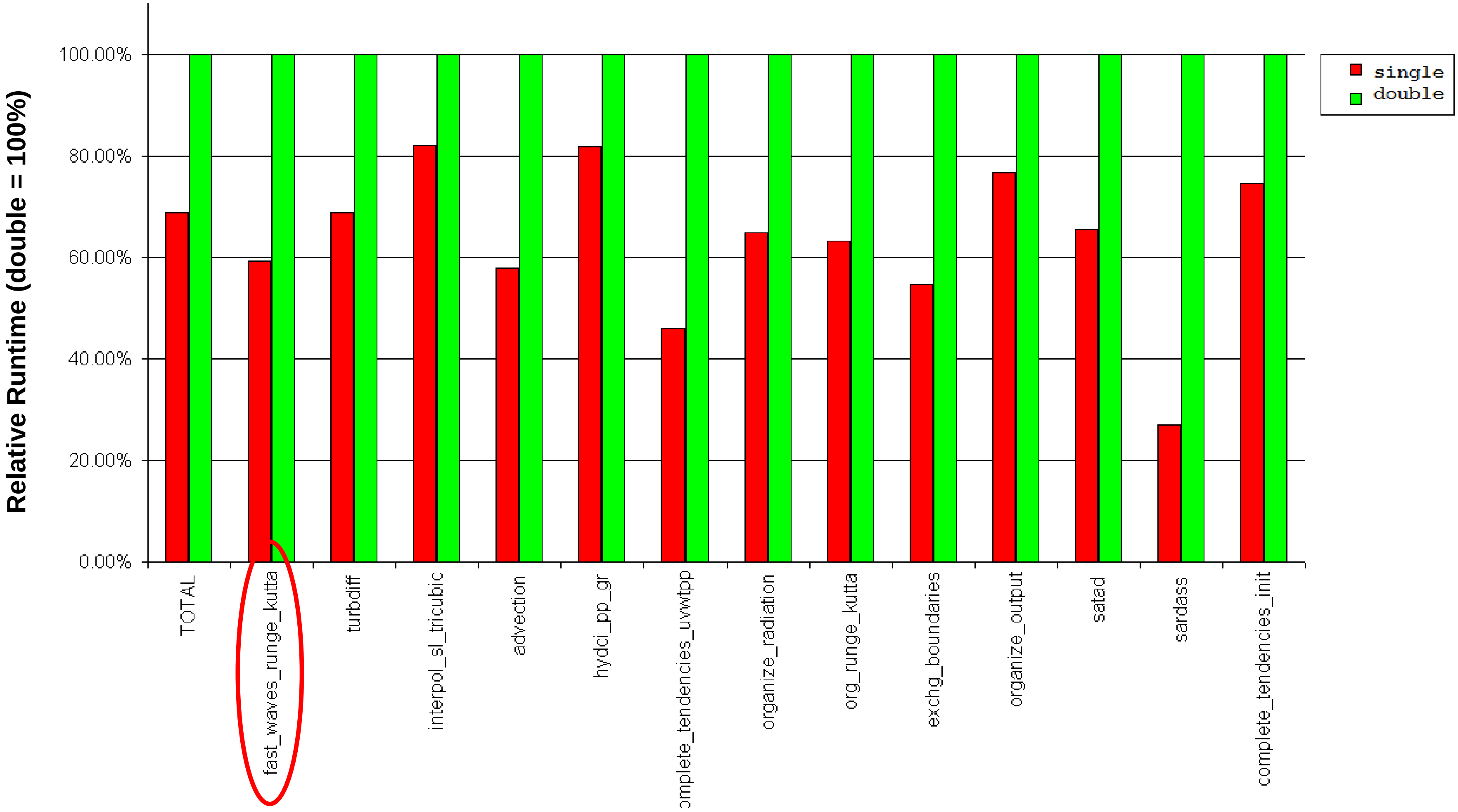
Analysis of error propagation in the fast waves solver of COSMO

Oliver Fuhrer (MeteoSwiss)
Katharina Riedinger (Universität Innsbruck)
Tobias Gysi (Supercomputing Systems AG)

