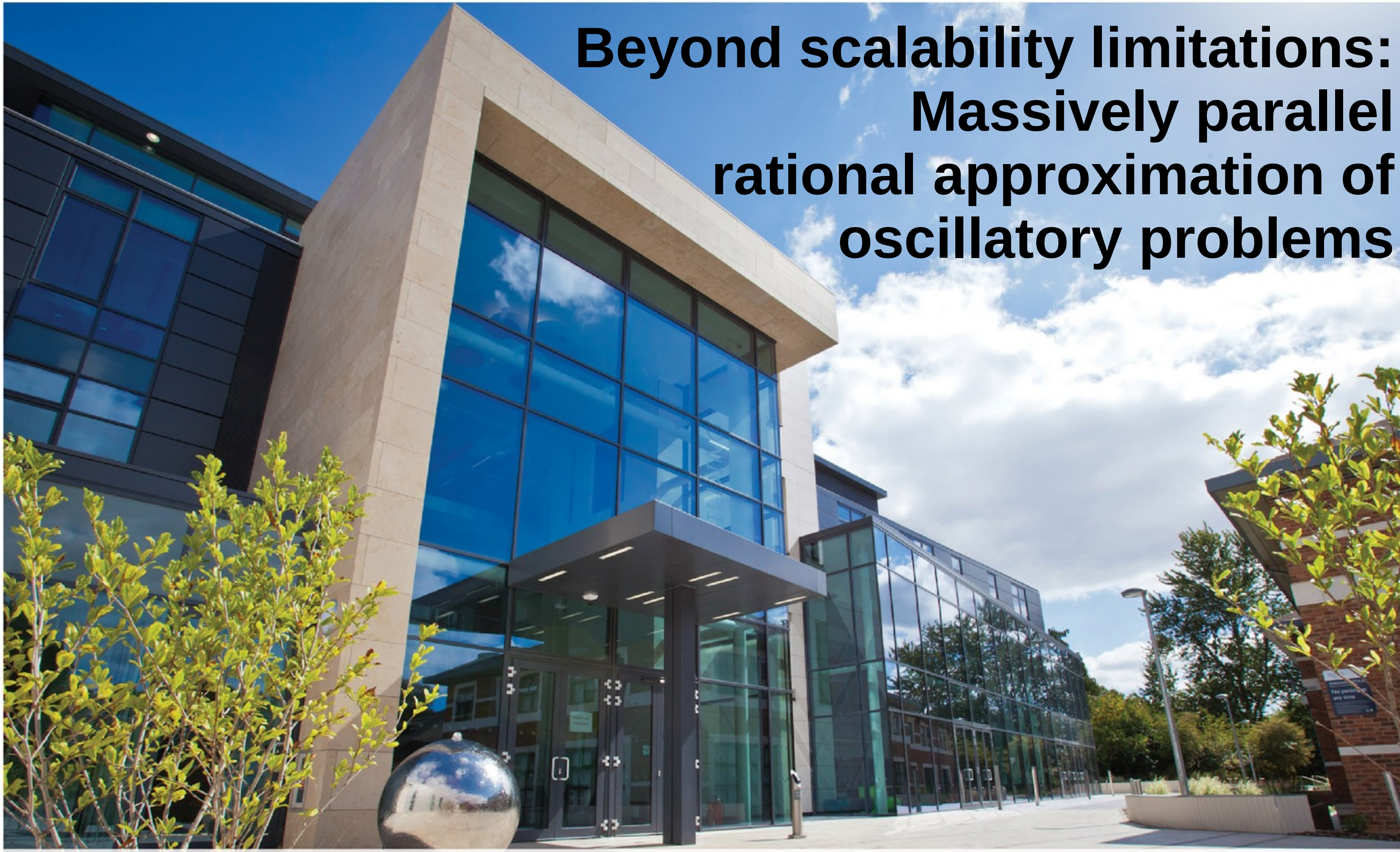
Beyond scalability limitations: Massively parallel rational approximation of oscillatory problems

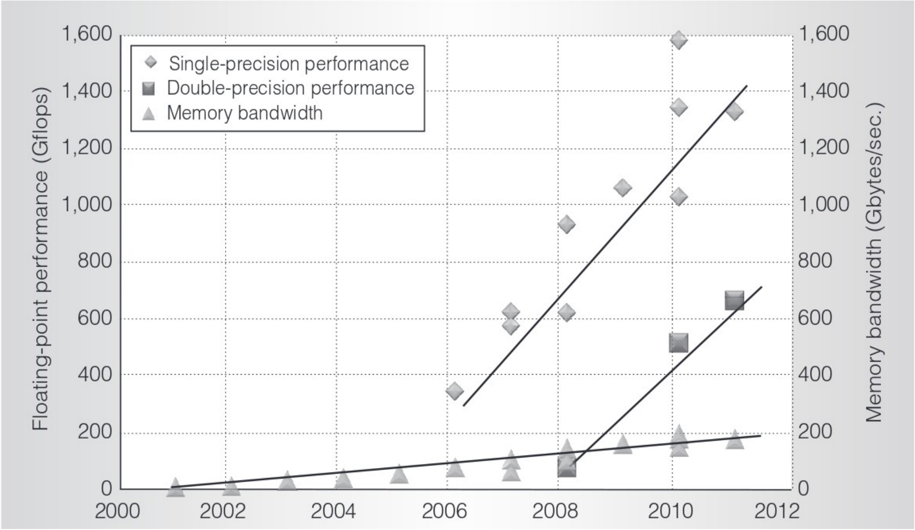


UNIVERSITY OF EXETER

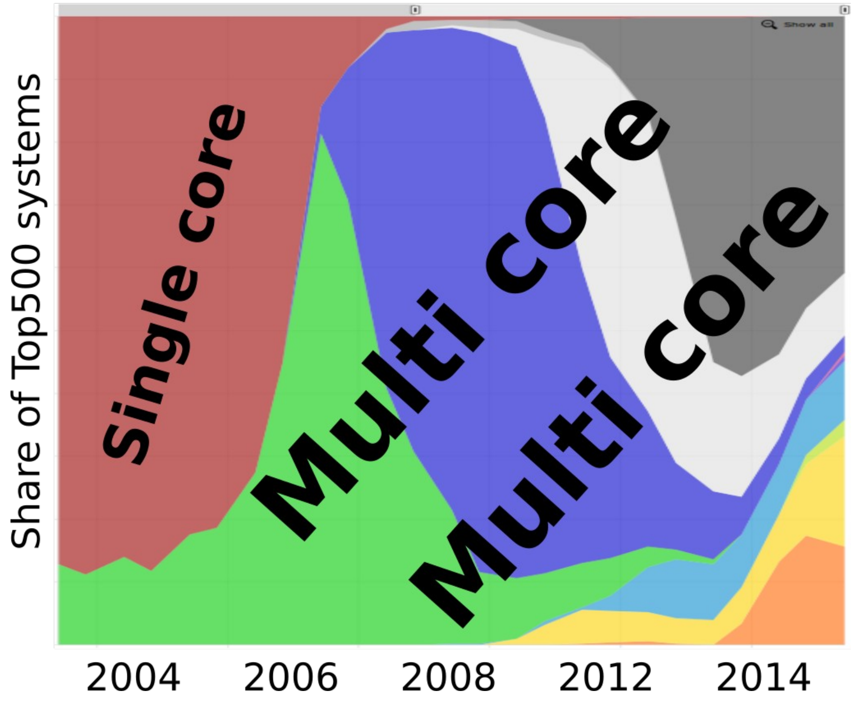
Martin Schreiber <M.Schreiber@exeter.ac.uk>
with Terry Haut, Pedro Peixoto, Beth Wingate

Computational limitations: HPC trend towards data parallelism

- Increase in frequency stagnated
- Performance increase dominated by increase in data parallelism
  => multi-core, SIMD
- Bandwidth/Latency bottlenecks increasing:



