Breaking through the barrier of time integration for climate and weather simulations

Martin Schreiber

In collaboration with many others:
Jed Brown, Finn Capelle, François Hamon, Richard Loft, Thibaut Lunet, Michael Minion, Matthew Normile, Nathanaël Schaeffer, Joao G.C. Steinstraesser, Pedro S. Peixoto, Daniel Ruprecht, Raphael Schilling

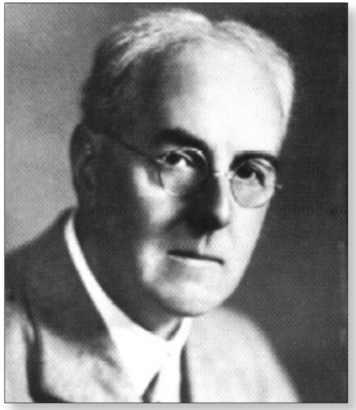
NHR PerfLab Seminar
March 26th 2024, Erlangen, Germany (virtual)

Université Grenoble Alpes

Inria

Laboratoire Jean Kuntzmann
Mathématiques Appliquées - Informatique

Computational weather forecasts (My first two heroes)



Lewis Fry Richardson (1881 - 1953)

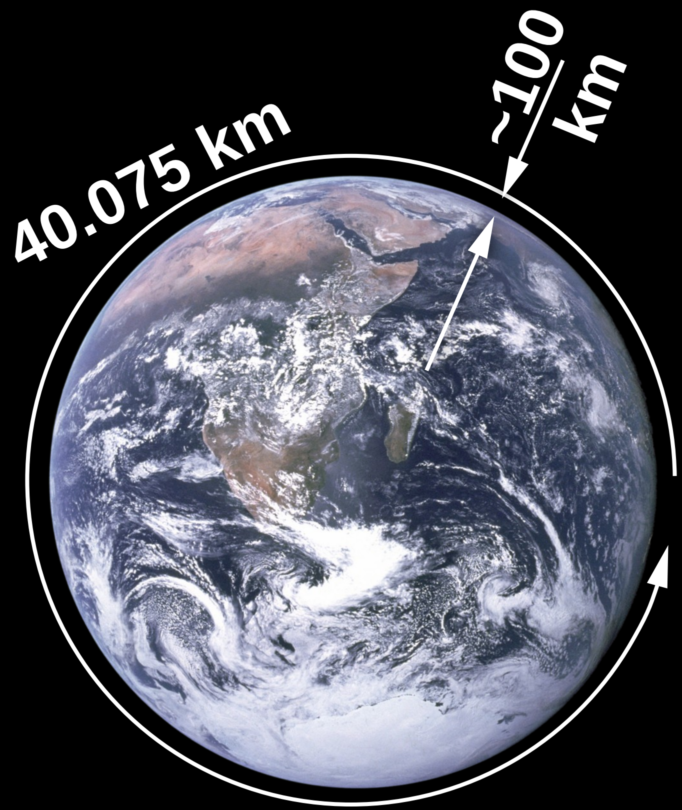
- **Visionary** about numerical weather forecasts by humans
- Computer technology not yet invented
- Computers = “s.o. who computes”

John von Neumann (1903 - 1957)
(stability analysis, von-Neumann architecture etc.)

- Genius in various areas
- Development of **1st “weather” forecast**
  - Using ENIAC (hard-wired & without von-Neumann arch.)
  - Simplified **single layer vorticity equation**

Images: https://en.wikipedia.org/, https://www.historyofinformation.com/image.php?id=15

Discretizing the atmosphere of the earth



Example

- ECMWF HRES (in 2023)
- Horizontal: **9km** resolution (O1280)
- Vertical: 137 levels, scales of **20m to 6km**

=> Different numerical properties in vertical & horizontal dimension

Horizontal (2D plane or on the sphere)

- Discretizations in space
- Mimetic properties
- ...
- Time integration methods

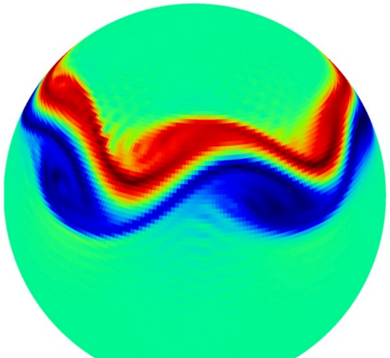
Research direction: Horizontal

3D Euler equation in **rotating frame** (non-closed, Lagrangian form)

$$\frac{\mathrm{D}\rho}{\mathrm{Dt}} = -\rho\nabla\cdot\mathbf{V}$$
$$\frac{\mathrm{D}\mathbf{V}}{\mathrm{Dt}} = -2\boldsymbol{\Omega}\times\mathbf{V} - \frac{1}{\rho}\nabla\mathrm{p} - \nabla\Phi + \mathbf{F}$$

Applying **simplifications**...

- Hydrostatic balance
- Constant density
- Depth integration
- ...



E.g. barotropic instability benchmark

…leads to **shallow-water equation** (closed, Lagrangian)

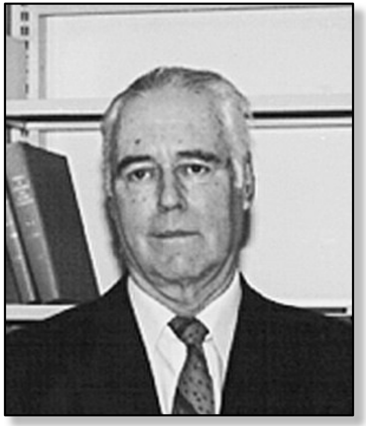
$$\frac{\mathrm{D}\Phi}{\mathrm{Dt}} = -\Phi\nabla\cdot\mathbf{V}$$
$$\frac{\mathrm{D}\mathbf{V}}{\mathrm{Dt}} = -\nabla\Phi - 2\Omega\sin\phi\mathbf{k}\times\mathbf{V}$$

Used to study **spatial** and **temporal discretizations**

- Includes **rotation effect**
- Includes (probably) all **terms relevant for time integration**

Properties of each term (and the 3rd hero)

$$\begin{bmatrix} \frac{\partial \Phi'}{\partial t} \\ \frac{\partial \mathbf{V}}{\partial t} \end{bmatrix} = \underbrace{\begin{bmatrix} -\overline{\Phi}\nabla \cdot \mathbf{V} \\ -\nabla \Phi \end{bmatrix}}_{L_g(U)} + \underbrace{\begin{bmatrix} 0 \\ -f\mathbf{k} \times \mathbf{V} \end{bmatrix}}_{L_c(U)} + \underbrace{\begin{bmatrix} -\mathbf{V} \cdot \nabla \Phi' \\ -\mathbf{V} \cdot \nabla \mathbf{V} \end{bmatrix}}_{N_a(U)} + \underbrace{\begin{bmatrix} -\Phi' \nabla \cdot \mathbf{V} \\ \end{bmatrix}}_{N_r(U)}$$



Source: https://en.wikipedia.org/

André Robert (1929 – 1993)

- **Linear**(ized) terms: **Implicitly treated** (1969)
  - Max. stable timestep size increase: 7x - 10x
  - Similar speedup with spherical harmonics
- **Non-linear** terms: **Semi-Lagrangian** treatment (1981)
  - Stable timestep size increase: 4x - 6x
  - **Wallclock** time reduction of about 2x

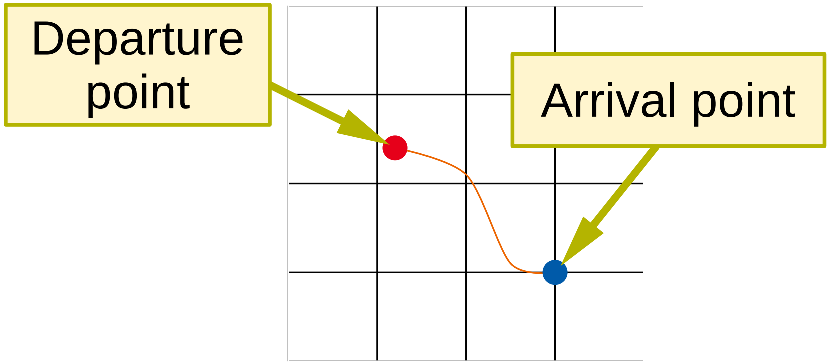
Nowadays state-of-the-art (ECMWF, MetOffice, Météo-France, etc.)

