The use of inexact hardware to improve weather and climate predictions

Peter Düben, Tim Palmer and many more

University of Oxford and ECMWF

The future of Earth System models on supercomputers

- Individual processors will not be faster.
- We will run Earth System models on a huge number of processing units in parallel (up to 10^6 already today).
- Scalability and performance will influence decisions in model development (but we do not always know what's best for a specific hardware).
- Power consumption will be a big problem for future high performance computing.
- Hardware errors and hardware faults will happen frequently.

The future of Earth System models on supercomputers

- Individual processors will not be faster.
- We will run Earth System models on a huge number of processing units in parallel (up to 10^6 already today).
- Scalability and performance will influence decisions in model development (but we do not always know what's best for a specific hardware).
- Power consumption will be a big problem for future high performance computing.
- Hardware errors and hardware faults will happen frequently.

The free lunch is over.

A reduction in numerical precision using inexact hardware

- Double precision is used in almost all weather and climate models.
- Inexact hardware trades precision against computing cost and allows a reduction of power consumption, an increase in performance, or a reduction in data storage.
- If inexact hardware could be used in weather and climate models, this would allow simulations at higher resolution and possibly more accurate forecasts.

A reduction in numerical precision using inexact hardware

- Double precision is used in almost all weather and climate models.
- Inexact hardware trades precision against computing cost and allows a reduction of power consumption, an increase in performance, or a reduction in data storage.
- If inexact hardware could be used in weather and climate models, this would allow simulations at higher resolution and possibly more accurate forecasts.

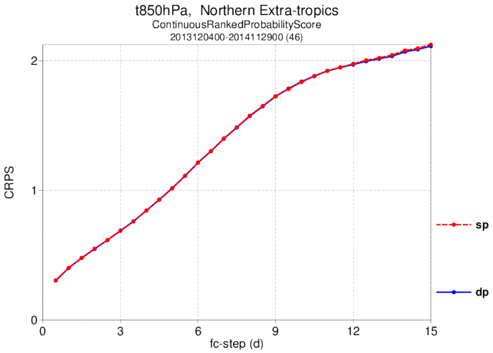
How can we trade precision against computing cost?

- Easy: double → single → half.
- Easy: Reduction of precision in data storage.
- Hard work: Field Programmable Gate Arrays (FPGAs).
- Future perspective: Flexible precision hardware and hardware with frequent hardware faults.

Outline

1. **Five studies to convince you that precision can be reduced in Earth System modelling.**
2. Five examples that adjust numerical precision to model error and model uncertainty.

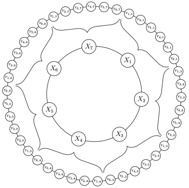
Study 1: IFS forecast model in single precision



- Filip Váňa has investigated single precision at ECMWF.
- Ensemble forecasts and long-term simulations in double and single precision at T399 resolution are almost identical.
- ≈40% speed-up.
- More tests are needed.

Düben et al. MWR 2015, Váňa et al. submitted to MWR

Study 2: Lorenz ’96 on Field Programmable Gate Arrays (FPGAs)



- We implemented the Lorenz ’96 model on FPGAs in cooperation with the group of Wayne Luk at Imperial College.
- We scale the size of the Lorenz model to the size of a high performance computing application with more than 100 million degrees-of-freedom.
- Simulations with reduced precision (17 bits for X; 14 bits for Y) are two times faster compared to simulations in single precision.
- The impact of the precision reduction is comparable to a parameter change of 1%.

Düben et al. JAMES 2015, Russel et al. FCCM 2015.

Study 3: Reduced precision in an atmosphere model

- We calculate weather forecasts with a spectral dynamical core (full dynamics - no physics).
- Floating point precision is reduced to 20 bits (instead of 64) using an emulator in almost the entire model.
- We estimate savings for inexact hardware in cooperation with computer scientists (the groups of Krishna Palem Rice University, Christian Enz EPFL and John Augustine IITM).

Study 3: Reduced precision in an atmosphere model

- We calculate weather forecasts with a spectral dynamical core (full dynamics - no physics).
- Floating point precision is reduced to 20 bits (instead of 64) using an emulator in almost the entire model.
- We estimate savings for inexact hardware in cooperation with computer scientists (the groups of Krishna Palem Rice University, Christian Enz EPFL and John Augustine IITM).

Resolution	Precision in number of bits	Normalised Energy Demand	Mean error Z500 at day 2
235 km	64	1.0	2.3
315 km	64	0.47	4.5
235 km	**20**	0.29	2.5

Study 3: Reduced precision in an atmosphere model

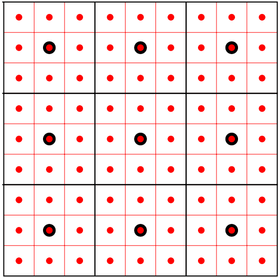
- We calculate weather forecasts with a spectral dynamical core (full dynamics - no physics).
- Floating point precision is reduced to 20 bits (instead of 64) using an emulator in almost the entire model.
- We estimate savings for inexact hardware in cooperation with computer scientists (the groups of Krishna Palem Rice University, Christian Enz EPFL and John Augustine IITM).

Resolution	Precision in number of bits	Normalised Energy Demand	Mean error Z500 at day 2
235 km	64	1.0	2.3
315 km	64	0.47	4.5
235 km	**20**	0.29	2.5

To save power a reduction in precision is much more efficient when compared to a reduction in resolution.

Düben et al. MWR 2015, Düben et al. DATE 2015.

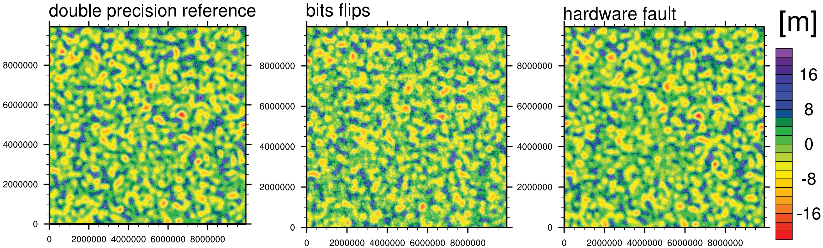
Study 4: Shallow water model with hardware faults



- We introduce a coarse backup grid to save prognostic fields.
- We test whether the fields on the backup grids are physically meaningful and restore erroneous values on the model grid, using the backup grid.
- We emulate soft errors in floating point operations and the loss of information in large areas of the model domain.
- The backup system generates 13% overheads in compute time.

Düben and Dawson in prep. for JAMES

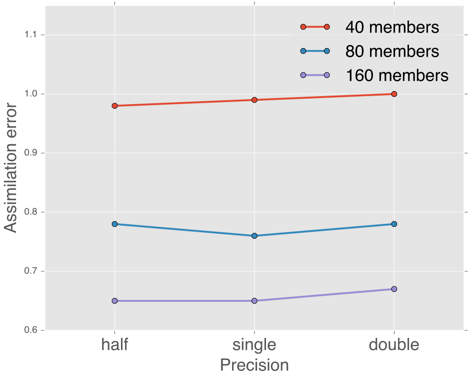
Study 4: Shallow water model with hardware faults



- We introduce a coarse backup grid to save prognostic fields.
- We test whether the fields on the backup grids are physically meaningful and restore erroneous values on the model grid, using the backup grid.
- We emulate soft errors in floating point operations and the loss of information in large areas of the model domain.
- The backup system generates 13% overheads in compute time.

