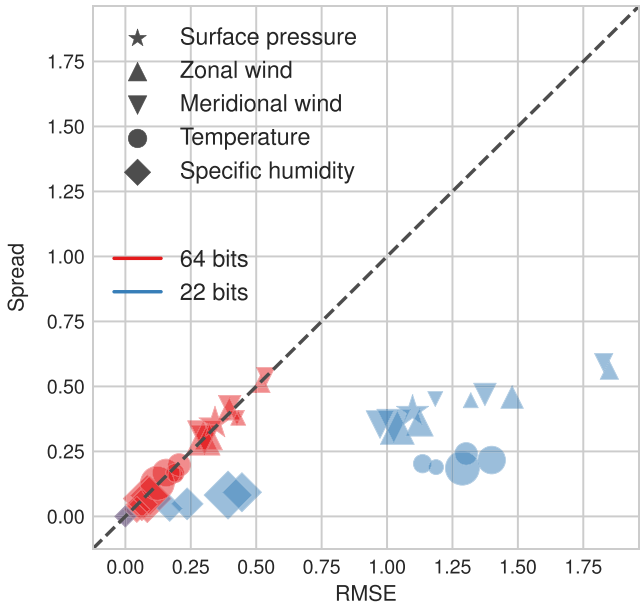

Figure 4. The analysis ensemble mean RMSE (abscissa) and spread (ordinate) of all prognostic fields at levels 0.95, 0.835, 0.51, 0.34, and 0.095, normalized by the respective observation error, for both the 64- and 22-bit configurations of SPEEDY. The larger markers correspond to the lower sigma levels and vice versa. The nature run is generated using the 64-bit configuration. RMSE = root-mean-square error; SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

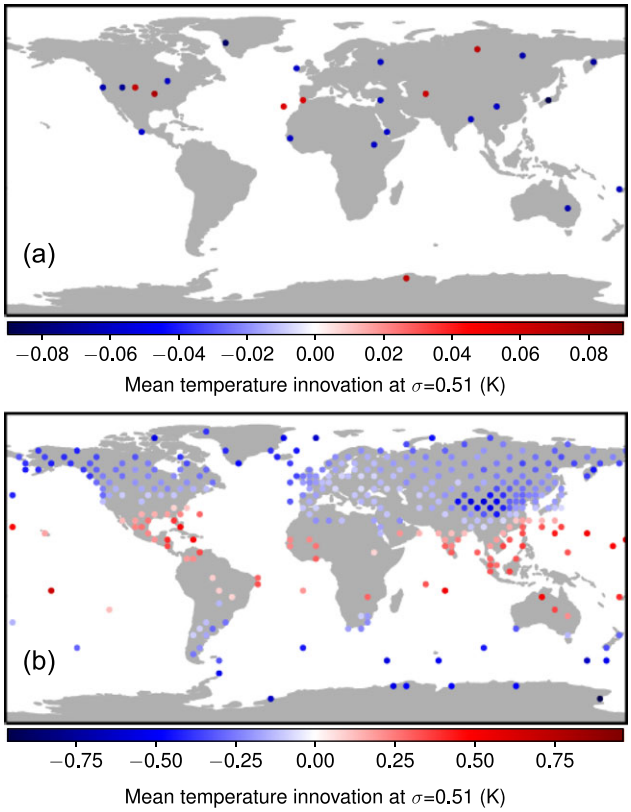

Figure 5. The temperature innovations at $\sigma = 0.51$ averaged over all assimilation cycles of the assimilation experiment shown in Figure 4 with (a) a 64-bit assimilation model and (b) a 22-bit assimilation model. Note the difference in the color scale between (a) and (b). Only statistically significant innovation biases are shown.

scale of 0.1 sigma levels. We keep these two constant throughout the paper, reasoning that the optimal localization 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, $\boldsymbol{x}_i'^b$:

$$\boldsymbol{x}_i'^{a,\mathrm{RTPP}} = (1-\alpha)\boldsymbol{x}_i'^a + \alpha\boldsymbol{x}_i'^b \tag{2}$$