Semi-Lagrangian method (in a nutshell)

$$\begin{bmatrix} \frac{D\Phi'}{Dt} \\ \frac{D\mathbf{V}}{Dt} \end{bmatrix} = \underbrace{\begin{bmatrix} -\overline{\Phi}\nabla\cdot\mathbf{V} \\ -\nabla\Phi \end{bmatrix}}_{L_g(U)} + \underbrace{\begin{bmatrix} 0 \\ -f\mathbf{k}\times\mathbf{V} \end{bmatrix}}_{L_c(U)} + \underbrace{\begin{bmatrix} -\Phi'\nabla\cdot\mathbf{V} \\ \end{bmatrix}}_{N_r(U)}$$

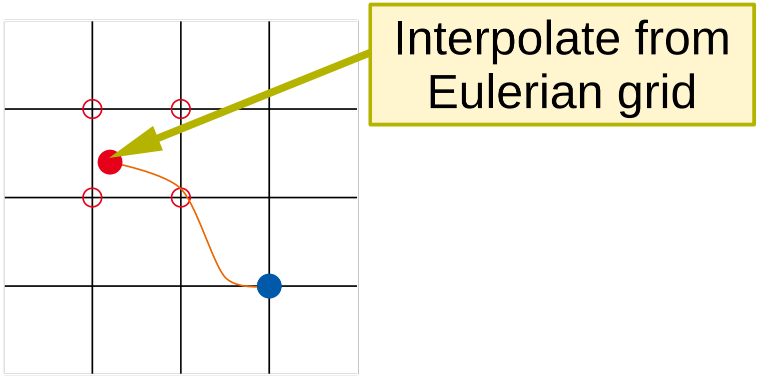
Step 1: Lagrangian view

Trace **departure** point backwards



Step 2: Eulerian view

Interpolate values from grid



- We use Stable Extrapolation Two-Time-Level Scheme (**SETTLS**)
- Well established scheme in **ECMWF's** dynamical core

Exponential integration Part I

With Richard Loft, Nathanael Schaeffer

Exponential integration with REXI

- Focus only on **linear PDE**

$$U_t(t) = LU(t)$$

- Exponential integration leads to

$$U(t + \Delta t) = e^{\Delta t L} U(t)$$

- **How to approximate efficiently** $\psi_0(K) = e^K$?

  E.g. using Krylov-based methods, polynomial approximations, or...

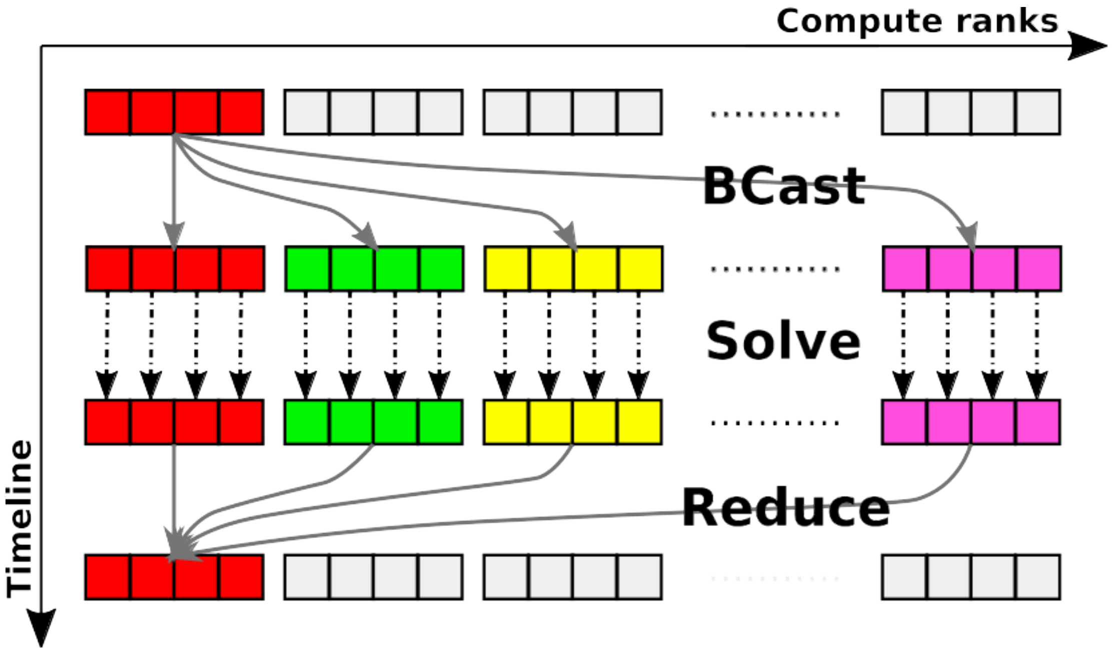
- **Rational Approximation of Exponential Integrators (REXI)**:

$$\begin{aligned} U(t + \Delta t) &= e^{\Delta t L} U(t) \\ &\approx \sum_n \beta_n (\Delta t L - I\alpha_n)^{-1} U(t) \end{aligned}$$

Exponential integration with REXI

- Replace exp() with **parallelizable part** => **parallel-in-time**

$$U(t+\Delta t) \approx \sum_n \beta_n (\Delta t L - I\alpha_n)^{-1} U(t)$$



WARNING: REXI terms solved with **spherical harmonics.** Otherwise much more challenging!!!

Rational approximation of exp. integration (REXI)

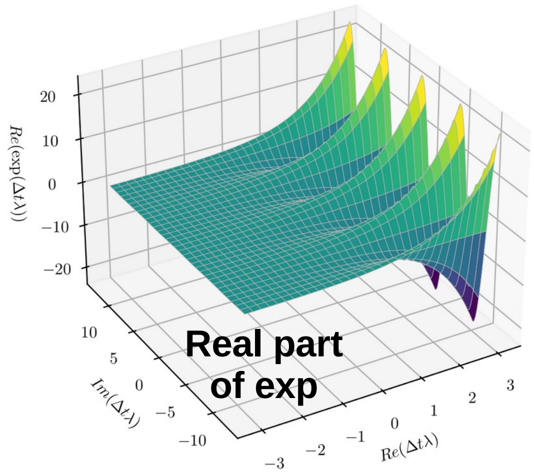
- For an **ODE** $u_t = \lambda u$ we can use a linear combination of rational terms

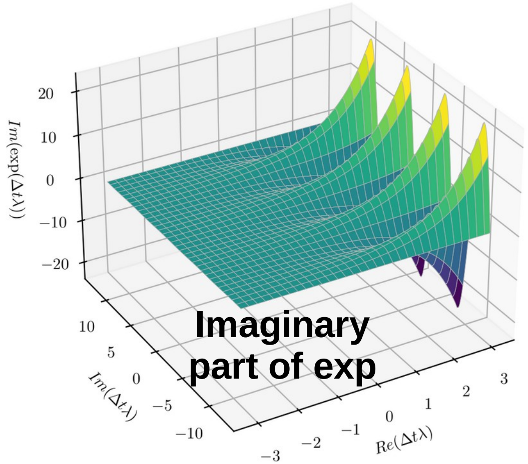
$$
\begin{aligned}
u(t+\Delta t) &= \exp(\Delta t \lambda) u(t) \\
&\approx \left(\sum_n \beta_n (\Delta t \lambda - \alpha_n)^{-1} \right) u(t)
\end{aligned}
$$

  to approximate the exp function

- We can do this similarly for a **PDE** $U_t = LU$

$$
\begin{aligned}
U(t+\Delta t) &= e^{\Delta t L} (\Delta t L) \, U(t) \\
&\approx \sum_n \beta_n (\Delta t L - I \alpha_n)^{-1} U(t)
\end{aligned}
$$



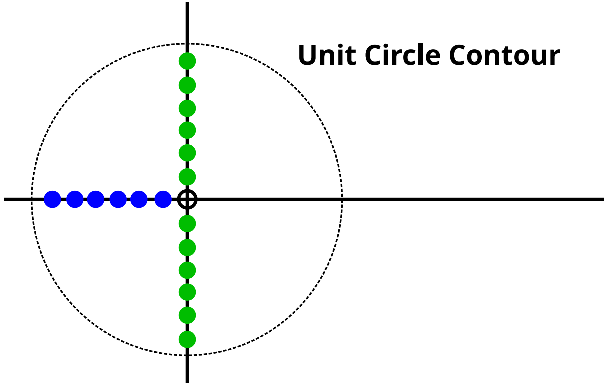


CI-REXI: Cauchy Contour Integral

- Cauchy contour integral is defined as

$$f(x_0) = \frac{1}{2\pi i} \oint_\Gamma \frac{f(z)}{z - x_0} dz$$

with $\boxed{f(z) = \exp(z)}$ to be approximated by REXI



- E.g., using a **circle contour** around the origin and a **trapezoidal rule** we get

$$\exp(x) \approx \sum_{n=1}^{N} \frac{\beta_n}{x + \alpha_n}$$

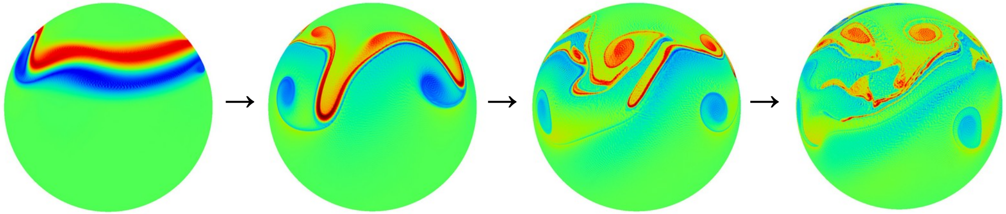
$$\begin{aligned} \alpha_n &= -(R \exp(i\theta_n)) \\ \beta_n &= -\frac{1}{N}(R \exp(i\theta_n)) \exp(R \exp(i\theta_n)) \end{aligned}$$