Source: Top500.org

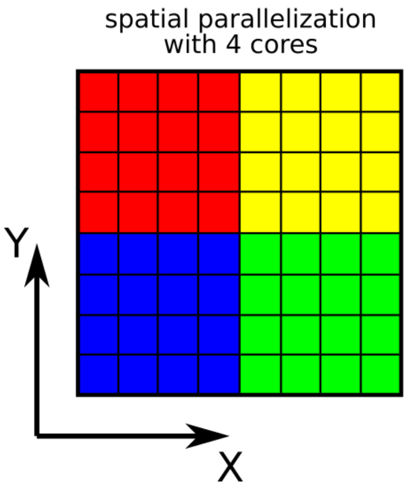


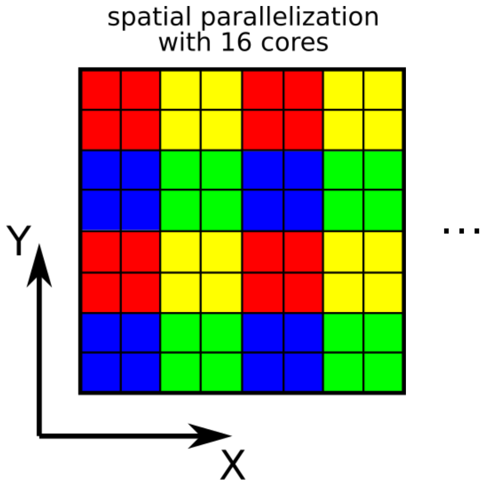
GPUs and the future of parallel computing
Stephen W. Keckler, William J. Dally, Brucek Khailany, Michael Garland, David Glasco

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Further challenges + limitations for real time requirements (between strong and weak scaling)

- **Challenges**: Varying core frequency, system noise, caching side-effects / false sharing, fault tolerance, memory-bandwidth limitation, threading overheads (starting), synchronization overheads, hardware scaling, communication, etc. are limiting performance for strong scaling problems





- What to do if scalability limitation is reached or we need run the simulations faster?

Img Source: http://portal.uni-freiburg.de/aam/abtlg/wissmit/agkr/muellert/finite-volume-schemes-for-the-shallow-water-equations-on-the-sphere

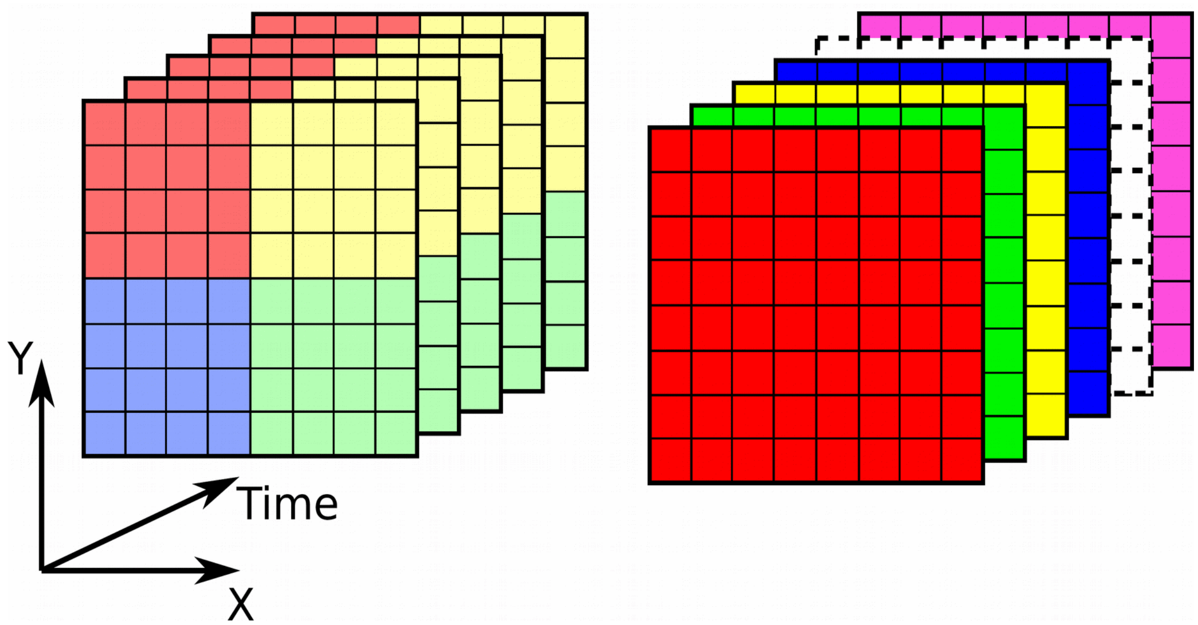
Martin Schreiber <M.Schreiber@exeter.ac.uk>

Parallelization-in-time

Use time as an additional “dimension” of parallelism

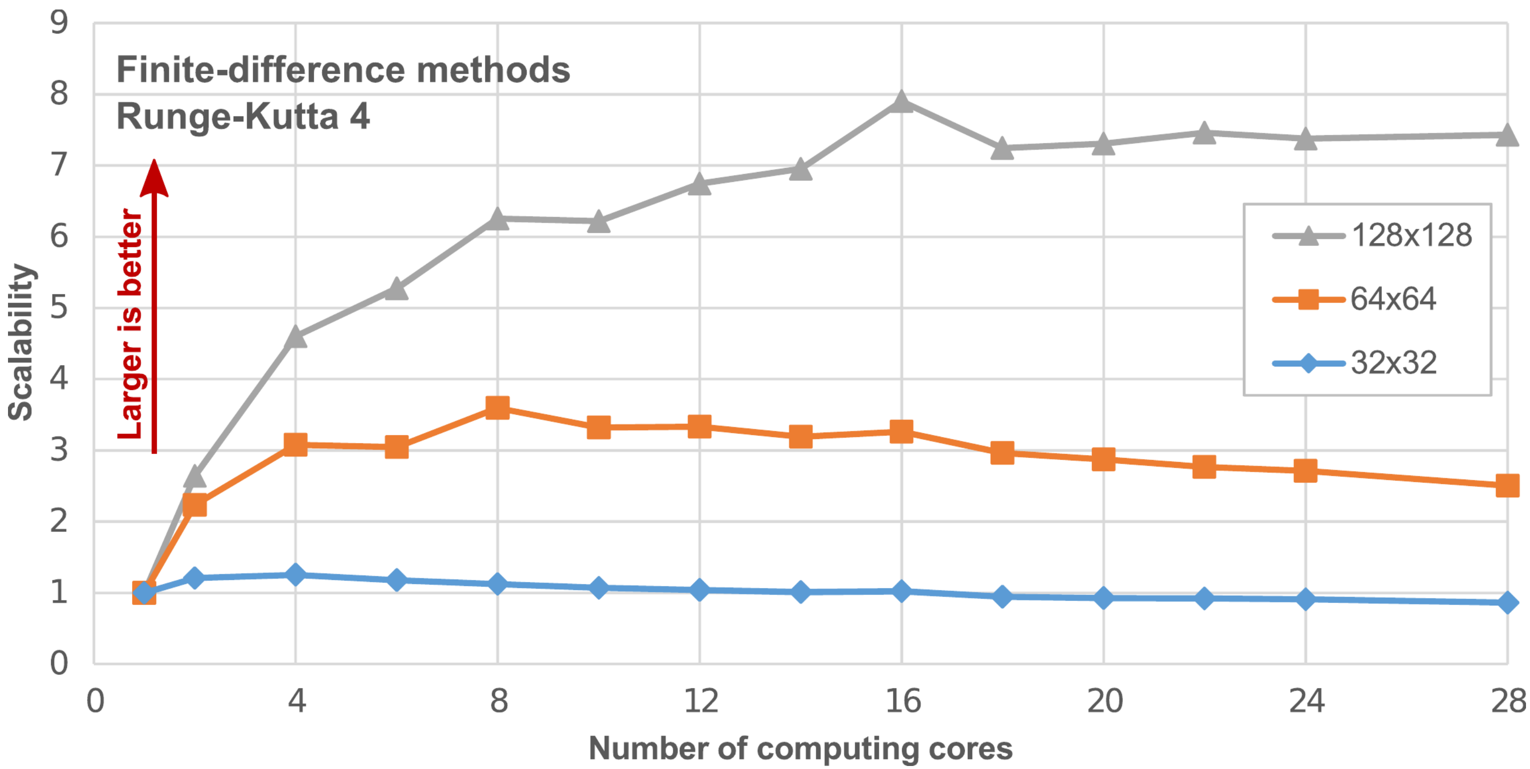
Standard time stepping:
Compute time step,
advance in time

Parallelization in time:
Potential to compute solutions
ahead of current time step



Martin Schreiber <M.Schreiber@exeter.ac.uk>

Example: Lin. Shallow-water equations



- **2x 14 cores, Intel Xeon(R) CPU E5-2697, no hyperthreading, compact affinities**
- **Finite-difference method**, linear parts of SWE, Runge-Kutta 4, C-grid (staggered)
- Shared-memory parallelization only, no distributed-memory communication overheads
- **Scalability limited**