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 1. The posterior spread is therefore *relaxed* toward 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:

$$\mathrm{W}^{\mathrm{RTPP}} = (1-\alpha)\mathrm{W} + \alpha\mathrm{I} \tag{3}$$

where W is the original weight matrix, $\mathrm{W}^{\mathrm{RTPP}}$ is the corrected weight matrix, and I is the identity matrix. This formulation is identical to that presented in equation (2) (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 nonuniform observation network. For a given component of the state vector, j, the degree of inflation, f, can be written as

$$f = \frac{(1-\alpha)x_{i,j}'^a + \alpha x_{i,j}'^b}{x_{i,j}'^a} = 1 - \alpha + \alpha\frac{x_{i,j}'^b}{x_{i,j}'^a}. \tag{4}$$

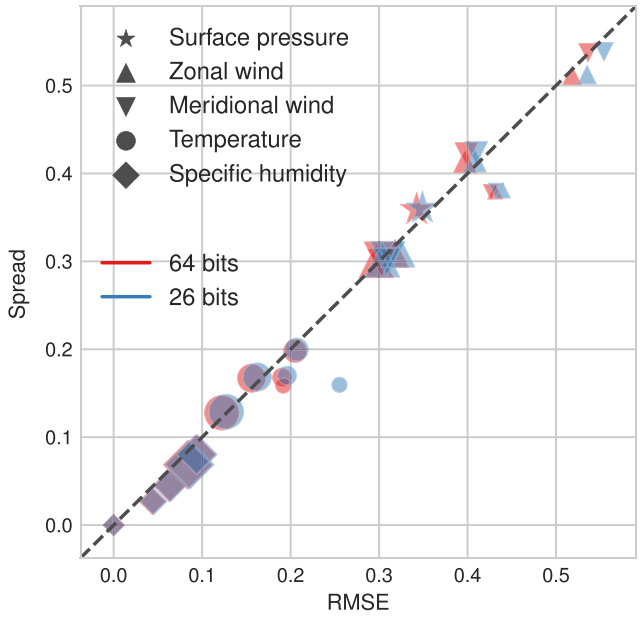

Figure 6. The analysis ensemble mean RMSE and spread of all prognostic fields, normalized by the respective observation error, for both the 64- and 26-bit configurations of SPEEDY, as in Figure 4. The nature run is generated using the 64-bit configuration. RMSE = root-mean-square error; SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

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 Hatfield et al. (2018), where f is a constant with a value slightly greater than 1.

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

$$f = \frac{(1-\alpha)x'^{a,\mathrm{RTPP}}_{i,j}}{x'^{a,\mathrm{RTPP}}_{i,j} - \alpha x'^{b}_{i,j}}. \tag{5}$$

Using equation (5), 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 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.

3. Perfect Model Results

First, 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- 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 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 normalized by the respective observation errors given in Table 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

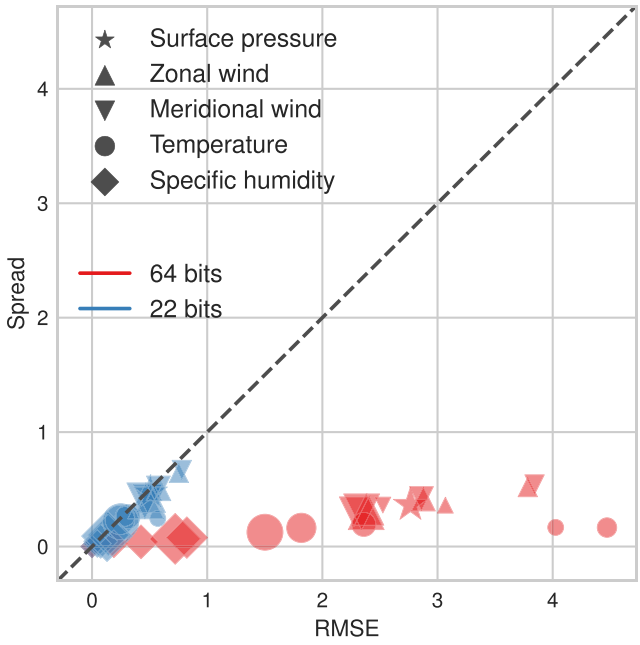

Figure 7. The analysis ensemble mean RMSE and spread of all prognostic fields, normalized by the respective observation error, for both the 64- and 22-bit configurations of SPEEDY, as in Figure 4. The nature run is generated using the 22-bit configuration. RMSE = root-mean-square error; SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

both fields before computing the RMSE did not significantly alter these results or any of the results in section 4 either.

