Toward space-time parallel simulations of phase-field models

Oktober 9, 2018 | Ruth Schöbel & Robert Speck
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH

The consortium

Uwe Glatzel
Applications

Carsten Gräser
Numerics

Oliver Sander
Numerics

Robert Speck
HPC

Jiri Kraus
HPC

Marc-Andre Keip
Applications

BMBF-Project ParaPhase

High-order space-time parallel phase-field simulations

- interface problems can be modeled with phase-field equations
- wide range of applications, e.g. fracture propagation in ceramics, drying soil etc.
- some key phenomena only emerge when the domain is large and simulation time long enough
- significant computing cost ⇒ parallelism in space and time

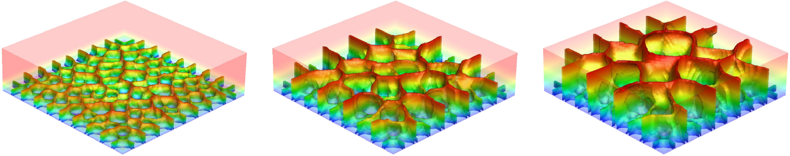


Figure: Modeling of fracture propagation in dry soil

BMBF-Project ParaPhase

High-order space-time parallel phase-field simulations

- interface problems can be modeled with phase-field equations
- wide range of applications, e.g. fracture propagation in ceramics, drying soil etc.
- some key phenomena only emerge when the domain is large and simulation time long enough
- significant computing cost $\Rightarrow$ parallelism in space and time

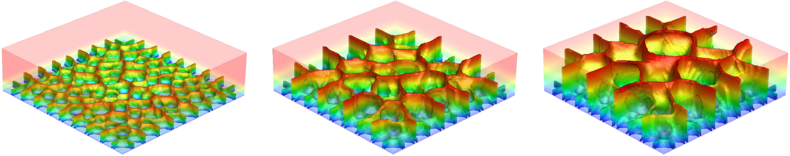
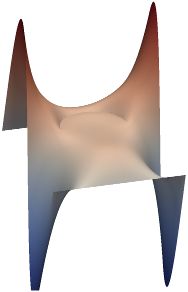


Figure: Modeling of fracture propagation in dry soil

A not so simple toy problem (in space)



Model problem:

$$\begin{aligned} -\Delta u &= f && \text{on } \Omega := [0,1]^2 \\ u &\leq u_+ && \text{on } \Omega \\ u &= g && \text{on } \partial\Omega \end{aligned}$$

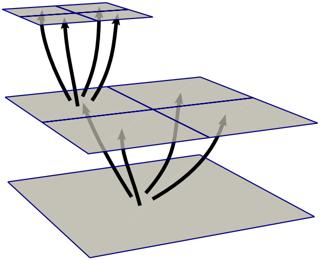
Our approach: multigrid

- very efficient (mostly)
- supported by theory
- parallelizable

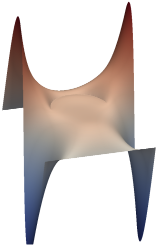
Parallel multigrid with DUNE

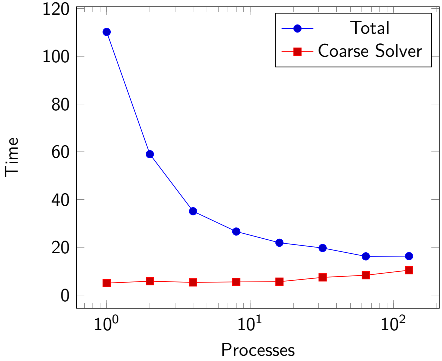
DUNE = Distributed and Unified Numerics Environment

- modular C++ library
- easy implementation of methods like Finite Elements
- slim interfaces, HPC-ready, easy to extend