Martin Schreiber <M.Schreiber@exeter.ac.uk>

A brief and (definitively incomplete!) overview of Parallelization in time

- Parallelization-in-time aims at new degrees of parallelization
- Variety of methods available:
  - **“Iterative” in time:**
    - Spectral deferred corrections (SDC)
    - Revisionist Integral Deferred Correction (RIDC)
    - Parareal
  - **“Direct” in time with exp. Integrators:**
    - ParaExp (direct without non-linearities)
    - *Rational approximation of an exponential integrator (REXI)*
  - Mixture (e.g. use ParaExp/REXI with Parareal)

[Gander] 50 Years of Time Parallel Time Integration, Martin J. Gander
[Parareel] Résolution d’EDP par un schéma en temps «pararéel»,
Jacques-Louis LIONS, Yvon MADAY, Gabriel TURINICI

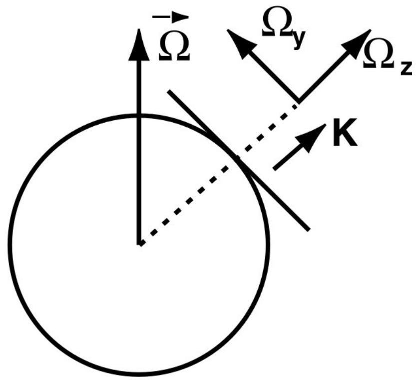
Parallelization in time for linear part of shallow water equations on f-plane

Martin Schreiber <M.Schreiber@exeter.ac.uk>

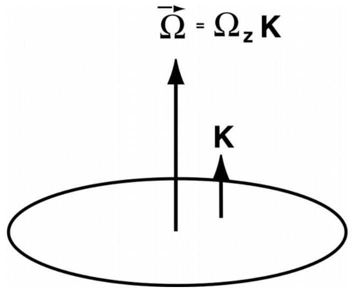
Shallow-water equation on f-plane

Shallow-water equations are used as test cases to develop dynamical cores for climate and weather simulations

- Coriolis force for simulation on the sphere



- Consider only small area with constant Coriolis frequency



- **Advective (non-conservative) formulation with** $U := (\eta, u, v)^T$

$$U_t = L(U) + N(U)$$

$$L(U) := \begin{pmatrix} & & \\ -g\partial_x & & f \\ -g\partial_y & -f & \end{pmatrix} U$$

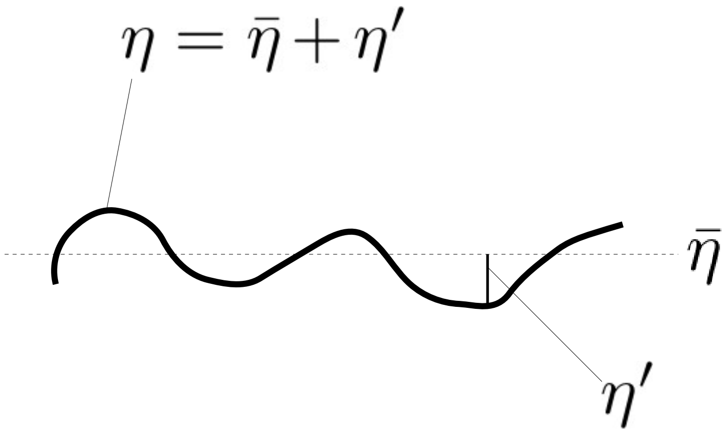
$$N(U) := \begin{pmatrix} -(\eta u)_x - (\eta v)_y \\ -u u_x - v u_y \\ -u v_x - v v_y \end{pmatrix}$$

Image source:
Atmospheric and oceanic fluid dynamics, Geoffrey K. Vallis, http://weknowyourdreams.com/earth.html

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Shallow-water equation on f-plane

- Mean height $\bar{\eta}$
- Perturbation η'
- Fluid height



- Linear operator:
  Use mean height + perturbation

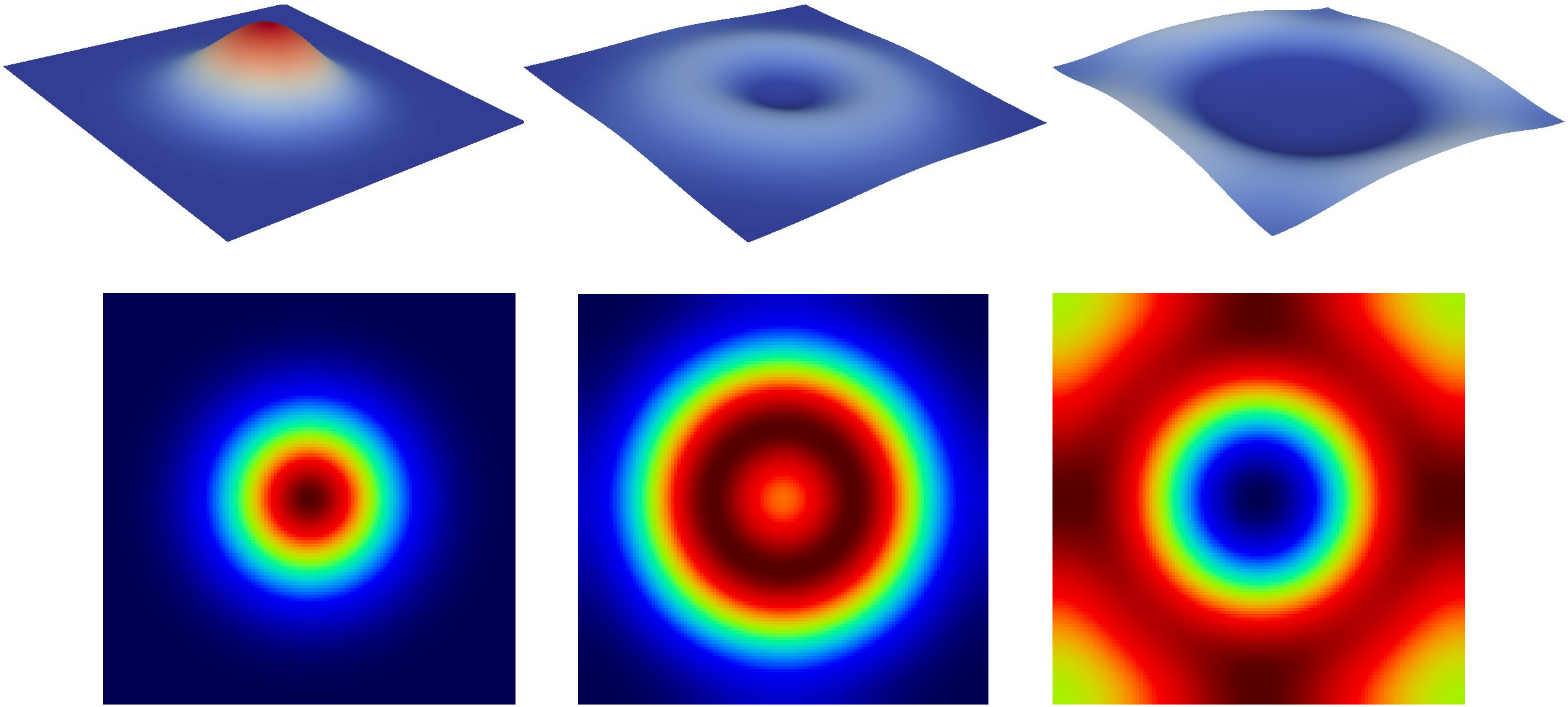
$$L(U) := \begin{pmatrix} 0 & -\bar{\eta}\partial_x & -\bar{\eta}\partial_y \\ -g\partial_x & 0 & f \\ -g\partial_y & -f & 0 \end{pmatrix} U$$

- In this work, we **neglect the non-linear parts** (future work)

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Pictures! Pictures! Pictures!

Height field on biperiodic plane



Martin Schreiber <M.Schreiber@exeter.ac.uk>

Exponential integrators

Exponential integrator?!?