Overview

- Motivation
- Floating point precision
- Fast waves solver
- Analysis
- Discussion

Motivation



Example: Operational COSMO-2 on Cray XT4

Motivation

- Move less information

```
real(kind=8) :: a    ! I am 8 Bytes
real(kind=4) :: b    ! I am 4 Bytes
```

- Fit more information into cache

- Lower precision arithmetic is faster

```
a = a+a-a*a*a   ! Wow, time flies!
b = b+b-b*b*b   ! That was fast!
```

Questions

- How much arithmetic precision do we really need in COSMO?
- Where are the hotspots which are sensitive to arithmetic precision?
- Can we modify these to work also with low precision arithmetic?

Source of Errors

- **Model errors**
  - “Under the assumption that cows are spherical we can write Equation 2 as follows…”

- **Approximation errors**
  - Discretization errors
  - Convergence errors

- **Round-off errors**
  i.e. computers use a finite number of digits to represent any number

Requirement: The error in the model solution is not dominated by round-off errors.

Floating point numbers

- Most computers follow the IEEE 754 standard

$$x = (-1)^s \cdot c \cdot b^q$$

$$-12.345 = (-1)^1 \cdot 12345 \cdot 10^{-3}$$

s sign
c signicand (coefficient)
b base
q exponent

- Examples

Name	Size	Decimal digits	Minimum number	Maximum number
half precision	2 Bytes	3.3	10^{-5}	10^{4}
single precision	4 Bytes	7.2	10^{-38}	10^{38}
double precision	8 Bytes	16.0	10^{-308}	10^{308}
quadruple precision	16 Bytes	34.0	10^{-4932}	10^{4932}

Round-off Errors

- Cancellation error (e.g. finite differences)

$$\frac{\partial T}{\partial x}\Delta x \approx T_1 - T_2 = 293.18 - 292.90 = 0.28$$

- Round-off error accumulation (e.g. linear systems)

M x = d

- Overflow error (e.g. exponents)

$$\frac{a^4}{b^3} = a\left(\frac{a}{b}\right)^3$$

Fast Waves Solver

- Update horizontal velocities (explicit)

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + f^u \qquad \frac{\partial v}{\partial t} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + f^v$$

- Update w, T, p' (vertically implicit)

$$\left.\begin{aligned} \frac{\partial w}{\partial t} &= -\frac{1}{\rho}\frac{\partial p'}{\partial x} + B + f^w \\ \frac{\partial p'}{\partial t} &= g\rho_0 w - p\frac{c_p}{c_v}\left(D_h + \frac{\partial w}{\partial z}\right) + f^{p'} \\ \frac{\partial T'}{\partial t} &= -p\frac{1}{\rho\, c_v}\left(D_h + \frac{\partial w}{\partial z}\right) + f^T \end{aligned}\right\} \quad \mathbb{M} \cdot \vec{\mathrm{w}} = \vec{\mathrm{d}}$$