Strong scaling for obstacle problem

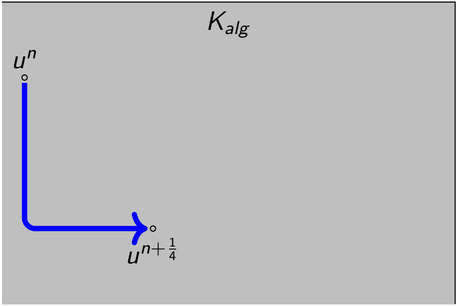




- Parallel version of TNNMG for obstacle problem
- Sequential solver for coarse grid problem
- Number of iterations increases from 20 to 38

Multigrid for obstacle problems

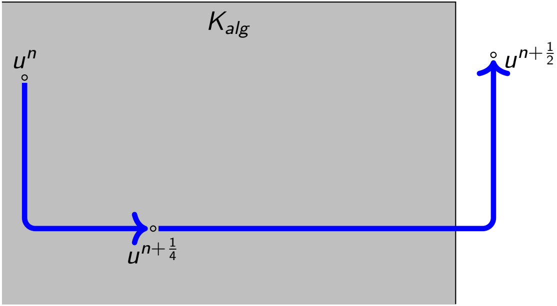
The TNNMG algorithm



- Pre-smoothing: projected Gauss-Seidel
- Linear multigrid step for reduced problem
- Project on K_{alg}
- Line search to ensure monotonicity

Multigrid for obstacle problems

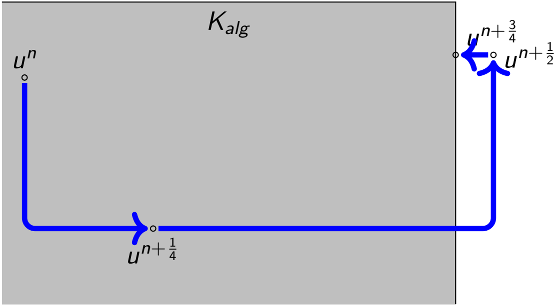
The TNNMG algorithm



- Pre-smoothing: projected Gauss-Seidel
- Linear multigrid step for reduced problem
- Project on K_{alg}
- Line search to ensure monotonicity

Multigrid for obstacle problems

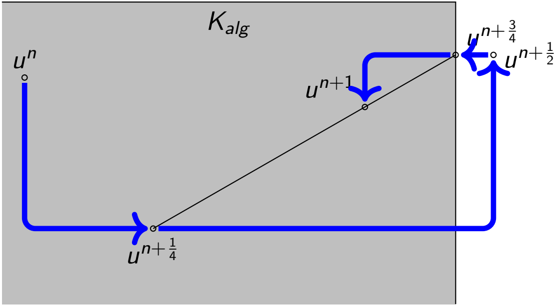
The TNNMG algorithm



- Pre-smoothing: projected Gauss-Seidel
- Linear multigrid step for reduced problem
- Project on K_{alg}
- Line search to ensure monotonicity

Multigrid for obstacle problems

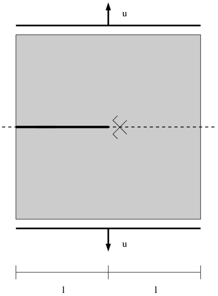
The TNNMG algorithm

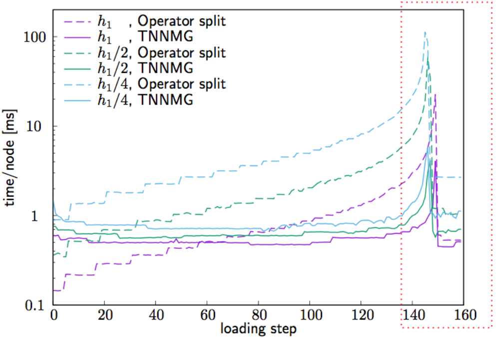


- Pre-smoothing: projected Gauss-Seidel
- Linear multigrid step for reduced problem
- Project on K_{alg}
- Line search to ensure monotonicity

Application: Notched domain under tension

A Boundary Value Problem



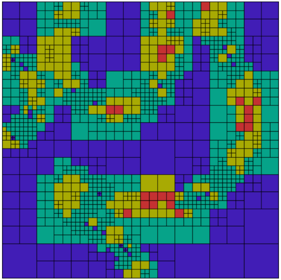


- Single-edge-notched domain under tensile loading
- Deformation by a linear increasing non-homogeneous Dirichlet boundary condition at the top
- TNNMG is much faster!