- For linear operator L, the exponential integrator is given by

$$\vec{u}(t+\Delta t) := e^{\Delta t\, L}\, \vec{u}(t) \qquad L(U) := \begin{pmatrix} 0 & -\bar{\eta}\partial_x & -\bar{\eta}\partial_y \\ -g\partial_x & 0 & f \\ -g\partial_y & -f & 0 \end{pmatrix} U$$

- Exponential integrators are a **non-standard time integrators**: Different to RKn, Leapfrog, …
- **No error** in time
- **Arbitrary long time step size**
  => Does not suffer of small time steps for highly oscillatory solution
- **Challenging** to find efficient solver

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Computing exponential integrators

- Efficient computation of exponential integrator?
- Straightforward Eigenvector decomposition too expensive (not feasible for large matrix) and too memory demanding

$$e^{\mathbf{A}} = \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{A}^k = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\mathbf{T} \mathbf{\Lambda} \mathbf{T}^{-1} \right)^k \qquad e^{\mathbf{A}} = \mathbf{T} \begin{bmatrix} e^{\lambda_1} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n} \end{bmatrix} \mathbf{T}^{-1}$$

- Different solver strategies exist: Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later, Cleve Moler, Charles Van Loan
- We use the property that EVs are all imaginary: **exp(i x)** (skew Hermitian matrix L)

$$L(U) := \begin{pmatrix} 0 & -\bar{\eta}\partial_x & -\bar{\eta}\partial_y \\ -g\partial_x & 0 & f \\ -g\partial_y & -f & 0 \end{pmatrix} U$$

- See also: High-order time-parallel approximation of evolution operators Terry Haut, T. Babb, P. G. Martinsson, B. Wingate

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Step 1) Approximation of exp(ix)

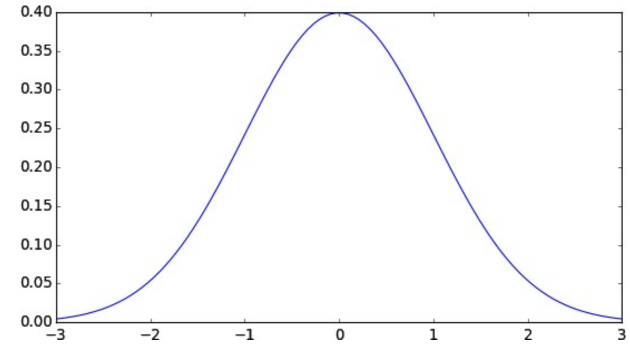
- Use Gaussian as basis function

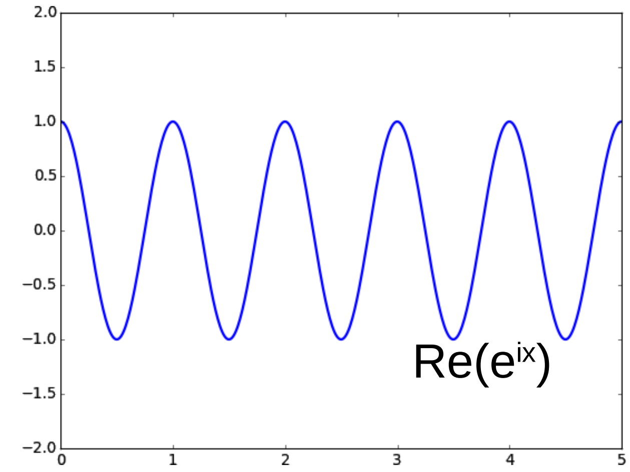
$$\psi_h(x) := (4\pi)^{-\frac{1}{2}} e^{-x^2/(4h^2)}$$

- Use superposition to approximate **e**^{ix}

$$e^{ix} \approx \sum_{m=-M}^{M} b_m \psi_h(x + mh)$$

h Sampling accuracy (cf. Nyquist theorem)
M Number of samples

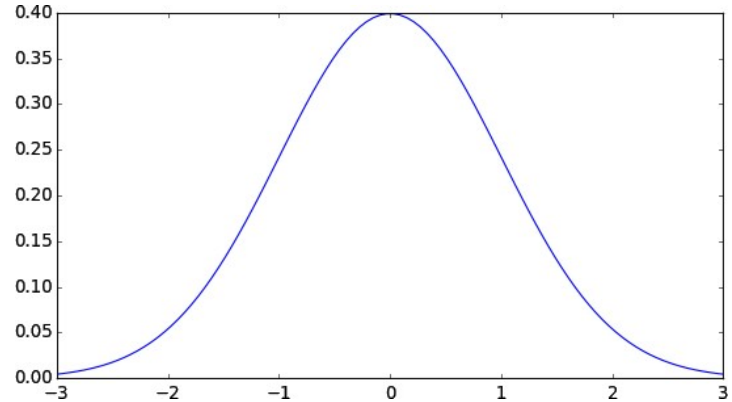




On approximate approximations using Gaussian kernels, V. Maz'ya, G. Schmidt

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Step 2) Approximation of Gaussian basis function



Use rational approximation or Gaussian basis function

$$\psi_h(x) := (4\pi)^{-\frac{1}{2}} e^{-x^2/(4h^2)}$$

Tabulated values

$$\psi_h(x) \approx Re\left(\sum_{l=-L}^{L} \frac{a_l}{i\frac{x}{h} + (\mu + i\,l)} \right)$$

Fast and accurate con-eigenvalue algorithm for optimal rational approximations., T. S. Haut & G. Beylkin

Approximation of an approximation

$$b_m = e^{-imh} e^{h^2}$$

- Approximation of function

$$e^{ix} \approx \sum_{m=-M}^{M} b_m \sum_{l=-L}^{L} Re\left(\frac{h a_l}{ix + h(\mu + i(m+l))}\right)$$

Approximation of Gaussian basis function

Approximation of exp(ix)

- Specialize on exp(ix):

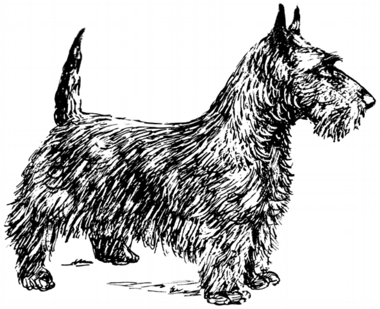
$$e^{ix} \approx \sum_{n=-N}^{N} Re\left(\frac{\beta_n^{Re}}{ix + \alpha_n}\right) + i\, Re\left(\frac{\beta_n^{Im}}{ix + \alpha_n}\right)$$

High-order time-parallel approximation of evolution operators, Terry Haut, T. Babb, P. G. Martinsson, B. Wingate

Rational approximation of an exponential integrator: (T)REXI

- Approximation of real values of exp(ix):

$$e^{ix} \approx \sum_{n=-N}^{N} Re\left(\frac{\beta_n}{ix+\alpha_n}\right)$$


Terry(er)

- Approximation to linear operator L:

Complex shifted poles

$$U(\tau) = e^{\tau L}U(0) \approx \sum_{n=0}^{N} \gamma_n^{Re} \underbrace{(\tau L + \alpha_n)}^{-1} U(0)$$

!!! System of equations to solve !!!

High-order time-parallel approximation of evolution operators, Terry Haut, T. Babb, P. G. Martinsson, B. Wingate

Solving $(L+alpha)^{-1}$

$$L(U) := \begin{pmatrix} 0 & -\bar{\eta}\partial_x & -\bar{\eta}\partial_y \\ -g\partial_x & 0 & f \\ -g\partial_y & -f & 0 \end{pmatrix} U$$

- Step 1)
  Reformulate as Helmholtz equation and solve for height:

$$((\alpha^2 + f^2) - g\bar{\eta}\Delta)\eta = \frac{f^2 + \alpha^2}{\alpha}\eta_0 - \bar{\eta}\delta_0 - \frac{f\bar{\eta}}{\alpha}\zeta_0$$

- Step 2)
  Directly compute velocity (u,v):

$$U = A_\alpha^{-1}(U_0 - g\nabla\eta)$$

$$A_\alpha^{-1} = \frac{1}{f^2 + \alpha^2}\begin{pmatrix} \alpha & f \\ -f & \alpha \end{pmatrix}$$

Time for results

Analytical & parallel performance results

Martin Schreiber <M.Schreiber@exeter.ac.uk>

SWEET:

Shallow Water Equations Environment for Test, Awesome!