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 nonzero 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 0 where the standard error is computed from the sample standard deviation of the innovation time series. Figure 5 shows the statistically significant mean temperature innovation biases at $\sigma = 0.51$ (approximately 500 hPa) for the same experiment as in Figure 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 3 times higher than when using the 64-bit configuration. Figure 4 also indicates that while the 64-bit configuration has a good match of spread to RMSE, the 22-bit configuration suffers from substantial overdispersiveness. 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 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 6 shows the same result as in Figure 4; only a 26-bit configuration of SPEEDY (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-standardized 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

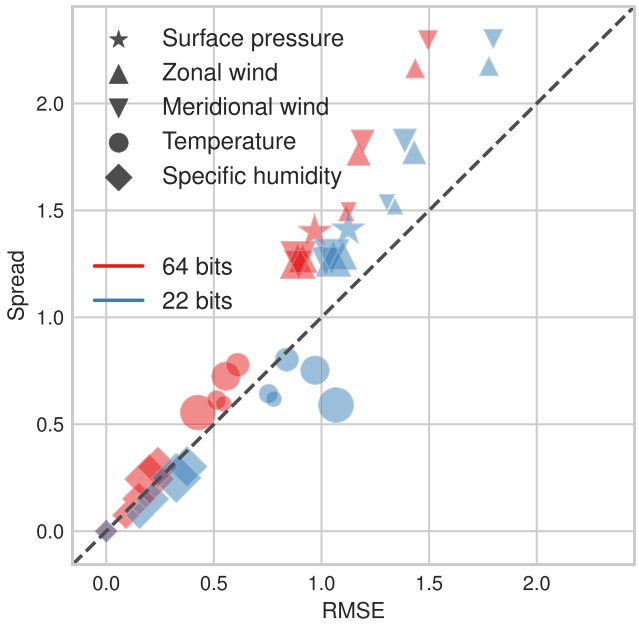

Figure 8. The analysis ensemble mean RMSE and spread of all prognostic fields, normalized by the respective observation error, for both the 64- and 22-bit configurations of SPEEDY, as in Figure 4. The nature run is generated as in Figure 4, but the assimilation model has a modified diffusion scheme. RMSE = root-mean-square error; SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

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 significand width from 52 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- 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 7. Compared with Figure 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. Imperfect Model Results

4.1. 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 to 4.8 hr. We also doubled the maximum damping time for extra diffusion in the stratosphere from 12 to 24 hr. 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.

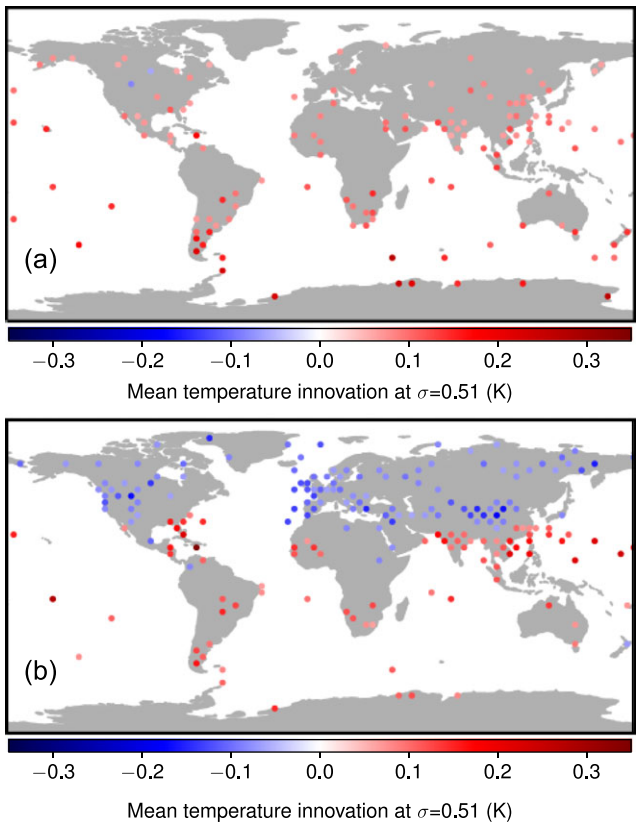

Figure 9. The temperature innovations at $\sigma = 0.51$ averaged over all assimilation cycles of the assimilation experiment shown in Figure 8 with (a) a 64-bit assimilation model and (b) a 22-bit assimilation model. The nature run is generated as in Figure 5, but the assimilation model has a modified diffusion scheme. Only statistically significant innovation biases are shown.