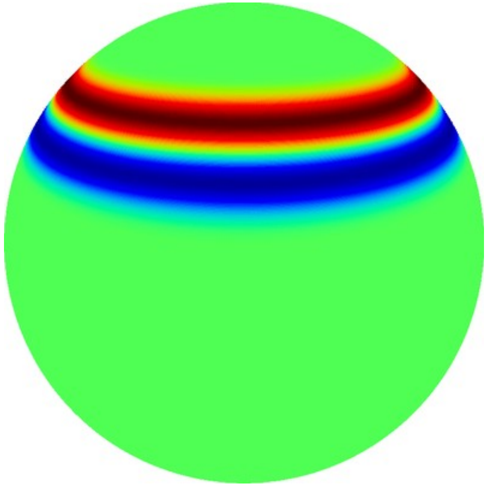
and finally

$$\boxed{U(t + \Delta t) = \exp(\Delta t L) U(t) \approx \sum_{n=1}^{N} \beta_n (\Delta t L + I\alpha_n)^{-1} U(t)}$$

Barotropic instability benchmark

- **Non-linear shallow-water equations, barotropic instability**
- **Cauchy Contour Integration REXI method**
- Reference solution: Simulation time of 122h, 4th order Runge-Kutta

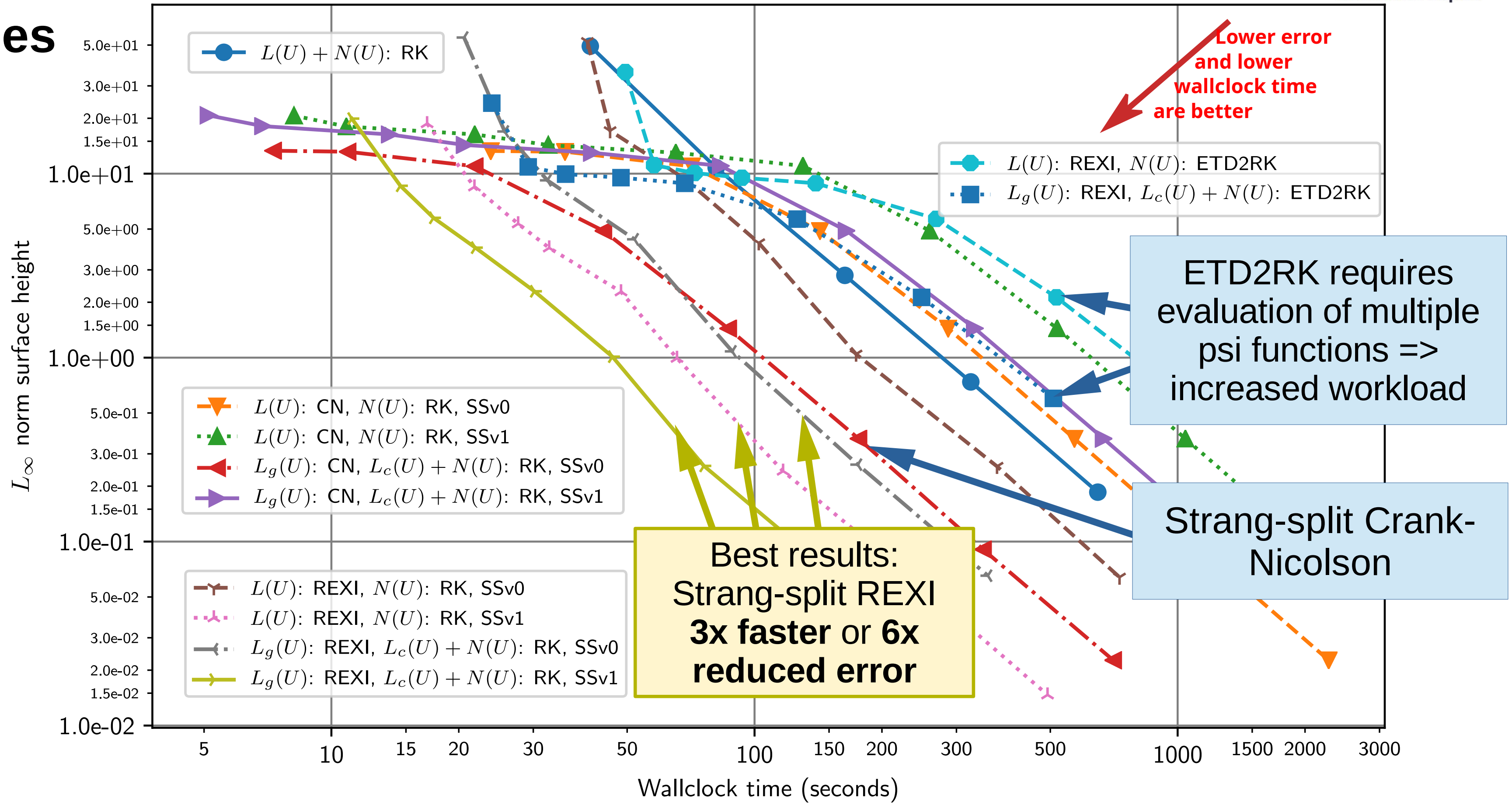




J. Galewsky, R. Scott, L. Polvani (2004) An initial-value problem for testing numerical models of the global shallow-water equations

Wallclock time for shifted Circle CI-REXI

N=128 poles



Schreiber, M., Schaeffer, N., & Loft, R. (2019). ***Exponential integrators with parallel-in-time rational approximations for the shallow-water equations on the rotating sphere****. Parallel Computing, 85, 56–65. https://doi.org/10.1016/j.parco.2019.01.005*

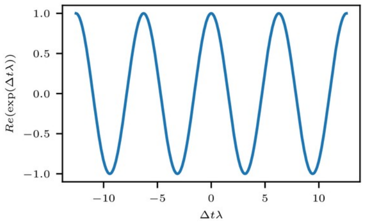
Exponential integration Part II

With Jed Brown

(And partly Finn Capelle, Matthew Normile, Raphael Schilling)

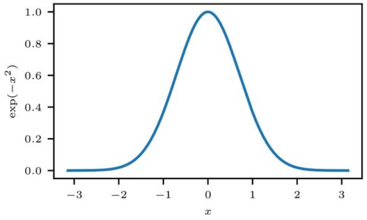
T-REXI: Terry et al. method

1) Approximate exponential by **linear combination of Gaussian kernels** (via Fourier space)



$$\mathrm{Re}\left(e^{i\lambda}\right) \approx \mathrm{Re}\left(\sum_{m=-M}^{M} b_m^{Re} G_h(\lambda + mh)\right)$$

2) Approximate Gaussian kernel by **linear combination of rational functions**



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

3) **Combination** of (1) and (2)

$$e^{i\lambda} \approx \sum_{n=-2N}^{2N} \left(\frac{\beta_n}{i\lambda + \alpha_n}\right)$$

Result:

- Approximation of **purely oscillatory function**
- Arbitrarily long time steps

T. S. Haut et al. (2015) - A high-order time-parallel scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator

From Butcher table to REXI...