(can be used for more than just SWE)

Framework for 2D simulations on regular Cartesian grid

Unified equation programming model:

- Use same notation for spectral and FD methods
- Example: `h += dt*(u*h + v*h);`
- Support for different grid layouts, e.g. Arakawa A- and C-grids

Space-discretization:

- Spectral methods (Similar to SPH-basis of ECMWF model)
- Finite differences (Similar to FD in ENDGame, MPAS)

Time-stepping methods:

- Euler
- Runge-Kutta 2,3,4
- **REXI**

Parallelization:

- **Space**: OpenMP
- **Time**: MPI/OpenMP

Fast trigonometric-based spectral solvers:

- Allows computing certain solutions directly
- E.g. for Poisson, (specific) Helmholtz problems

SWEET is Open Source: https://github.com/schreiberx/sweet
(Code should be part of HPC publication to allow reproducibility of results)

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Evaluation of (T)REXI: Accuracy results

$$f(x) \approx \sum_{m=-M}^{M} b_m \psi_h(x + mh)$$

- How to choose h and M?
  - Number of samples **M**
  - Sampling accuracy **h** (cf. Nyquist theorem)
- What are the dependencies of h and M?
  - **N**: Resolution NxN
  - **L** operator (frequencies)
  - Used solver for $(\mathbf{L}+a)^{-1}$ (spectral or finite differences)
  - Time step size **dt**

Martin Schreiber <M.Schreiber@exeter.ac.uk>

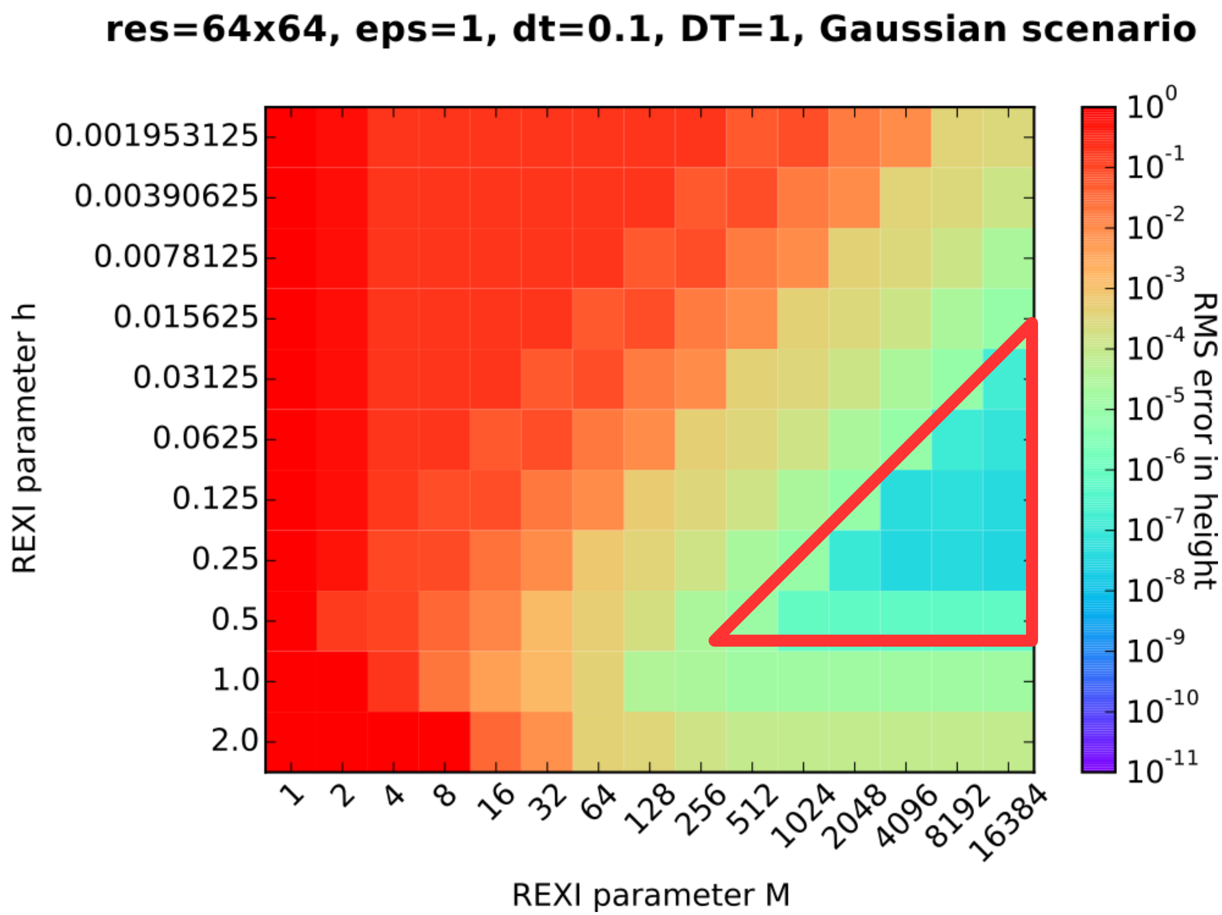
Evaluation of (T)REXI for varying h & M

Test environment:

- Spectral method
- Spectral solver for REXI term $(L+a)^{-1}$

Target:
Maximize h (accuracy) to **minimize M** (computational workload)