Further improvements: High-Order DG for Phase-Fields

Phase-Field and its Discretization

- Problems typically involve smooth and non-smooth parts
- adaptively refine grid to better resolve non-smooth areas
- for smooth parts, employ ansatz functions of higher order

- for smooth parts, employ ansatz functions of higher order
- Discontinuous Galerkin for more flexibility and data-locality
- hybrid use of classical matrix-based and matrix-free methods

Adding time

Complexity $+= 1$

What if our problems are time-dependent?

- example: Allen-Cahn or Cahn-Hilliard equations: diffuse interface model for phase transition and separation phenomena
- TNNMG allows semi- or fully implicit time-stepping with provable convergence
- parallel TNNMG = parallelization in space for time-dependent phase-field problems

Adding time

Complexity $+=1$

What if our problems are time-dependent?

- example: Allen-Cahn or Cahn-Hilliard equations: diffuse interface model for phase transition and separation phenomena
- TNNMG allows semi- or fully implicit time-stepping with provable convergence
- parallel TNNMG = parallelization in space for time-dependent phase-field problems

Are we done, then?

Limits of purely spatial parallelization

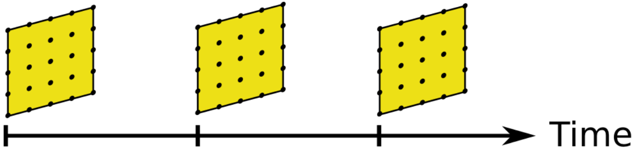


Figure: Time-stepping to solve time-dependent partial differential equations.

- Spatial parallelization reduces runtime **per time-step**
- Strong scaling saturates eventually because of communication
- Costs for **more time-steps** are not mitigated

Limits of purely spatial parallelization

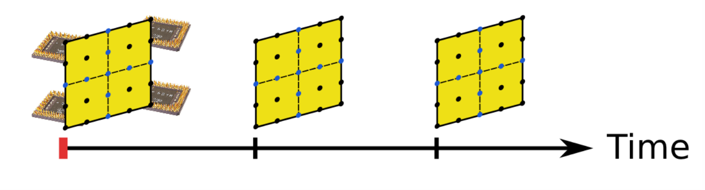


Figure: Time-stepping to solve time-dependent partial differential equations.

- Spatial parallelization reduces runtime **per time-step**
- Strong scaling saturates eventually because of communication
- Costs for **more time-steps** are not mitigated

Limits of purely spatial parallelization

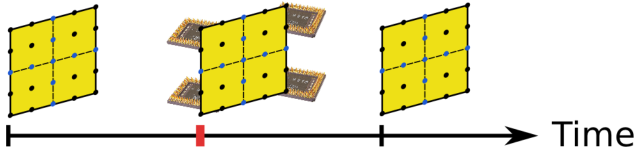


Figure: Time-stepping to solve time-dependent partial differential equations.

- Spatial parallelization reduces runtime **per time-step**
- Strong scaling saturates eventually because of communication
- Costs for **more time-steps** are not mitigated

Limits of purely spatial parallelization

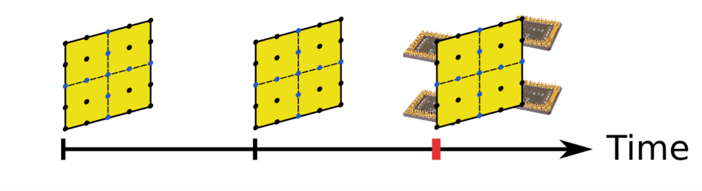


Figure: Time-stepping to solve time-dependent partial differential equations.