- **Rewrite implicit Runge-Kutta**, e.g., based on collocation methods to

$$Y = U + \Delta t A L Y \Leftrightarrow Y = (I - \underbrace{\Delta t A}_{S} L)^{-1} U$$

and **diagonalize S** leading to $S = \Sigma \Lambda \Sigma^{-1}$ with a few more steps

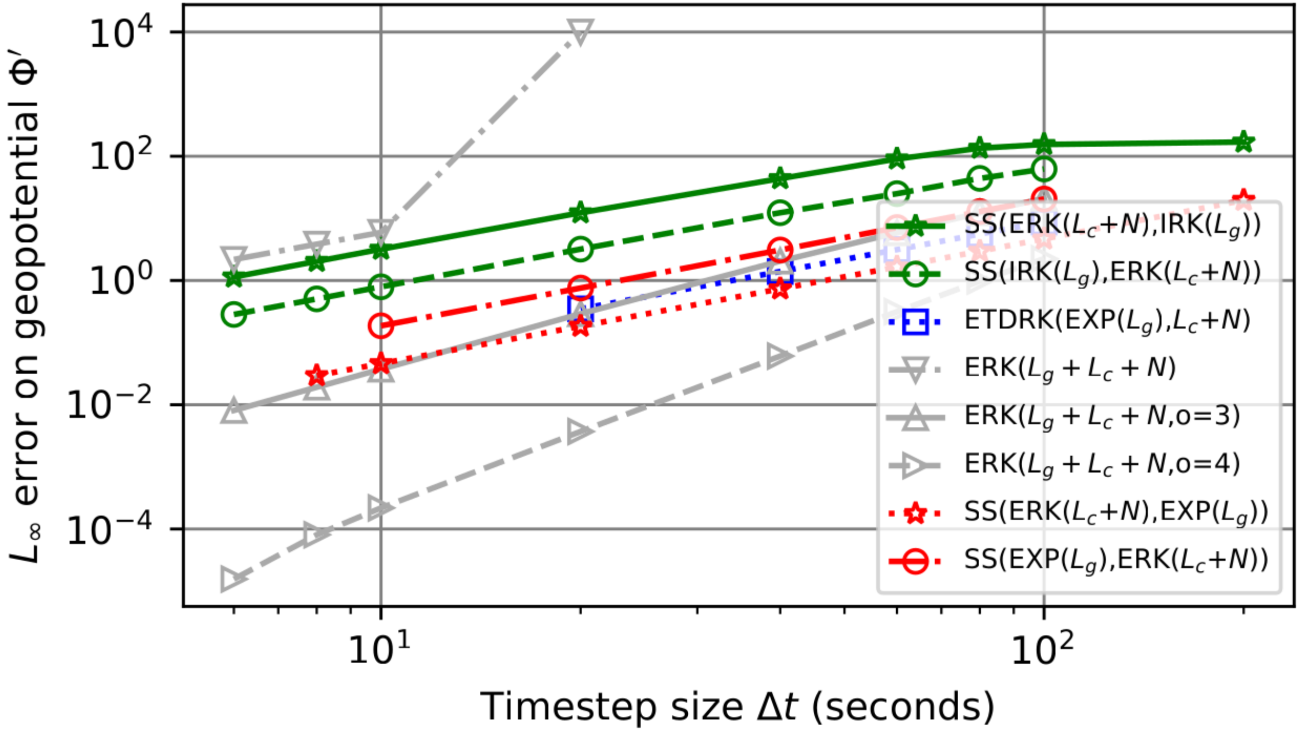
- Results in **unified REXI** formulation

$$U(t + \Delta t) = \gamma U(t) + \sum_{s=1}^{S} \beta_s (\alpha_s + \Delta t L)^{-1} U(t)$$

- Properties
  - Replaces a fully implicit RK method by **independent backward Euler steps**
  - Again leads to **complex-valued poles**
  - **Inherit all beneficial properties** from standard collocation IRK methods

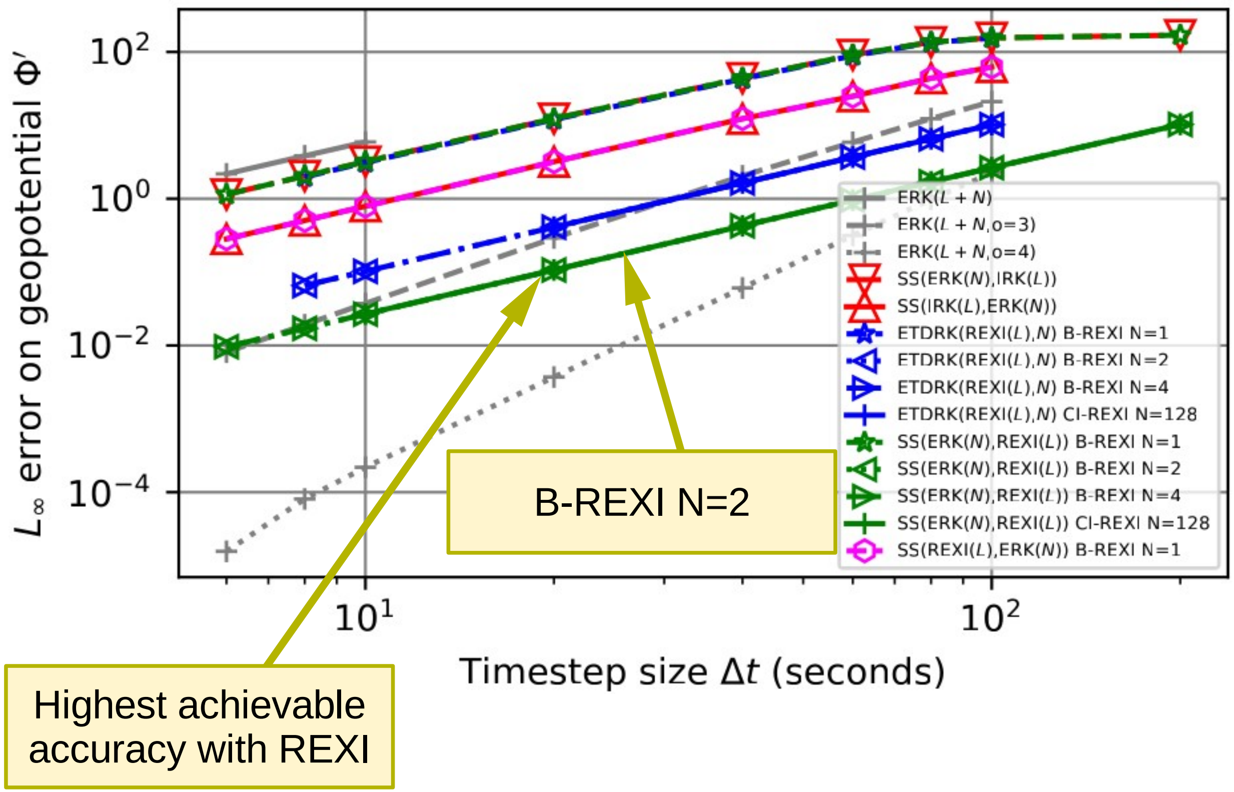
Jed Brown, "Fast solvers for implicit Runge-Kutta"', Copper Mountain Conference, 2014, https://jedbrown.org/files/20140408-FastIRK.pdf
Butcher, J. C. (1976). On the implementation of implicit Runge-Kutta methods. BIT, 16(3), 237–240. https://doi.org/10.1007/BF01932265
Bickart, T. A. (1977). An Efficient Solution Process for Implicit Runge-Kutta Methods. 14(6), 1022–1027.

Fairy tale: Exponential integration is awesome!
(Fairy tale if comparing errors on geopotential)



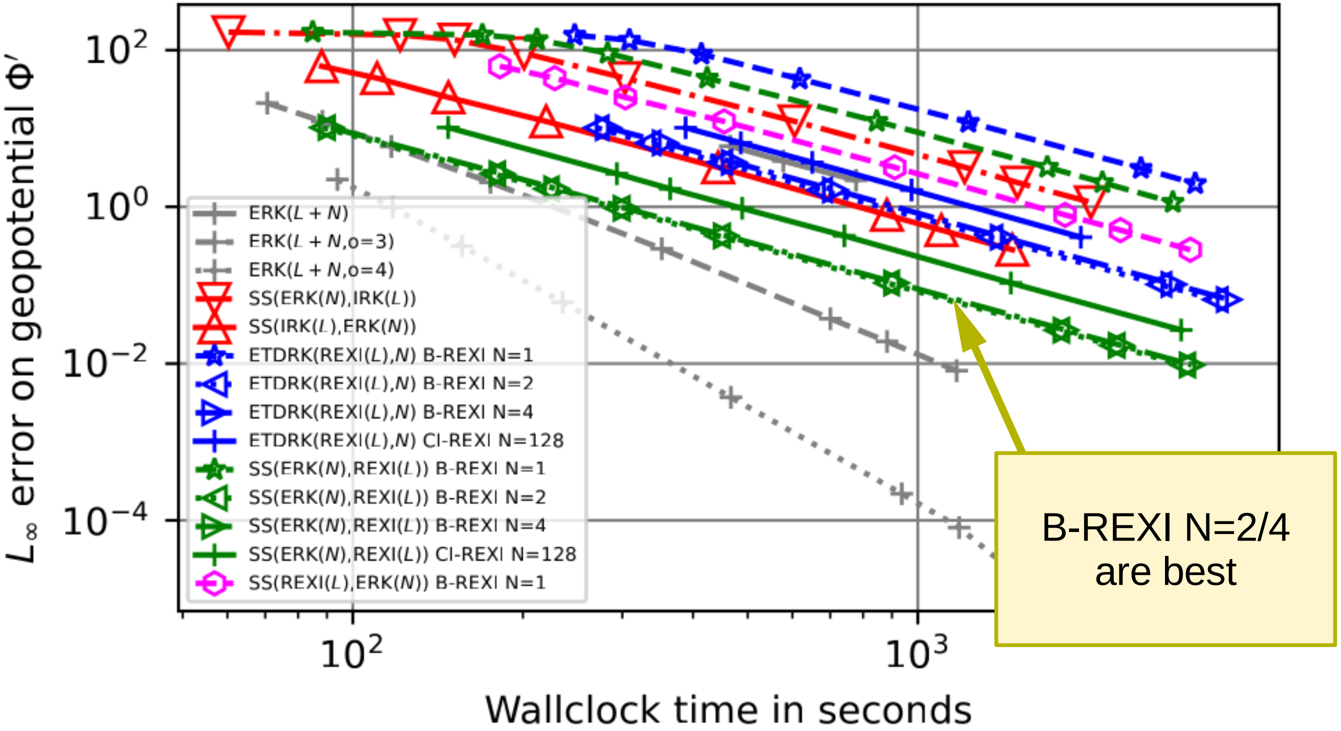
- Results based on **direct exponentiation**
- Lessons learned: **Classical 4th order Runge-Kutta is best**!