Figure 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, there is a clear reduction in the error gap between the two configurations (the 22- and 64-bit models). Furthermore, according to Figure 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 strengthen this conclusion, another form of model error should be considered. In the next section, we consider changing the resolution of the nature run.

4.2. 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 background 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 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.

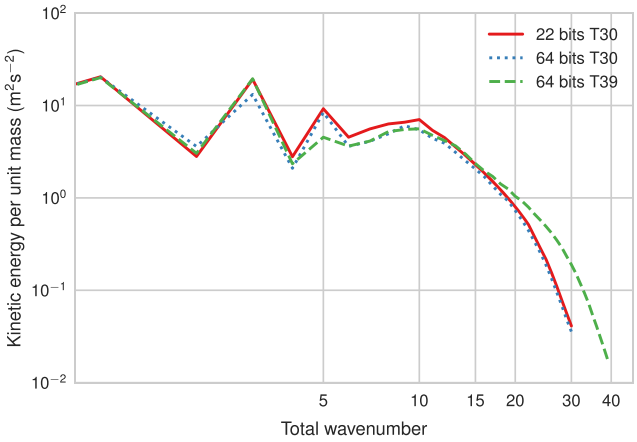

Figure 10. The kinetic energy spectra for SPEEDY at T30 with 64- and 22-bit arithmetic and at T39 resolution with 64-bit arithmetic. SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

Figure 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- and 22-bit configurations is now very small, and the overall error difference is less than 1%, on average. Furthermore, according to Figure 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 12. In other parts of the globe, such as Europe, the 64- 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 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

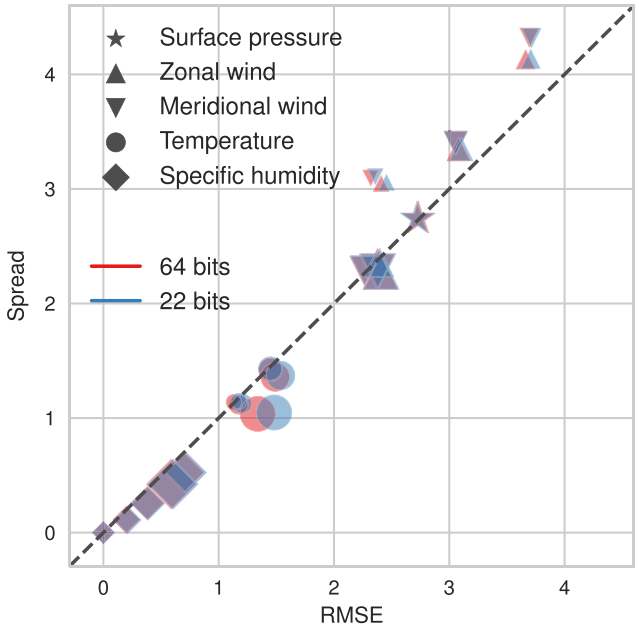

Figure 11. The analysis ensemble mean RMSE and spread of all prognostic fields, normalized by the respective observation error, for both the 64- and 22-bit configurations of SPEEDY, as in Figure 4. The nature run is T39 resolution (64-bit), but the assimilation model is T30 resolution. RMSE = root-mean-square error; SPEEDY = Simplified Parametrizations, primitivE-Equation DYnamics.