- Spatial parallelization reduces runtime **per time-step**
- Strong scaling saturates eventually because of communication
- Costs for **more time-steps** are not mitigated

Limits of purely spatial parallelization

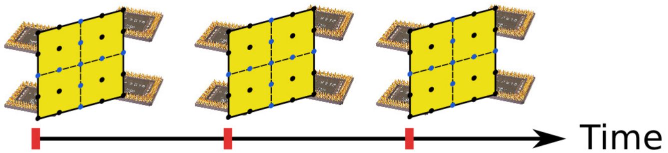


Figure: Time-stepping to solve time-dependent partial differential equations.

- Spatial parallelization reduces runtime **per time-step**
- Strong scaling saturates eventually because of communication
- Costs for **more time-steps** are not mitigated

→ Can we compute multiple time-steps simultaneously?

A quick algebraic introduction to PFASST

Basic building block: spectral deferred corrections (SDC)

Consider the Picard form of an initial value problem on $[T_n, T_{n+1}]$

$$u(t) = u_0 + \int_{T_n}^{t} f(u(s))ds,$$

discretized using spectral quadrature rules with nodes t_m:

$$u_m = u_0 + \Delta t QF(u) \approx u_0 + \int_{T_l}^{t_m} f(u(s))ds,$$

then SDC methods can be seen as (approximative) Gauss-Seidel iteration to solve this collocation problem for all u_m.

A quick algebraic introduction to PFASST

Basic building block: spectral deferred corrections (SDC)

Consider the Picard form of an initial value problem on $[T_n, T_{n+1}]$

$$u(t) = u_0 + \int_{T_n}^{t} f(u(s))ds,$$

discretized using spectral quadrature rules with nodes t_m:

$$(I - \Delta t Q F)(\vec{u}) = \vec{u}_0$$

then SDC methods can be seen as (approximative) Gauss-Seidel iteration to solve this collocation problem for all u_m.

A quick algebraic introduction to PFASST

Basic building block: spectral deferred corrections (SDC)

Consider the Picard form of an initial value problem on $[T_n, T_{n+1}]$

$$u(t) = u_0 + \int_{T_n}^{t} f(u(s))ds,$$

discretized using spectral quadrature rules with nodes t_m:

$$(I - \Delta t Q F)(\vec{u}) = \vec{u}_0$$

then SDC methods can be seen as (approximative) Gauss-Seidel iteration to solve this collocation problem for all u_m.

$\Rightarrow$ Use this for block smoothing in space-time multigrid = PFASST

A quick algebraic introduction to PFASST

Parallel Full Approximation Scheme in Space and Time

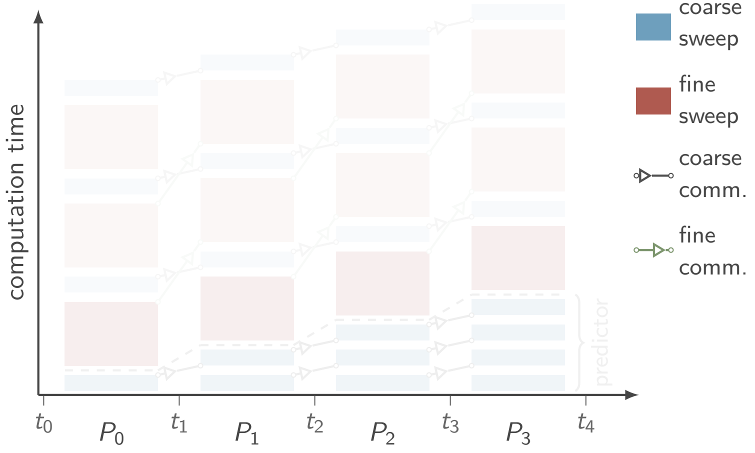
We now glue L time-steps together, using N to transfer information from step l to step $l+1$. We get:

$$\begin{pmatrix} I-\Delta tQF & & & \\ -N & I-\Delta tQF & & \\ & \ddots & \ddots & \\ & & -N & I-\Delta tQF \end{pmatrix} \begin{pmatrix} \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_L \end{pmatrix} = \begin{pmatrix} \vec{u}_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