REXI investigation: Timestep size vs. error



- These results use **REXI on full linear term**
- Lessons learned:
  - B-REXI N=2 has lowest computational effort
  - 4th order implicit method competitive to exp. integration

… and wallclock time isn't your friend!



- Lessons learned:
  RK4 rocks!
- But could be a fairy tale as well:
  - **higher resolutions** could change the results
  - **Only** considering **geopotential**
  - **Many** more things (physical properties)

Parallel Spectral Deferred Corrections (pSDC)

With Thibaut Lunet, Daniel Ruprecht

Spectral Deferred Correction (SDC) - Part I: Picard

- A solution can be written in Picard form as

$$u(t) = u(a) + \int_a^t f(u(s), s)ds$$

- With quadrature points across time step and quadrature, we obtain

$$u(\tau_m) = u_0 + \sum_{j=1}^{M} q_{m,j} f(u(\tau_j), \tau_j) \text{ for } m = 1, \dots, M.$$

- In matrix notation, we get

$$U(\tau) = U_0 + QF\left(U(\tau), \tau\right)$$

and with **fixed point** notation for **iteration k**

$$U^{k+1}(\tau) = U^k + \left(U_0 - U^k + QF\left(U^k(\tau), \tau\right)\right).$$

Martin Schreiber, Université Grenoble Alpes / Inria / Laboratoire Jean Kuntzmann

Spectral Deferred Correction (SDC) - Part II: SDC

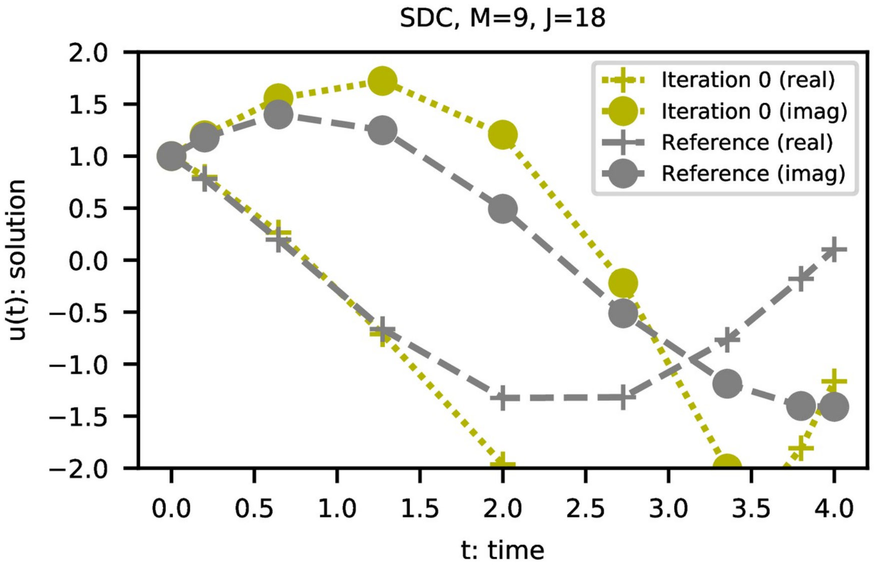
- Take arbitrary time integrator “I”
- Iterative form for k-th iteration (zero-to-node form)

$$U^{k+1} = U_0 + E\left(I\left[F(U^{k+1},\tau)\right] - I\left[F(U^{k},\tau)\right]\right) + QF\left(U^{k}\right)$$

$$E = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

with **arbitrary time integrator “I”**

- **Each iteration** leads to **additional order**
- Variants:
  - Explicit SDC, Implicit SDC
  - IMEX SDC
  - (Exponential SDC)
- But: **sequential over nodes and iterations**



For IMEX, see D. Ruprecht and R. Speck. Spectral Deferred Corrections with Fast-wave Slow-wave Splitting. SIAM Journal on Scientific Comp., 2016
Tommaso Buvoli. A Class of Exponential Integrators Based on Spectral Deferred Correction. pages 1–22, 2015.

Picard iteration with preconditioner

- With preconditioned Picard iteration using wisely chosen “P”:

$$U^{k+1} = U^k + P \underbrace{\left(U_0 - (I - \Delta t QF)\, U^k\right)}_{=\text{residual}} \qquad P = (I - \Delta t Q_\Delta F)^{-1}$$

- Finally, we obtain

$$(I - \Delta t Q_\Delta F)\, U^{k+1} = U_0 + \Delta t\, (Q - Q_\Delta)\, F U^k.$$

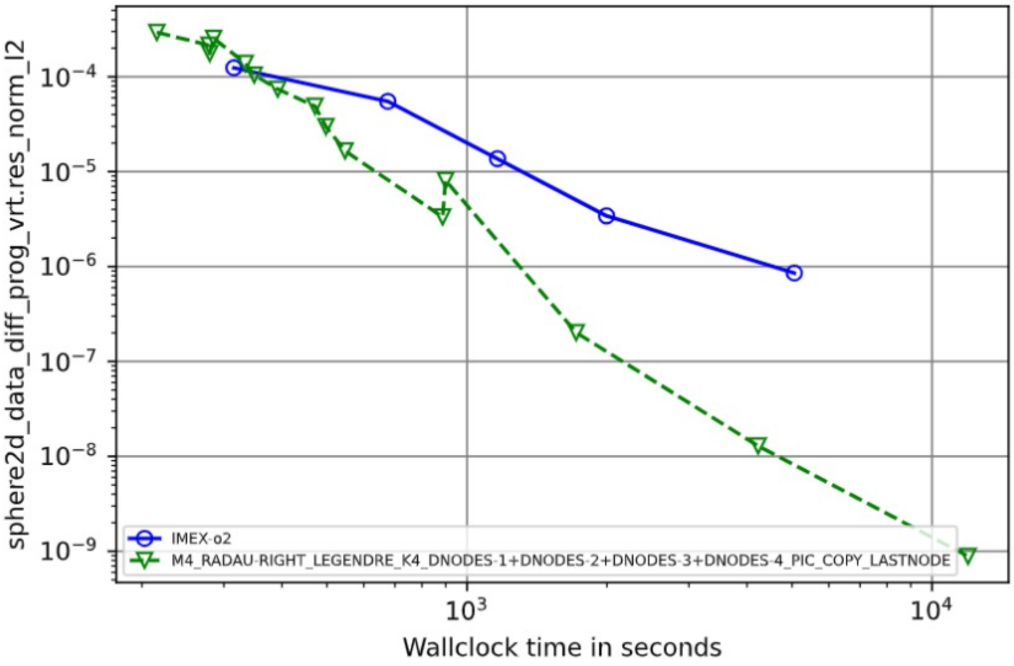
where Q_Δ can be arbitrarily chosen

- If **diagonal** Q_Δ, the preconditioner can be applied in **parallel** => pSDC
- If SDC implementation available, pSDC is almost trivial to use

Robert Speck, “Parallelizing spectral deferred corrections across the method”, 2017

Results for IMEX pSDC

- Hardware:
  - **Shared memory** parallelization
  - **4 NUMA domains**
- Math:
  - IMEX vs. pSDC
  - **Diagonal implicit preconditioner**
- Results:
  - Speedups if very high accuracy is required
  - Cross-over point



