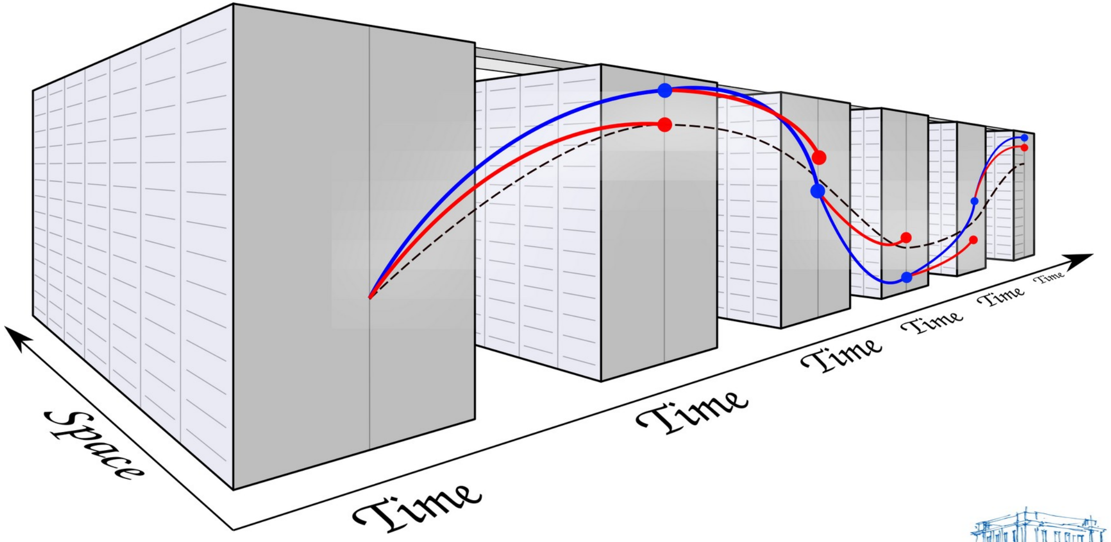
Next-Generation Time Integration targeting Weather and Climate Simulations
Part I: Rational Approximation of Exponential Integrators

Martin Schreiber
Computer Architectures and Parallel Systems (of Equations)
Technical University of Munich

This presentation is based on collaborations with
  Terry Haut
  Richard Loft
  Nathanael Schaeffer
Cheers!



Seminar “Numerical methods for atmospheric and oceanic modelling: recent advances and future prospects”
ECMWF, Reading, UK, September 18th, 2020

Application focus: weather/climate simulations

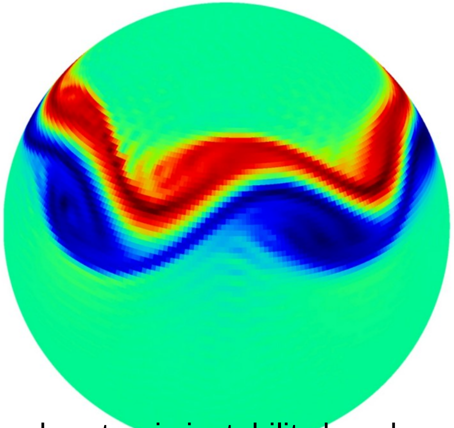
- **Weather** simulations:
  - Ongoing **demands for increase in accuracy** via **increase in resolution**
  - Leads to smaller and hence **more time steps**
  - Challenge: Forecasts need to be **completed within a specific time frame** (about 1 hour)
- **Climate** simulations:
  - **low resolution, but long simulation time span** (e.g. for Paleoclimate) requires weeks or months of wall clock time
  - Challenge: **reduce wallclock time**

Are there alternative concepts to improve accuracy vs. wallclock time?

Test case: single-layer atmospheric model

- First steps towards new formulations:
  - Use **single-layer** atmospheric model
  - Can be related to **shallow-water equations**

$$\frac{\partial \Phi}{\partial t} = -\nabla \cdot (\Phi \mathbf{V})$$
$$\frac{\partial \mathbf{V}}{\partial t} = -\nabla \Phi - f\mathbf{k} \times \mathbf{V} - \mathbf{V} \cdot \nabla \mathbf{V}$$



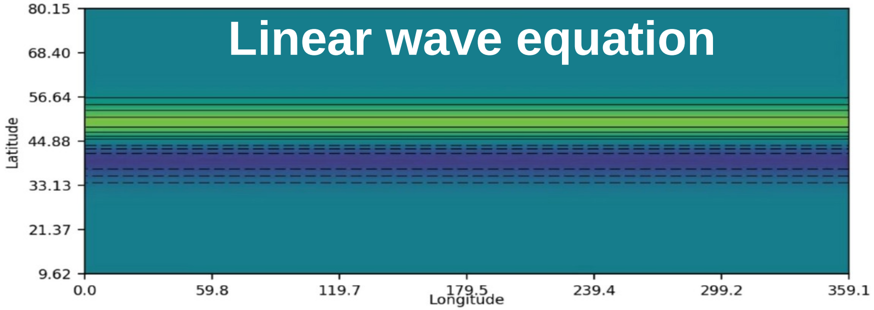
E.g. barotropic instability benchmark
Relative vorticity field

- Shallow-water equations (SWE) are **commonly used to study horizontal discretization aspects** as a first step to develop new methods for dynamical cores
- **Discretization and time integration in this work**
  - **Numerics aligned with ECMWF's model, e.g. Spherical Harmonics**
  - **Plenty of details skipped in the following slides**, in particular efficient solvers for Helmholtz (and Helmholtz-like) problems.

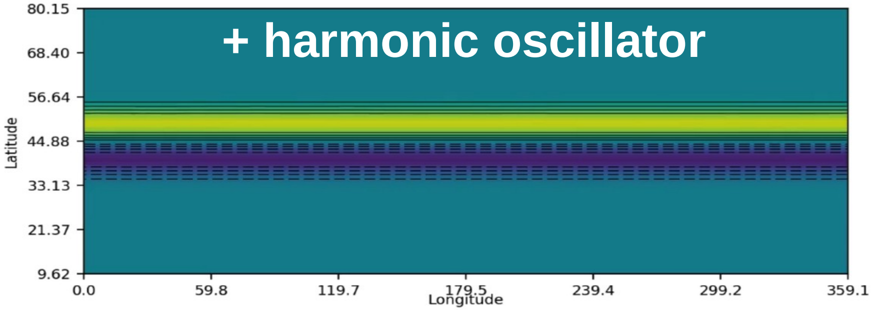
Splitting SWE into “subproblems”

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

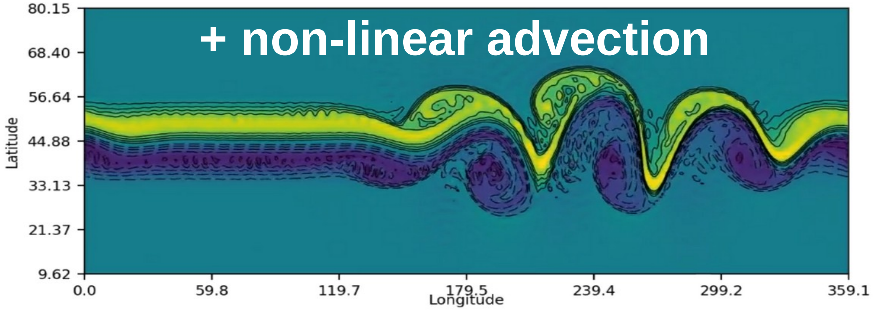
$$\frac{\partial U}{\partial t} = L_g(U)$$



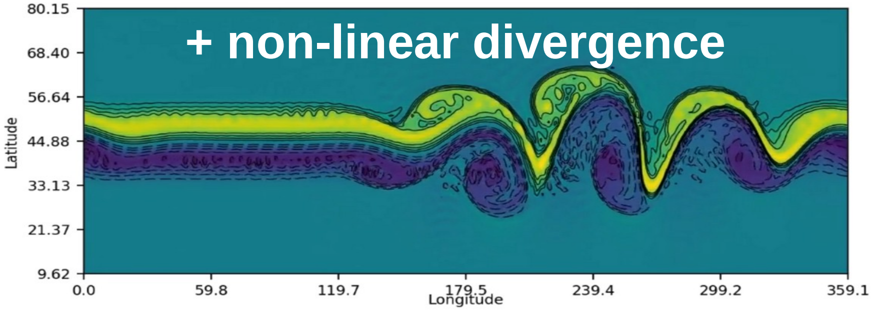
$$\frac{\partial U}{\partial t} = L_g(U) + L_f(U)$$



$$\frac{\partial U}{\partial t} = L_g(U) + L_f(U) + N_a(U)$$



$$\frac{\partial U}{\partial t} = L_g(U) + L_f(U) + N_a(U) + N_d(U)$$



Overview of different research directions

- **EXP:** Exponential time integration
  - T-REXI
  - CI-REXI
  - …

Part of this presentation

- **SL:** Semi-Lagrangian methods
  - State of the art
  - Combined with Parareal
- PFASST
  - **Spectral deferred correction** (SDC)
  - **Multi-level**
  - **Parallel-in-time** corrections (Parareal)

- **Combinations** of methods:
  SL / EXP / SDC / Parareal
- "**Tuning**" of methods (Convergence / accuracy)
- **EXP Machine learning** with ANN
- (Quantum Computing)

Overview

- Introduction
- **Exponential integration**
  - **Introduction**
  - CI-REXI (Cauchy Contour Integral method)
  - Studies with SWE
    - Linear SWE: Error vs. wall clock time studies
    - Non-linear PDEs: Error vs. wall clock time studies
- Summary & outlook

Exponential integrators: Linear case $\frac{\partial U}{\partial t} = LU$

- **Direct solution** at arbitrary point in time given by

$$U(t) = e^{tL}U(0) = Qe^{t\Lambda}Q^{-1}U(0)$$

- Note, that we can / should **never setup the Eigenvector matrix "Q" explicitly**!

- Important properties
  - **No errors in time!**
  - Theoretically,
    - infinitely large time steps (**no CFL** limitation)
    - going **forward / backward in time** (for oscillatory systems)

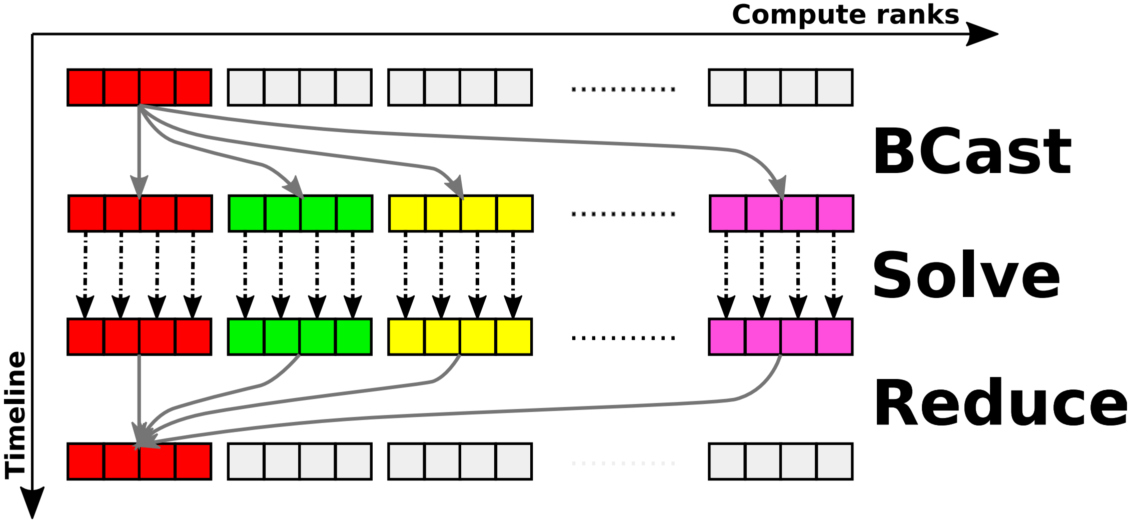
- Exponential is
  - simple to **solve for ODE systems** (Setup "Q" matrix), however it is
  - **challenging to find efficient solver** for PDEs (Infeasible to setup "Q")

- Approach taken here:
  Rational approximation of exponential integrators (**REXI**)

REXI: Formulation & Parallelization

- **Linear PDE:** With some linear algebra, we get a sum over independent terms
- Replace exponential with **parallelizable part** => **parallel-in-time** (see right picture)
- **WARNING**: We solve the REXI terms with spherical harmonics. This is much more challenging with other spatial discretizations!!!

$$U(t) = e^{tL}U(0) \approx \sum_n \beta_n (I\alpha_n + tL)^{-1} U(0)$$



A tool for the best REXI method

- There are various **REXI methods**:
  - T-REXI
  - CI-REXI
  - B-REXI
  - EL-REXI
- Each method has **various parameters**

- **Different requirements** from **application side**:
  - Oscillatory / diffusive linear PDE
  - Stability requirements
  - Filtering requirements
  - Workload (number of REXI terms)
  - Consistency
  - ...

Results in **high-dimensional optimization space** which is hard to explore
=> Software-supported exploration (coming soon)

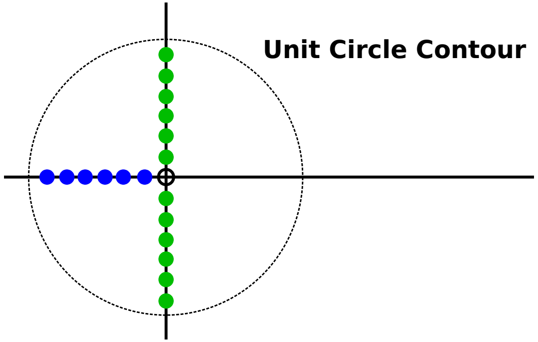
For sake of time, **example provided by Cauchy Contour integral method**

Overview

- Introduction
- Exponential integration
  - Introduction
  - **CI-REXI (Cauchy Contour Integral method)**
  - Studies with SWE
    - Linear SWE: Error vs. wall clock time studies
    - Non-linear PDEs: Error vs. wall clock time studies
- Summary & outlook

Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

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 $f(z) = \exp(z)$ to be approximated by REXI

- Using e.g. a **circle contour** around the origin and a **trapezoidal rule** we get

$$\alpha_n = -(R\exp(i\theta_n))$$

$$\beta_n = -\frac{1}{N}(R\exp(i\theta_n))\exp(R\exp(i\theta_n))$$

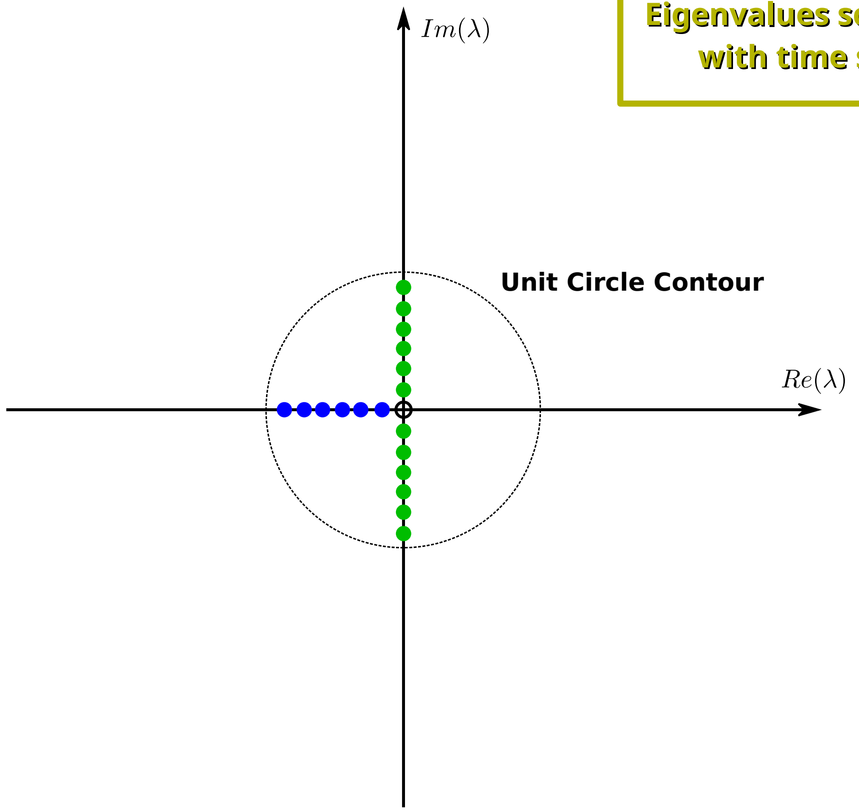
and finally $\exp(x) \approx \sum_{n=1}^{N} \frac{\beta_n}{x + \alpha_n}$

$$U(t) = \exp(tL)U(0) \approx \sum_{n=1}^{N} \beta_n (tL + I\alpha_n)^{-1} U(0)$$

T. Buvoli (2015) A Class of Exponential Integrators Based on Spectral Deferred Correction
L.N. Trefethen, J.A.C. Weideman (2014) The Exponentially Convergent Trapezoidal Rule
C. Clancy and P. Lynch (2011) - Laplace transform integration of the SWE. Part I: Eulerian formulation and Kelvin waves
Sheen, D., Sloan, I. H., & Thomée, V. (1999). A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature. Mathematics of Computation, 69(229)

CI-REXI: Small time steps

Eigenvalues scale linearly with time step size

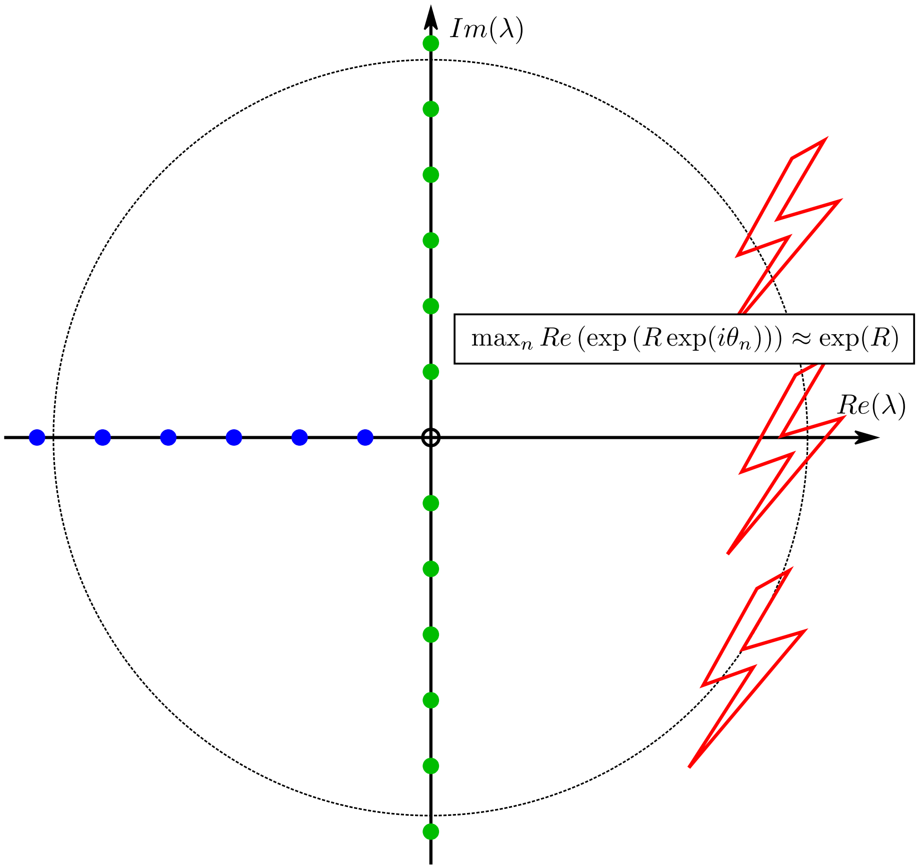


(2019) M. Schreiber, N. Schaeffer, R. Loft, Exponential Integrators with Parallel-in-Time Rational Approximations for Shallow-Water Equations on the Rotating Sphere, Parallel Computing, Elsevier

Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

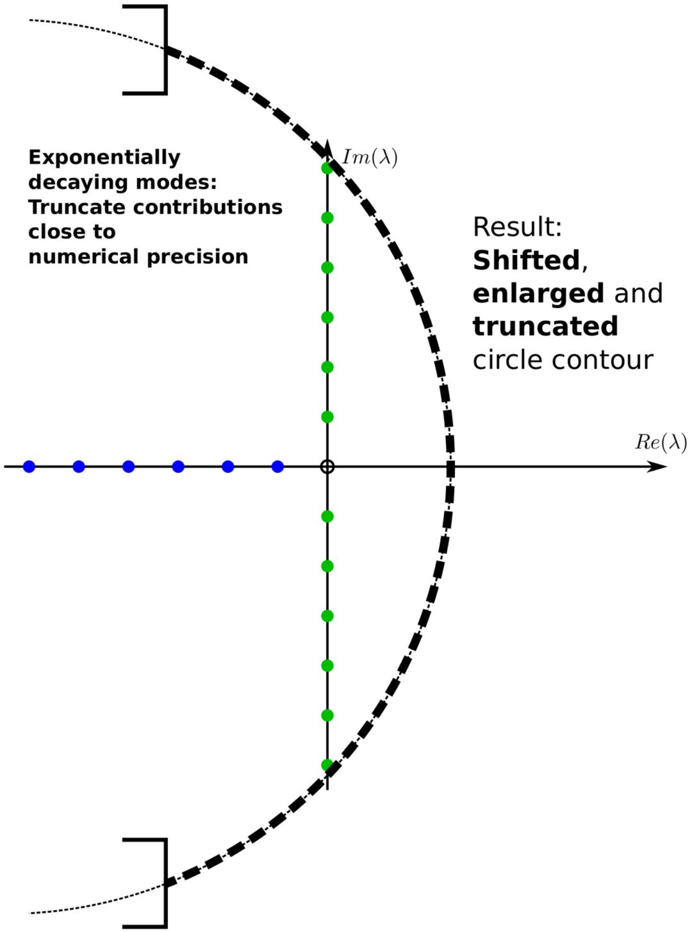
CI-REXI: Large time steps

$$U(t) \approx \sum_{n=1}^{N} \beta_n \left(tL + I\alpha_n\right)^{-1} U(0)$$



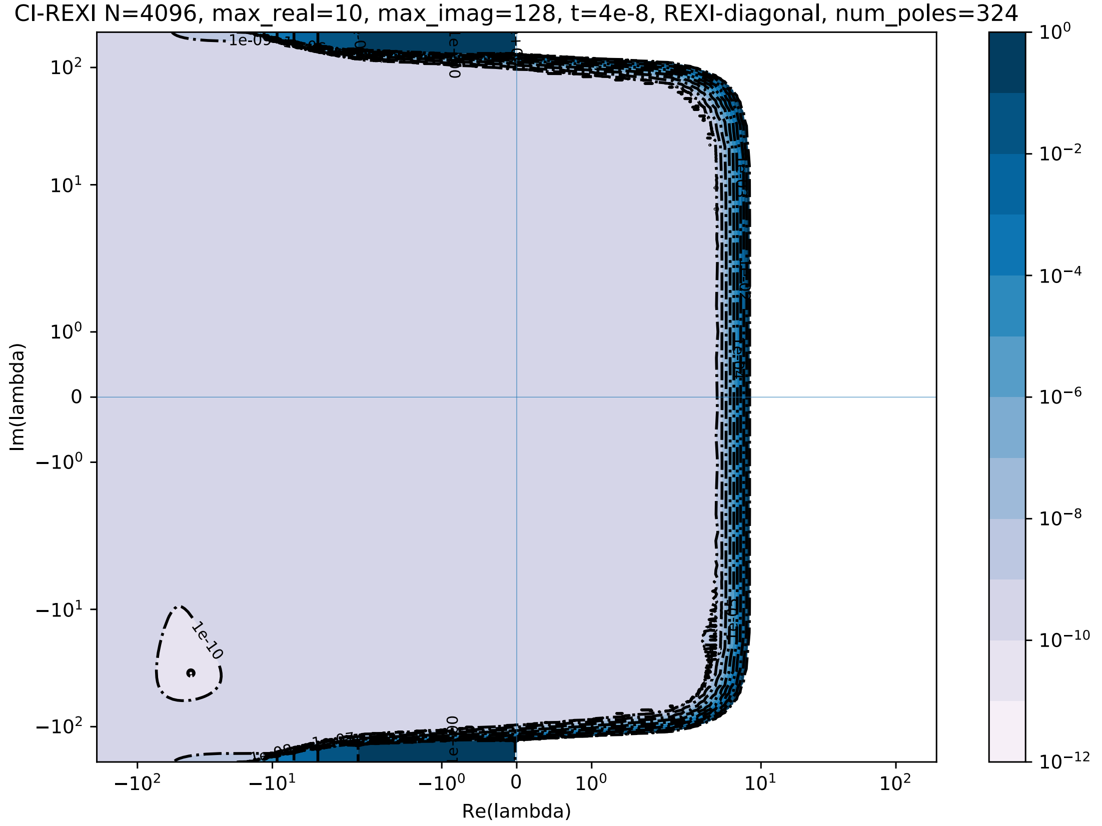
(2019) M. Schreiber, N. Schaeffer, R. Loft, Exponential Integrators with Parallel-in-Time Rational Approximations for Shallow-Water Equations on the Rotating Sphere, Parallel Computing, Elsevier

CI-REXI



(2019) M. Schreiber, N. Schaeffer, R. Loft, Exponential Integrators with Parallel-in-Time Rational Approximations for Shallow-Water Equations on the Rotating Sphere, Parallel Computing, Elsevier

Error L_∞ plot for CI-REXI



Error plots for CI-REXI with N=4096, maxImag=128, maxReal=10 and eps=4e−8 for the pole filter, resulting in N=324 poles. Results are extremely accurate, close to machine precision (10^{-10}).

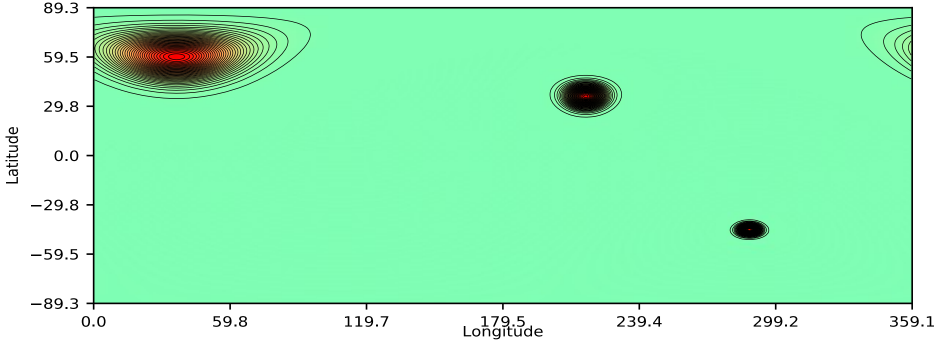
Overview

- Introduction
- Exponential integration
  - Introduction
  - CI-REXI (Cauchy Contour Integral method)
  - **Studies with SWE**
    - Linear SWE: Error vs. wall clock time studies
    - Non-linear PDEs: Error vs. wall clock time studies
- Summary & outlook

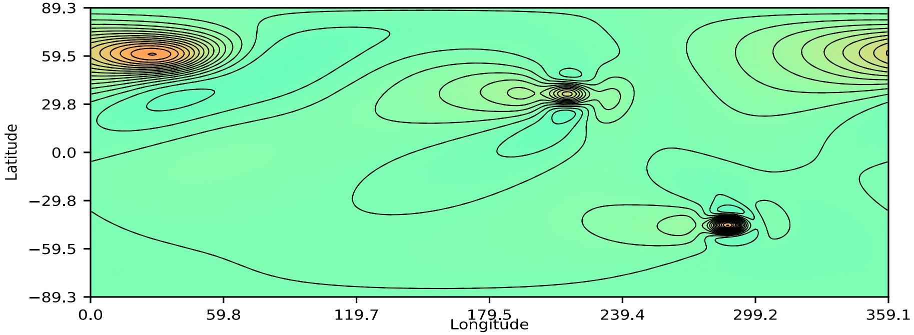
Benchmarks: Propagation of three Gaussian bumps with linearized single-layer SWE model

- Test propagation of waves **across rotating sphere over 1.5 days**

t=0: Three Gaussian bumps on height as initial conditions



t=1.5 days: Reference solution with 4^{th} order explicit Runge-Kutta

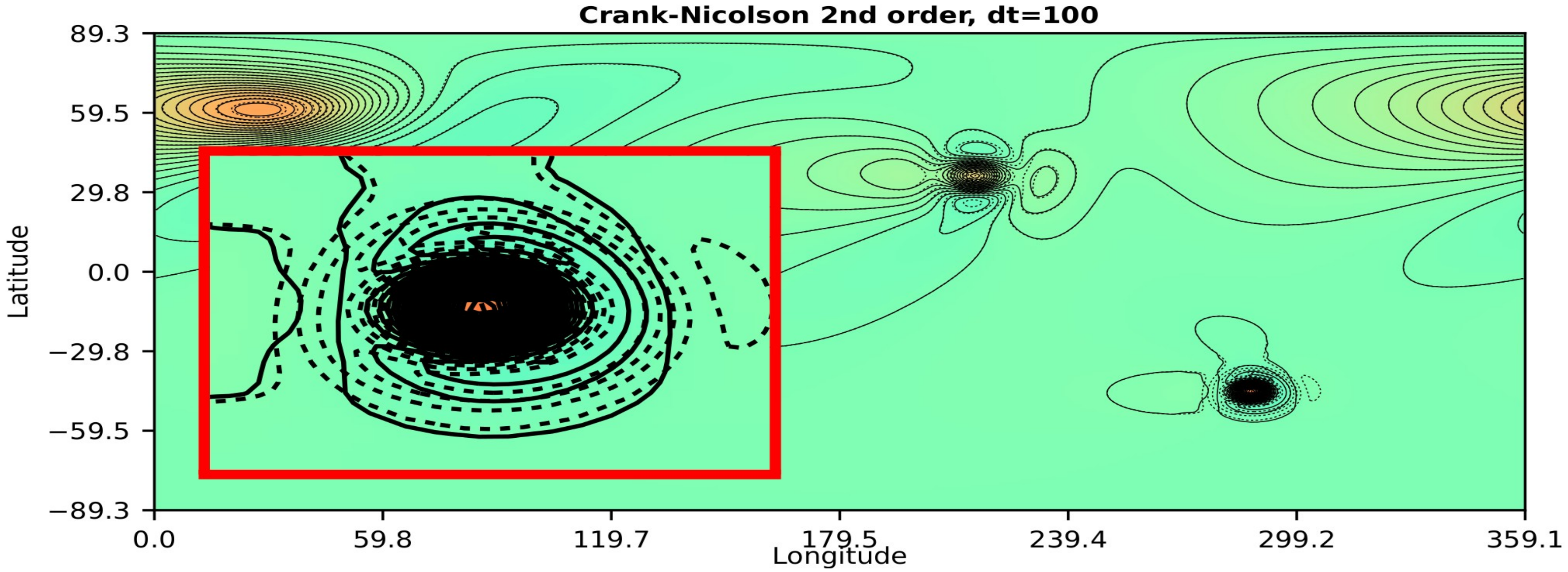


- Benchmarks based on **full linear operator, including Coriolis effect**
- Video: https://youtu.be/mmaj0l2ZO9k
- **Terry's REXI coefficients (T-REXI)** are used in the following slides

Schreiber, M., & Loft, R. (2018). A parallel time integrator for solving the linearized shallow water equations on the rotating sphere. Numerical Linear Algebra with Applications, 26(2). https://doi.org/10.1002/nla.2220

Crank-Nicolson (implicit), Δt = 100s

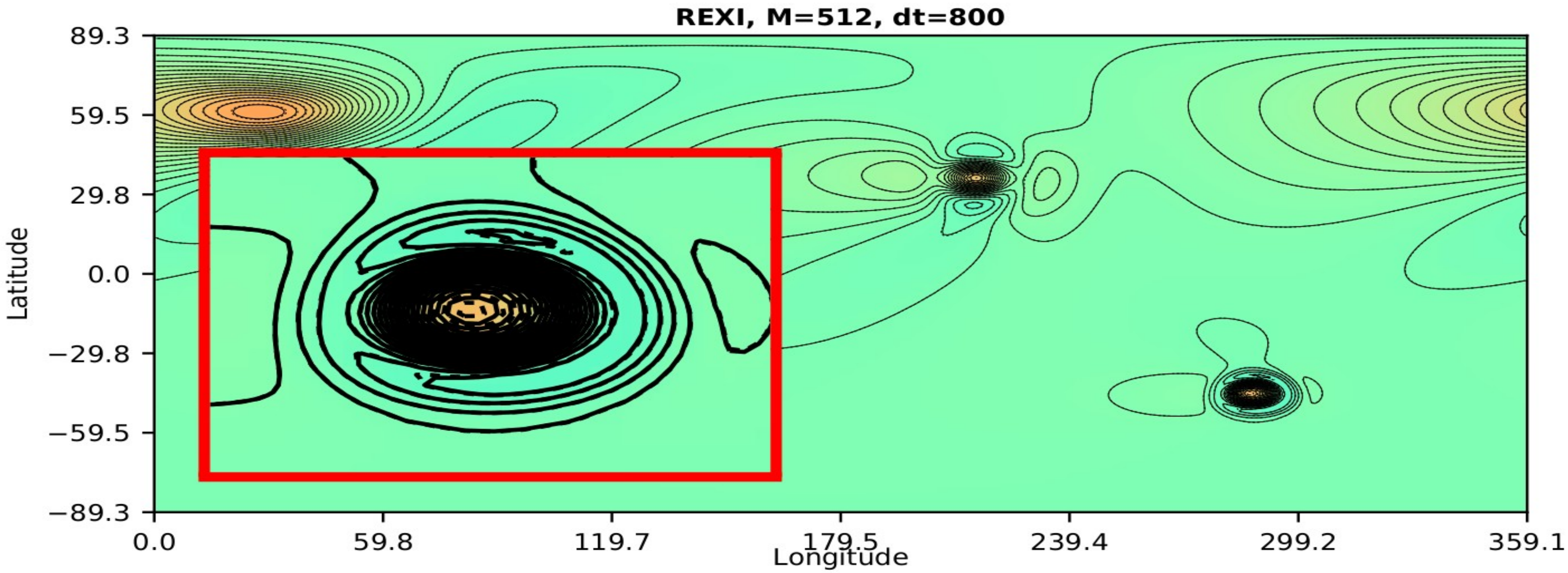
- Dashed lines: Reference solution
- Dispersion errors: Clearly visible
- **Larger time step sizes: Errors significantly increase**



Schreiber, M., & Loft, R. (2018). A parallel time integrator for solving the linearized shallow water equations on the rotating sphere. Numerical Linear Algebra with Applications, 26(2). https://doi.org/10.1002/nla.2220

T-REXI with M=512, Δt = 800s

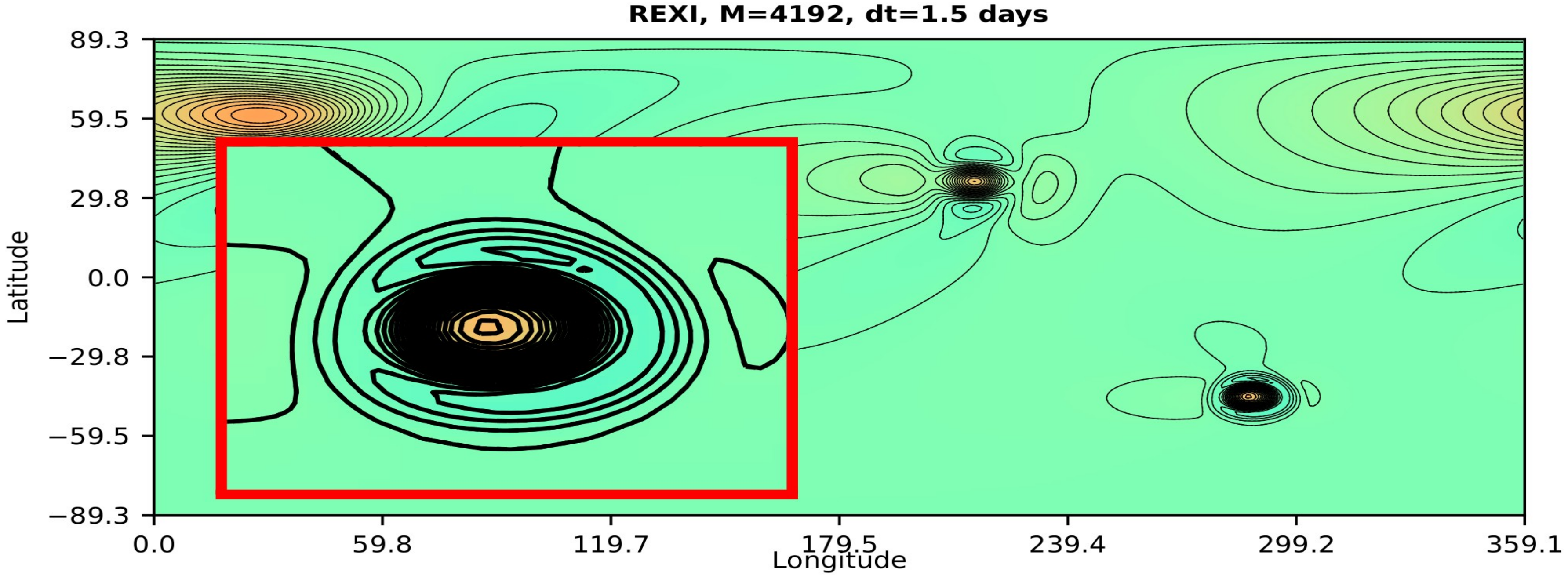
- Dashed lines: Reference solution
- Dispersion errors: **Hardly any errors visible**
- Larger time step sizes? => Next slide



Schreiber, M., & Loft, R. (2018). A parallel time integrator for solving the linearized shallow water equations on the rotating sphere. Numerical Linear Algebra with Applications, 26(2). https://doi.org/10.1002/nla.2220

T-REXI with M=4192, Δt = 1.5 days

- Dashed lines: Reference solution
- **Dispersion errors: Not visible, extremely accurate!**
- Larger time step sizes: already large enough…