- Optimum of h close to 0.2
- M for this test case close to 1024

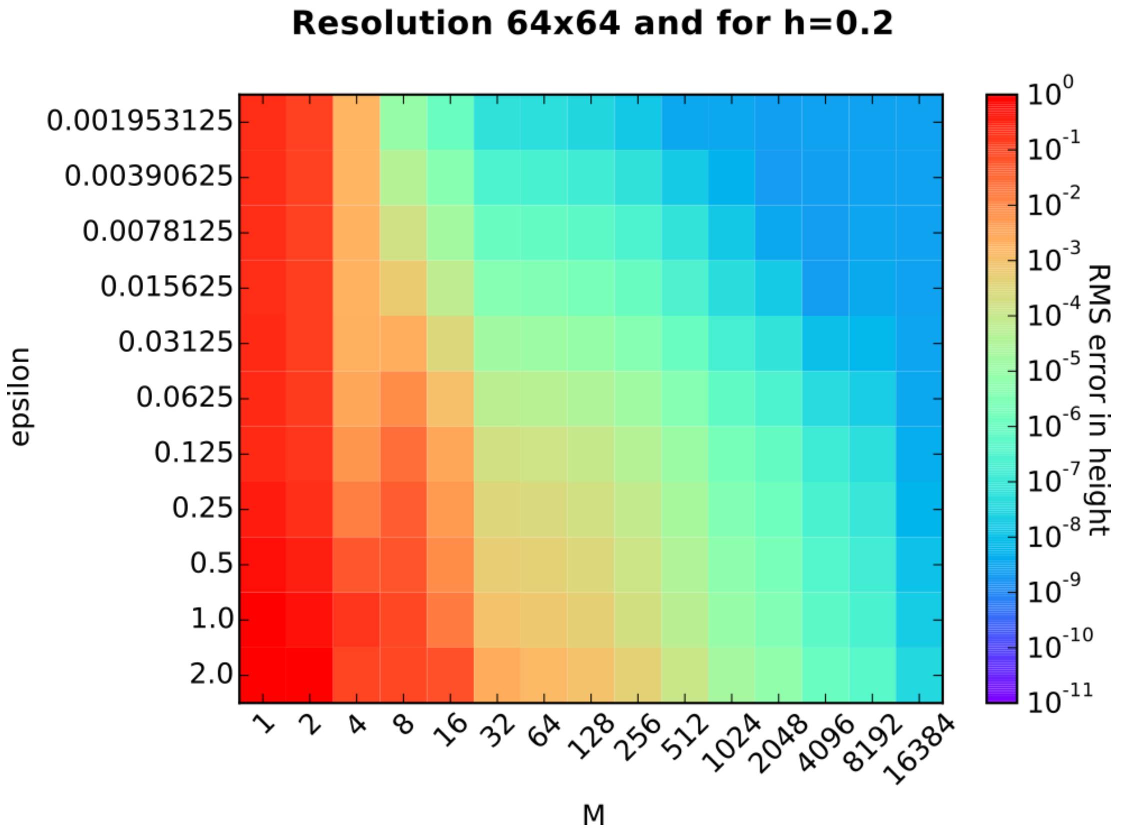


Martin Schreiber <M.Schreiber@exeter.ac.uk>

Parameter studies for oscillatory behavior

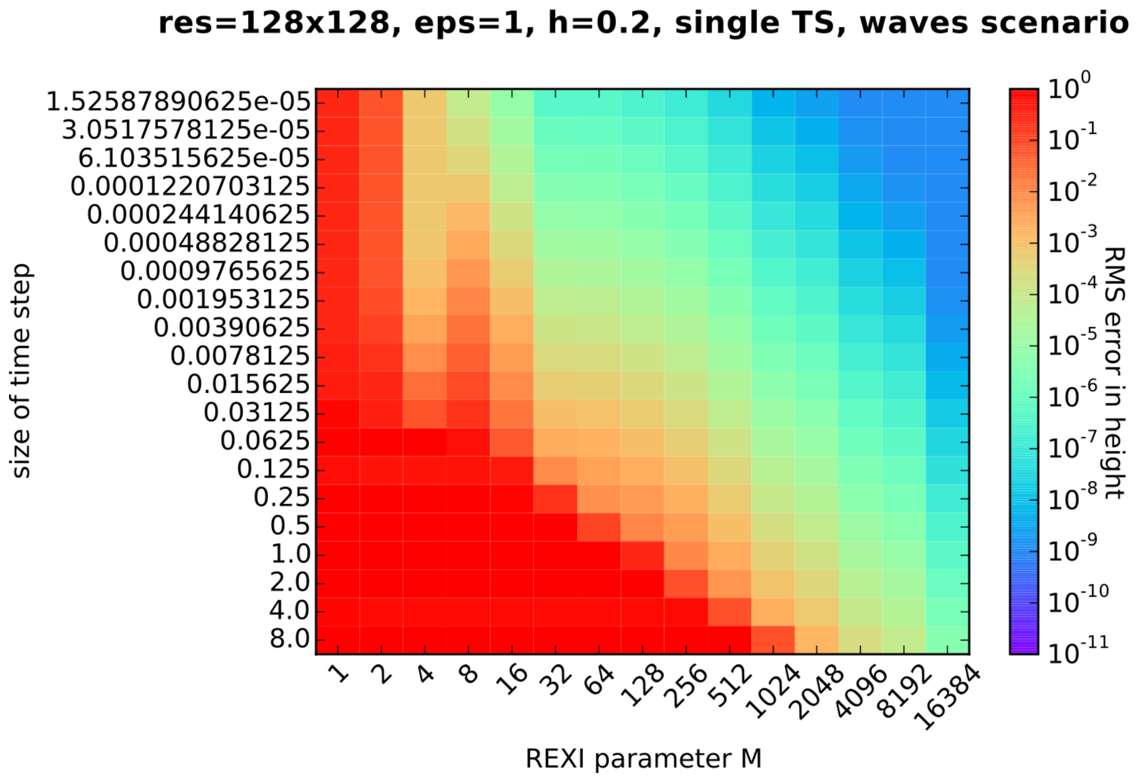
- Larger epsilon
  => More oscillations
- Increasing number of poles M required for larger epsilon

$$U_t = \epsilon L(U)$$



Dependency of M to size of large time step

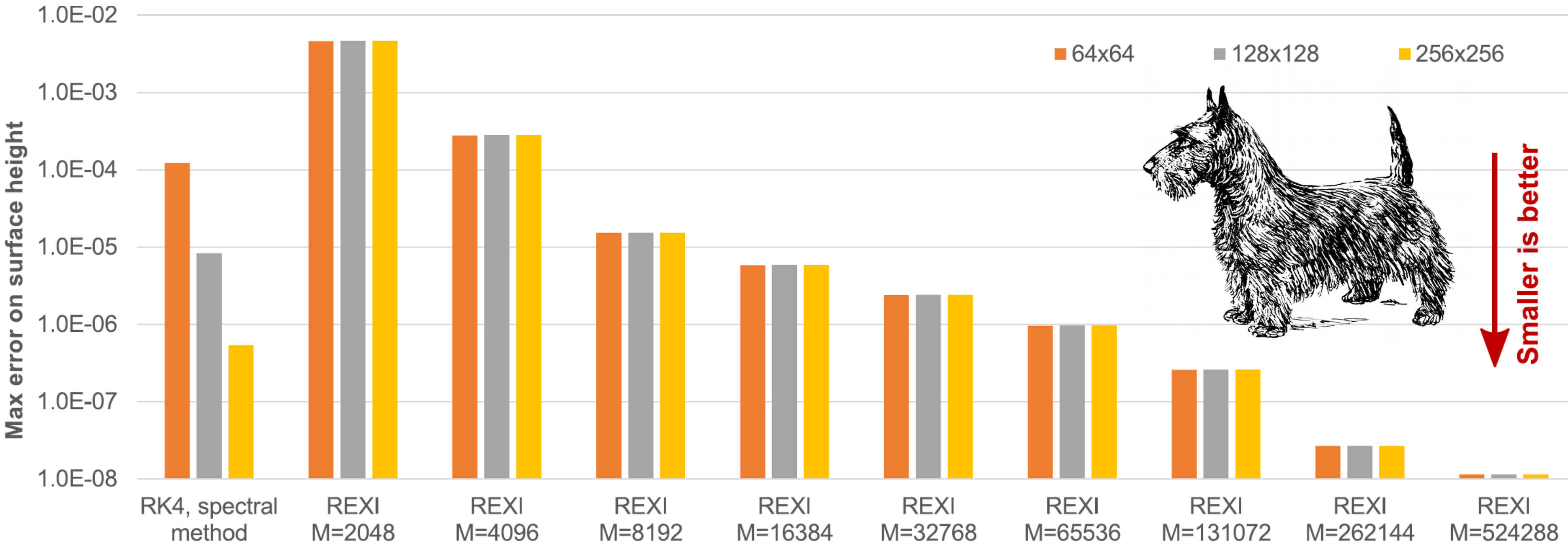
- Larger time step => Larger M
- We can observe a linear dependency of M to dT
- Larger time step sizes result in an increase in parallelism!



Martin Schreiber <M.Schreiber@exeter.ac.uk>

Spectral methods vs. (T)REXI

dt=5.0, T=50.0, CFL=0.3, RK4



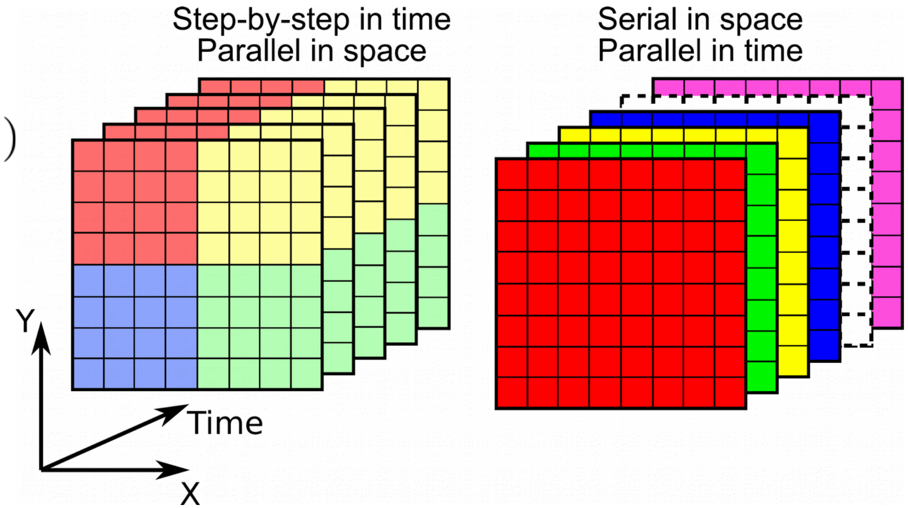
- Waves initial distribution
- Possibility of higher accuracy compared to solution with spectral method (Errors in time stepping dominate)

Martin Schreiber <M.Schreiber@exeter.ac.uk>

Performance, performance!
New dimension of parallelization

- New Parallelization degree over the sum!