Düben and Dawson in prep. for JAMES

Study 5: Data assimilation with reduced precision

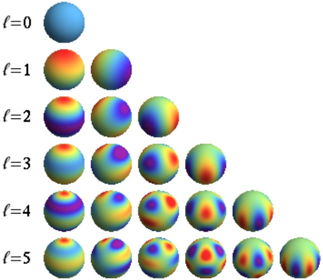


- We study data assimilation with an Ensemble Kalman filter at reduced numerical precision (Samuel Hatfield).
- The Figure shows the assimilation error for a Lorenz'95 model.
- It is better to use a large ensemble at low precision than a small ensemble at high precision at the same computational cost.
- 4DVAR may be very sensitive to a reduction in precision.

Outline

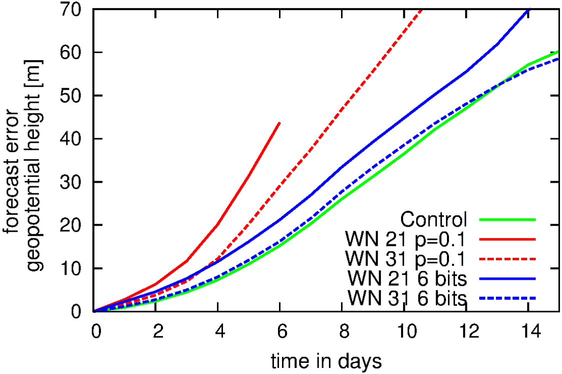
1. Five studies to convince you that precision can be reduced in Earth System modelling.
2. **Five examples that adjust numerical precision to model error and model uncertainty.**

Example 1: A scale-selective approach



- Spectral models allow to treat different scales at different levels of precision.
- We can push the small scales harder than the large scales.
- This is intuitive due to the high inherent uncertainty in small scale dynamics (parametrisation, viscosity, data-assimilation,...).
- The smallest scales are the most expensive once.

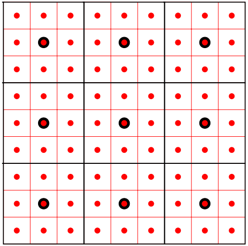
Example 1: A scale-selective approach

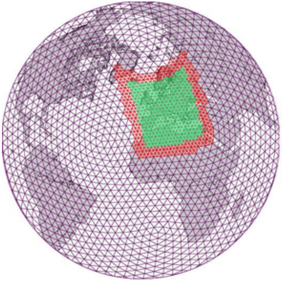


Scale separation in a spectral dynamical core.

Düben, MacNamara and Palmer JCP 2014; Düben and Palmer MWR 2014

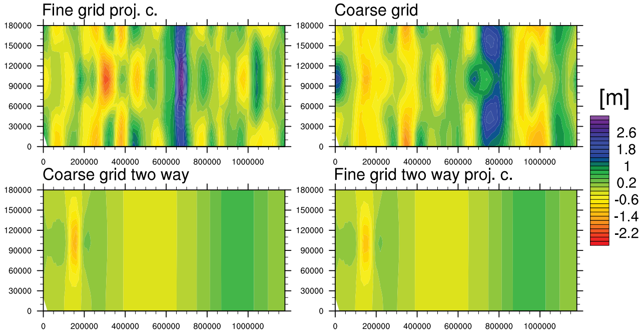
Example 1: A scale-selective approach





- Using scale dependent precision: High precision on the coarse grid, low precision on the fine grid.
- Coarse grid: Stability
- Fine grid: Variability.
- The approach is motivated by two-way nesting.
- Tests with a C-grid shallow water model.

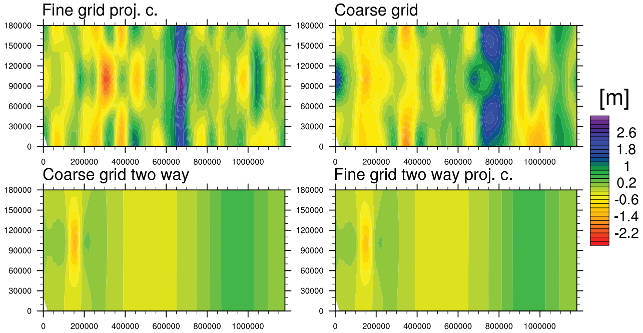
Example 1: A scale-selective approach



Height field of an isolated mountain test after 200,000 timesteps.

While one way nesting works fine in both directions, two way nesting does not work since small-scale structures are smoothed away.

Example 1: A scale-selective approach

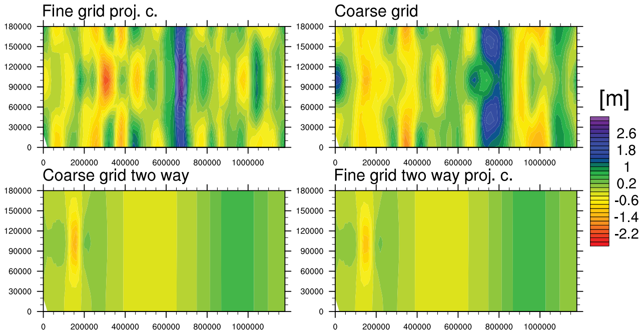


Height field of an isolated mountain test after 200,000 timesteps.

While one way nesting works fine in both directions, two way nesting does not work since small-scale structures are smoothed away.

Suggestions for improvements are very welcome!

Example 1: A scale-selective approach



Height field of an isolated mountain test after 200,000 timesteps.

While one way nesting works fine in both directions, two way nesting does not work since small-scale structures are smoothed away.

Suggestions for improvements are very welcome!

Maybe we should not perform nesting within the domain?

Example 2: Stochastic parametrisation schemes and rounding errors

- Rounding errors will generate a forcing that is added to the differential equations that is uncorrelated in space and time.
- We can influence the forcing by changing either the level of precision, or the model setup (time stepping scheme etc.).
- Stochastic parametrisation schemes use random forcing with specific mean and variability for physical reasons.

Example 2: Stochastic parametrisation schemes and rounding errors

- Rounding errors will generate a forcing that is added to the differential equations that is uncorrelated in space and time.
- We can influence the forcing by changing either the level of precision, or the model setup (time stepping scheme etc.).
- Stochastic parametrisation schemes use random forcing with specific mean and variability for physical reasons.

Can we design noise from rounding errors to replace the random forcing of stochastic parametrisation schemes.

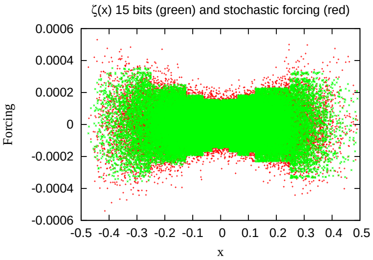
Example 2: Stochastic parametrisation schemes and rounding errors

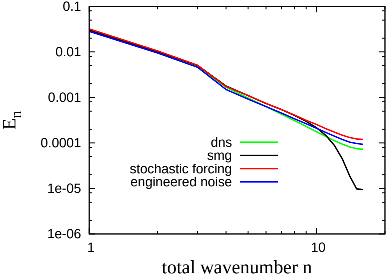
- Rounding errors will generate a forcing that is added to the differential equations that is uncorrelated in space and time.
- We can influence the forcing by changing either the level of precision, or the model setup (time stepping scheme etc.).
- Stochastic parametrisation schemes use random forcing with specific mean and variability for physical reasons.

Can we design noise from rounding errors to replace the random forcing of stochastic parametrisation schemes.