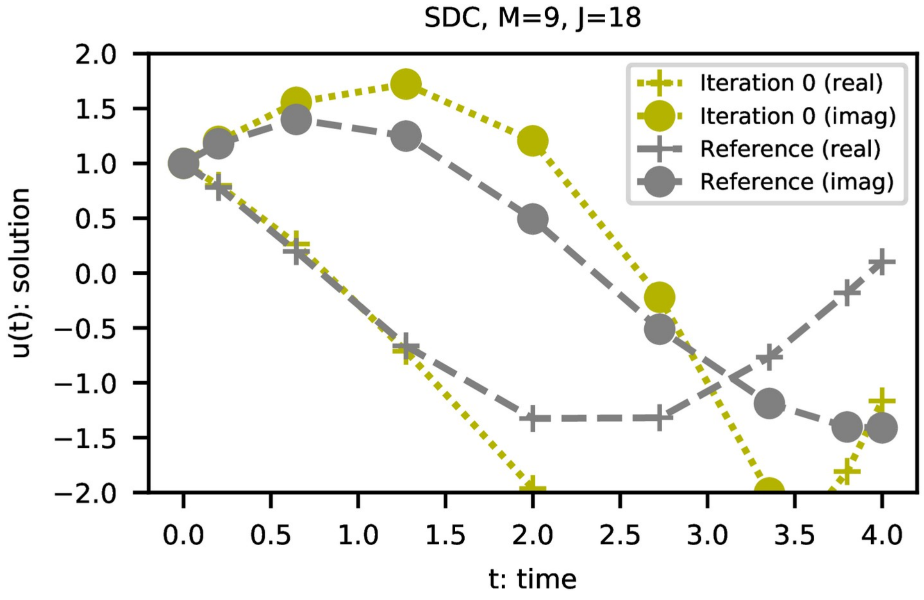
Philip Freese, Sebastian Götschel, Thibaut Lunet, Daniel Ruprecht, Martin Schreiber
"Parallel performance of shared memory parallel spectral deferred corrections", (submitted) 2024

Parallel Full Approximation Scheme in Space and Time (PFASST)

With François Hamon, Michael Minion

PFASST:
Parallel Full Approximation Scheme in Space and Time

- **Spectral Deferred Correction**
  Why? Higher order
  Iterative within the time step
- **Multi-level** in space
- **Parallel across time steps**



Dutt, A., Greengard, L., & Rokhlin, V. (1998). Spectral Deferred Correction Methods for Ordinary. 40(2), 1–26.

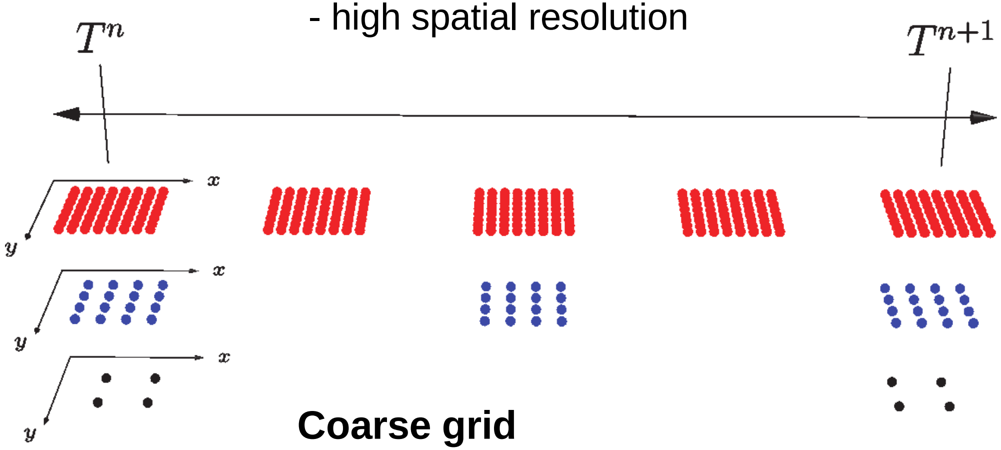
PFASST:
Parallel Full Approximation Scheme in Space and Time

- **SDC** Iterative within the time step
  Why? Higher order

- **Multi-level** in space
  Why? Overcoming CFL condition

- **Parallel across time steps**

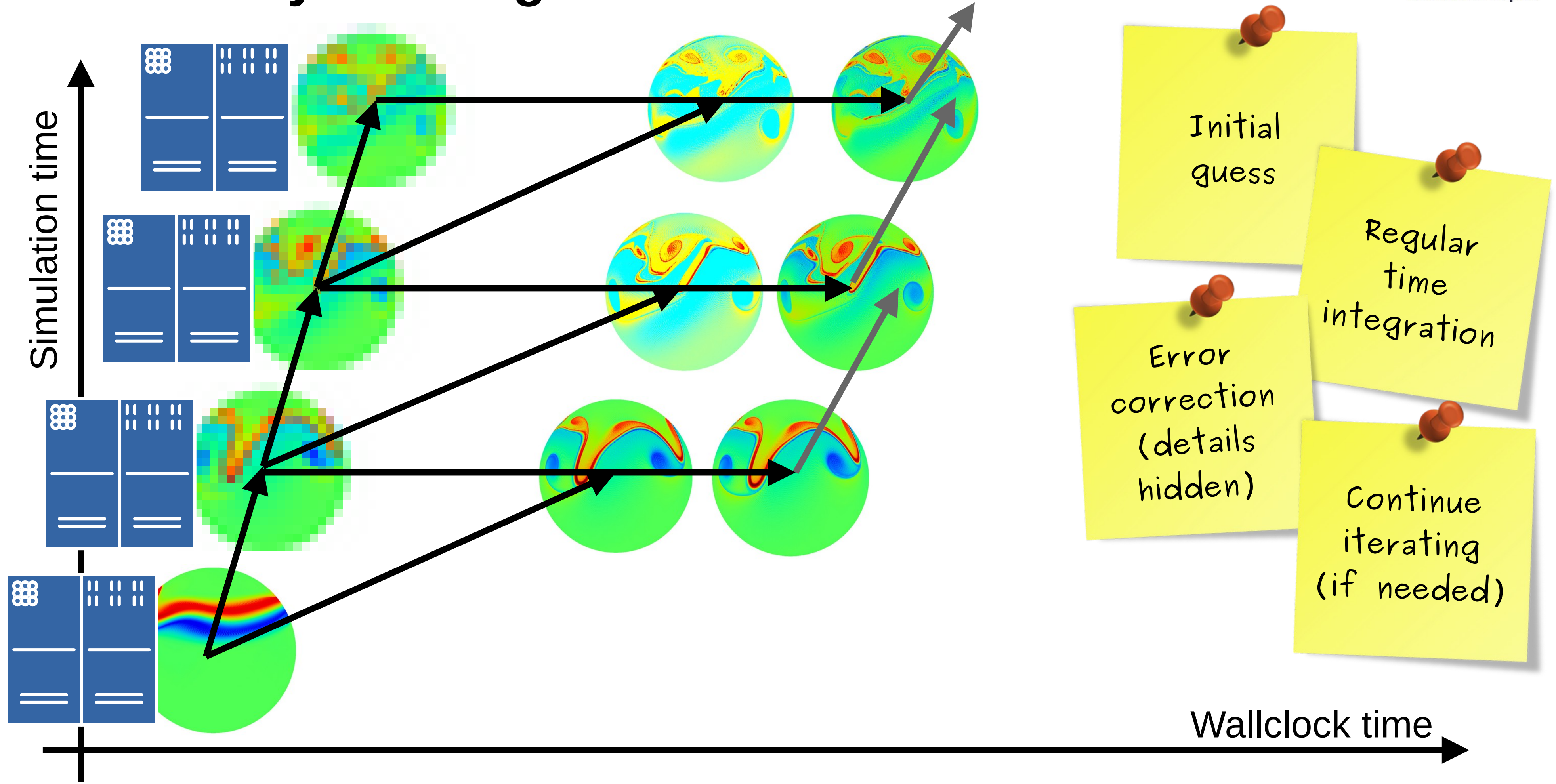
Fine grid
- smaller time step sizes
- high spatial resolution



Coarse grid
- larger time step sizes
- lower spatial resolution

Image: M. L. Minion, et al. Interweaving PFASST and parallel Multigrid

One of many PinT algorithms: Parareal

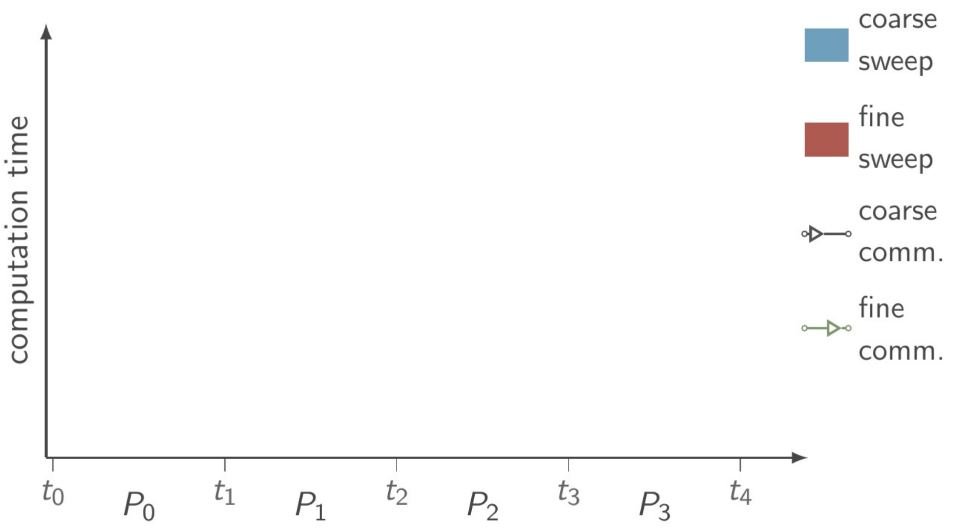


PFASST:
Parallel Full Approximation Scheme in Space and Time

- **SDC** Iterative within the time step
  Why? Higher order

- **Multi-level** in space
  Why? Fast first guess

- **Parallel across time steps**
  Why? Exploit increased parallelism
  (Similar to Parareal)