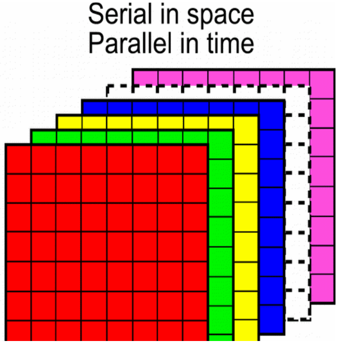
$$e^{\tau L}U(0) \approx \sum_{n=0}^{N} \gamma_n^{Re} \left(\tau L + \alpha_n\right)^{-1} U(0)$$

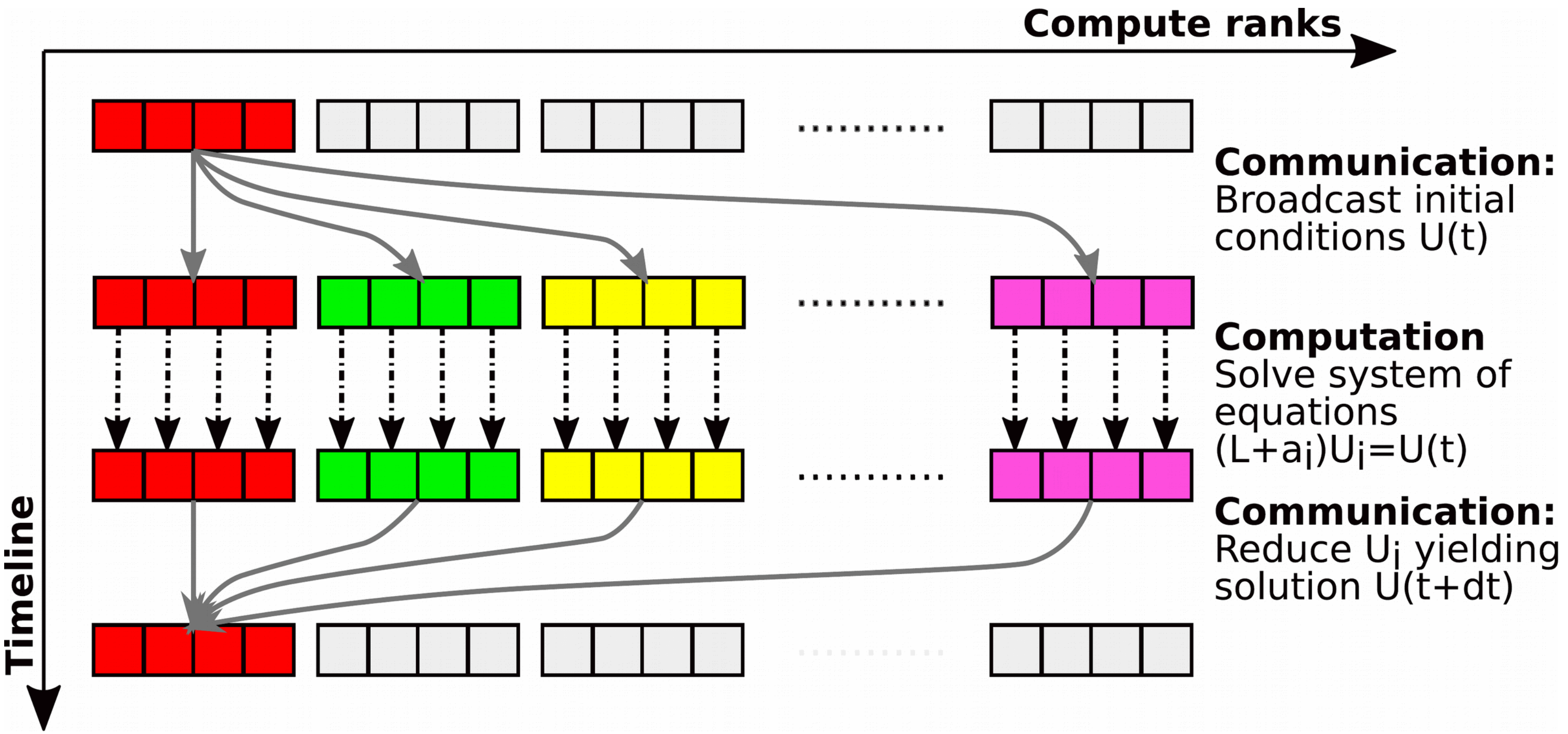


- Each term totally independent
- Parallelization
  - in **space** and sequential in time or
  - in **time** and sequential in space
  - both **in time and space**

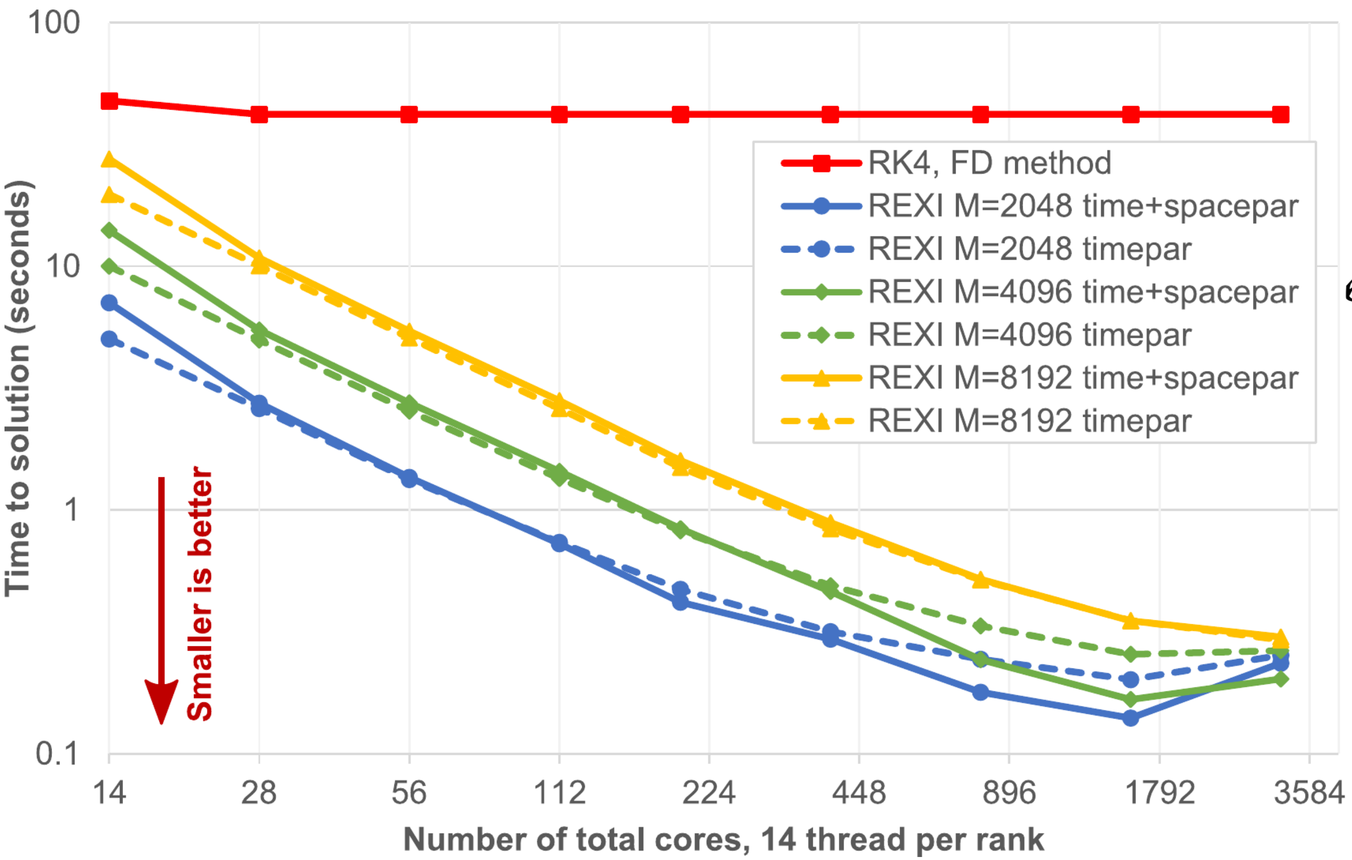
Martin Schreiber <M.Schreiber@exeter.ac.uk>