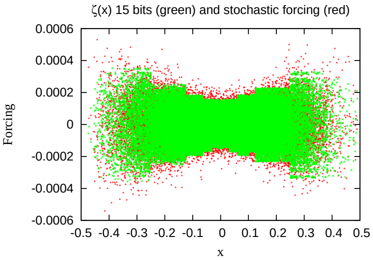
We study a Burgers equation model by Dolaptchiev, Timofeyev and Achatz with a stochastic parametrisation schemes for turbulent closure.

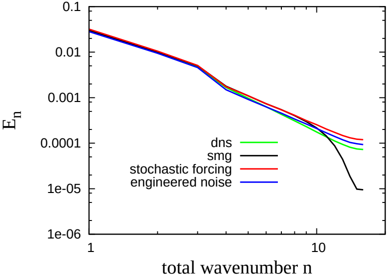
Example 2: Stochastic parametrisation schemes and rounding errors





Example 2: Stochastic parametrisation schemes and rounding errors

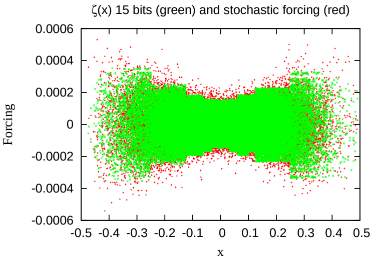


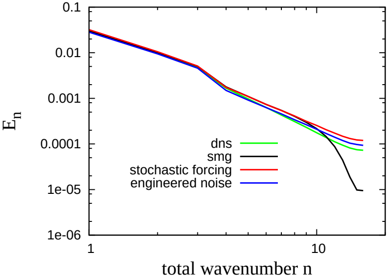


We can replace the stochastic forcing of a stochastic parametrisation scheme with a “rounding error forcing”.

→ Rounding errors can represent sub-grid-scale variability.

Example 2: Stochastic parametrisation schemes and rounding errors

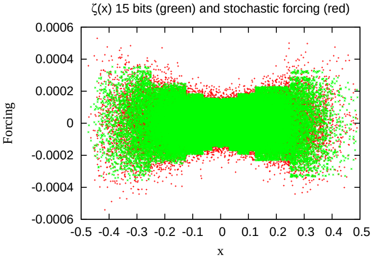


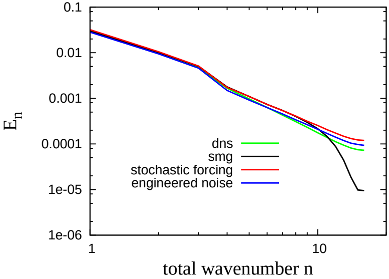


We can replace the stochastic forcing of a stochastic parametrisation scheme with a “rounding error forcing”.
→ Rounding errors can represent sub-grid-scale variability.

We can base ensemble simulations on rounding error forcings.

Example 2: Stochastic parametrisation schemes and rounding errors





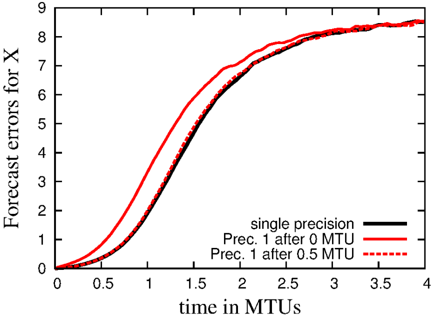
We can replace the stochastic forcing of a stochastic parametrisation scheme with a “rounding error forcing”.
→ Rounding errors can represent sub-grid-scale variability.

We can base ensemble simulations on rounding error forcings.

This study is extremely idealised.

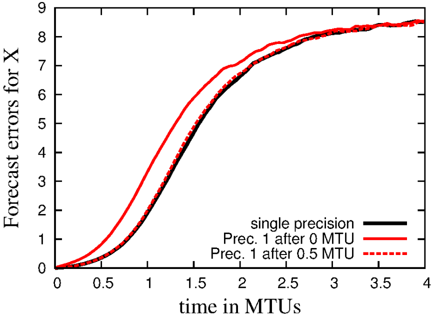
Düben and Dolaptchiev TCFD 2015

Example 3: A reduction of precision with lead time



Lorenz 95 on FPGAs: Prec. 1 is using 6 bits in the exponent and 11 bits in the significand.

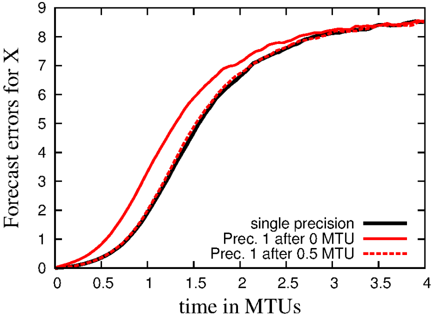
Example 3: A reduction of precision with lead time



Lorenz 95 on FPGAs: Prec. 1 is using 6 bits in the exponent and 11 bits in the significand.

Precision can be reduced with forecast lead time.

Example 3: A reduction of precision with lead time



Lorenz 95 on FPGAs: Prec. 1 is using 6 bits in the exponent and 11 bits in the significand.

Precision can be reduced with forecast lead time.

How can we find the optimal precision adjustment with lead time?

Düben et al. JAMES 2015

Example 3: A reduction of precision with lead time

Let’s consider precision in data storage.

Example 3: A reduction of precision with lead time

Let's consider precision in data storage.

The uncertainty of initial conditions provides the level for precision at the beginning of the forecast.

Example 3: A reduction of precision with lead time

Let’s consider precision in data storage.

The uncertainty of initial conditions provides the level for precision at the beginning of the forecast.

Error growth due to initial perturbations is roughly exponential.

Example 3: A reduction of precision with lead time

Let’s consider precision in data storage.

The uncertainty of initial conditions provides the level for precision at the beginning of the forecast.

Error growth due to initial perturbations is roughly exponential.

Precision of the representation of numbers will reduce exponentially with the number of bits.

Example 3: A reduction of precision with lead time

Let’s consider precision in data storage.

The uncertainty of initial conditions provides the level for precision at the beginning of the forecast.

Error growth due to initial perturbations is roughly exponential.

Precision of the representation of numbers will reduce exponentially with the number of bits.

→ Precision should be reduced linearly with forecast lead time. The rate of the precision reduction should be proportional to the leading Lyapunov exponent.

Example 3: A reduction of precision with lead time

Let’s consider precision in data storage.

The uncertainty of initial conditions provides the level for precision at the beginning of the forecast.

Error growth due to initial perturbations is roughly exponential.

Precision of the representation of numbers will reduce exponentially with the number of bits.

$\rightarrow$ Precision should be reduced linearly with forecast lead time. The rate of the precision reduction should be proportional to the leading Lyapunov exponent.

This would reduce data volume by a factor of two.

Example 3: A reduction of precision with lead time

Let’s consider precision in data storage.

The uncertainty of initial conditions provides the level for precision at the beginning of the forecast.

Error growth due to initial perturbations is roughly exponential.

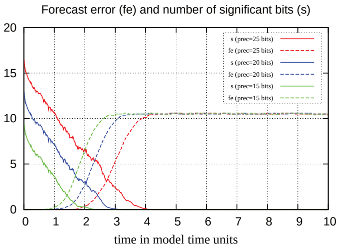
Precision of the representation of numbers will reduce exponentially with the number of bits.

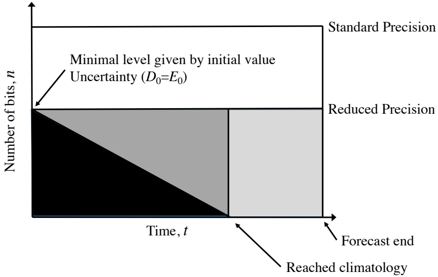
→ Precision should be reduced linearly with forecast lead time. The rate of the precision reduction should be proportional to the leading Lyapunov exponent.