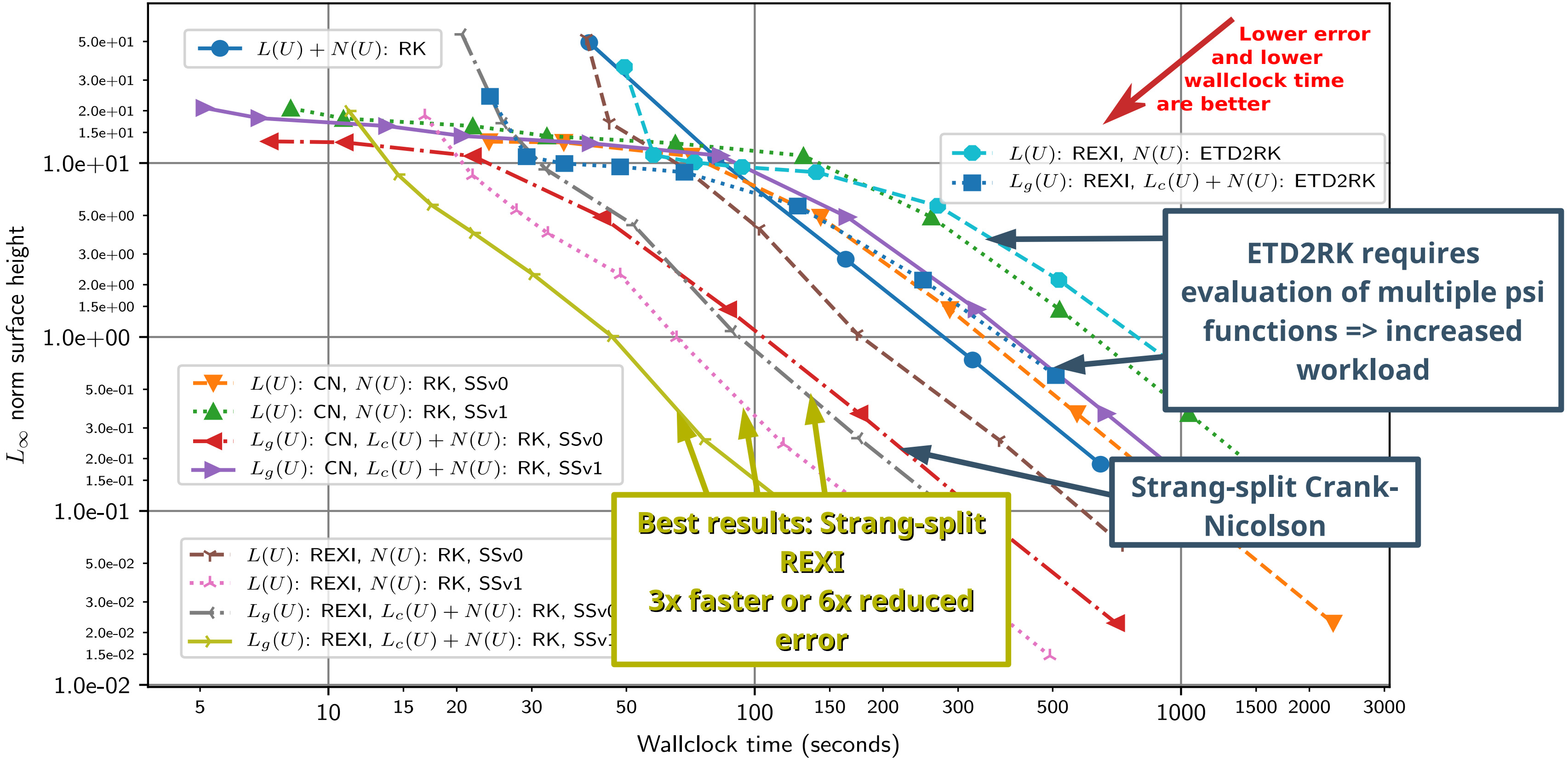
Schreiber, M., & Loft, R. (2018). A parallel time integrator for solving the linearized shallow water equations on the rotating sphere. Numerical Linear Algebra with Applications, 26(2). https://doi.org/10.1002/nla.2220

Overview

- Introduction
- Exponential integration
  - Introduction
  - CI-REXI (Cauchy Contour Integral method)
  - **Studies with SWE**
    - **Linear SWE: Error vs. wall clock time studies**
    - **Non-linear PDEs: Error vs. wall clock time studies**
- Summary & outlook

With Nathanael Schaeffer & Richard Loft

Barotropic instability: Error vs. wallclock time

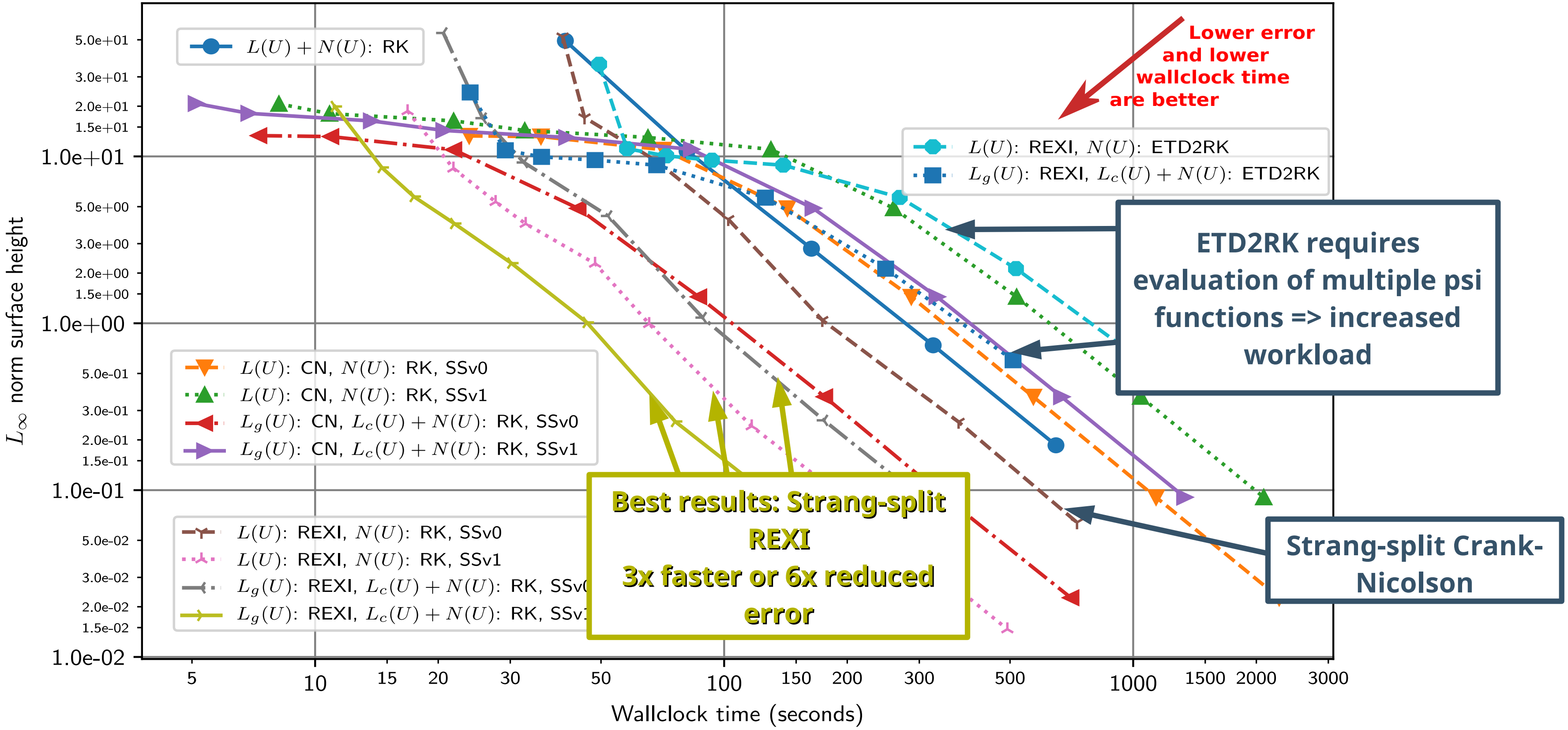


Schreiber, M., Schaeffer, N., & Loft, R. (2019). Exponential integrators with parallel-in-time rational approximations for the shallow-water equations on the rotating sphere. Parallel Computing, 85, 56–65. https://doi.org/10.1016/j.parco.2019.01.005

Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

With Nathanael Schaeffer & Richard Loft

Barotropic instability: Error vs. wallclock time

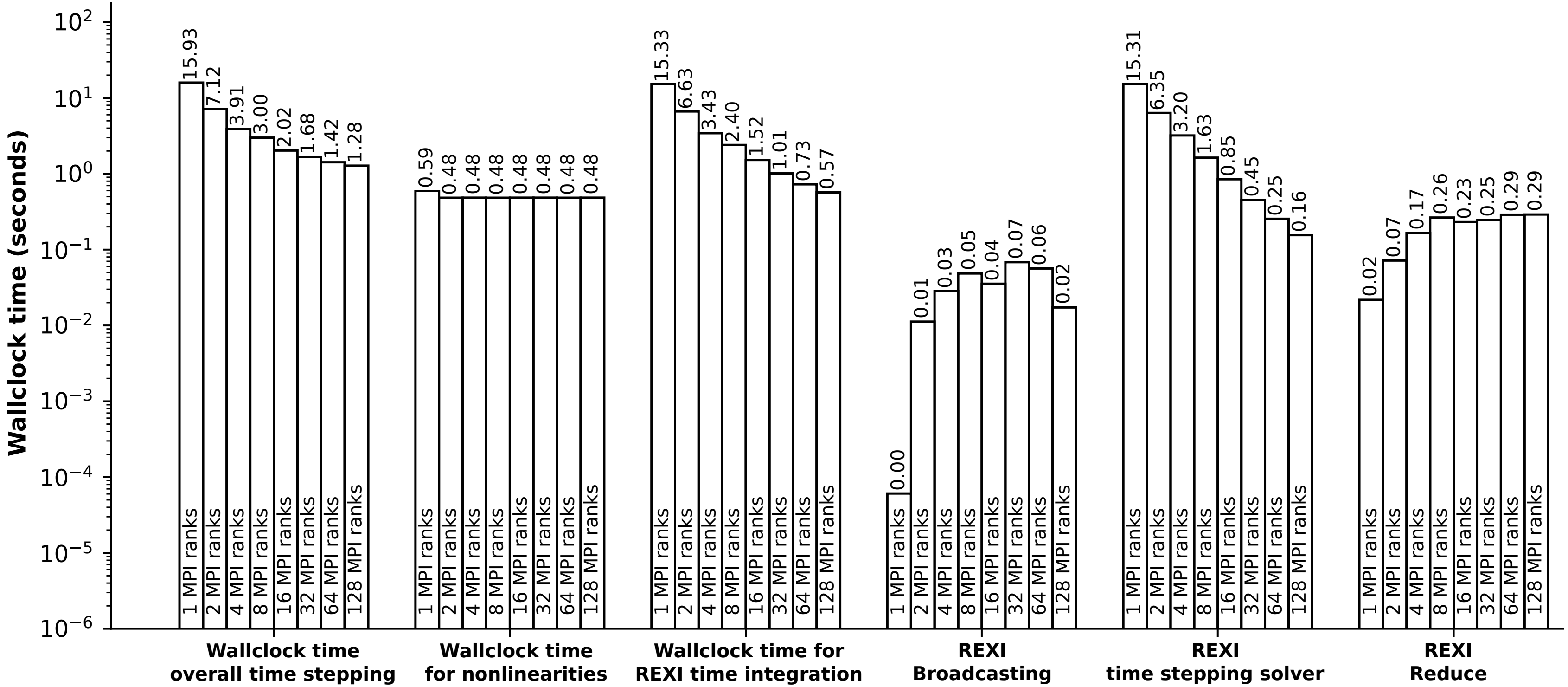


Schreiber, M., Schaeffer, N., & Loft, R. (2019). Exponential integrators with parallel-in-time rational approximations for the shallow-water equations on the rotating sphere. Parallel Computing, 85, 56–65. https://doi.org/10.1016/j.parco.2019.01.005

Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

Scalability breakdown

Based on best performing method $L_g(U) : \mathrm{REXI}, L_c(U) + N(U) : \mathrm{RK}, \mathrm{SSv1}$



REXI communication does (in this case) not dominate overall simulation time
More followup work on this soon (as part of KONWIHR funding)

Overview

- Introduction
- Exponential integration
  - Introduction
  - CI-REXI (Cauchy Contour Integral method)
  - Studies with SWE
    - Linear SWE: Error vs. wall clock time studies
    - Non-linear PDEs: Error vs. wall clock time studies
- **Summary & outlook**

Summary

- REXI with **linear equations**:
  No limitation on time step size with T-REXI method

- REXI with **nonlinear** equations:
  Improved error vs. wallclock time results

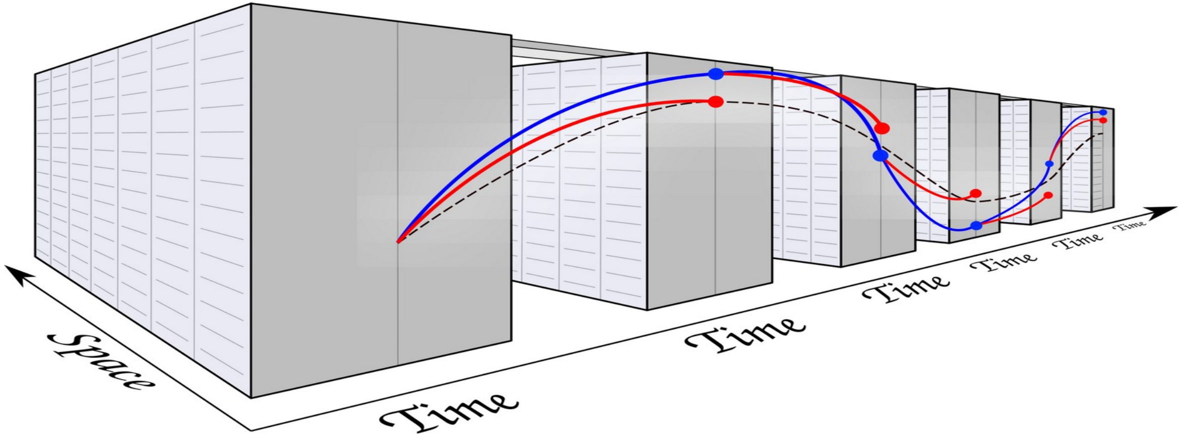
Future work

- REXI:
  - Efficient **REXI solvers** for non-global-spectral discretizations
    (work on fast direct solvers)
  - CI-REXI: Extension to **flexible polynomial-based contours**

- Semi-Lagrangian:
  - Proof-of-concept for EXP-SL on plane already shown (with Pedro Peixoto)
  - **EXP-SL** integration **on the sphere** is work in progress

Next-Generation Time Integration targeting Weather and Climate Simulations Part II: SDC, ML-SDC & PFASST

Martin Schreiber
Computer Architectures and Parallel Systems (of Equations)
Technical University of Munich

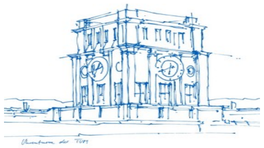


This presentation is based on collaborations with

Francois Hamon
Michael Minion

Cheers!

Seminar “Numerical methods for atmospheric and oceanic modelling: recent advances and future prospects”
ECMWF, Reading, UK, September 18th, 2020



Overview of different research directions

- **EXP:** Exponential time integration
  - T-REXI
  - CI-REXI
  - …
- **SL:** Semi-Lagrangian methods
  - State of the art
  - Combined with Parareal

Part of this presentation

- PFASST
  - **Spectral deferred correction** (SDC)
  - **Multi-level**
  - **Parallel-in-time** corrections (Parareal)

- **Combinations** of methods:
  SL / EXP / SDC / Parareal
- "**Tuning**" of methods (Convergence / accuracy)
- **EXP Machine learning** with ANN
- (Quantum Computing)

Overview

- Introduction

- **Spectral deferred Correction**
  - Introduction
  - Issues with DC
  - SDC example
- ML-SDC
  - Introduction
  - Results
- PFASST
  - Introduction
  - Results
-
- Summary & outlook

SDC: Idea of Deferred/Defect Corrections

- Let an **Cauchy problem** be given by an ODE

$$\frac{dU(t)}{dt} = f(t, U(t))$$

  with the initial condition

$$u(a) = u_0$$

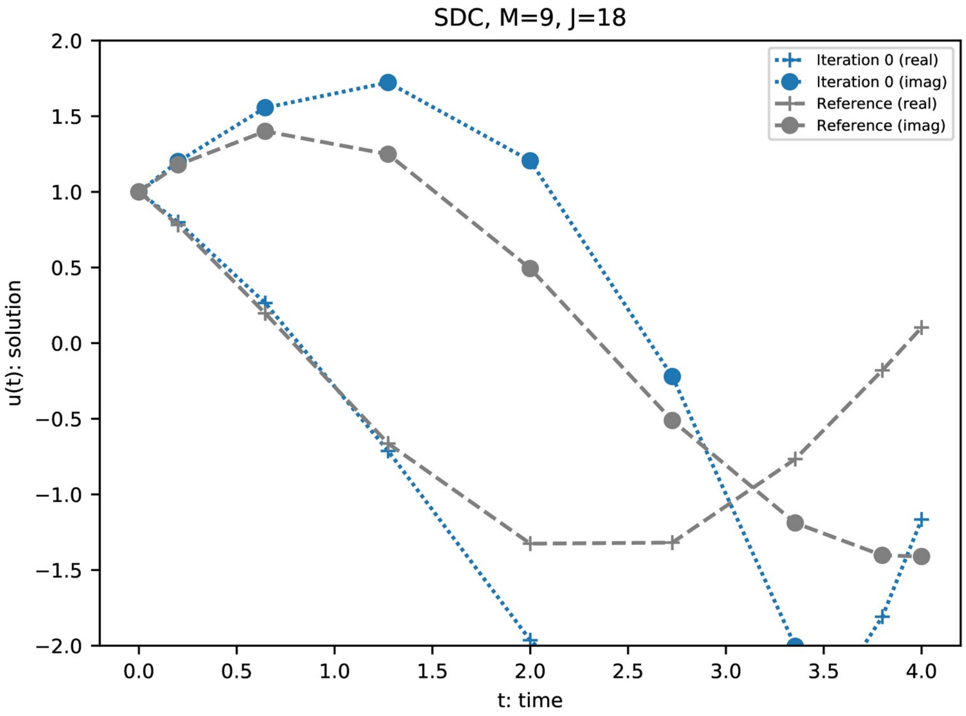
  at the beginning of each time step
- **Spectral Deferred Correction** (SDC) is **based on whatever "standard time integrator"** R there is

$$U_{n+1} = R(U_n, t, \Delta t_n)$$

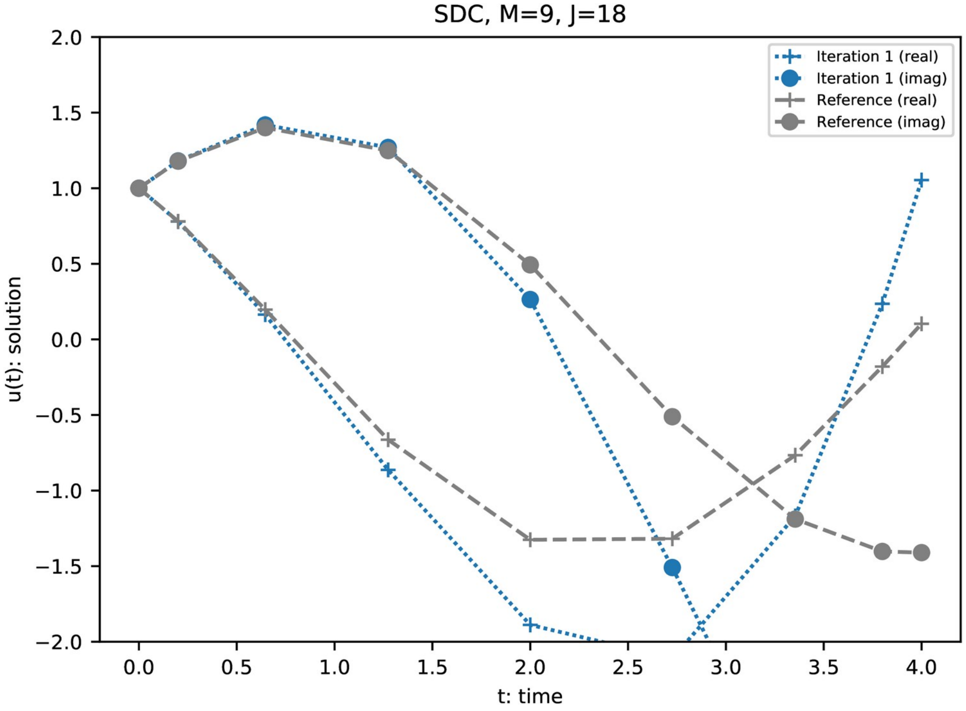
  which could be explicit, implicit, IMEX, exponential, basically whatever you want to put in.
- From a software point of view, **SDC can be used in a black-box fashion** by just providing "R".
- SDC then assembles the "R" evaluations in order to get a **higher-order time integration method**.