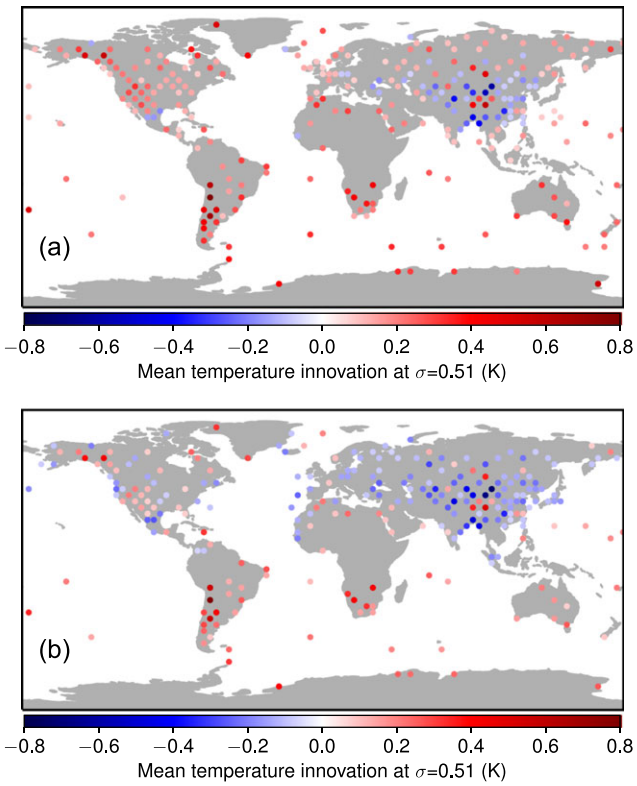

Figure 12. The temperature innovations at $\sigma = 0.51$ averaged over all assimilation cycles of the assimilation experiment shown in Figure 11 with (a) a 64-bit assimilation model and (b) a 22-bit assimilation model. The nature run is T39 resolution (64-bit), but the assimilation model is T30 resolution. Only statistically significant innovation biases are shown.

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.

5. Conclusion

Reducing the precision of a model may at first seem counterintuitive. 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 modeling of the atmosphere is inherently probabilistic. 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 modeling). These latter sources of uncertainty provide a margin of error within which to *fit* the rounding error incurred by lowering precision.

In this paper 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 the floating-point significand in the SPEEDY model from 52 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, although the quality of the analyses is substantially degraded with such a drastic reduction in precision, this only occurs in a perfect model setup. When some model uncertainty is introduced, the difference in performance between the 22- and 64-bit models is reduced substantially.

We note that our results obtained from reducing the precision of the assimilation model are not necessarily transferable to the model used to produce actual weather forecasts. The assimilation model is restarted every 6 hr 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 a LETKF) may require a different precision to the assimilation model as the computational profile is quite different, consisting of mostly matrix operations.

One limitation of our study is that we did not reduce the floating-point exponent width. The exponent, which is composed of 11 bits for double-precision and 8 bits for single-precision, determines the range of representable numbers. 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, e.g., Váňa et al., 2017, who ran a spectral atmospheric model at a much higher resolution entirely in single-precision with no overflow or underflow problems). If we could reduce the exponent further, to 5 bits, this would be of interest, as then our model would be using entirely half-precision arithmetic. Unlike our 22-bit floating-point type, half-precision is part of the IEEE754 standard and is increasingly supported in hardware, predominantly graphics processing units. The Tensor Core on Nvidia *Volta* GPUs is, for example, 9 times faster when multiplying half-precision matrices than when multiplying single-precision matrices on the previous generation of hardware, *Pascal* (Nvidia, 2017). Naturally this should be of interest to the atmospheric modeling community. Unfortunately, we were not able to run SPEEDY at half-precision as the 5-bit exponent does not provide enough range for all arithmetic operations to be computed without overflows and underflows. The range of half-precision floats is only around 11 orders of magnitude, with a minimum value of around 10^{-8}. Spectral wind tendencies are usually well below this.

There may be a limited use for half-precision arithmetic in certain parts of the model code. This could necessitate a rescaling of model variables such that they fit inside the half-precision range. In spectral models, for example, the Legendre and fast Fourier transforms required to transform grid point variables to spectral space and back constitute a significant fraction of the model cost. These operations are linear, and so any rescaling before transformation could be undone by dividing by the same rescaling factor after the transformation. Such a technique has been employed in order to perform neural network training using half-precision arithmetic (Micikevicius et al., 2017). This will be the subject of a future paper.

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 speedup 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. In the future, we would like to demonstrate that an even greater cost saving is possible, through a limited use of half-precision arithmetic, with beneficial impacts on forecasting skill.