This would reduce data volume by a factor of two.

Limitations: Linear error growth of model error, seasonal predictions,...

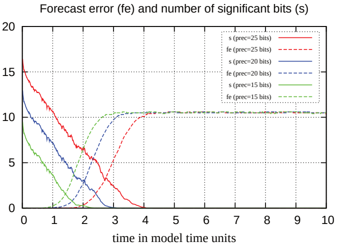
Example 3: A reduction of precision with lead time

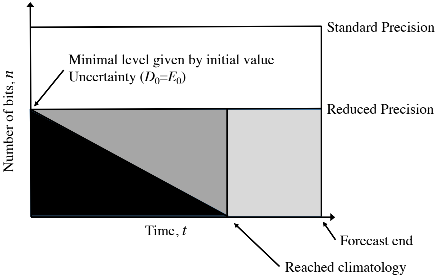




- We identify the number of bits that are not influenced by rounding errors using the verificarlo tool and emulated precision (Fenwick Cooper and Christophe Denis).
- The expected reduction of bits is consistent with the timescale of forecast errors for the Lorenz’95 model.
- The number of significant bits reduces linearly with time.

Example 3: A reduction of precision with lead time

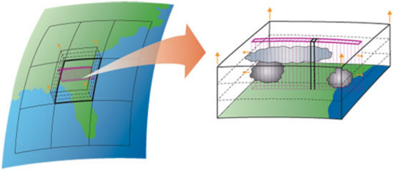




- We identify the number of bits that are not influenced by rounding errors using the verificarlo tool and emulated precision (Fenwick Cooper and Christophe Denis).
- The expected reduction of bits is consistent with the timescale of forecast errors for the Lorenz’95 model.
- The number of significant bits reduces linearly with time.

Promising! However, more tests are needed.

Example 4: A precision analysis to adjust model complexity



- Superparametrisation is running a two-dimensional cloud resolving model in each grid-cell of a global simulation.
- Superparametrisation improves tropical predictions (MJO, Indian monsoon,...) but it is very expensive.
- We use a superparametrised single column model of IFS (many thanks to Filip and Marat).
- We integrate the cloud resolving model using emulated inexact hardware.

Figure source: http://www.ucar.edu/communications/quarterly/summer06/cloudcenter.jsp

Example 4: A precision analysis to adjust model complexity

Parameter	Precision	Double precision	Reduced precision	Error
cp=specific heat of air	7	1004.0	1004.0	0.000%
ggr=gravitational acceleration	7	9.81	9.8125	0.025%
lcond=latent heat of condensation	14	2.5104e+06	2.510464e+06	0.003%
lfus=latent heat of fusion	7	3.336e+05	3.33824e+05	0.067%
lsub=latent heat of sublimation	12	2.8440e+06	2.844160e+06	0.006%
rv=gas constant water vapour	8	461.0	461.0	0.000%
diffelq=diffusivity water vapour	7	2.21e-05	2.2053719e-05	0.209%
therco=thermal conductivity of air	8	2.40e-02	2.3986816e-02	0.055%
muelq=dynamic viscosity of air	3	1.717e-05	1.7166138E-5	0.022%

- We automate the search for reduced precision to find the optimal level of precision for individual parameters.
- We compare model errors due to reduced precision with ensemble spread.
- Precision can be reduced significantly. However, estimates for savings are difficult.

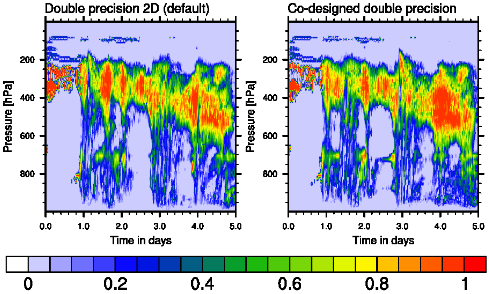
Example 4: A precision analysis to adjust model complexity

Parameter	Precision	Double precision	Reduced precision	Error
cp=specific heat of air	7	1004.0	1004.0	0.000%
ggr=gravitational acceleration	7	9.81	9.8125	0.025%
lcond=latent heat of condensation	14	2.5104e+06	2.510464e+06	0.003%
lfus=latent heat of fusion	7	3.336e+05	3.33824e+05	0.067%
lsub=latent heat of sublimation	12	2.8440e+06	2.844160e+06	0.006%
rv=gas constant water vapour	8	461.0	461.0	0.000%
diffelq=diffusivity water vapour	7	2.21e-05	2.2053719e-05	0.209%
therco=thermal conductivity of air	8	2.40e-02	2.3986816e-02	0.055%
muelq=dynamic viscosity of air	3	1.717e-05	1.7166138E-5	0.022%

- We automate the search for reduced precision to find the optimal level of precision for individual parameters.
- We compare model errors due to reduced precision with ensemble spread.
- Precision can be reduced significantly. However, estimates for savings are difficult.

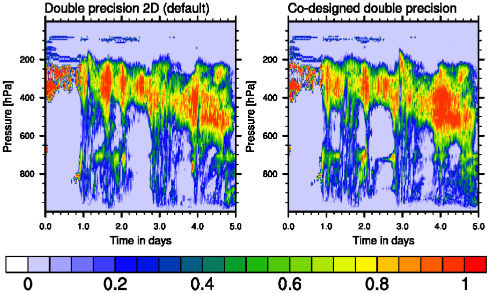
Long term aim: We will use the results of the precision analysis to adjust “fine-scale” stochastic parametrisation schemes.

Example 4: A precision analysis to adjust model complexity



- We find that precision can be reduced significantly in the turbulent kinetic energy scheme and for the high orders of the water vapour saturation curve.
- We remove those parts from the model.
- The new model setup is approximately 12% faster.

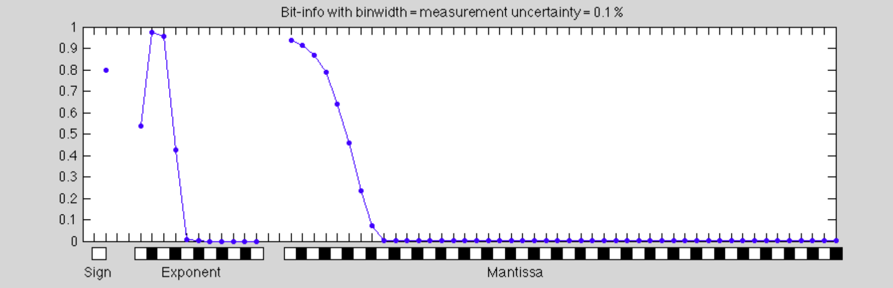
Example 4: A precision analysis to adjust model complexity



- We find that precision can be reduced significantly in the turbulent kinetic energy scheme and for the high orders of the water vapour saturation curve.
- We remove those parts from the model.
- The new model setup is approximately 12% faster.

A precision analysis can help to adjust model complexity.

Example 5: Information content in bit representation



- The calculation of information entropy allows to measure information content of individual bits (Stephen Jeffress).
- The Figure above shows the information content of bits for prognostic variables of a one-scaled Lorenz'95 model.
- A study of information entropy could help to identify the optimal way to store observation and model data.

Conclusions

The results suggest that...

- double precision as default is overcautious. A reduction in precision is promising huge savings.
- savings can be reinvested to allow higher resolution or more ensemble members to improve forecasts.
- a reduction in precision is much more efficient when compared to a reduction in resolution to save power.
- model simulations can be secured against bit flips and hardware faults with only limited overheads.
- precision should be reduced with spatial scale and forecast lead time.
- a precision analysis helps to adjust stochastic parametrisation schemes and vice versa.
- a precision analysis can help to improve models and to adjust model complexity.