SDC Example: Initial guess

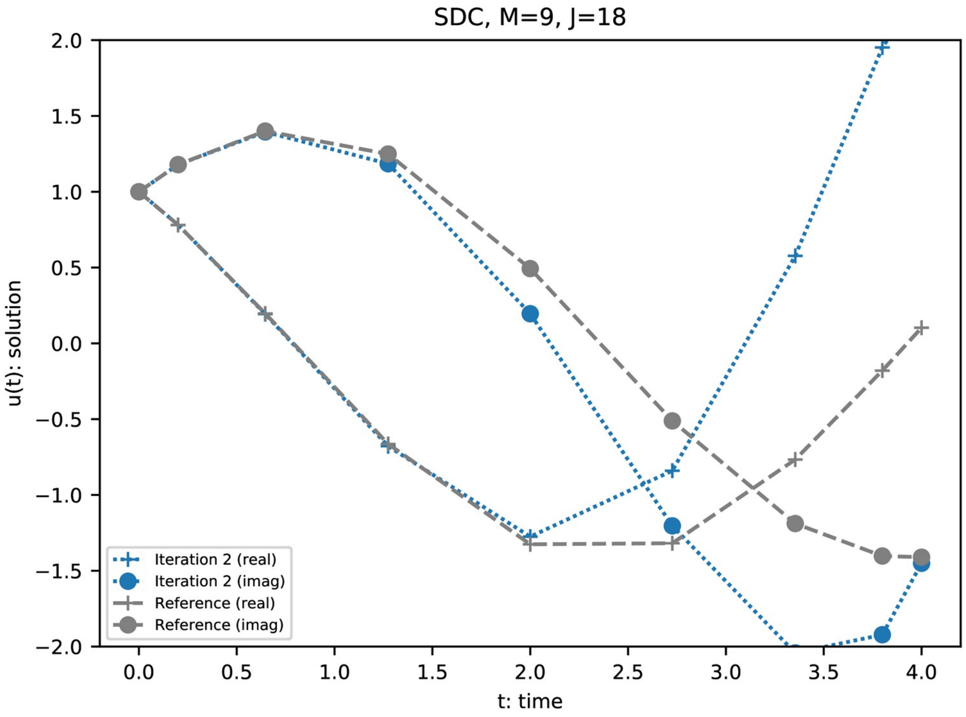
- Use k-th order accurate standard time integrator
- Linear oscillator



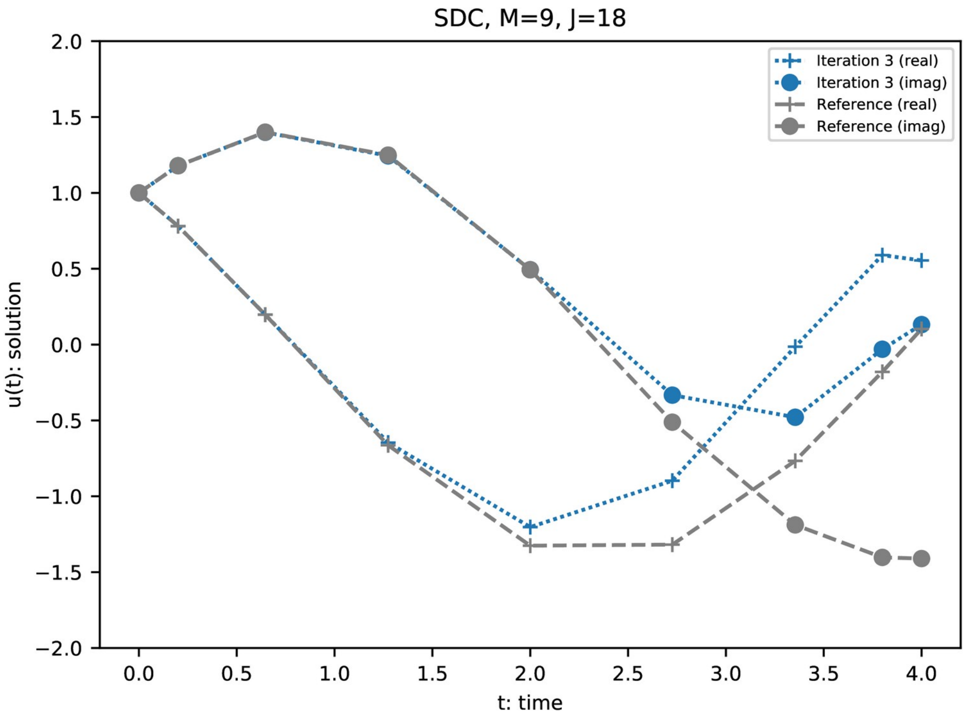
SDC Example: 1^{st} correction iteration