Condition Number

- Consider perturbed system M w = d

$$(M + \delta M)(w + \delta w) = d + \delta d$$

- Condition number determines upper bound of error growth

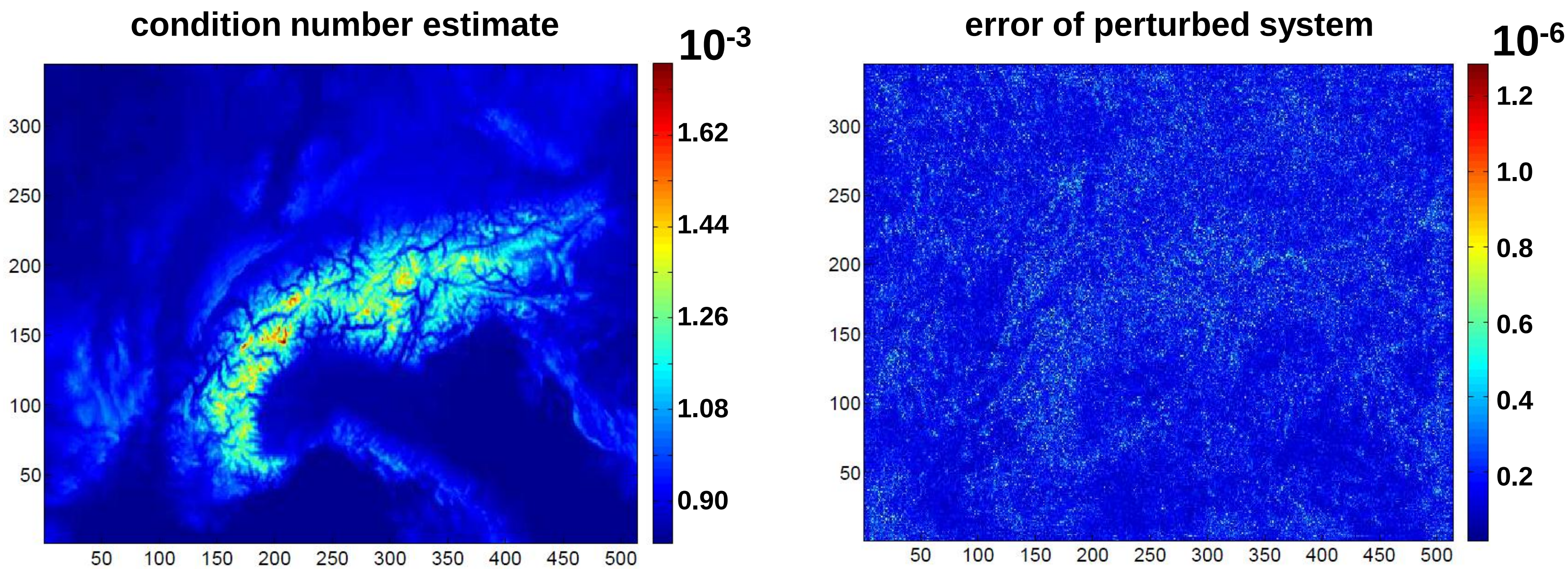
$$\frac{|\delta w|}{|w|} \leq \frac{\kappa(M)}{1 - \kappa(M)|\delta M|/|M|}\left(\frac{|\delta M|}{|M|} + \frac{|\delta d|}{|d|}\right)$$

$$\frac{|\delta w|}{|w|} \leq \kappa(M)\frac{|\delta d|}{|d|} \qquad \text{if } \delta M = 0$$

- Condition numbers are typically $O(10^4)$

Worst vs. Effective Error

- Perturb system with error O(10-7) and check result



Riedinger (2012)

- How can we be sure that this result is robust?