Implications for ECMWF

- Data compression (information entropy and a reduction in precision with forecast lead-time).
- Study the parallel processing of different model components and accept a reduction in precision (e.g., ocean/atmosphere/ice or dynamics/physics).
- Single precision and half precision (Pascal GPUs by NVIDIA and Intel Knights Mill).
- Study multi-grid approaches.
- Increase local computation, reduce communication.

How to approach full-blown GCMs?
An emulator for reduced precision

Method:
We define a new reduced-precision type that behaves like a floating point number, but reduces the precision when it is operated on, this allows the emulation of reduced precision and specific setups of inexact hardware in large models (such as IFS) with no need for extensive changes of model code.

Example:
Emulated 5 bit significand with reduced precision "+"

Standard Fortran:

```
REAL :: a,b,c
a = 1.442221
b = 2.136601
c = a+b
```

→ *c=3.578822*

Reduced precision declarations:

```
TYPE(reduced_precision) :: a,b,c
a = 1.442221
b = 2.136601
c = a+b
```

→ *c=3.562500*

Dawson and Düben in prep. for GMD

References

PD Düben, J Joven, A Lingamneni, H McNamara, G De Micheli, KV Palem, TN Palmer, Phil. Trans. A, 2014

PD Düben, TN Palmer, H McNamara, JCP, 2014

TN Palmer, PD Düben, H McNamara, Phil. Trans. A, 2014

PD Düben, TN Palmer, Mon. Weath. Rev., 2014

PD Düben, J Schlachter, Parishkrati, S Yenugula, J Augustine, C Enz, K Palem and TN Palmer, DATE, 2015

F Russell, PD Düben, X Niu, W Luk, TN Palmer, FCCM, 2015

PD Düben, S Jeffress, TN Palmer, EMIT, 2015

PD Düben, SI Dolaptchiev, TCFD, 2015

PD Düben, F Russel, X Niu, W Luk, TN Palmer, JAMES, 2015

Example 1: A scale-selective approach

$$\partial_t \mathbf{u} + \mathbf{u} \cdot (\nabla \mathbf{u}) + f\mathbf{k} \times \mathbf{u} + g\nabla h = \alpha_u \left(\mathbf{u}_{c/f} - \mathbf{u}\right)$$

$$\partial_t h + \nabla \cdot (H\mathbf{u}) = \alpha_h \left(h_{c/f} - h\right)$$

u two dimensional velocity, f is the Coriolis parameter, **k** vertical unit vector, g gravitational acceleration, h surface elevation, H height of the fluid column, α nesting parameter

1. We use bilinear mapping to calculate the relaxation term for the differential equations on both grids.
2. The code gets messy for a 2d periodic staggered grid, but it is straight forward to do it.
3. Computational overhead due to the coarse grid is very small and we use $3\Delta X$ for the coarse grid.

Why should we think about an adjustment of precision?

This is what we want to represent in an atmosphere model.

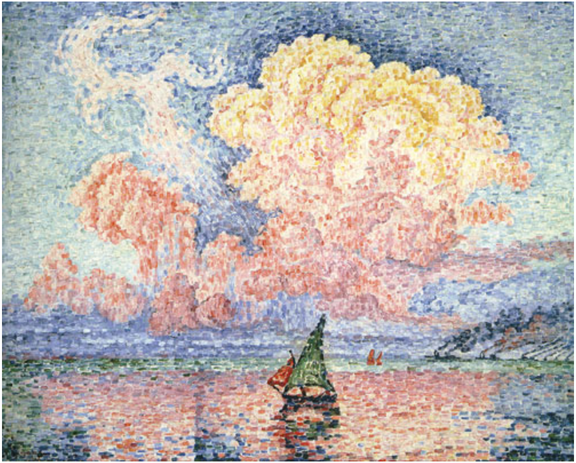
Why should we think about an adjustment of precision?

This is how we represent the atmosphere.

Steve Greaves

Why should we think about an adjustment of precision?

Can we represent the atmosphere like this?



Antibes 1916

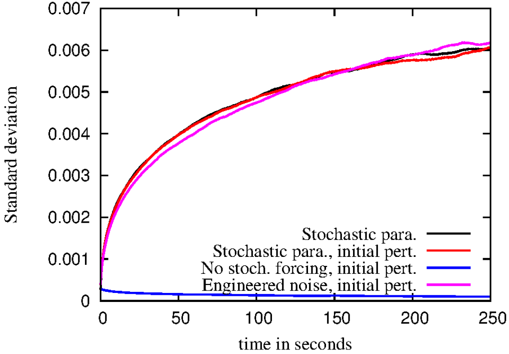
A precision analysis to improve models

If we perform a detailed precision analysis for weather and climate models...

- ...we can secure high precision simulations against rounding errors and improve portability.
- ...we can identify and remove model parts that do not have a strong influence on model dynamics.
- ...we can use the strength of acceptable rounding errors to inform stochastic parametrisation schemes.
- ...we can use rounding errors to generate ensemble spread.
- ...we can use the optimal level of precision to quantify model uncertainty.

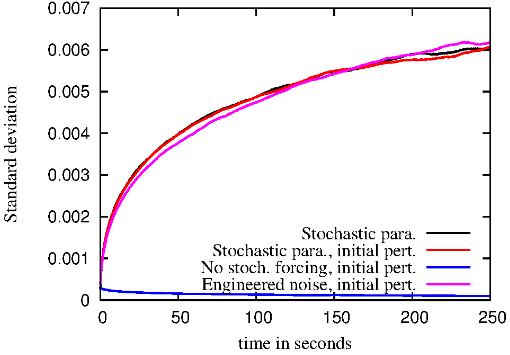
A study of rounding errors will provide plenty of information on model error and model uncertainty.

Example 2: Stochastic parametrisation schemes and rounding errors



If we add a tiny amount of random noise to the initial conditions, rounding errors will be uncorrelated in space and time.

Example 2: Stochastic parametrisation schemes and rounding errors



If we add a tiny amount of random noise to the initial conditions, rounding errors will be uncorrelated in space and time.

Rounding errors can be used to represent sub-grid-scale variability and generate ensembles.

Example 2: Parameter uncertainty in superparametrisation

Parameter	Precision	64 bits	Reduced prec.	Error
specific heat	7	1004.0	1004.0	0%
grav. acceleration	5	9.81	9.75	0.6%
latent heat of condensation	4	2.5104e+06	2.490368e+06	0.8%
latent heat of fusion	0	3.336e+05	0.262144e+05	21%
latent heat of sublimation	3	2.8440e+06	2.883584e+06	1.4%
gas constant	0	461.0	512.0	11%
diffusivity water vapour	1	2.21e-05	2.2888184e-05	3.6%
thermal conductivity	1	2.40e-02	2.34375e-02	2.3%
dynamic viscosity	0	1.717e-05	1.5258789E-5	11%
...	...	...	...	...

Example 2: Parameter uncertainty in superparametrisation

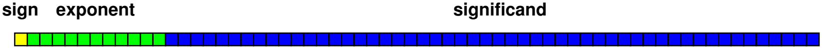
Parameter	Precision	64 bits	Reduced prec.	Error
specific heat	7	1004.0	1004.0	0%
grav. acceleration	5	9.81	9.75	0.6%
latent heat of condensation	4	2.5104e+06	2.490368e+06	0.8%
latent heat of fusion	0	3.336e+05	0.262144e+05	21%
latent heat of sublimation	3	2.8440e+06	2.883584e+06	1.4%
gas constant	0	461.0	512.0	11%
diffusivity water vapour	1	2.21e-05	2.2888184e-05	3.6%
thermal conductivity	1	2.40e-02	2.34375e-02	2.3%
dynamic viscosity	0	1.717e-05	1.5258789E-5	11%
...	...	...	...	...