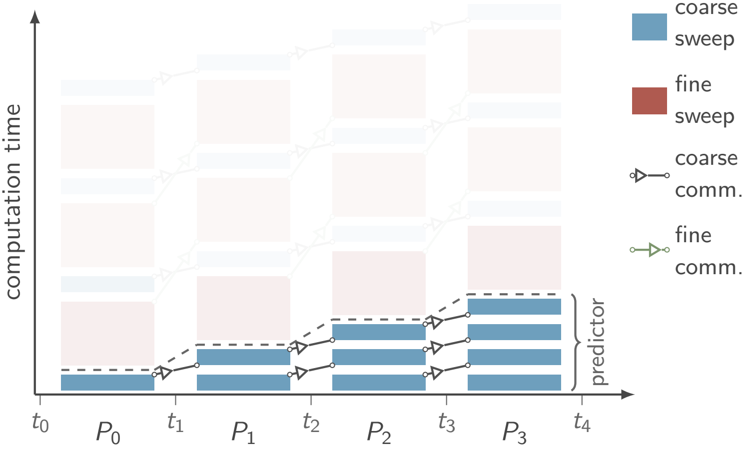
PFASST:

- use (linear/FAS) multigrid to solve this system iteratively
- fine-level smoother: parallel block Jacobi with SDC in the blocks
- coarse-level: serial block Gauß-Seidel with SDC in the blocks