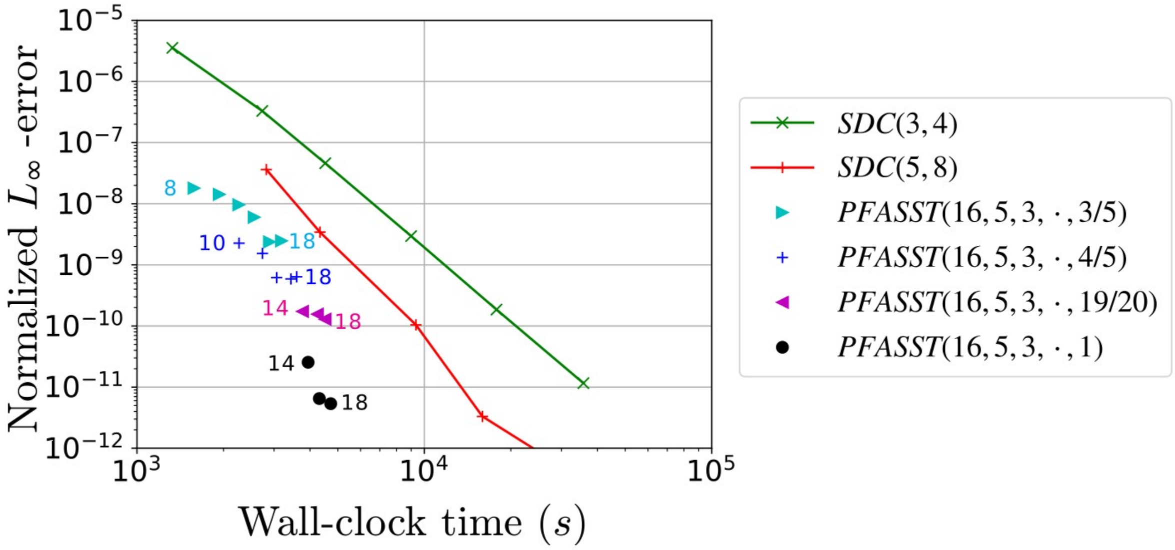


Picture from https://github.com/f-koehler/

Emmett, M. and Minion, M. L. (2012). Toward an efficient parallel in time method for partial differential equations. Communications in Applied Mathematics and Computational Science

PFASST wallclock time results with IMEX

- Benchmark: Barotropic instability benchmark on the sphere after 144h
- **Point sets: For increasing number of iterations (sweeps)**



SDC(
 # nodes,
 # sweeps
)

PFASST(
 # processors,
 # fine nodes,
 # coarse nodes,
 # MLSDC sweeps,
coarsening ratio
)

Plenty of other optimizations for PFASST not yet exploited!

Viscosity required for stabilization :-(

- **PFASST is Pareto optimal** for all time step sizes

F. Hamon, M. Schreiber, M. Minion (2020) Parallel-in-Time Multi-Level Integration of the Shallow-Water Equations on the Rotating Sphere

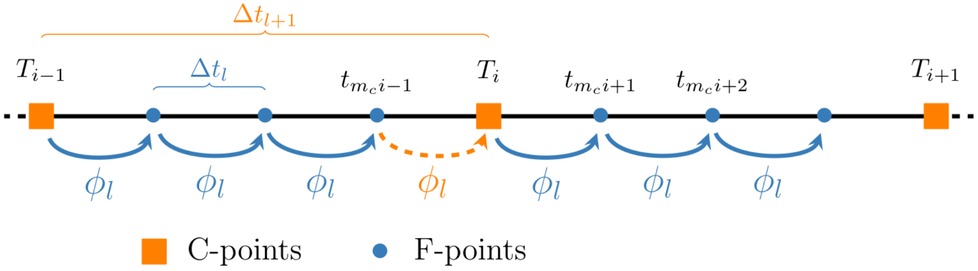
Multigrid in Time (MGRIT)

With Pedro S. Peixoto, Joao Steinstraesser

Multigrid in time (MGRIT)

- Previous issues with PFASST:
  - **#1: Fully higher-order:**
    - Nice, but computationally quite expensive with SDC
    - Not really required for climate/weather
  - **#2: Requires viscosity for stability :-(**

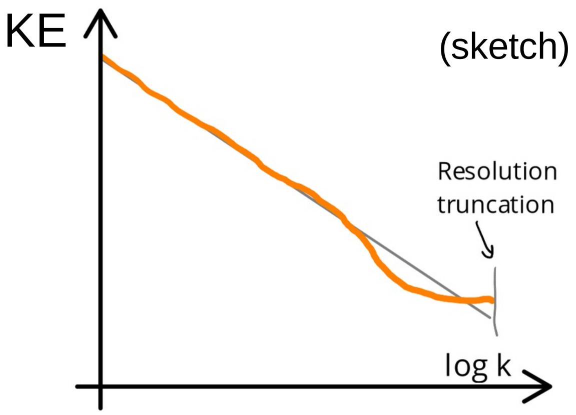
- **#1: Switch to MGRIT**
  - Away from higher-order method: Allows arbitrary time integrators
  - Parallel-in-time
  - For the following results / slides:
    - Fine grid: **IMEX**
    - Coarse grid: **IMEX** or **SETTLS**