The minimal level of precision provides plenty of information on parameter uncertainty.

Two examples for approaches to inexact hardware

Double precision

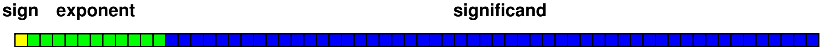
Floating point numbers are represented with 64 bits and 15 decimal places. 1 sign bit, 11 bits in the exponent and 52 bits in the significand.



Two examples for approaches to inexact hardware

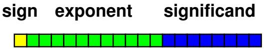
Double precision

Floating point numbers are represented with 64 bits and 15 decimal places. 1 sign bit, 11 bits in the exponent and 52 bits in the significand.



Pruning

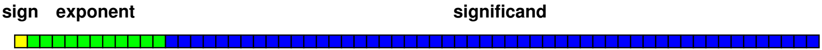
Parts of the CPU that are hardly used or do not have a strong influence on significant bits are removed. Errors are limited to the significand of floating point numbers.



Two examples for approaches to inexact hardware

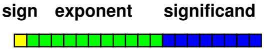
Double precision

Floating point numbers are represented with 64 bits and 15 decimal places. 1 sign bit, 11 bits in the exponent and 52 bits in the significand.



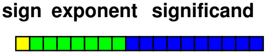
Pruning

Parts of the CPU that are hardly used or do not have a strong influence on significant bits are removed. Errors are limited to the significand of floating point numbers.

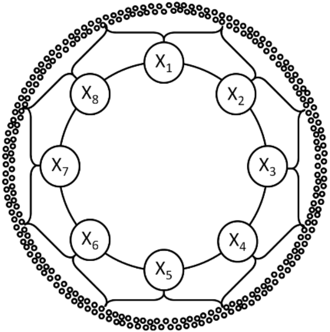


Field Programmable Gate Array (FPGA)

FPGAs are integrated circuits that can be configured by the user. Numerical precision can be customised to the application.

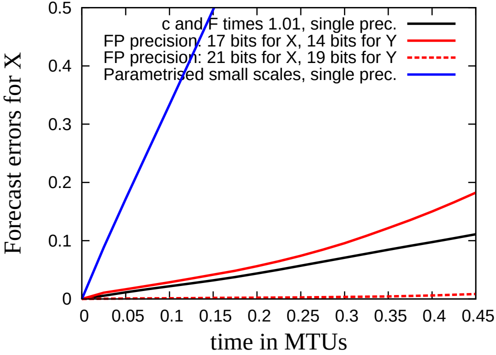


Lorenz '96 on an FPGA

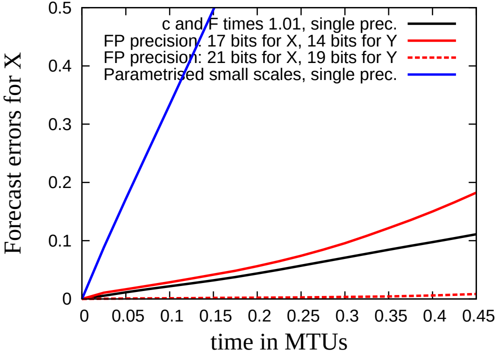


- We implemented the Lorenz '96 model on an FPGA in cooperation with Xinyu Niu, Francis Russel and Wayne Luk from Imperial College.
- We scale the size of the Lorenz model to the size of a high performance application (up to more than 100 million degrees-of-freedom).
- We compare results with reduced precision against results with

Lorenz ’96 on an FPGA

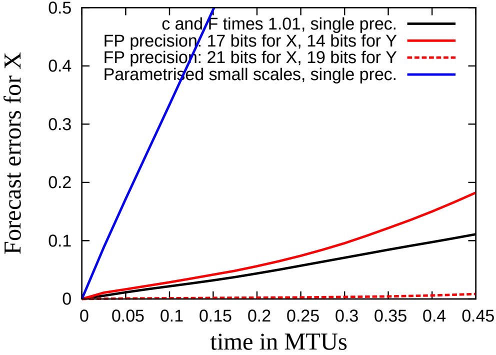


Lorenz '96 on an FPGA



Changes in weather type forecasts are comparably small when precision is reduced.

Lorenz ’96 on an FPGA



Changes in weather type forecasts are comparably small when precision is reduced.

The same holds for climate type forecasts.

Lorenz ’96 on an FPGA: Speed and Power

Hardware	Speed	Energy efficiency
CPU, 12 cores, single precision	1.0	1.0
CPU, 12 cores, double precision	0.5	-
FPGA, single precision	2.8	10.4
FPGA, 17 bits for X, 14 bits for Y	6.9	23.9
FPGA, 21 bits for X, 19 bits for Y	5.4	18.9

Lorenz ’96 on an FPGA: Speed and Power

Hardware	Speed	Energy efficiency
CPU, 12 cores, single precision	1.0	1.0
CPU, 12 cores, double precision	0.5	-
FPGA, single precision	2.8	10.4
FPGA, 17 bits for X, 14 bits for Y	6.9	23.9
FPGA, 21 bits for X, 19 bits for Y	5.4	18.9

We get significant savings in energy consumption and a significant increase in performance if we use FPGAs with reduced precision.

A short introduction to bit representation

- The computer represents an integer number as a string of 32 bits. Each bit represents a power of two:

$$102090 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 ... = \sum_{i=0}^{31} b_i 2^i$$

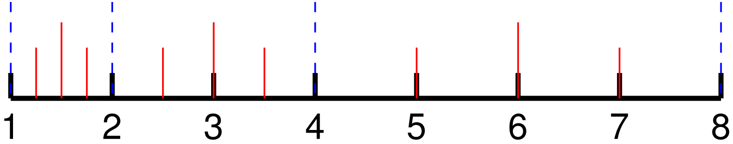
A short introduction to bit representation

- The computer represents an integer number as a string of 32 bits. Each bit represents a power of two:

$$102090 = 0{\cdot}2^0+1{\cdot}2^1+0{\cdot}2^2+1{\cdot}2^3+0{\cdot}2^4+0{\cdot}2^5+1{\cdot}2^6 ... = \sum_{i=0}^{31} b_i 2^i$$

- A real number *a* is represented as a 64 bit floating point number:

$$a = (-1)^S \left(1+\sum_{i=1}^{52} b_{-i} 2^{-i}\right) 2^E, \quad \text{where} \quad E = \left(\sum_{i=0}^{10} e_i 2^i\right) - 1023.$$



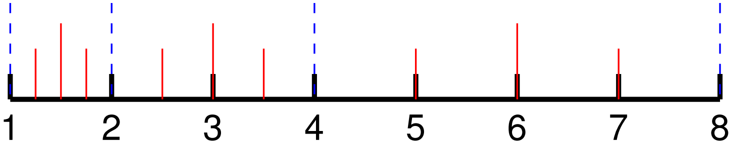
A short introduction to bit representation

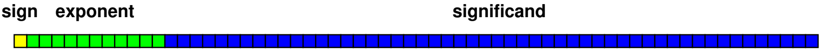
- The computer represents an integer number as a string of 32 bits. Each bit represents a power of two:

$$102090 = 0\cdot 2^0 + 1\cdot 2^1 + 0\cdot 2^2 + 1\cdot 2^3 + 0\cdot 2^4 + 0\cdot 2^5 + 1\cdot 2^6 \ldots = \sum_{i=0}^{31} b_i 2^i$$

- A real number a is represented as a 64 bit floating point number:

$$a = (-1)^S \left(1 + \sum_{i=1}^{52} b_{-i} 2^{-i}\right) 2^E, \quad \text{where} \quad E = \left(\sum_{i=0}^{10} e_i 2^i\right) - 1023.$$





How to approach full-blown GCMs?