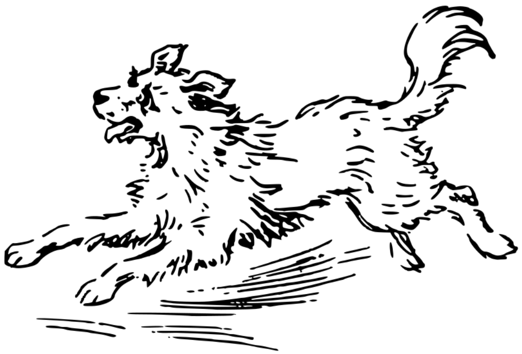
REXI:
One parallelization pattern





Performance: FD vs. (T)REXI





N=128x128

- **Two orders of magnitude faster with similar accuracy**

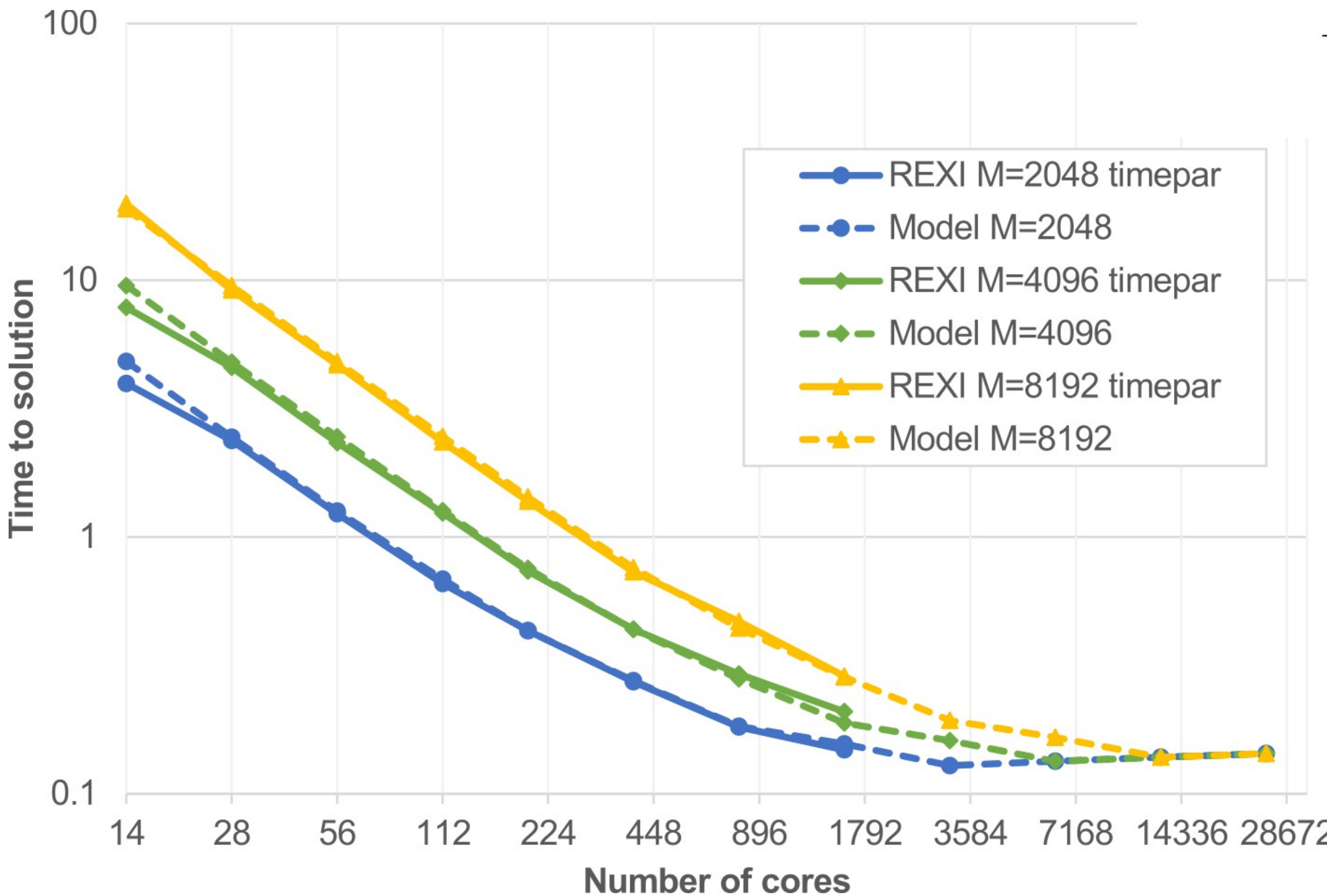
Helmholtz equation is directly solved in spectral space

Computed on Linux Cluster, LRZ / Technical University of Munich

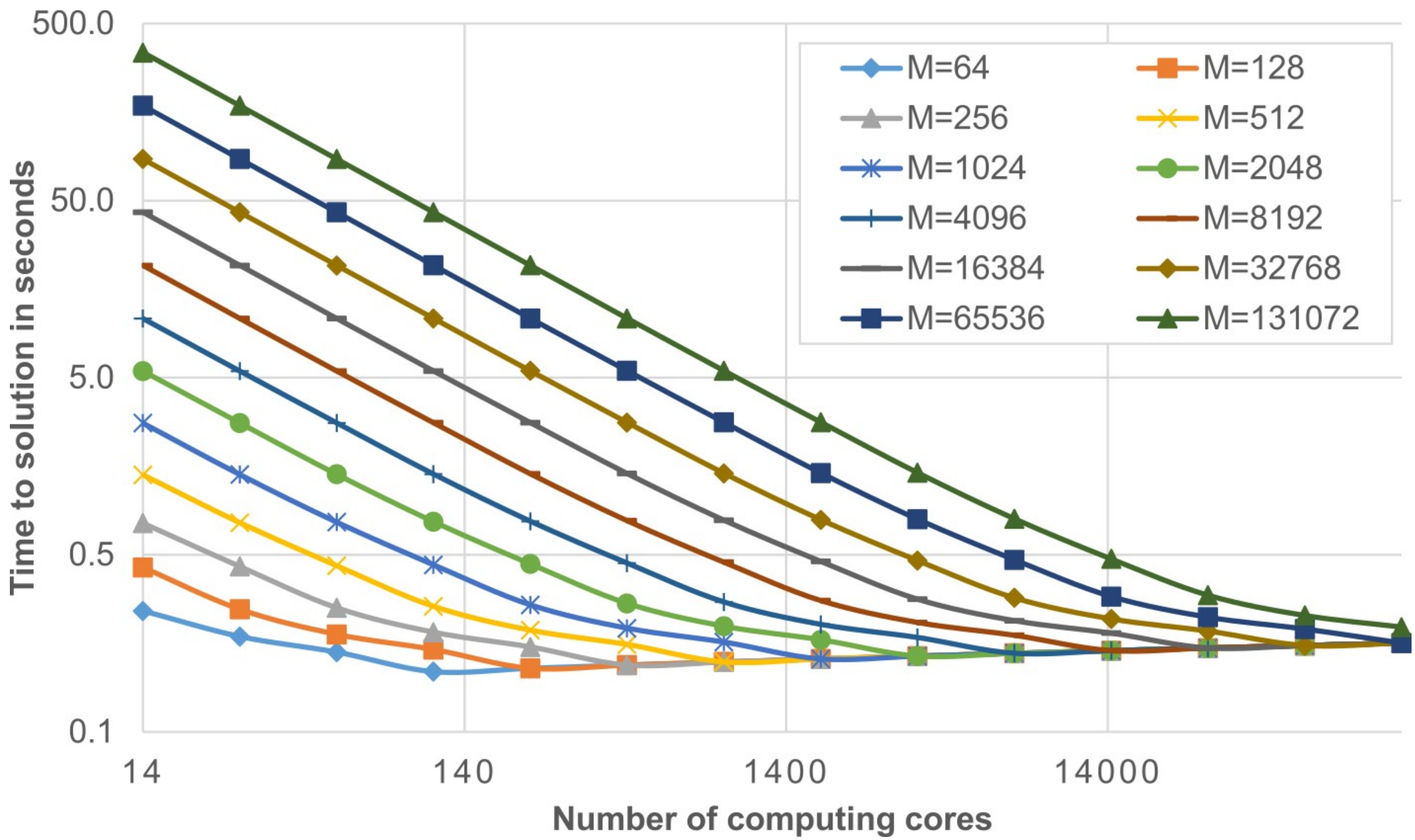
Martin Schreiber <M.Schreiber@exeter.ac.uk>

Performance modeling (FD methods)

$$T(C,M) = \underbrace{s_L}_{\text{serial part}} + \underbrace{s_B \log(C) + s_R \log(C)}_{\text{serial/parallel part}} + \underbrace{p_W \left\lceil \frac{W}{\min(C,W)} \right\rceil}_{\text{parallel part}}$$



Extrapolation with performance model



Martin Schreiber <M.Schreiber@exeter.ac.uk>

PinT it!

Summary

- Limitations of scalability in space
- Mathematical reformulation with exponential integrators typically computational expensive
- **(T)REXI** can be used to efficiently solve it massively parallel
- Significant speedups beyond scalability in space
- **Additional degree of parallelization** can be used for performance boost, by avoiding limited scalability in space
- If spectral solver is not applicable, efficient **iterative solver** for $(L-a)^{-1}$ required

Outlook

- Include **non-linear** part of SWE with semi-Lagrangian method
- Several new research areas arising:
  - Price to pay?
  - Additional energy consumption?
  - New heterogeneous network connectivities
  - Vectorization over elements in time
  - Fault tolerance
  - ...
- Long-term vision:
  Develop standard set of numerical PinT kernels for broader experiments

PhD position available – Please contact me

UNIVERSITY OF EXETER
www.exeter.ac.uk
Martin Schreiber <M.Schreiber@exeter.ac.uk>