Acronyms

SPEEDY	Simplified Parametrizations, primitivE-Equation DYnamics
LETKF	Local ensemble transform Kalman filter
RMSE	Root mean square error
EnKF	Ensemble Kalman filter
ECMWF	European Centre for Medium-Range Weather Forecasts
OSSE	Observing system simulation experiment
RTPP	Relaxation-to-prior perturbations

Acknowledgments
Sam Hatfield is funded by NERC grant NE/L002612/1 and also completed part of this work under the Japan Society for the Promotion of Science Summer Program. Peter Düben gratefully acknowledges funding from the ESiWACE project under grant 467 675191 and funding for his University Research Fellowship from the Royal Society. Matthew Chantry is funded by ONR grant DCR00480. The code for this study is available online (Hatfield, 2018) and is based on Takemasa Miyoshi's LETKF code, available at https://github.com/takemasa-miyoshi/letkf.

References

Amezcua, J., Kalnay, E., & Williams, P. D. (2011). The effects of the RAW filter on the climatology and forecast skill of the SPEEDY model. *Monthly Weather Review*, *139*(2), 608–619. https://doi.org/10.1175/2010MWR3530.1

Baek, S.-J., Szunyogh, I., Hunt, B. R., & Ott, E. (2009). Correcting for surface pressure background bias in ensemble-based analyses. *Monthly Weather Review*, *137*(7), 2349–2364. https://doi.org/10.1175/2008MWR2787.1

Bishop, C. H., Etherton, B. J., & Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review*, *129*(3), 420–436. https://doi.org/10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2

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

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

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

Düben, P. D., & Palmer, T. N. (2014). Benchmark tests for numerical weather forecasts on inexact hardware. *Monthly Weather Review*, *142*(10), 3809–3829. https://doi.org/10.1175/MWR-D-14-00110.1

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

Greybush, S. J., Kalnay, E., Miyoshi, T., Ide, K., & Hunt, B. R. (2011). Balance and ensemble Kalman filter localization techniques. *Monthly Weather Review*, *139*(2), 511–522. https://doi.org/10.1175/2010MWR3328.1

Hamill, T. M., & Whitaker, J. S. (2005). Accounting for the error due to unresolved scales in ensemble data assimilation: A comparison of different approaches. *Monthly Weather Review*, *133*(11), 3132–3147. https://doi.org/10.1175/MWR3020.1

Hamill, T. M., Whitaker, J. S., & Snyder, C. (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, *129*(11), 2776–2790. https://doi.org/10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2

Hamrud, M., Bonavita, M., & Isaksen, L. (2015). EnKF and hybrid gain ensemble data assimilation. Part I: EnKF implementation. *Monthly Weather Review*, *143*(12), 4847–4864. https://doi.org/10.1175/MWR-D-15-0071.1

Hatfield, S. (2018). samhatfield/letkf-speedy: Publication. https://doi.org/10.5281/zenodo.1198432

Hatfield, S., Subramanian, A., Palmer, T., & Düben, P. (2018). Improving weather forecast skill through reduced-precision data assimilation. *Monthly Weather Review*, *146*, 49–62. https://doi.org/10.1175/MWR-D-17-0132.1

Houtekamer, P. L., Deng, X., Mitchell, H. L., Baek, S.-J., & Gagnon, N. (2014). Higher resolution in an operational ensemble Kalman filter. *Monthly Weather Review*, *142*(3), 1143–1162. https://doi.org/10.1175/MWR-D-13-00138.1

Houtekamer, P. L., He, B., & Mitchell, H. L. (2014). Parallel implementation of an ensemble Kalman filter. *Monthly Weather Review*, *142*(3), 1163–1182. https://doi.org/10.1175/MWR-D-13-00011.1

Houtekamer, P. L., & Mitchell, H. L. (2005). Ensemble Kalman filtering. *Quarterly Journal of the Royal Meteorological Society*, *131*(613), 3269–3289. https://doi.org/10.1256/qj.05.135

Houtekamer, P. L., & Zhang, F. (2016). Review of the ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, *144*, 15–0440. https://doi.org/10.1175/MWR-D-15-0440.1

Hunt, B. R., Kostelich, E. J., & Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, *230*(1-2), 112–126. https://doi.org/10.1016/j.physd.2006.11.008

Kondo, K., & Miyoshi, T. (2016). 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. https://doi.org/10.1175/MWR-D-15-0388.1

Krishnamurti, T. N., Bedi, H. S., & Hardiker, V. M. (1998). Mathematical aspects of spectral models. In *An introduction to global spectral modeling* (pp. 71–103). Oxford University Press.