If we use the emulator to reduce precision, how can we find the minimal level of precision that can be used?

How to approach full-blown GCMs?

If we use the emulator to reduce precision, how can we find the minimal level of precision that can be used?

Shall we use the same precision level for a) the entire model, b) each module, c) each line of code, d) each variable?

How to approach full-blown GCMs?

If we use the emulator to reduce precision, how can we find the minimal level of precision that can be used?

Shall we use the same precision level for a) the entire model, b) each module, c) each line of code, d) each variable?

The more we fine-grain, the more we will be able to a) reduce precision, b) learn about the information content of specific variables and c) learn about technical issues and bad programming.

How to approach full-blown GCMs?

If we use the emulator to reduce precision, how can we find the minimal level of precision that can be used?

Shall we use the same precision level for a) the entire model, b) each module, c) each line of code, d) each variable?

The more we fine-grain, the more we will be able to a) reduce precision, b) learn about the information content of specific variables and c) learn about technical issues and bad programming.

But tests with a big model are expensive!

Information content and reduced precision

We like to think that the minimal level of precision that can be used is the “information content” of a variable (e.g. for π).

Information content and reduced precision

We like to think that the minimal level of precision that can be used is the “information content” of a variable (e.g. for π).

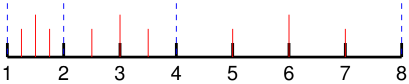
If this would be true and if we could identify this level of precision, this would be great help for model development and the development of stochastic parametrisation schemes.

Information content and reduced precision

We like to think that the minimal level of precision that can be used is the "information content" of a variable (e.g. for π).

If this would be true and if we could identify this level of precision, this would be great help for model development and the development of stochastic parametrisation schemes.

But this is not necessarily the case!
Example: Floating point representation of temperature in Kelvin (223-323) and in degree Celsius (-50-50) and model crashes.

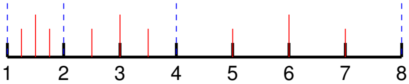


Information content and reduced precision

We like to think that the minimal level of precision that can be used is the "information content" of a variable (e.g. for π).

If this would be true and if we could identify this level of precision, this would be great help for model development and the development of stochastic parametrisation schemes.

But this is not necessarily the case!
Example: Floating point representation of temperature in Kelvin (223-323) and in degree Celsius (-50-50) and model crashes.



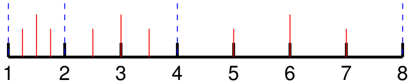
We need to look into details to understand the information content.

Information content and reduced precision

We like to think that the minimal level of precision that can be used is the "information content" of a variable (e.g. for π).

If this would be true and if we could identify this level of precision, this would be great help for model development and the development of stochastic parametrisation schemes.

But this is not necessarily the case!
Example: Floating point representation of temperature in Kelvin (223-323) and in degree Celsius (-50-50) and model crashes.

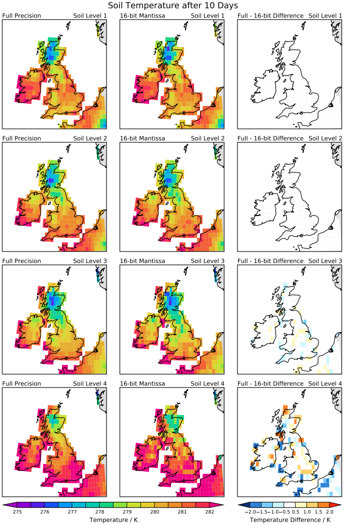


We need to look into details to understand the information content.

We can certainly learn something about the forcings.

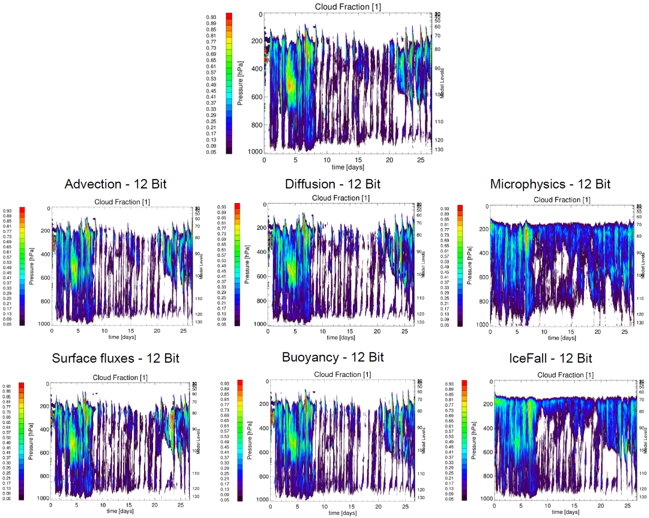
“Coarse-graining” in large models

Preliminary results with the emulator in the land surface scheme from Andrew Dawson.



“Coarse-graining” in large models

Preliminary results with the emulator in the cloud resolving model of the superparametrised IFS model from Aneesh Subramanian.



Advection - 12 Bit

Diffusion - 12 Bit

Microphysics - 12 Bit

Surface fluxes - 12 Bit

Buoyancy - 12 Bit

IceFall - 12 Bit

“Fine-graining” in large models

It is too expensive to run long term simulation with IFS to identify the right level of precision if we have many different precision levels!

“Fine-graining” in large models

It is too expensive to run long term simulation with IFS to identify the right level of precision if we have many different precision levels!

What is a cheap way to identify the minimal level of precision that can be used?

How to identify the right level of precision that should be used?

Tests with Lorenz ’95 show that short-term forecasts (50 timesteps) can provide useful information.

However:

- Each quantity needs to be checked against it’s own reference.
- Simulations are too sensitive after a couple of timesteps.

How to identify the right level of precision that should be used?

Tests with Lorenz ’95 show that short-term forecasts (50 timesteps) can provide useful information.

However:

- Each quantity needs to be checked against it’s own reference.
- Simulations are too sensitive after a couple of timesteps.

This is consistent with the seamless prediction approach.

My recipe to reduce precision in IFS

1. Introduce the emulator into the model.
2. Use short term simulations of 50 timesteps to “fine-grain” precision levels.
3. Compare each “prognostic quantity” of a subroutine against a high precision reference after 50 timesteps.
4. Find an appropriate quality control and automatise the search.

Tests with a C-grid shallow-water model

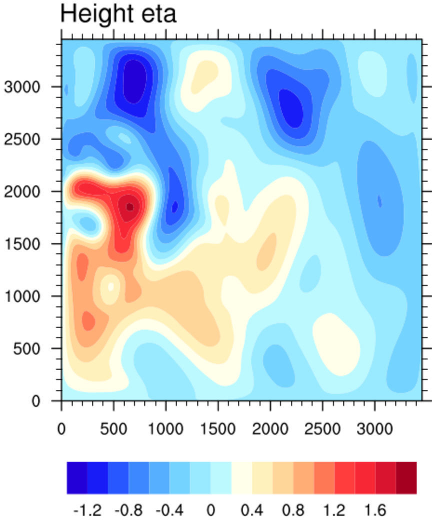
$$\partial_t \mathbf{u} + \mathbf{u} \cdot (\nabla \mathbf{u}) + f \mathbf{k} \times \mathbf{u} + g \nabla \eta = \nu \Delta \mathbf{u} + \boldsymbol{\tau}$$
$$\partial_t \eta + \nabla \cdot (h \mathbf{u}) = 0$$

u: horizontal velocity
f: Coriolis parameter
k: vertical unit vector
g: gravitational acceleration

η: surface elevation
ν: eddy viscosity
$\boldsymbol{\tau}$: forcing term
h: height of the fluid column

Tests with a C-grid shallow-water model

We run a Munk-double-gyre ocean testcase on a 128X128 grid with 500 meter depth.