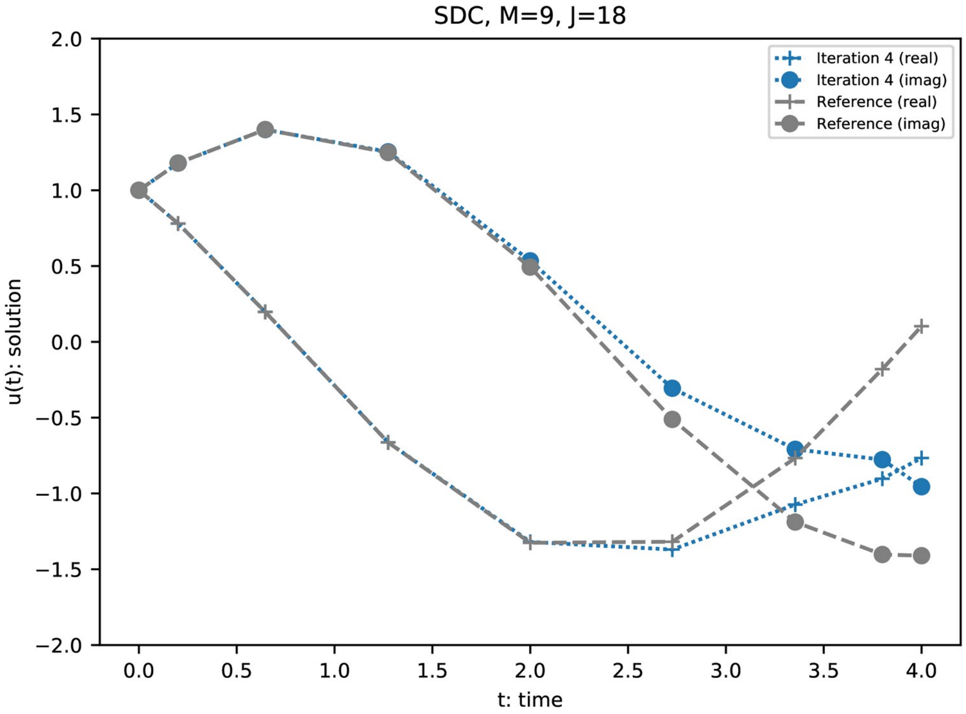
SDC Example: 2nd correction iteration



SDC Example: 3rd correction iteration

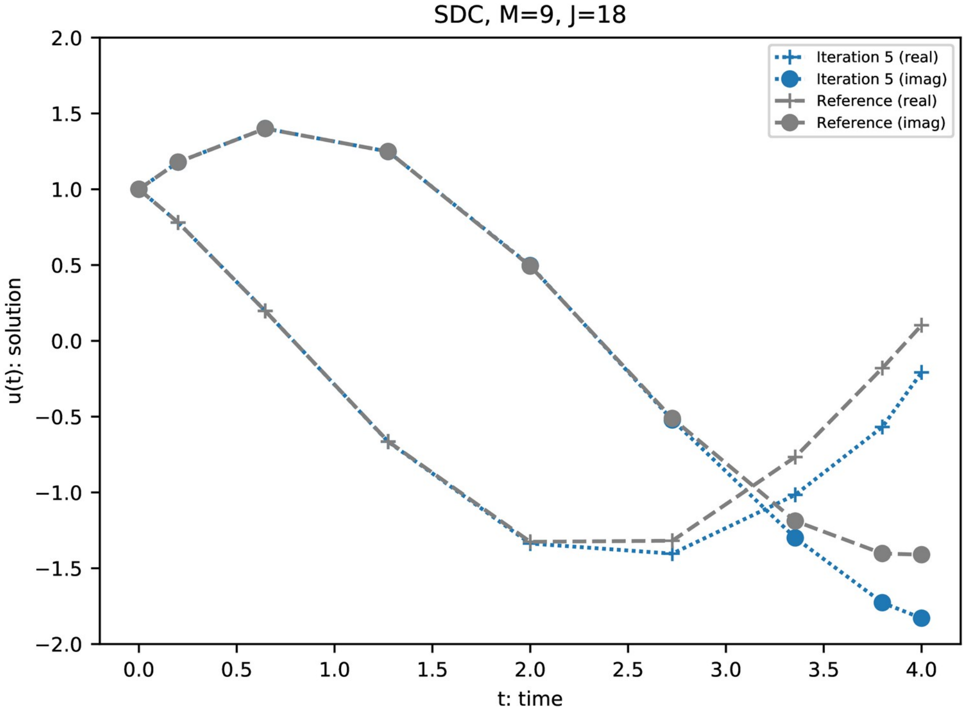


SDC Example: 4th correction iteration

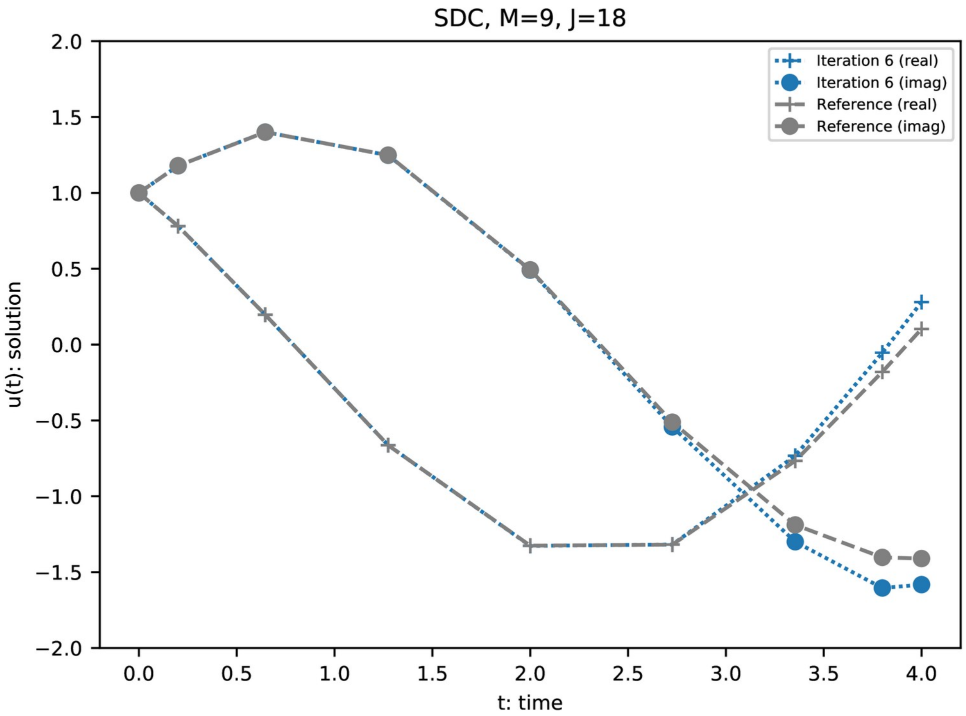


Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

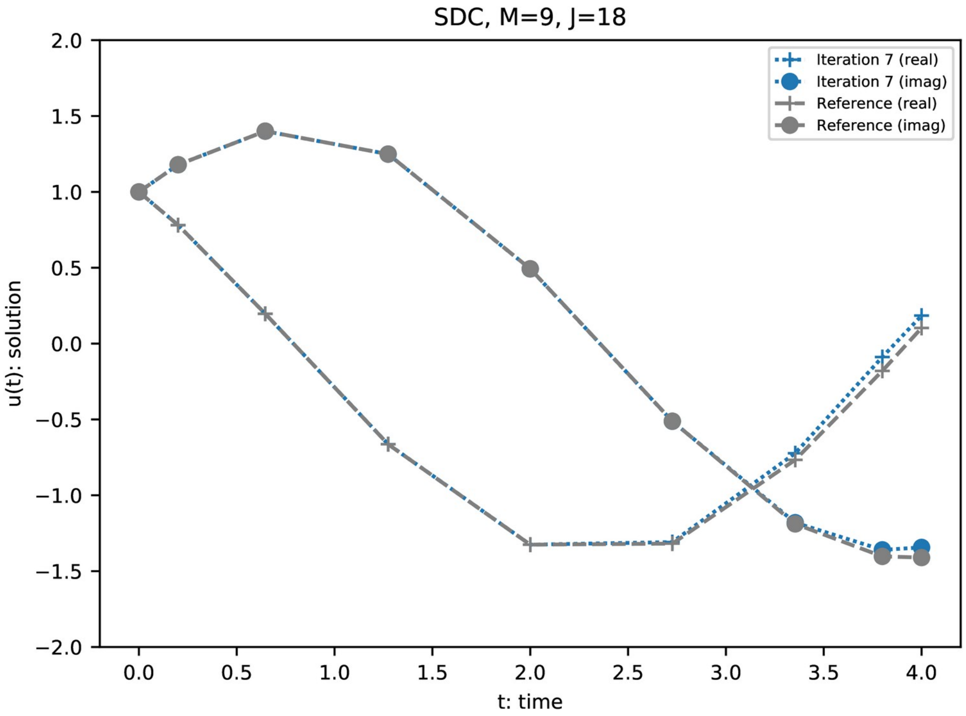
SDC Example: 5th correction iteration



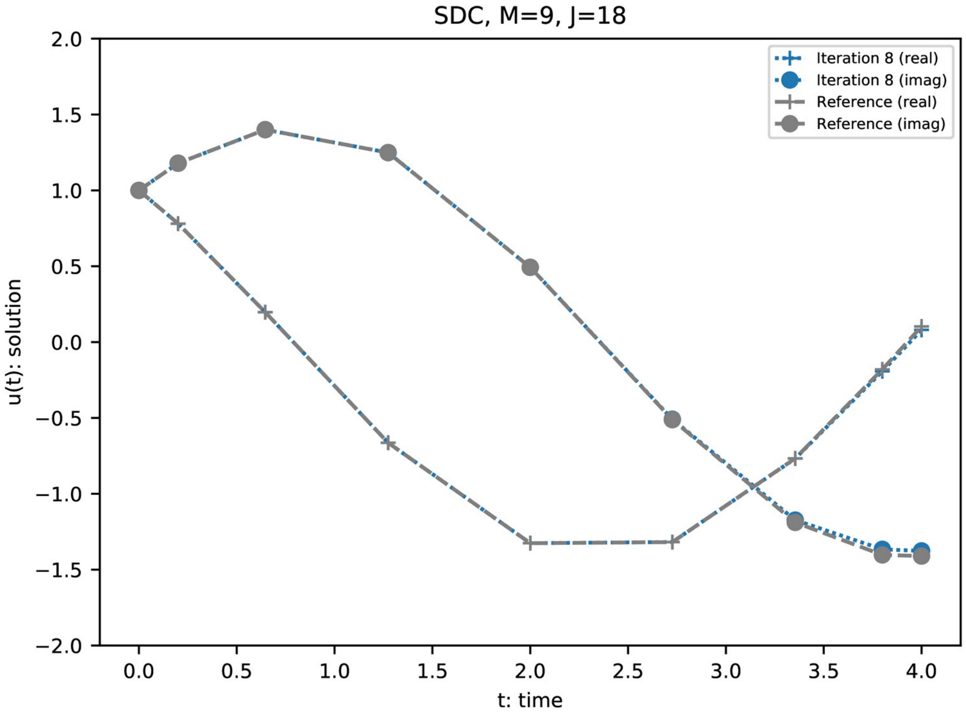
SDC Example: 6th correction iteration



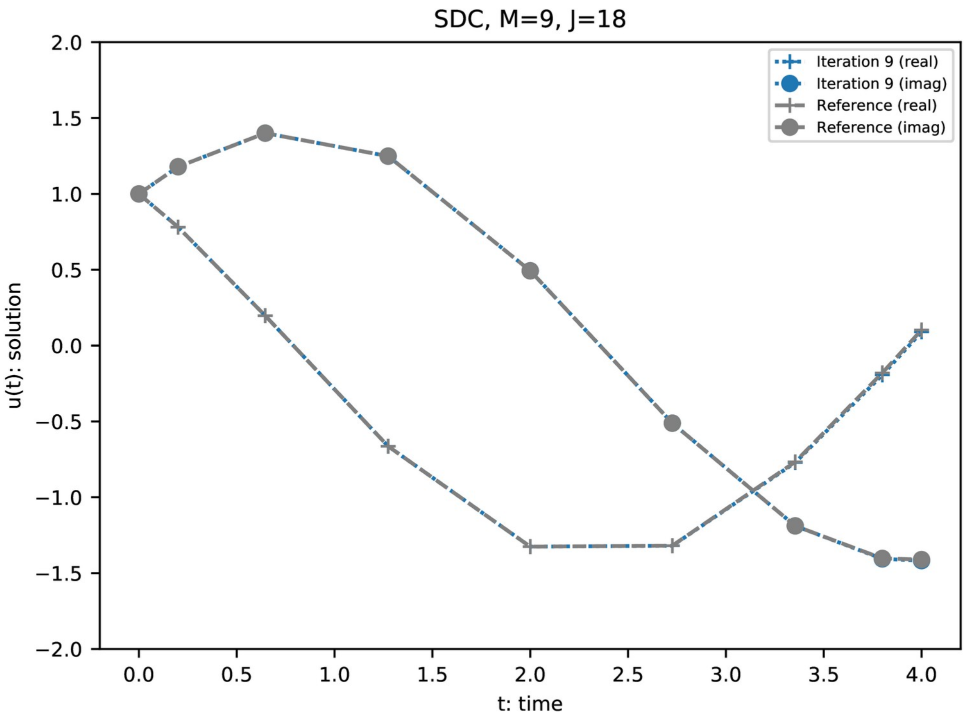
SDC Example: 7th correction iteration



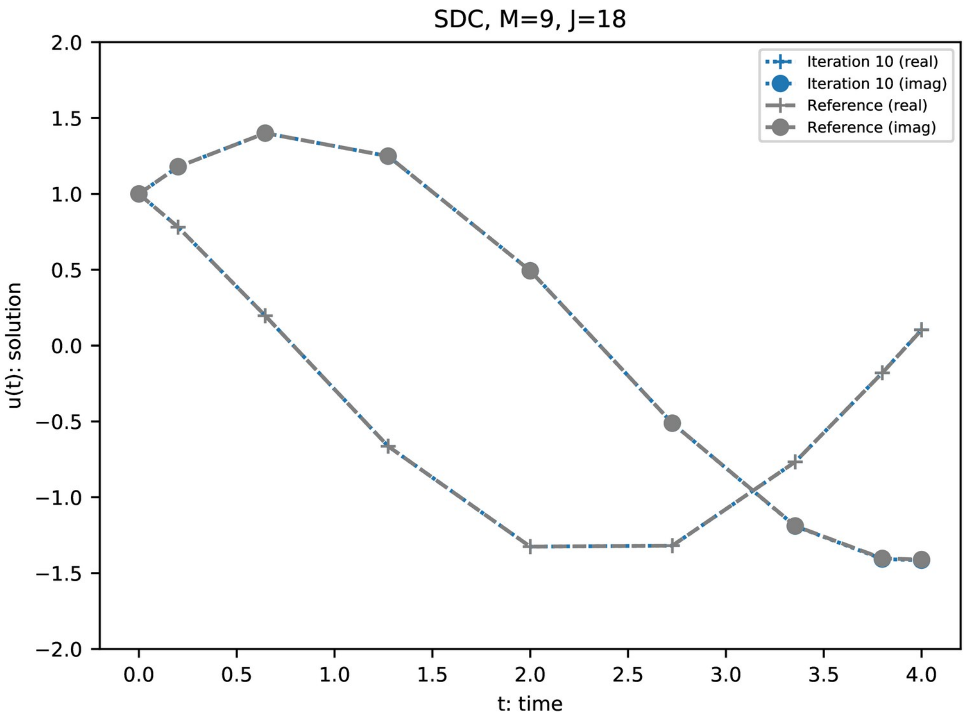
SDC Example: 8^{th} correction iteration



SDC Example: 9th correction iteration

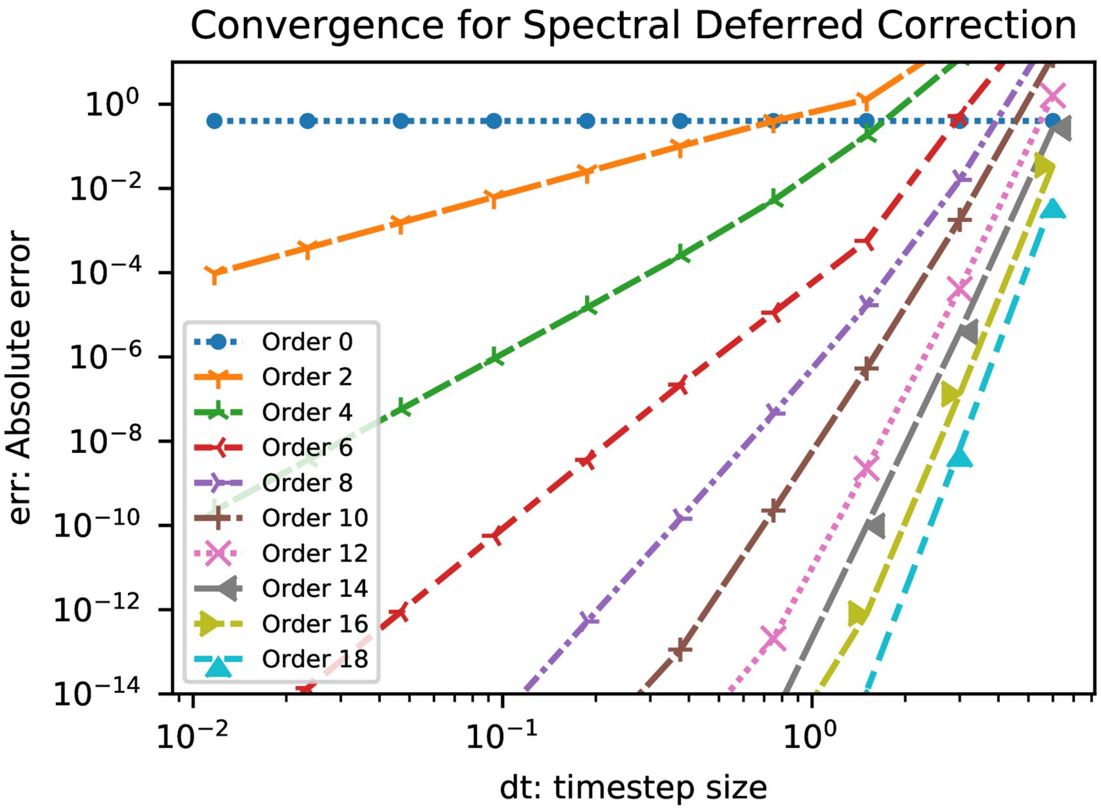


SDC Example: 10^{th} correction iteration



DC Convergence for different M

- Order = M+1 nodes



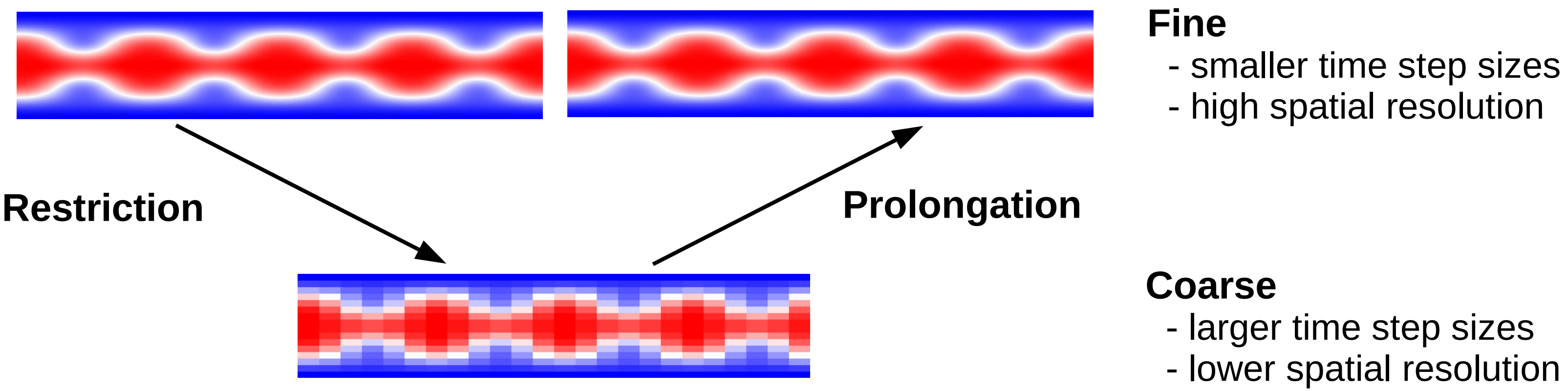
Overview

- Introduction

- Spectral deferred Correction
  - Introduction
  - Issues with DC
  - SDC example
- **ML-SDC**
  - Introduction
  - Results
- PFASST
  - Introduction
  - Results
-
- Summary & outlook

Multi-Level SDC

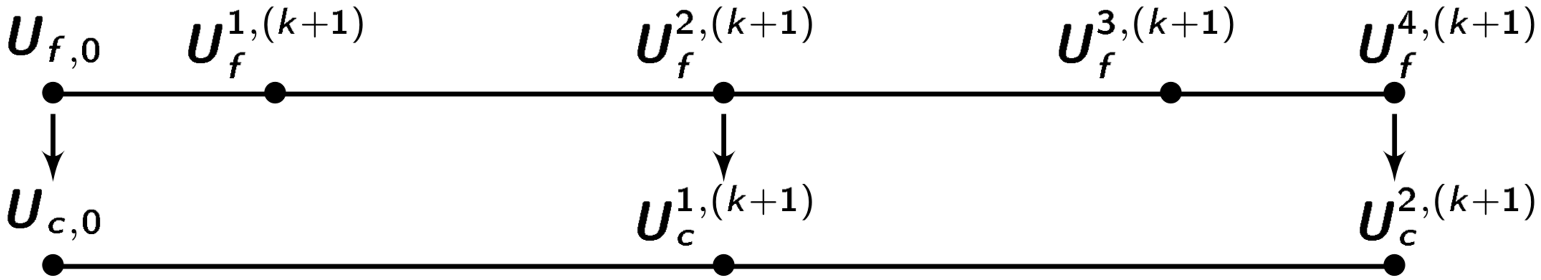
- Idea: Replace some of the computationally expensive SDC sweeps on high resolution with **computationally less expensive ones on a lower resolution**
- Using **full approximation scheme** (should be known from multi-grid methods)



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

Coarse/fine projections

- **Restriction**
  - Time:
    Point-wise injection at the GL SDC nodes



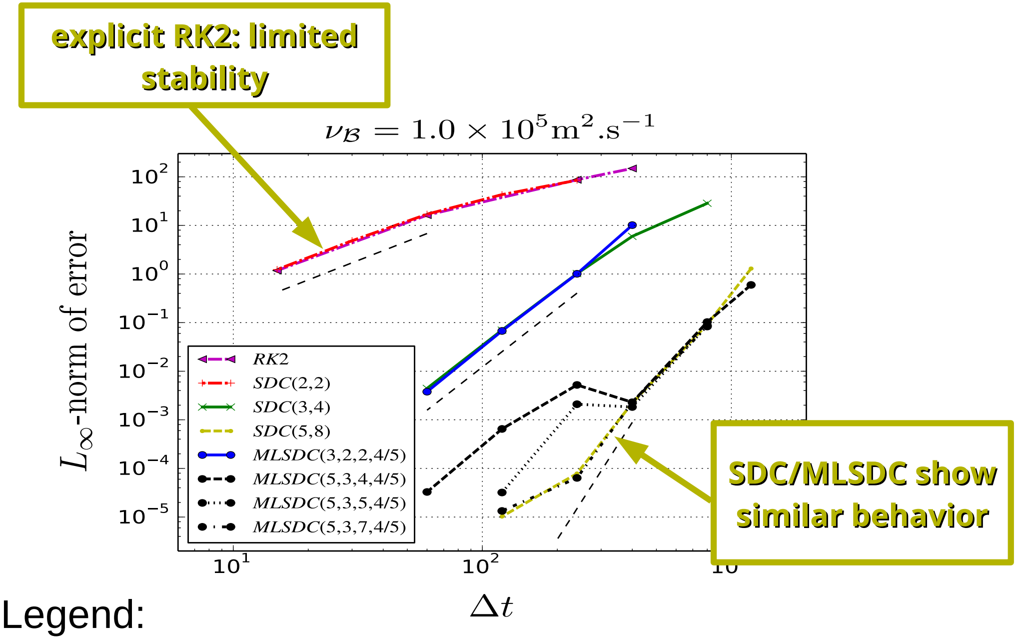
  - Space:
    Modal truncation based on spherical harmonics (“p”-adaptivity)

- **Prolongation**
  Time/Space: Canonical to restriction

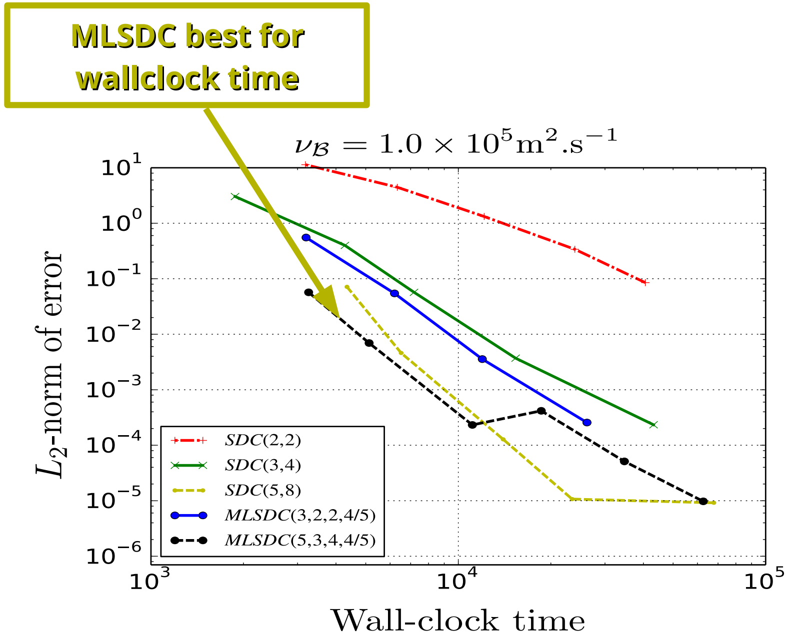
Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

Multi-Level Spectral Deferred Corrections

Error vs. timestep size



Error vs. wallclock time



Legend:
SDC(# nodes, # sweeps)
MLSDC(# fine nodes, # coarse nodes, # MLSDC sweeps, coarsening ratio)

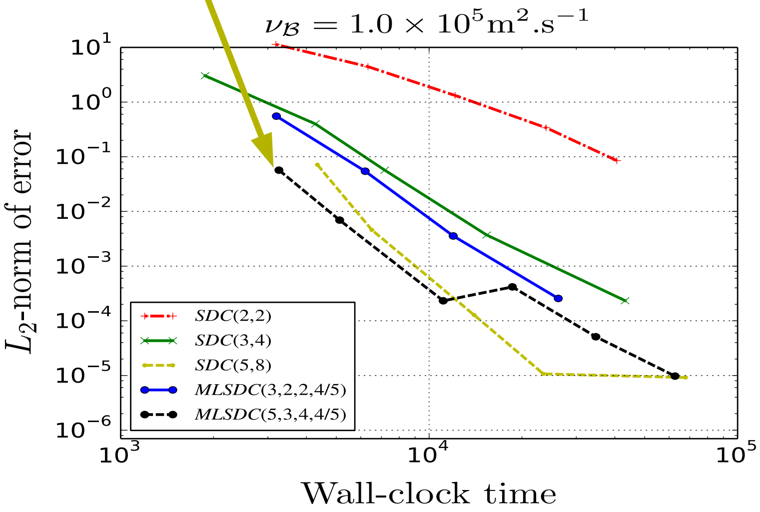
Hamon, F. P., Schreiber, M., and Minion, M. L. (2019). Multi-level spectral deferred corrections scheme for the shallow water equations on the rotating sphere. Journal of Computational Physics

Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

MLSDC: Error vs. wallclock time

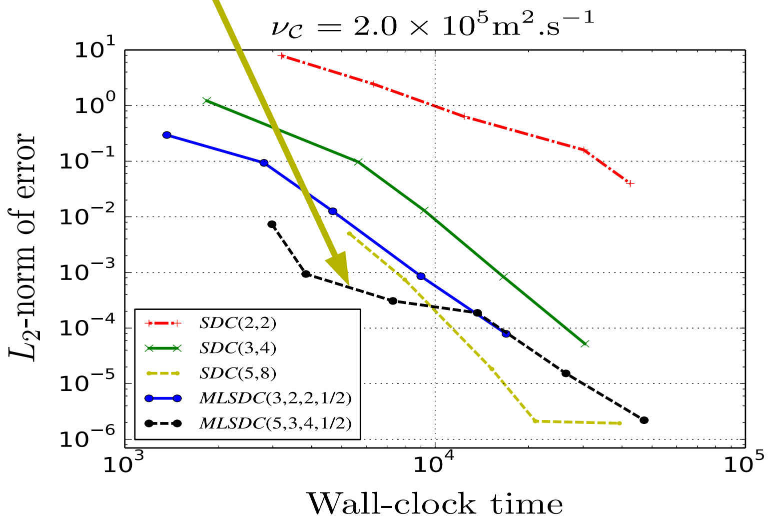
Moderate coarsening
+ Less viscosity required

MLSDC slightly better



Aggressive coarsening
+ More viscosity required!