Lien, G.-Y., Kalnay, E., & Miyoshi, T. (2013). Effective assimilation of global precipitation: Simulation experiments. *Tellus, Series A: Dynamic Meteorology and Oceanography*, *65*(1), 1–16. https://doi.org/10.3402/tellusa.v65i0.19915

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., et al. (2017). Mixed precision training, CoRR, abs/1710.0.

Miyoshi, T. (2005). Ensemble Kalman filter experiments with a primitive-equation global model (PhD thesis), Atmospheric & Oceanic Science Theses and Dissertations.

Miyoshi, T. (2011). The Gaussian approach to adaptive covariance inflation and its implementation with the local ensemble transform Kalman filter. *Monthly Weather Review*, *139*, 1519–1535. https://doi.org/10.1175/2010MWR3570.1

Miyoshi, T., Kondo, K., & Imamura, T. (2014). The 10,240-member ensemble Kalman filtering with an intermediate AGCM. *Geophysical Research Letters*, *41*, 5264–5271. https://doi.org/10.1002/2014GL060863

Miyoshi, T., Lien, G.-Y., Satoh, S., Ushio, T., Bessho, K., Tomita, H., et al. (2016). "Big data assimilation" toward post-petascale severe weather prediction: An overview and progress. *Proceedings of the IEEE*, *104*(11), 2155–2179. https://doi.org/10.1109/JPROC.2016.2602560

Molteni, F. (2003). Atmospheric simulations using a GCM with simplified physical parametrizations. I: Model climatology and variability in multi-decadal experiments. *Climate Dynamics*, *20*, 175–191. https://doi.org/10.1007/s00382-002-0268-2

NVIDIA (2017). NVIDIA Tesla V100 GPU architecture (Tech. Rep.). Retrieved from http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

Nakano, M., Yashiro, H., Kodama, C., Tomita, H., Nakano, M., Yashiro, H., et al. (2018). Single precision in the dynamical core of a nonhydrostatic global atmospheric model: Evaluation using a baroclinic wave test case. *Monthly Weather Review*, *146*, 17–0257. https://doi.org/10.1175/MWR-D-17-0257.1

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., et al. (2004). A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, *56*(5), 415–428. https://doi.org/10.3402/tellusa.v56i5.14462

Ruiz, J., & Pulido, M. (2015). Parameter estimation using ensemble-based data assimilation in the presence of model error. *Monthly Weather Review*, *143*(5), 1568–1582. https://doi.org/10.1175/MWR-D-14-00017.1

Ruiz, J., Pulido, M., & Miyoshi, T. (2013). Estimating model parameters with ensemble-based data assimilation: A review. *Journal of the Meteorological Society of Japan. Ser. II*, *91*(2), 79–99. https://doi.org/10.2151/jmsj.2013-201

Sluka, T. C., Penny, S. G., Kalnay, E., & Miyoshi, T. (2016). Assimilating atmospheric observations into the ocean using strongly coupled ensemble data assimilation. *Geophysical Research Letters*, *43*, 752–759. https://doi.org/10.1002/2015GL067238

Váňa, F., Düben, P., Lang, S., Palmer, T. N., Leutbecher, M., Salmond, D., & Carver, G. (2017). Single precision in weather forecasting models: An evaluation with the IFS. *Monthly Weather Review*, *145*(2), 495–502. https://doi.org/10.1175/MWR-D-16-0228.1

Whitaker, J. S., & Hamill, T. M. (2012). Evaluating methods to account for system errors in ensemble data assimilation. *Monthly Weather Review*, *140*(9), 3078–3089. https://doi.org/10.1175/MWR-D-11-00276.1

Zhang, F., Snyder, C., & Sun, J. (2004). Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble Kalman filter. *Monthly Weather Review*, *132*(5), 1238–1253. https://doi.org/10.1175/1520-0493(2004)132<1238:IOIEAO>2.0.CO;2