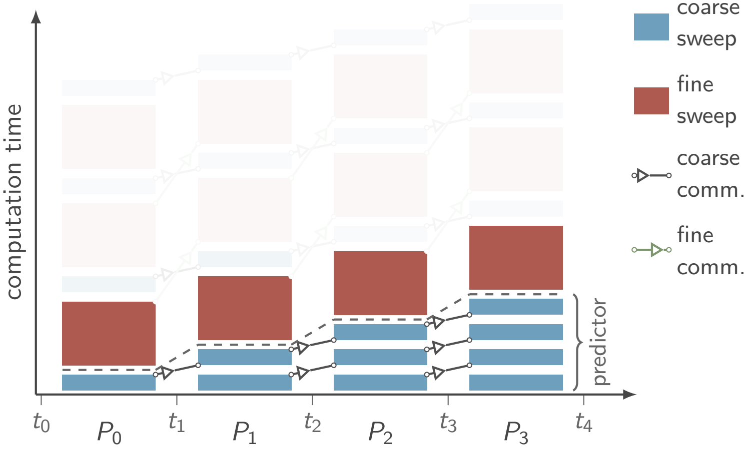
A quick visual introduction to PFASST



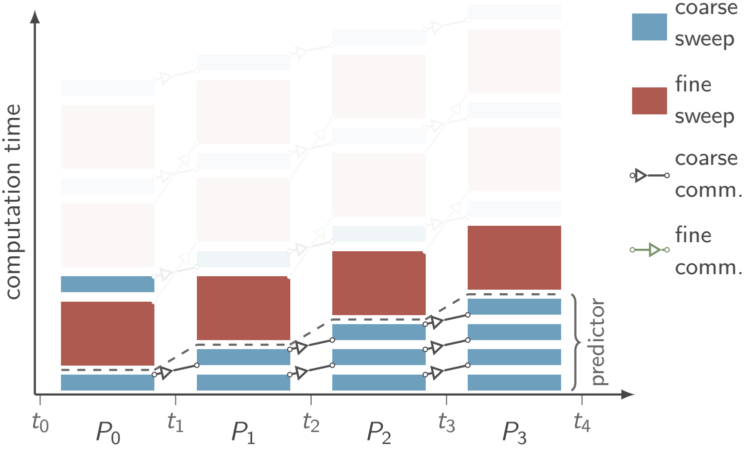
A quick visual introduction to PFASST



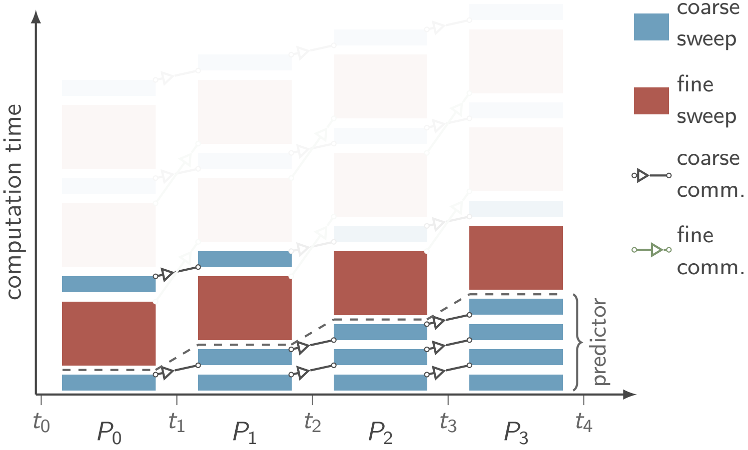
A quick visual introduction to PFASST



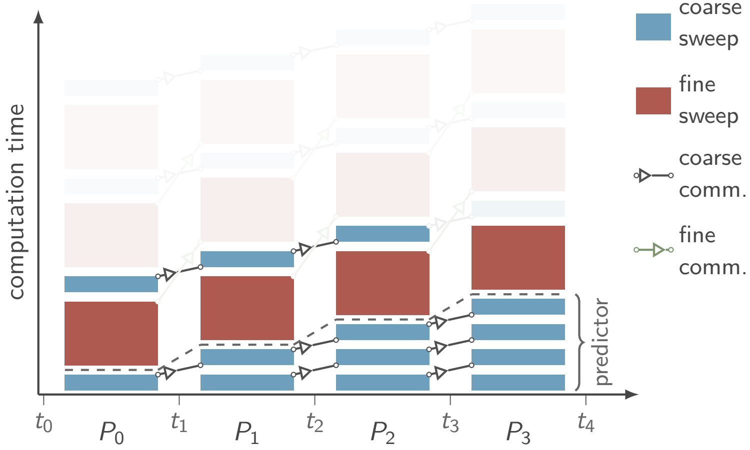
A quick visual introduction to PFASST