MLSDC best wallclock time vs. error



Legend:
SDC(# nodes, # sweeps)
ML-SDC(# fine nodes, # coarse nodes, # MLSDC sweeps, coarsening ratio)

Hamon, F. P., Schreiber, M., and Minion, M. L. (2019). Multi-level spectral deferred corrections scheme for the shallow water equations on the rotating sphere. Journal of Computational Physics

Overview

- Introduction
- Spectral deferred Correction
  - Introduction
  - Issues with DC
  - SDC example
- ML-SDC
  - Introduction
  - Results
- **PFASST**
  - Introduction
  - Results
- Summary & outlook

Parareal algorithm

- The first step is to **subdivide the time** integration interval $[0; T]$ into **coarse time steps** of size Δt

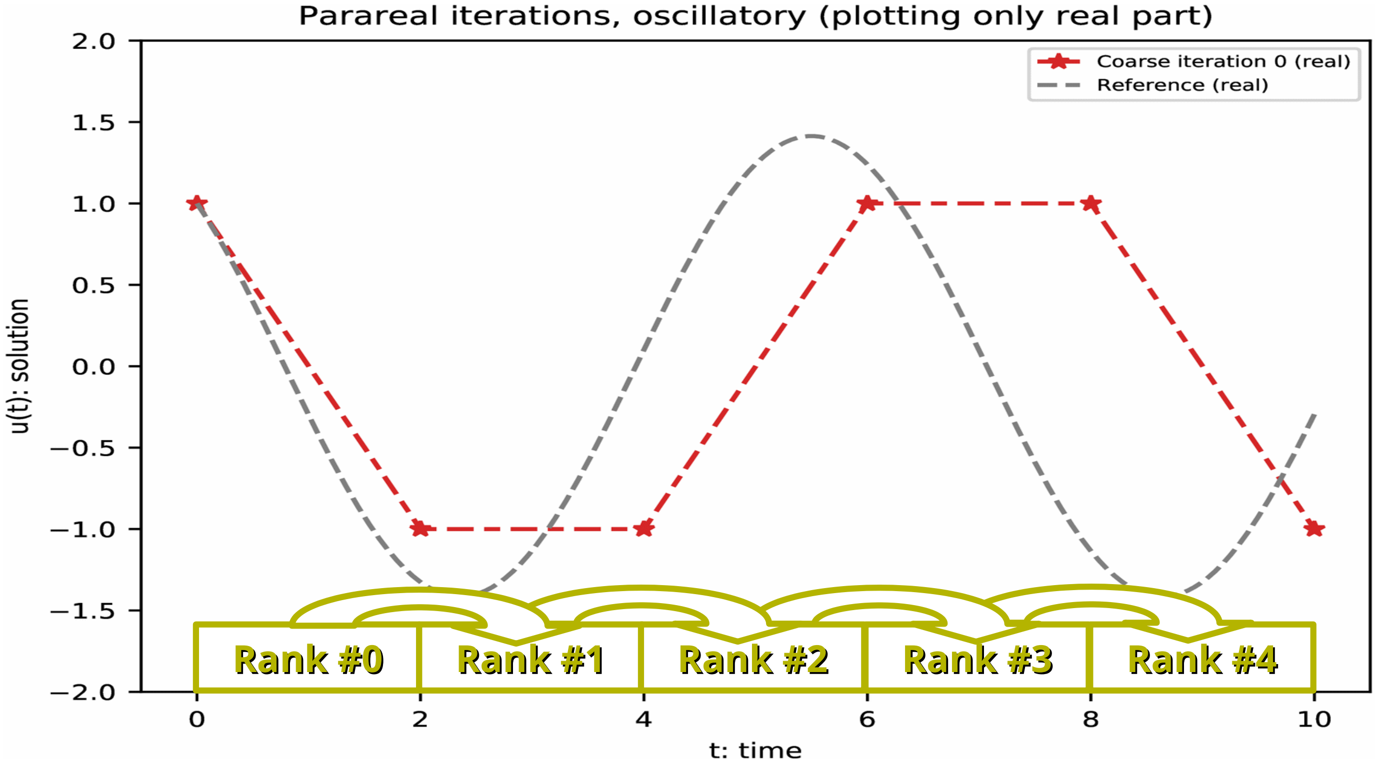
- We also require two **time integrators**
  - **Fine** time integrator $\mathcal{F}(u, t_n, t_{n+1}) = \mathcal{F}(u, \Delta t_n)$ and
  - **Coarse** time integrator $\mathcal{C}(u, t_n, t_{n+1}) = \mathcal{C}(u, \Delta t_n)$
    Needs to be **much faster than the fine integrator!**

- (Disclaimer: In the following slides, we will discuss the most basic version of the Parareal algorithm)

Lions, J. L., Maday, Y., & Turinici, G. (2001). Résolution d'EDP par un schéma en temps «pararéel ». Comptes Rendus de l'Academie Des Sciences - Series I: Mathematics, 332(7), 661–668. https://doi.org/10.1016/S0764-4442(00)01793-6

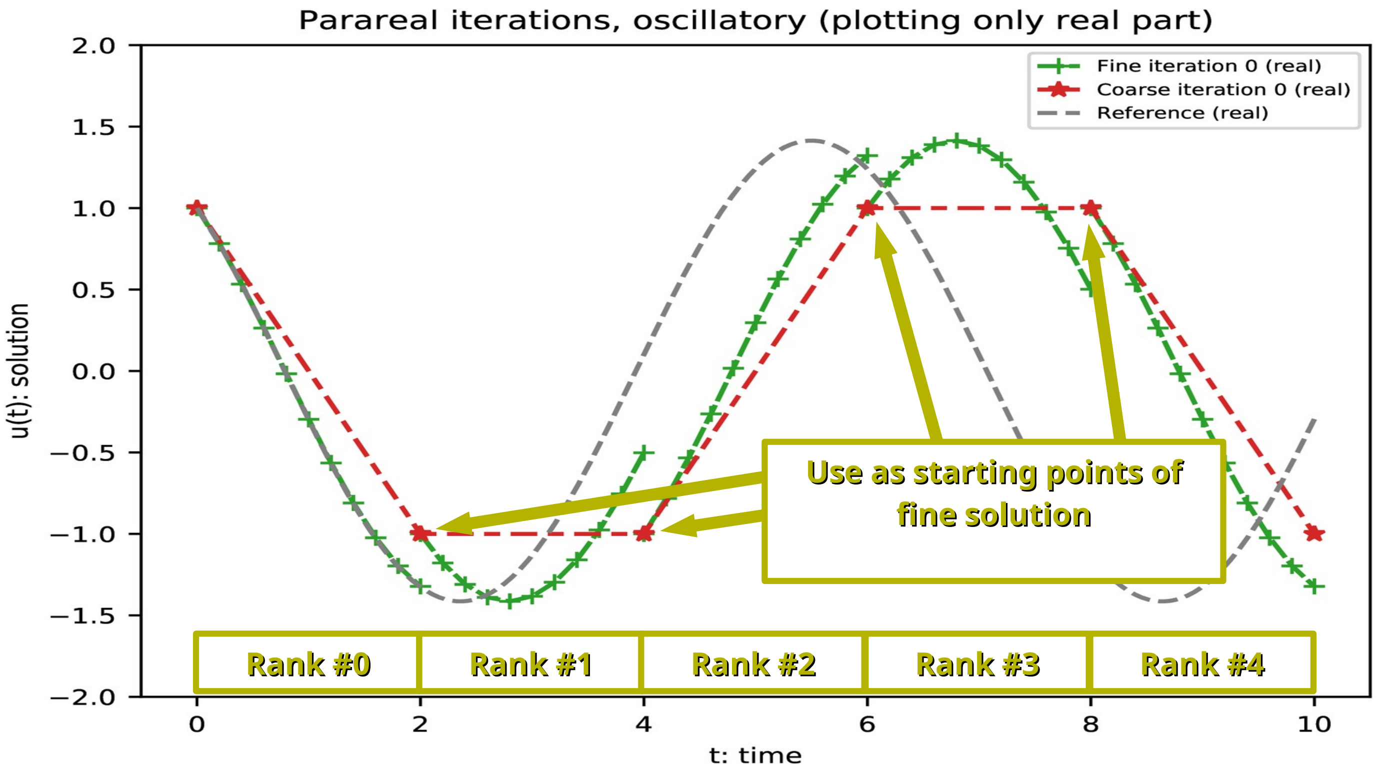
Parareal: Initial iteration

- Run **coarse time integrator**
- Purely **sequential**, hence requiring **fast coarse integrator**



Parareal: 1st iteration, fine time stepper

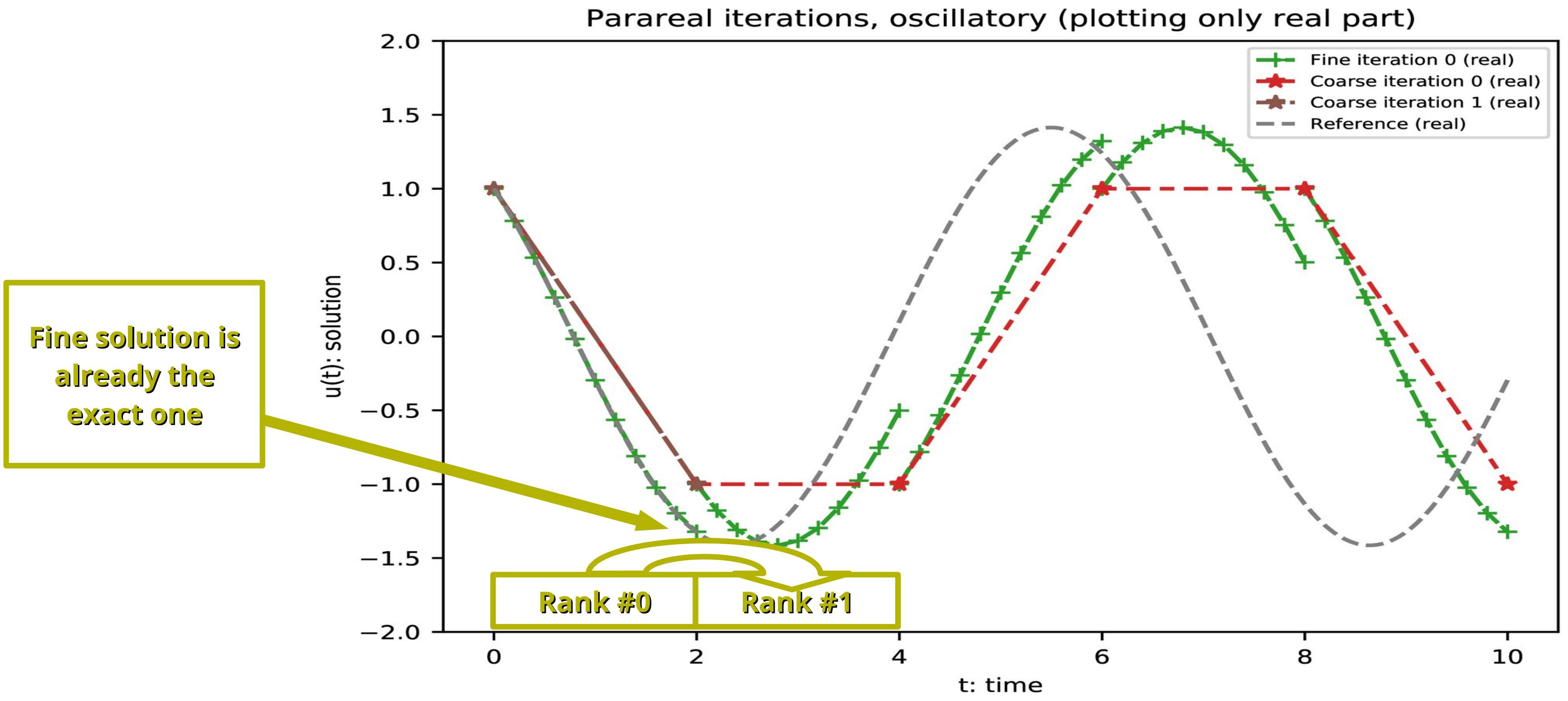
- Run fine time steppers
- **In parallel across all coarse time steps**



Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

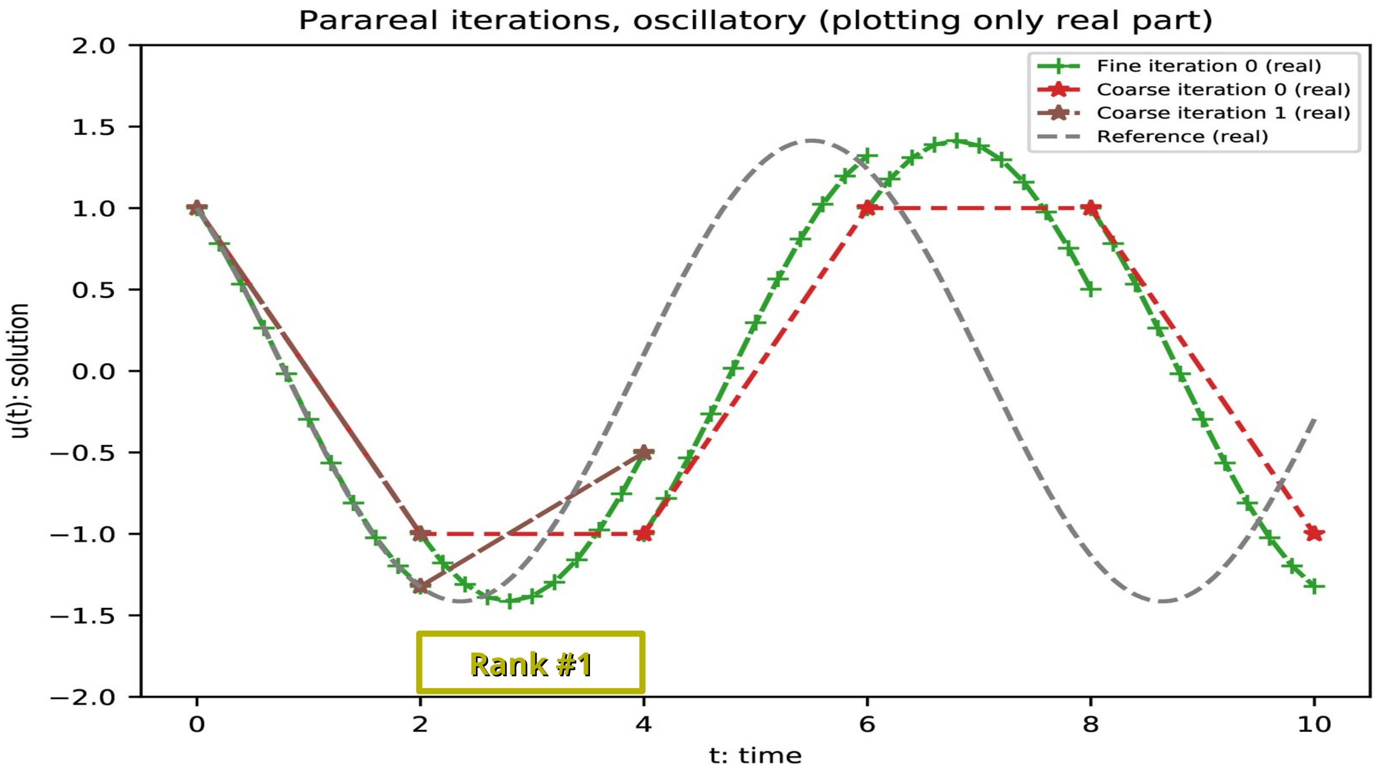
Parareal: 1st iteration, rank #0

- Nothing to do, already converged to fine time stepping solution



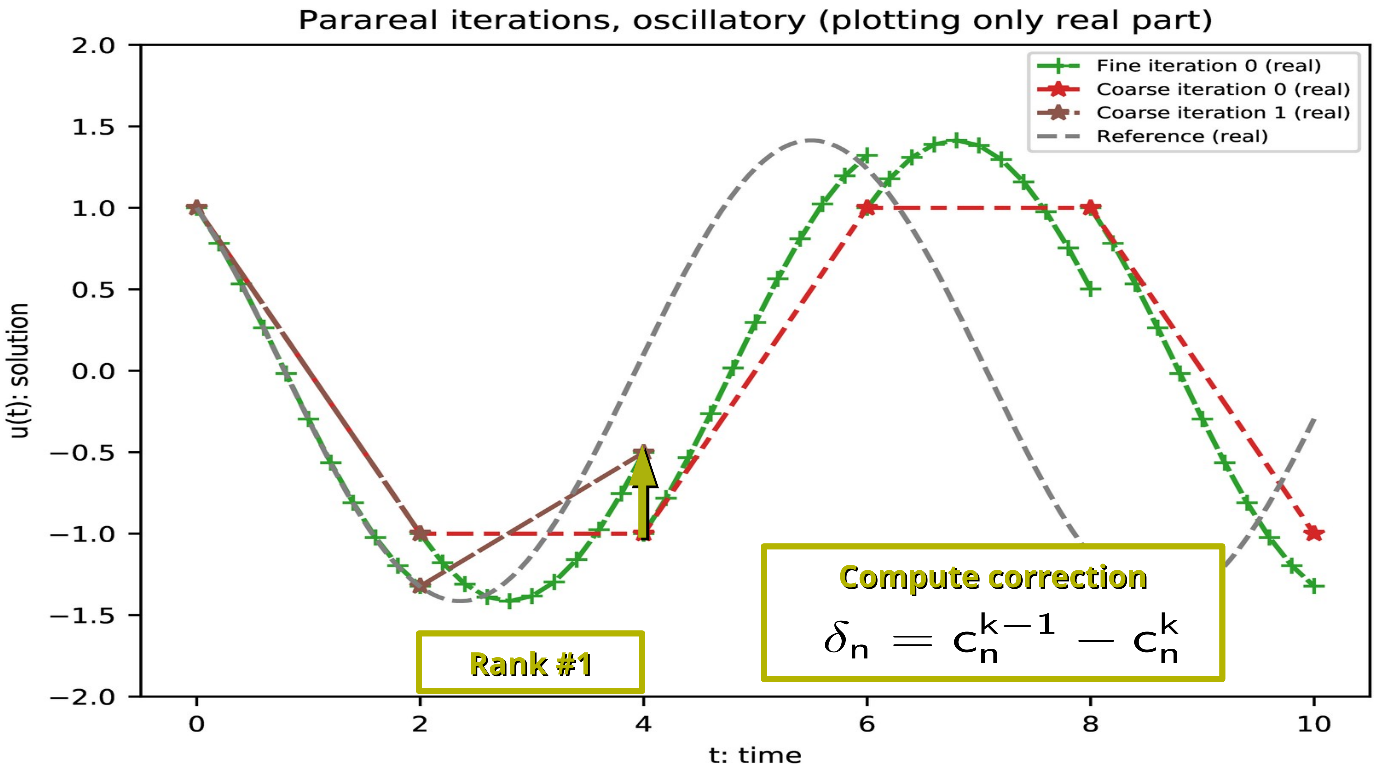
Parareal: 1st iteration, rank #1

- Run coarse time step with new initial value



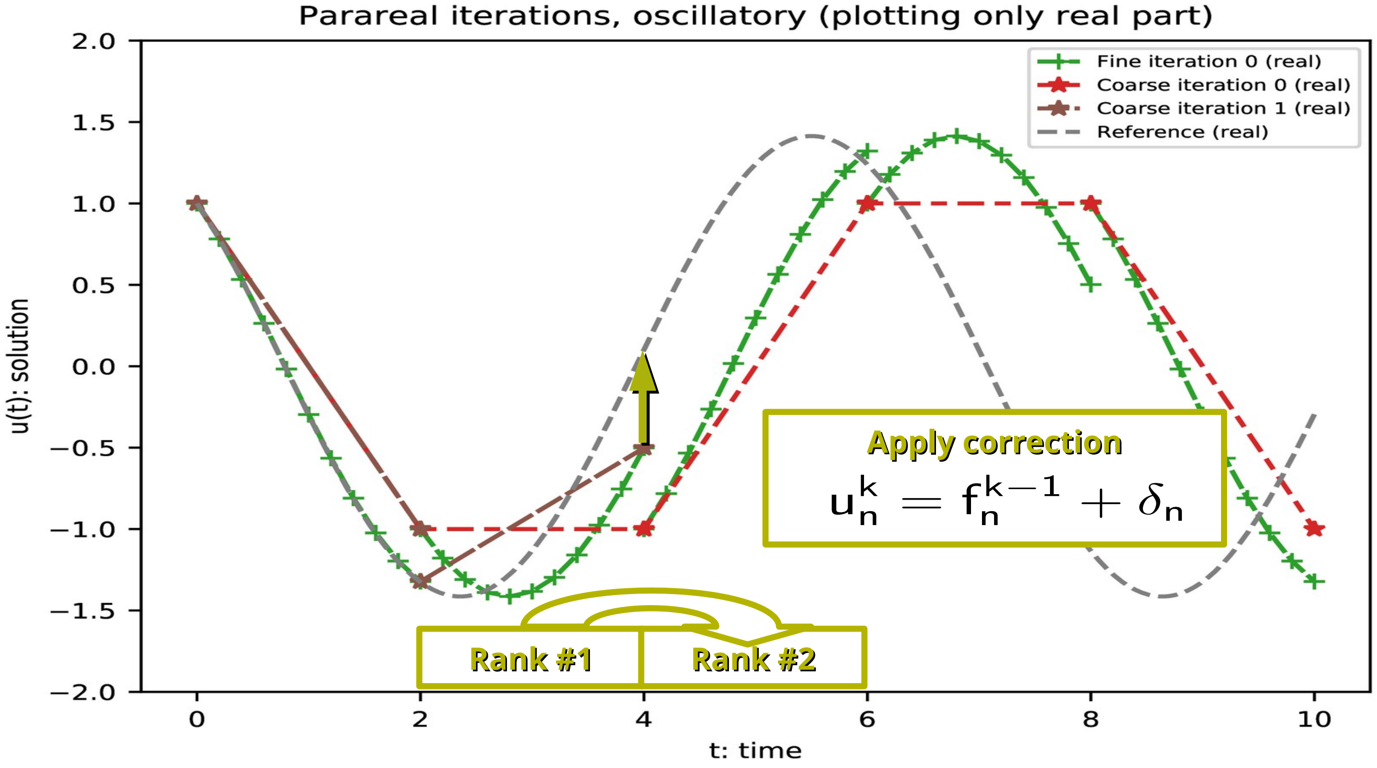
Parareal: 1st iteration, rank #1

- Compute correction



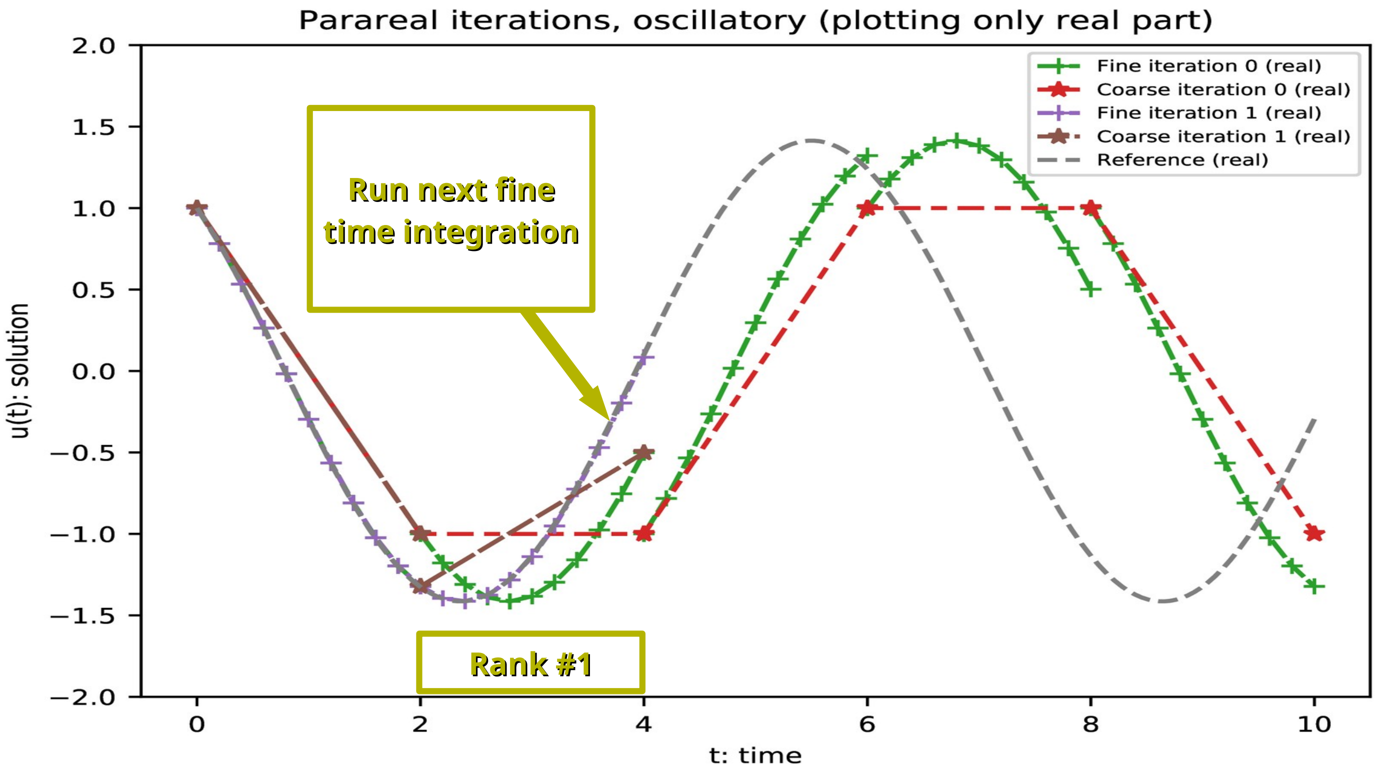
Parareal: 1st iteration, rank #1

- Compute correction & **send new corrected value** to next rank #2
- **Rank #2 starts iterating in parallel** (not yet shown here)



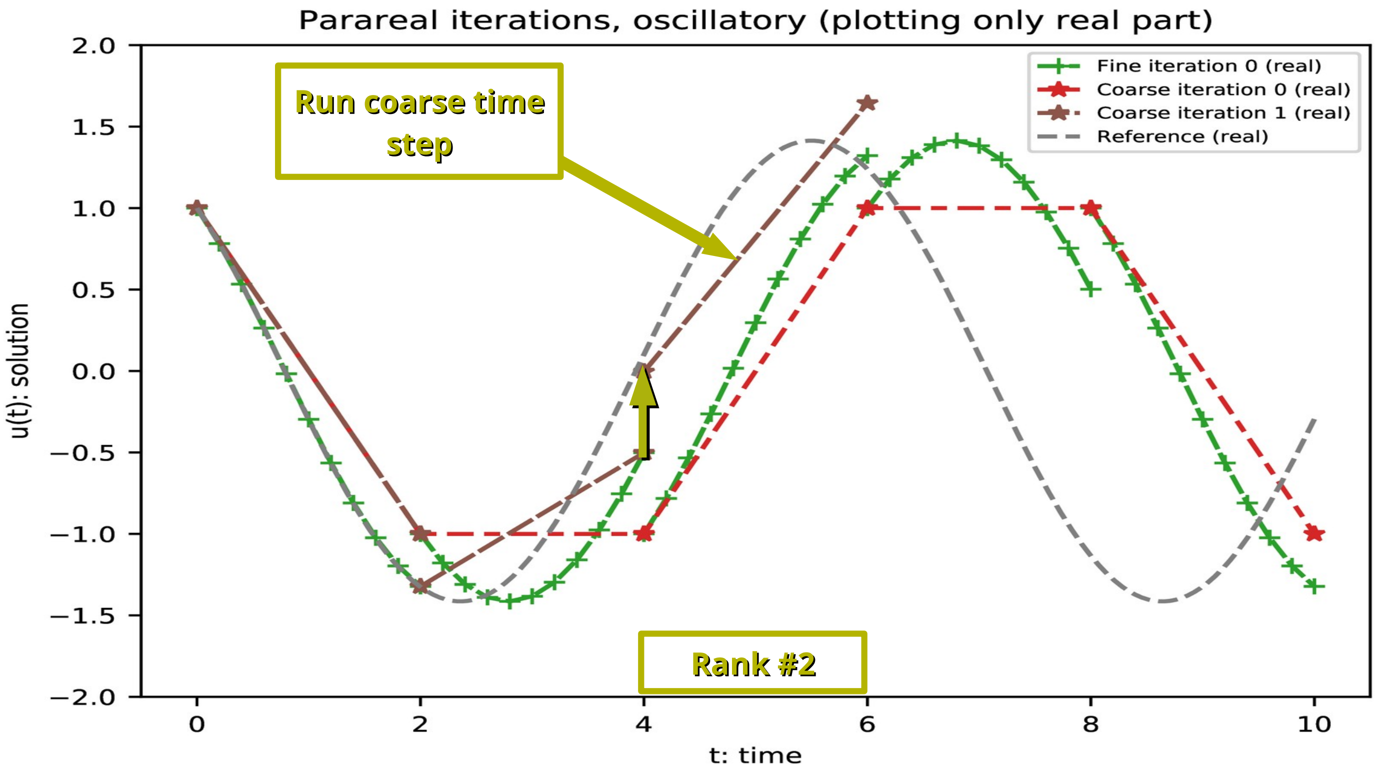
Parareal: 1st iteration, rank #1

- Run fine time stepping **parallel to rank #2 computations**
- Continue with next iteration



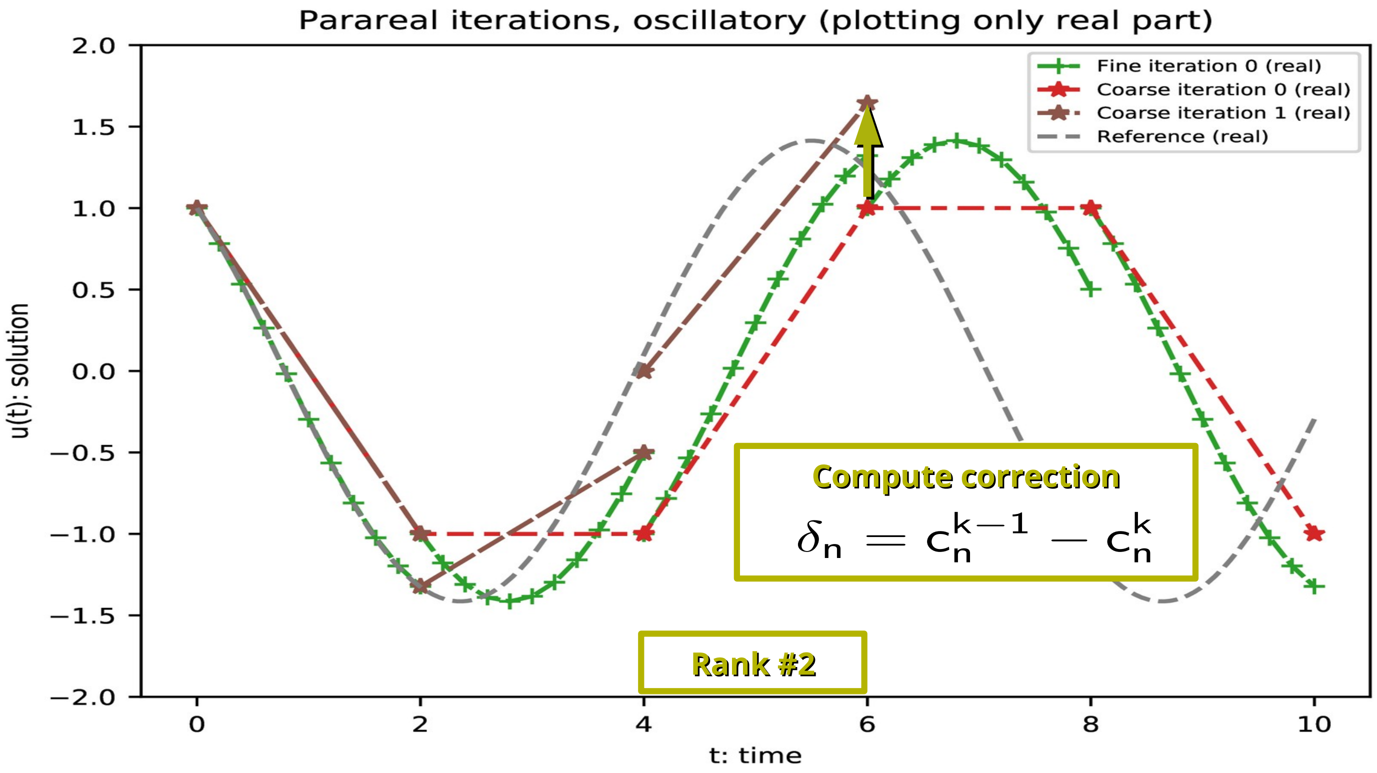
Parareal: 1st iteration, rank #2

- This happens in parallel to previous slide!
- Compute coarse time integration



Parareal: 1st iteration, rank #1

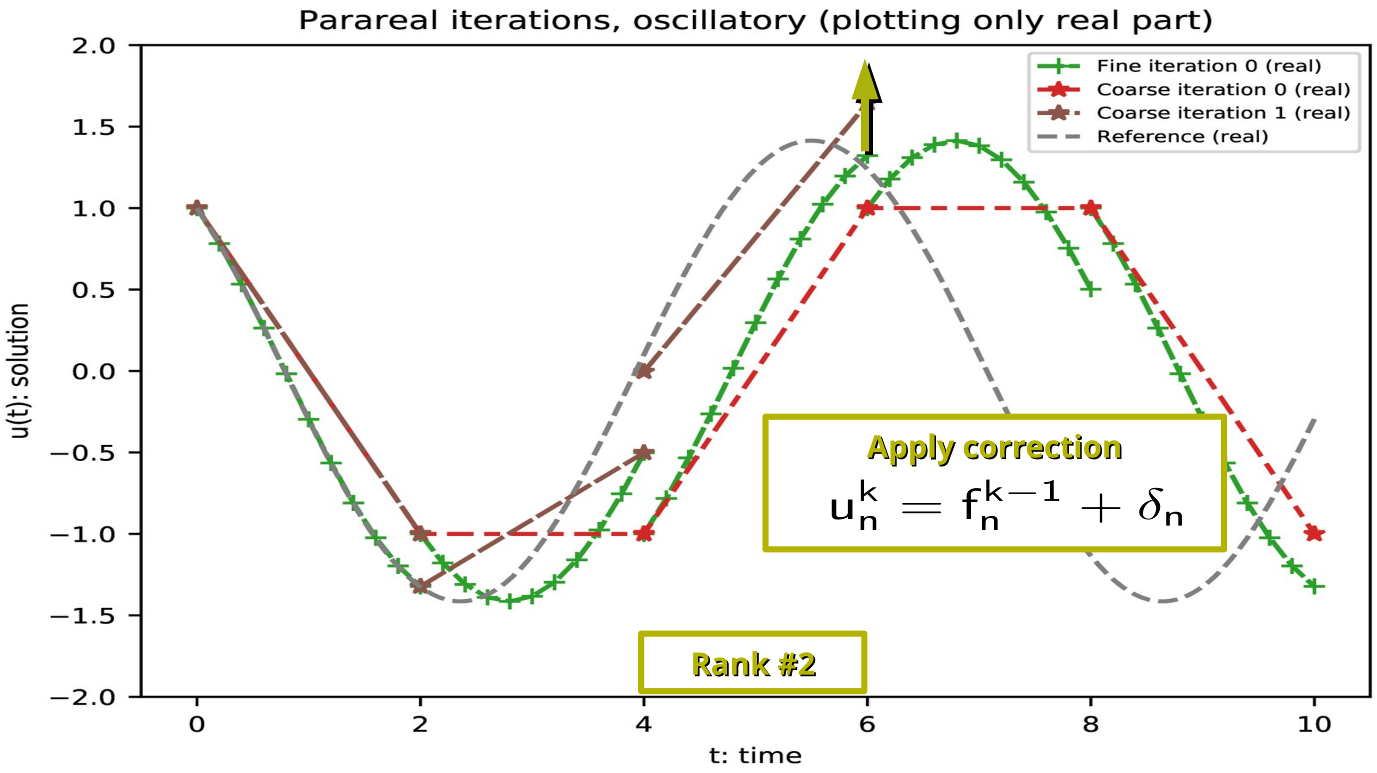
- Compute correction



Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

Parareal: 1st iteration, rank #2

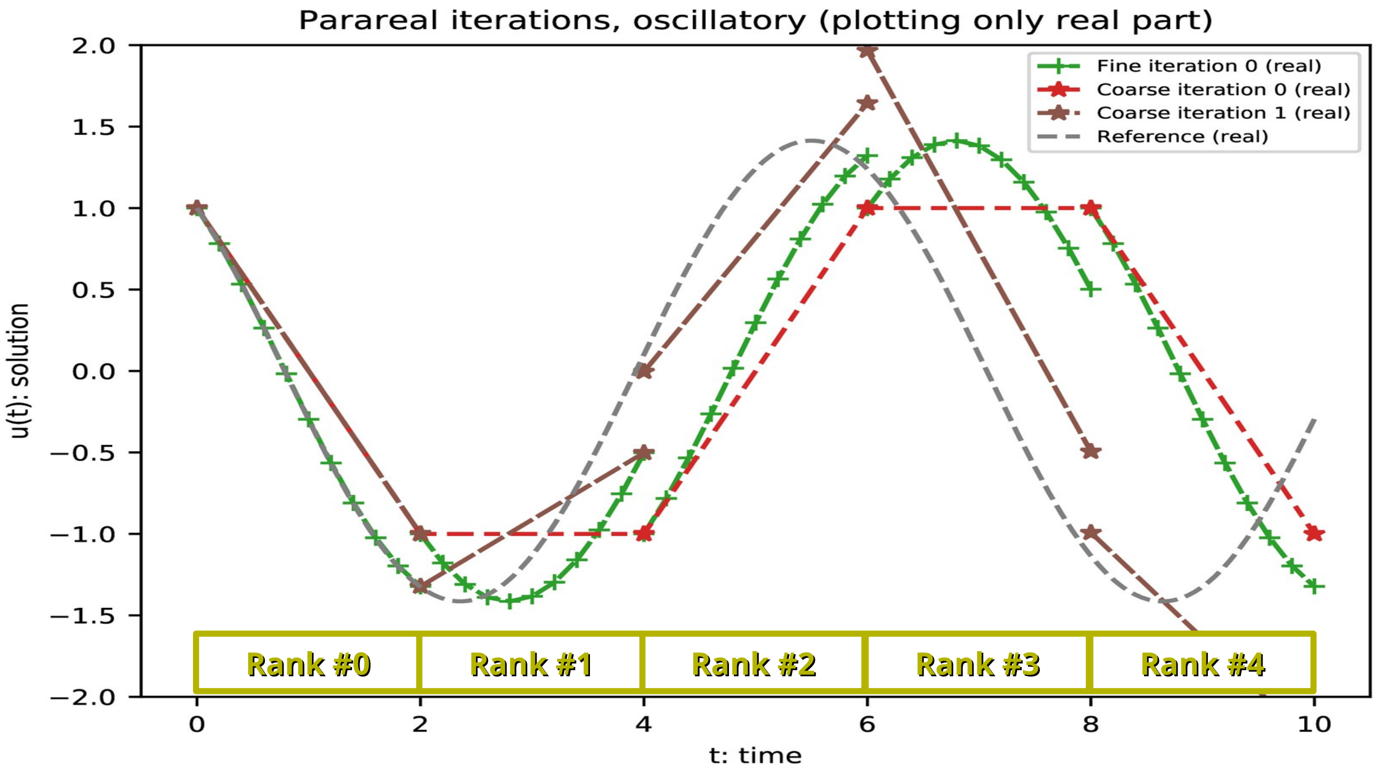
- Compute correction & **send new corrected value** to next rank #3
- **Rank #3 starts iterating in parallel** (not yet shown here)



Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

Parareal: 1^{st} iteration

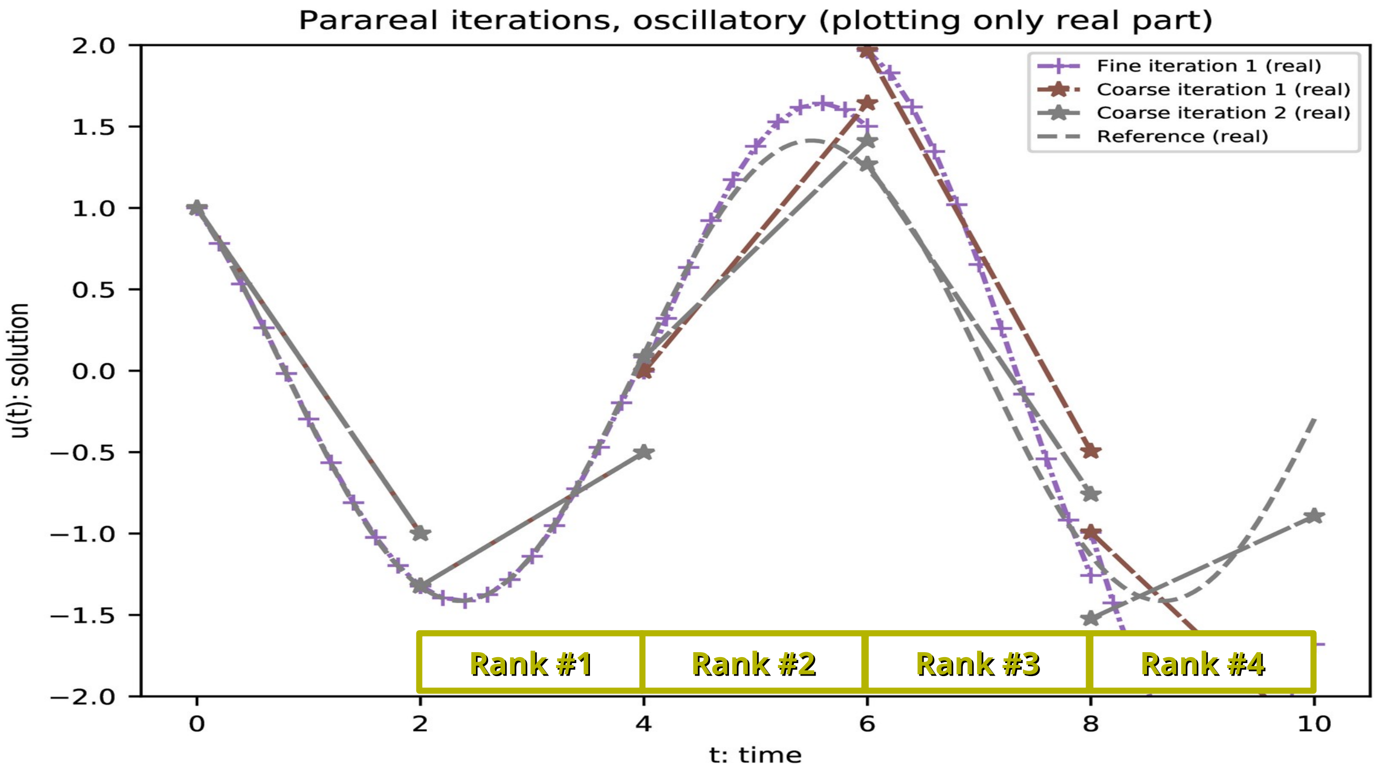
- Visualization of **full Parareal iteration from hereon**



Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

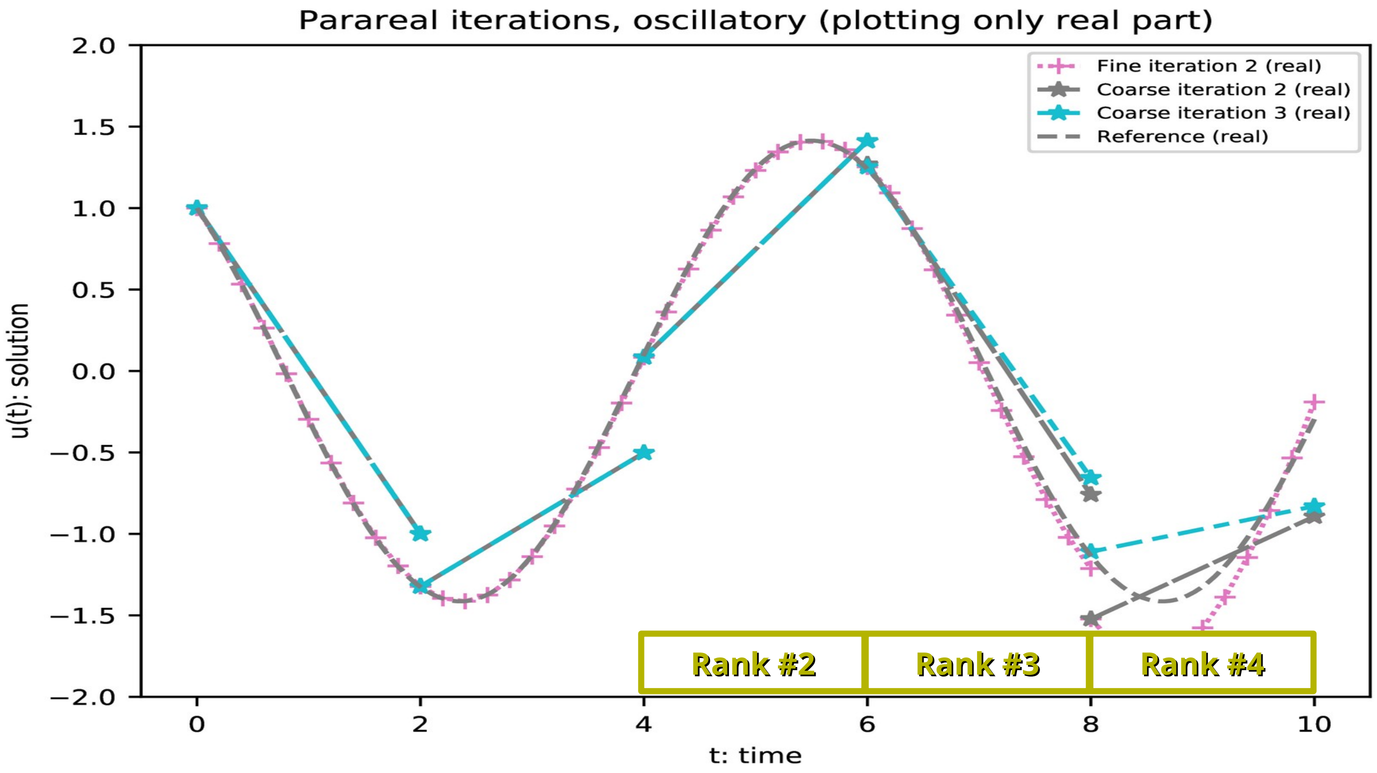
Parareal: 2nd iteration

- Not yet converged



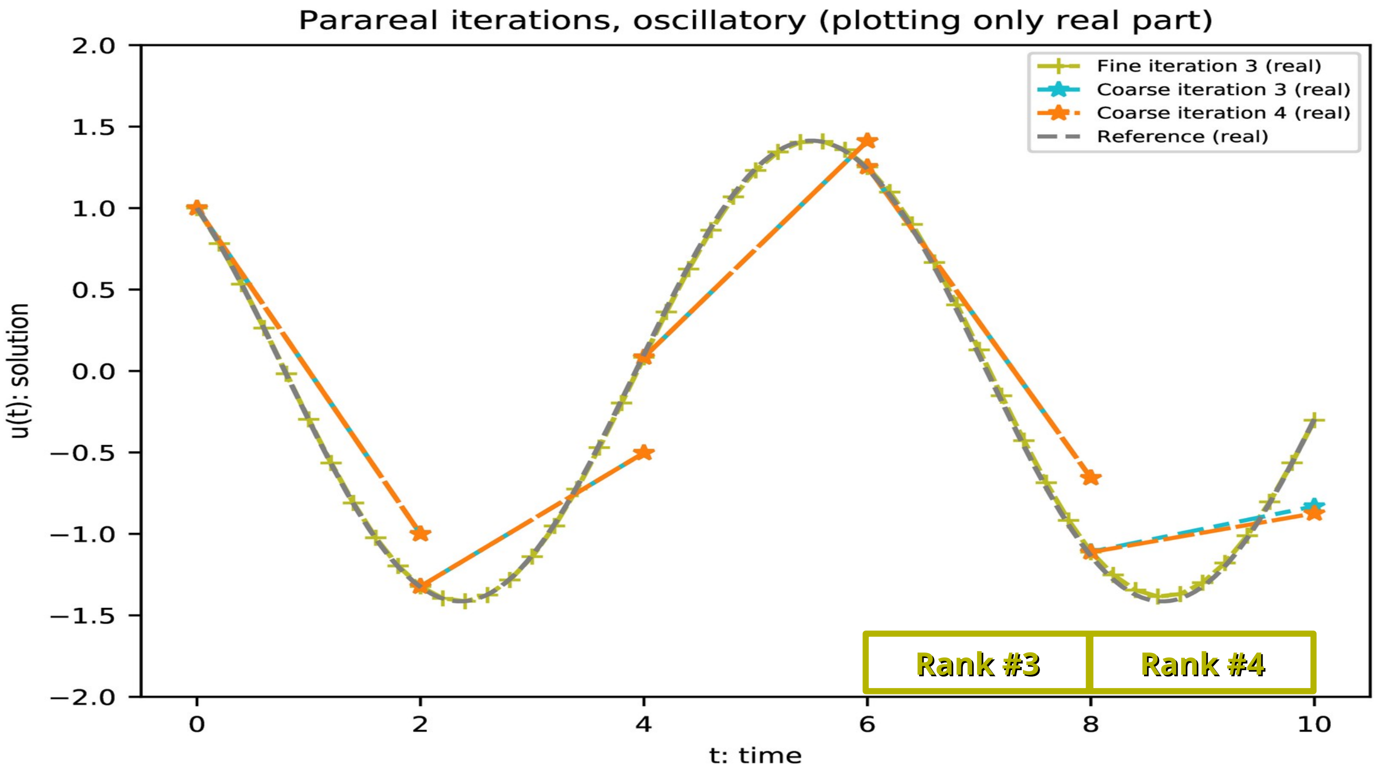
Parareal: 3rd iteration

- Relative high accuracy after **3 iterations**



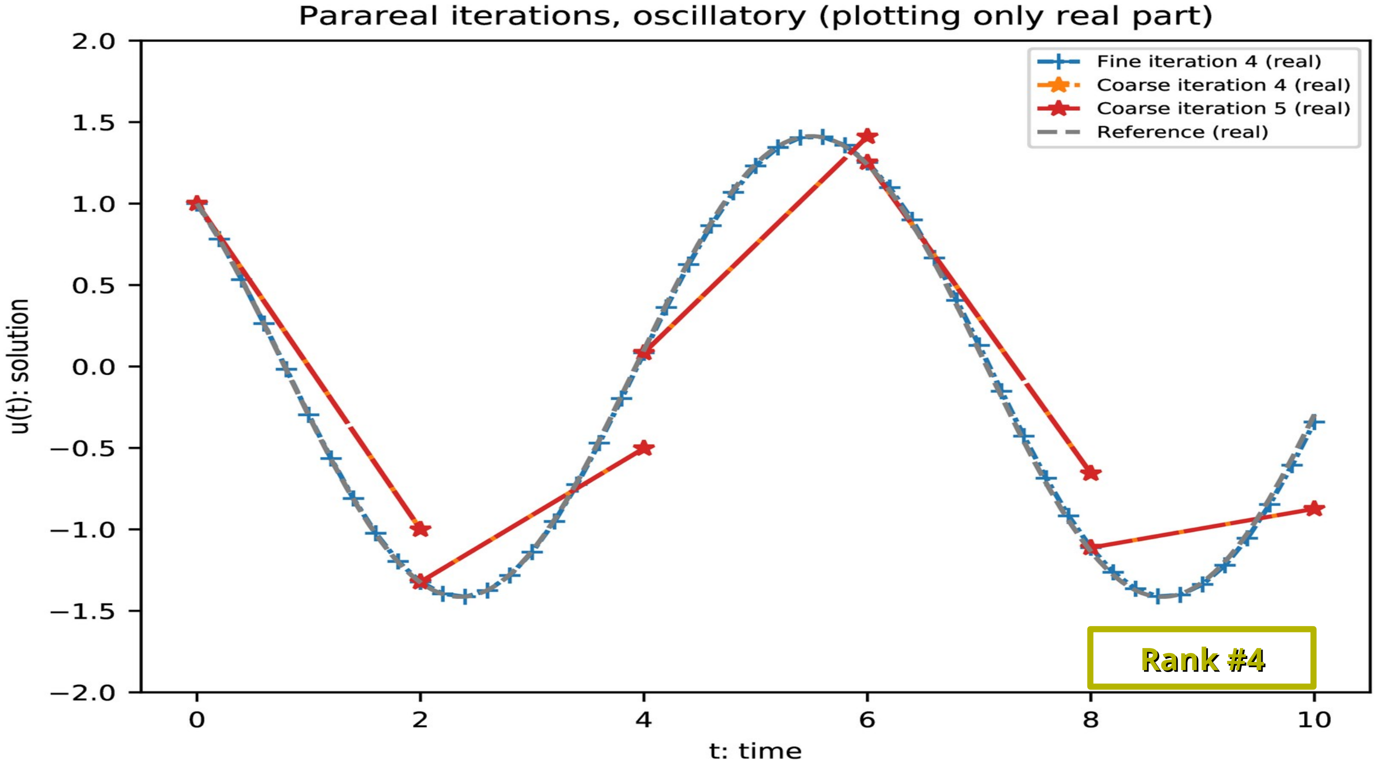
Parareal: 4th iteration

- Relative high accuracy after **4 iterations = 4 fine iterations**



Parareal: 5th iteration

- Solution converges exactly to the one of the non-Parareal fine integrator
- However, wallclock time would be worse than sequential time integration



PFASST: Parallel Full Approximation Scheme in Space and Time

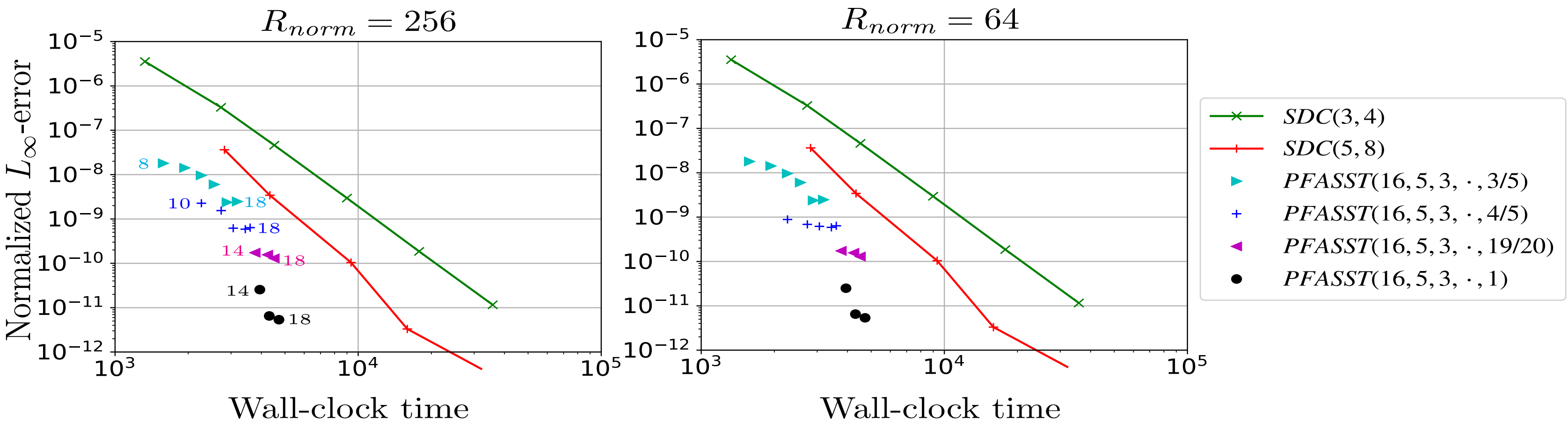
PFASST algorithmic parts

Algorithmic part	**Main motivation**
SDC Iterative within the time step	Higher order
Multi-level in space	Save compute time
Parallel across time steps	Exploit increased parallelism

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

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



Legend: R_{norm} = truncation for computing errors
SDC(# nodes, # sweeps)
PFASST(# processors, # fine nodes, # coarse nodes, # MLSDC sweeps, coarsening ratio)

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

F. Hamon, M. Schreiber, M. Minion (20??) Parallel-in-Time Multi-Level Integration of the Shallow-Water Equations on the Rotating Sphere

Overview

- Introduction

- Spectral deferred Correction
  - Introduction
  - Issues with DC
  - SDC example
- ML-SDC
  - Introduction
  - Results
- PFASST
  - Introduction
  - Results

- **Summary & Outlook**

Summary

- ML-SDC:
  - ML-SDC only partially performing better
  - Small viscosity/filter required for stability reasons
- PFASST:
  - Parallel-in-time
  - PFASST is eventually Pareto optimal for all time step sizes

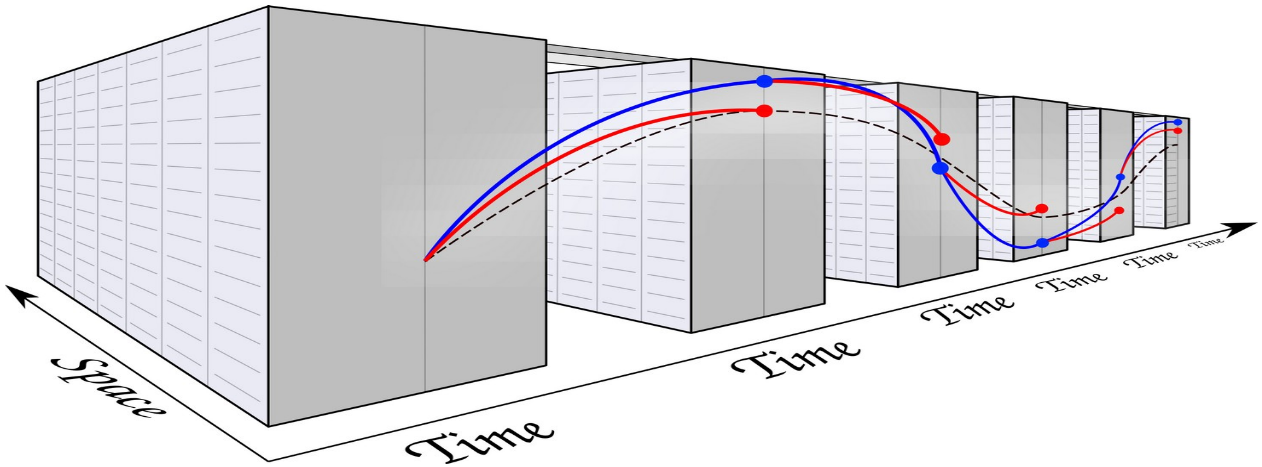
Future work

- Use PFASST with
  - Exponential and
  - Lagrangian methods
- And plenty of side tracks to get this into PFASST (e.g. EXP-SL)

Chair of Computer Architectures and Parallel Systems
Faculty of Informatics
Technical University of Munich

Thank you for your time

May the time be with you (in parallel!)



I’m available for any questions