Eigenvector/-value Analysis

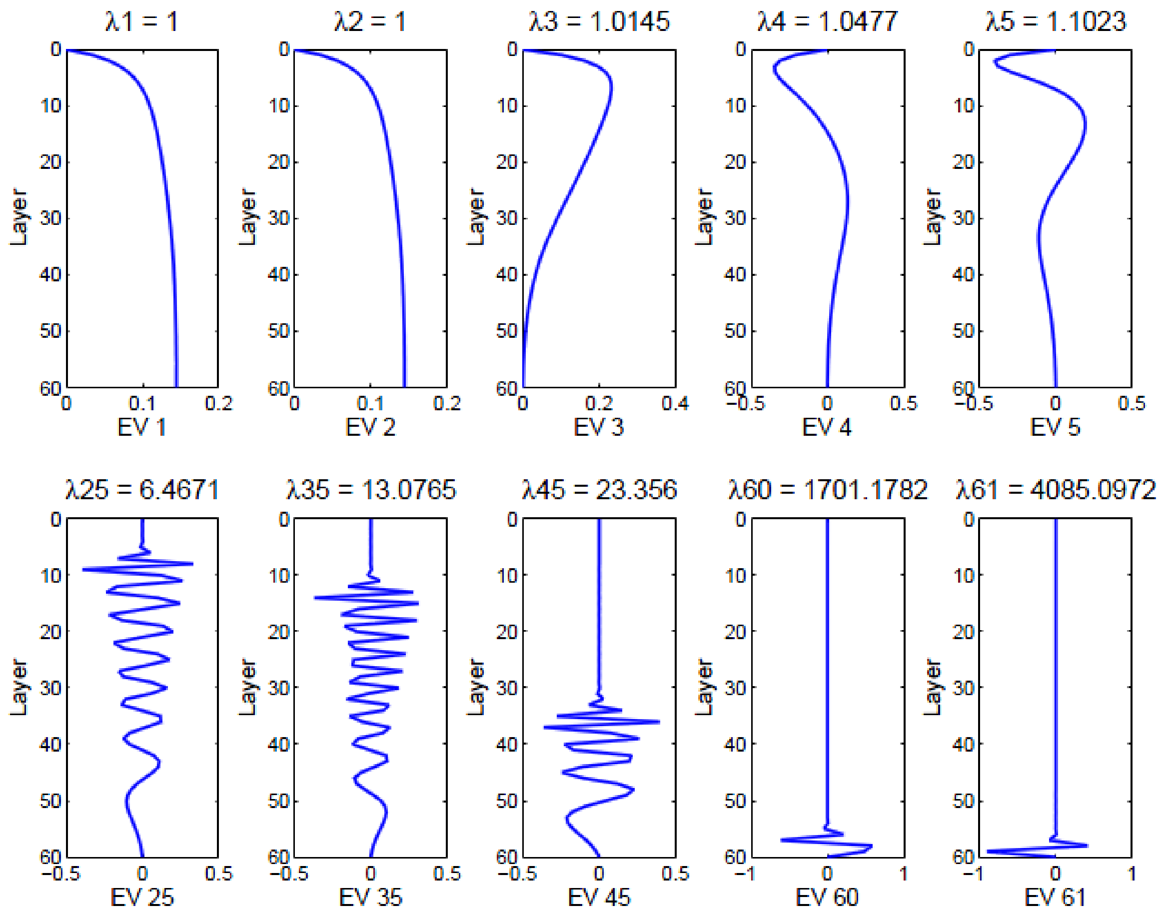
$$M\ (w + \delta w) = d + \delta d$$

$$M(e_1 + e_2) = \lambda_1 e_1 + \lambda_2 e_2$$

$$\Longrightarrow \quad \left|\frac{\delta w}{w}\right| = \left|\frac{\lambda_1}{\lambda_2}\right| \left|\frac{\delta d}{d}\right|$$

δw

w



Riedinger (2012)

Preconditioner

- Reduce condition number by pre-multiplying with matrix P

$$P\,M\,w = P\,d \qquad \text{e.g. } M^{-1}Mw = M^{-1}d$$

- Jacobi preconditioner

$$P = \begin{bmatrix} 1/m_{11} & & 0 \\ & \ddots & \\ 0 & & 1/m_{nn} \end{bmatrix} \quad \text{where} \quad M = \left[m_{ij}\right]$$

- Condition number reduced to $O(10^2)$ from $O(10^4)$

Performance Results

- Fast waves solver in rewritten HP2C code on GPU

Name	Without preconditioner	Jacobi Preconditioner	Diagonal Preconditioner
double precision	5.34 s	5.38 s	5.36 s
mixed precision	4.24 s	—	—
single precision	2.52 s	2.51 s	2.52 s

Riedinger (2012)

- Largest error of $O(10^{-4})$ occurs in computation of divergence

$$D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \quad \text{i.e.} \quad u(i+1) - u(i)$$

Discussion

- Reduction in arithmetic precision can substantially reduce the runtime
- Current model code has not been carefully written for lower precision
- Analysis of fast waves solver shows that reduction is possible, critical point is in computation of the divergence
- Preconditioning helps reduce maximum possible error

Questions

- Any feelings/reasons why we need double precision?
- Any ideas for writing divergence in perturbation form?
- How would you test the sensitivity of your application to numerical precision?