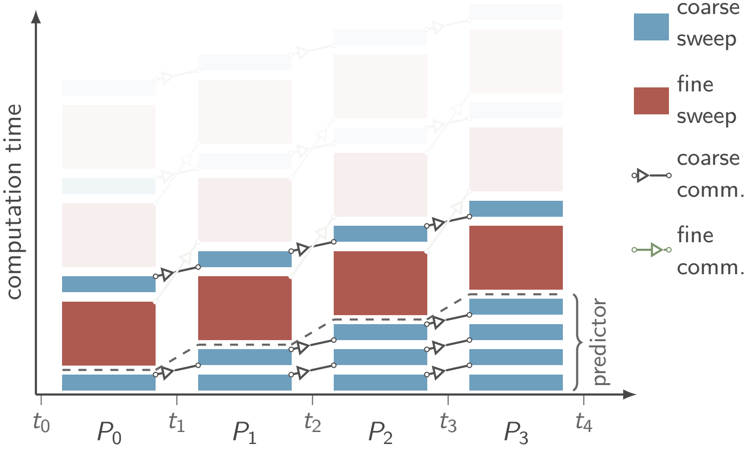
A quick visual introduction to PFASST



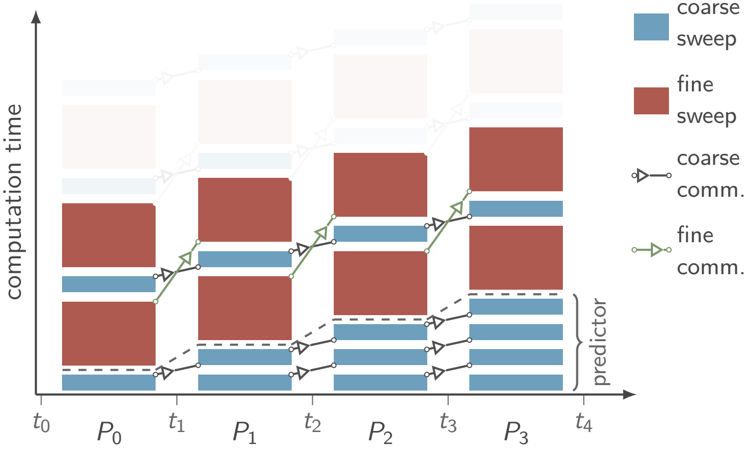
A quick visual introduction to PFASST



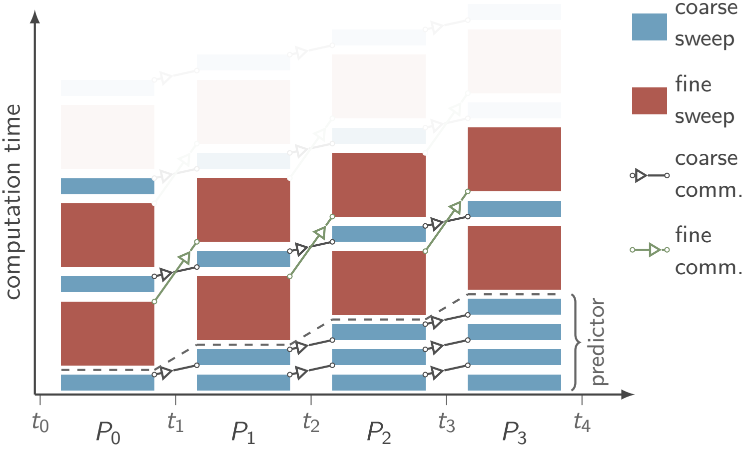
A quick visual introduction to PFASST



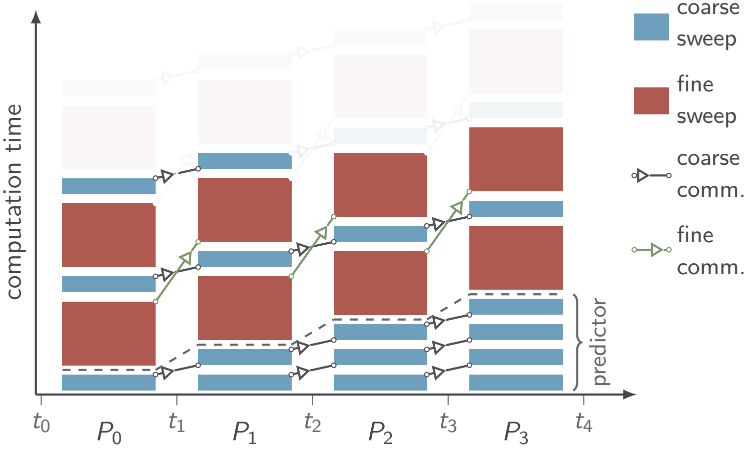
A quick visual introduction to PFASST