Tests with a C-grid shallow-water model

1. Our quality control: The mean difference between the double precision and the reduced precision simulation should be smaller than 0.1% of the standard deviation of u, v, and η.
2. We give each floating point field of the timestepping loop (29 in total) an individual level of precision (e.g. Δt, h, g,...).
3. We run a set of runs for which we reduce precision only for a specific variable. We start with 2 bits precision in the significand for the specific variable and increase this level until the run fulfils the quality control.
4. The results on the following slides are very preliminary!

Tests with a C-grid shallow-water model

Parameter	number of bits in the significand
η	12
u	12
v	12
τ_x	2
τ_y	2
dh	2
du	2
dv	2
ab	2
h0	2
fu	4
fv	4
b	7
zeta	2
vec	2
g	5
pi	2
f0	2
beta	2
ν	2
ah	2
x0	2
y0	2
dx	8
dy	8
dt	2
slip	2
sigmax	2
sigmay	2

We end up with precision levels that should be used for the significand of floating point numbers.

Precision can be reduced significantly!

We obtain information on the information content.

Tests with a C-grid shallow-water model

Parameter	number of bits in the significand
η	12
u	12
v	12
τ_x	2
τ_y	2
dh	2
du	2
dv	2
ab	2
h0	2
fu	4
fv	4
b	7
zeta	2
vec	2
g	5
pi	2
f0	2
beta	2
ν	2
ah	2
x0	2
y0	2
dx	8
dy	8
dt	2
slip	2
sigmax	2
sigmay	2

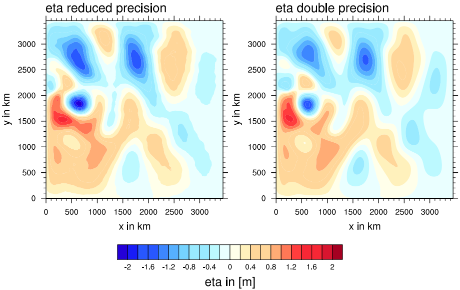
We end up with precision levels that should be used for the significand of floating point numbers.

Precision can be reduced significantly!

We obtain information on the information content.

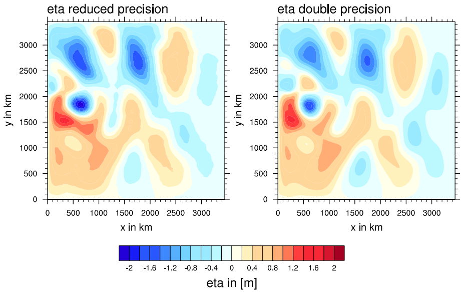
Expert knowledge is needed to obtain stable model simulations (increase precision for Δt and ab).

Tests with a C-grid shallow-water model



Height field after 28 days of simulations.

Tests with a C-grid shallow-water model



Height field after 28 days of simulations.

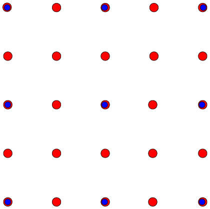
Results are very preliminary and more tests are needed!

How to approach full-blown GCMs?

How can we port a model to inexact hardware (whatever this may look like)?

- GCMs already face the challenge to be portable and scalable on different hardware architectures (e.g. GPU/CPU systems).
- If you want maximal performance, a rewrite of the most expensive parts of the model seems to be necessary already for exact hardware (flops/bits, parallelisations, GPUs,...).
- The use of a domain specific language will probably be the key!
- A study of inexact hardware would be much easier in a model which is based on a domain specific language.

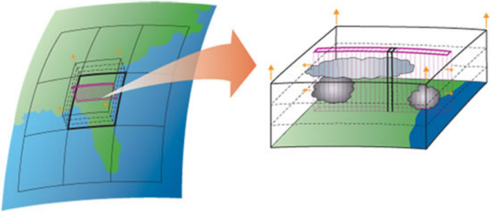
Future research: Multigrid/nested methods



Stephen Jeffress is looking into the use of inexact hardware in a C-grid shallow water model.

- A scale dependent allocation of computing resources on a hybrid CPU - FPGA architecture. High precision on the coarse grid, low precision on the fine grid.
- Coarse grid: Stability; Fine grid: Variability.
- Future studies could be based on two-way nesting!?

Future research: Super-parametrisation



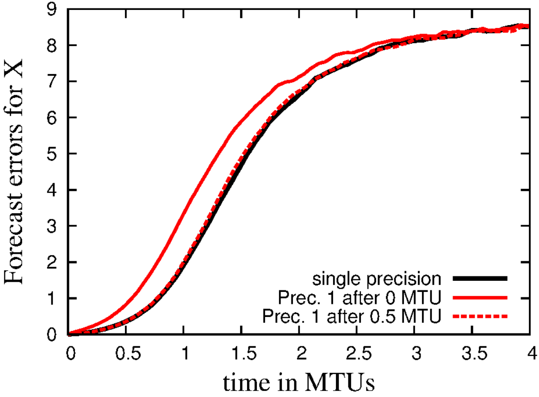
Aneesh Subramanian is looking into super-parametrisation.

1. We test a super-parametrisation scheme of atmospheric convection in the IFS of ECMWF which is similar to the scheme in the NCAR climate model.
2. Using super-parametrisation increases computational cost by a factor 60!
3. We calculate the cloud resolving model with emulated inexact hardware.

Reduced precision in an atmosphere model

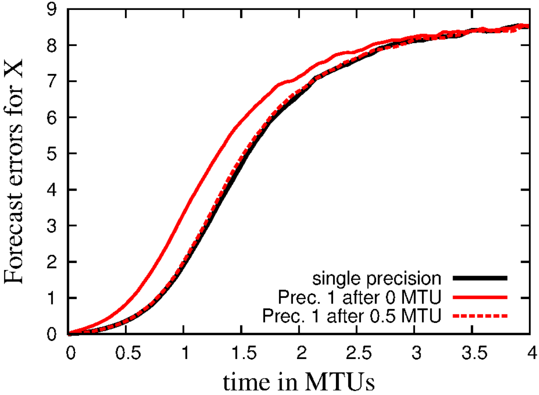
- In collaboration with Rice University (USA) and EPFL (Switzerland) we derived hardware setups of the floating point unit, memory and cache that show comparable error pattern
- Inexact floating point units are developed and synthesized with the same delay and area constraints as exact hardware
- DRAMsim is used to obtain energy estimates of the main memory
- We assume that the cache is built out of SRAM cells. Here, energy consumption scales linearly with the width of the word
- We do not consider instruction code memory access

Lorenz ’96 on an FPGA: Reduce precision with time



Prec 1 is using 6 bits in the exponents and 11 bits in the significand.

Lorenz ’96 on an FPGA: Reduce precision with time



Prec 1 is using 6 bits in the exponents and 11 bits in the significand.

Precision can be reduced with time in weather-type forecasts.