- **#2: Viscosity… next slide!**

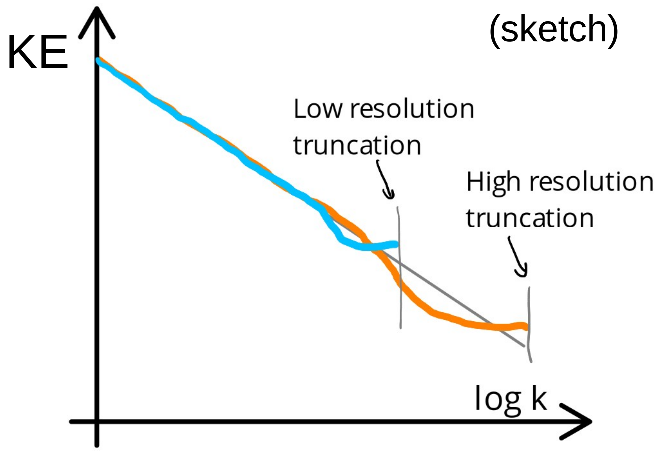
Filtering of non-linear parasitic modes

- Idea based on schematic plot of **kinetic energy spectrum**



- Up/downtailing at end of spectrum

Observation: Wrong modes nearby the spectrum

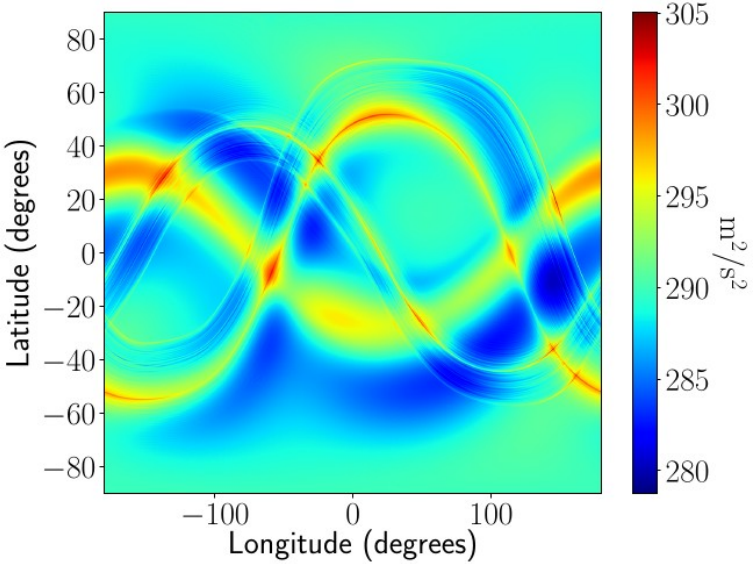


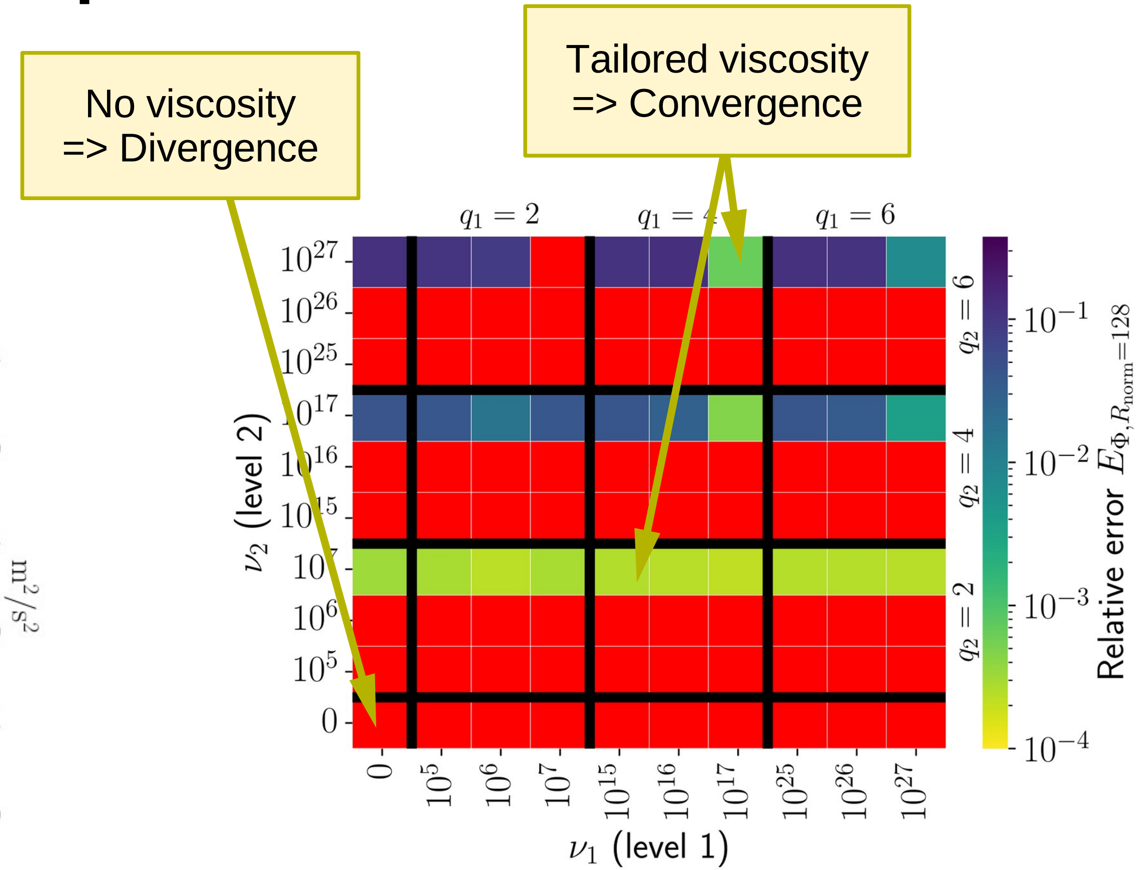
Idea: Filter out particularly these modes on the coarser grid

$$\boldsymbol{L}_\nu(\boldsymbol{U}) = (-1)^{\frac{q}{2}+1}\nu \begin{pmatrix} \nabla^q \Phi' \\ \nabla^q \xi \\ \nabla^q \delta \end{pmatrix}$$

Results with Gaussian bumps

- SL-SI-SETTLS (IMEX similar)
- 3 multi-level layers
- (Hyper)viscosity applied to coarser levels





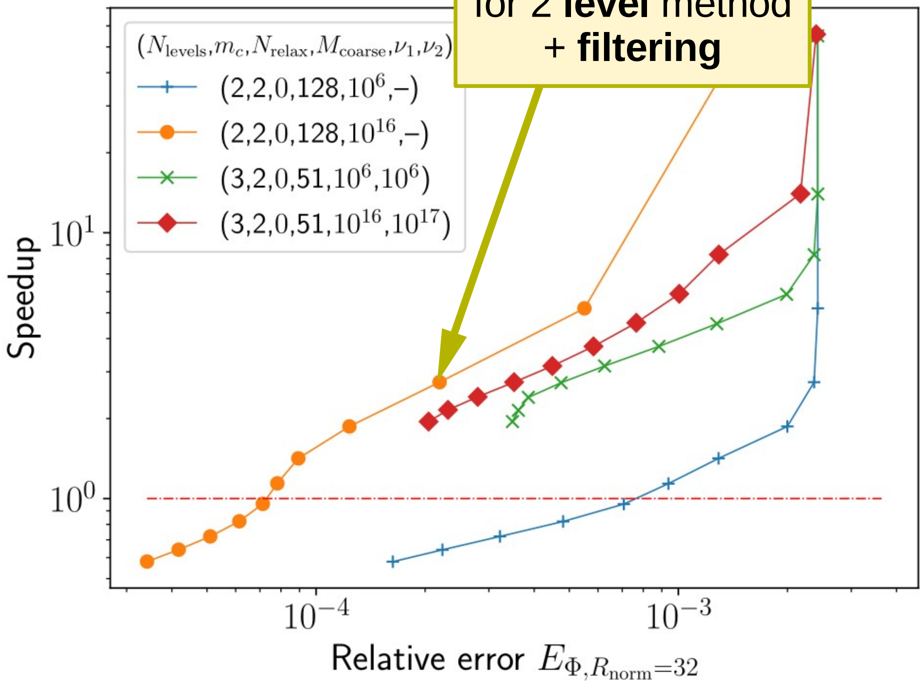
2023 J. G. C. Steinstraesser, P. S. Peixoto, M. Schreiber, Parallel-in-time integration of the shallow water equations on the rotating sphere using Parareal and MGRIT, Journal of Computational Physics, Elsevier

Speedups with Galewsky benchmark
Baseline: IMEX method

parallel-in-time processors = 64

- IMEX

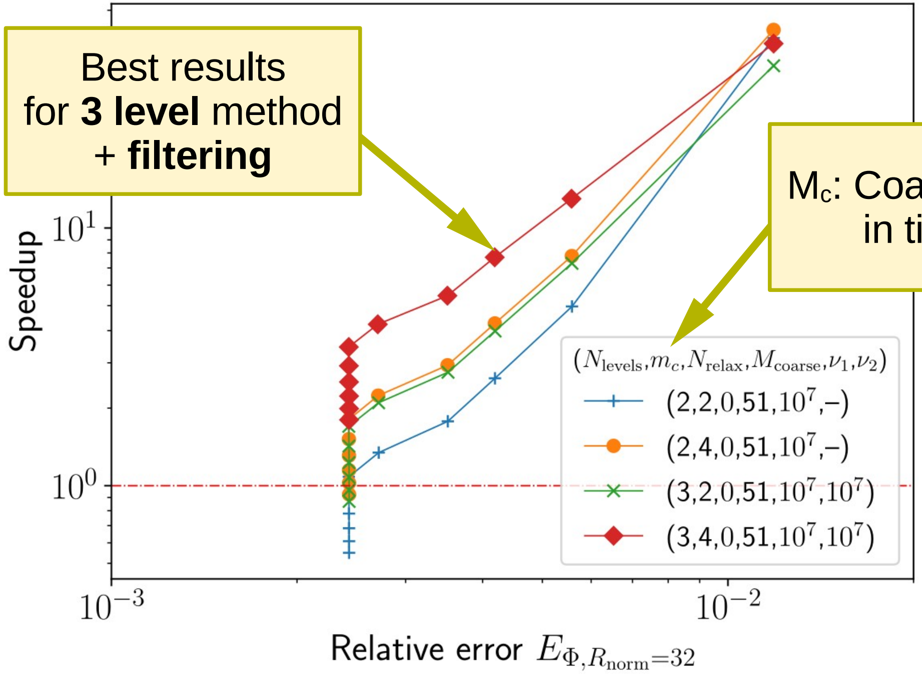
Best results for 2 **level** method + **filtering**



- SL-SI-SETTLS

Best results for **3 level** method + **filtering**

M_c: Coarsening in time



2023 J. G. C. Steinstraesser, P. S. Peixoto, M. Schreiber, Parallel-in-time integration of the shallow water equations on the rotating sphere using Parareal and MGRIT, Journal of Computational Physics, Elsevier

Summary

- **REXI:**
  - Arbitrarily long time step sizes
  - Helmholtz problem to solve
  - **Higher-order RK could be alternative to exp. integration**
- **PSDC:**
  - Parallel version of SDC provides performance boost
  - “For free” for existing SDC implementations
- **PFASST:**
  - **Pareto optimal**
  - But requires **viscosity** on all levels
- **MGRIT**:
  - Filtering on coarser levels to **avoid viscosity**
  - **> 2 levels**
  - Led to significant **speedups**

Thank you!