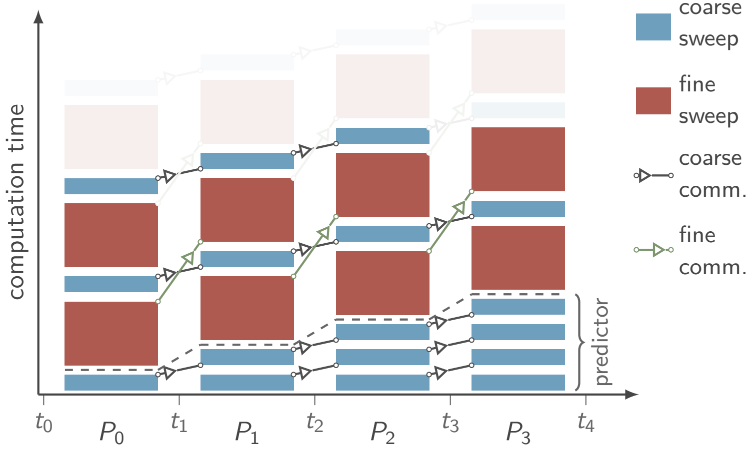
A quick visual introduction to PFASST



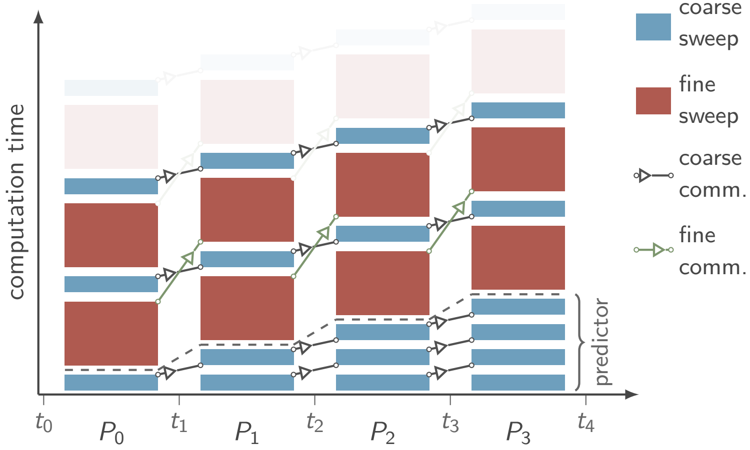
A quick visual introduction to PFASST



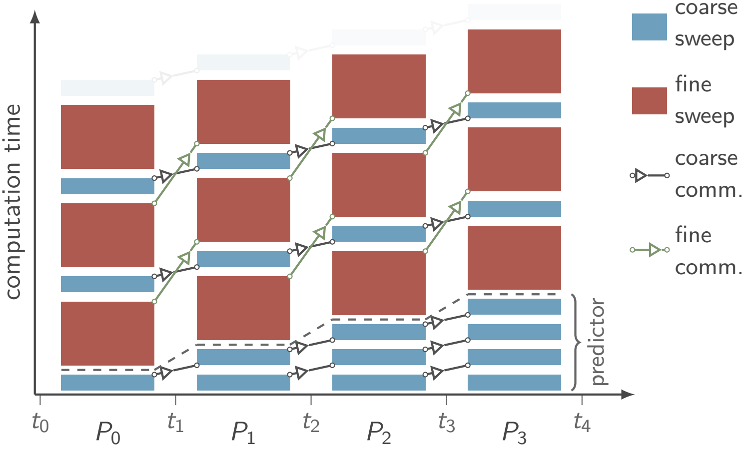
A quick visual introduction to PFASST



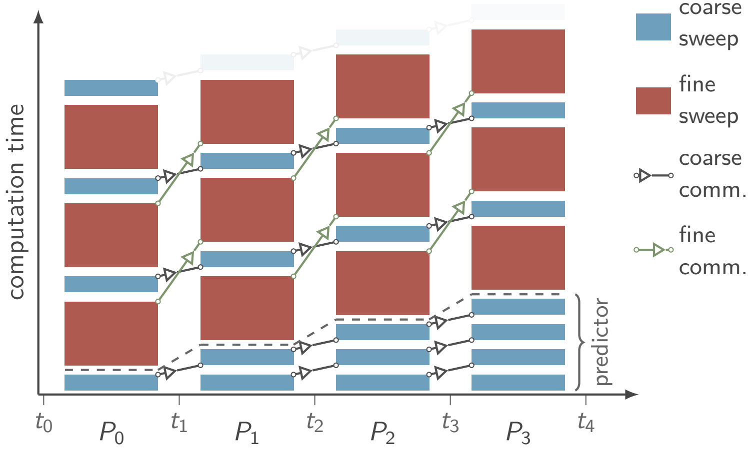
A quick visual introduction to PFASST



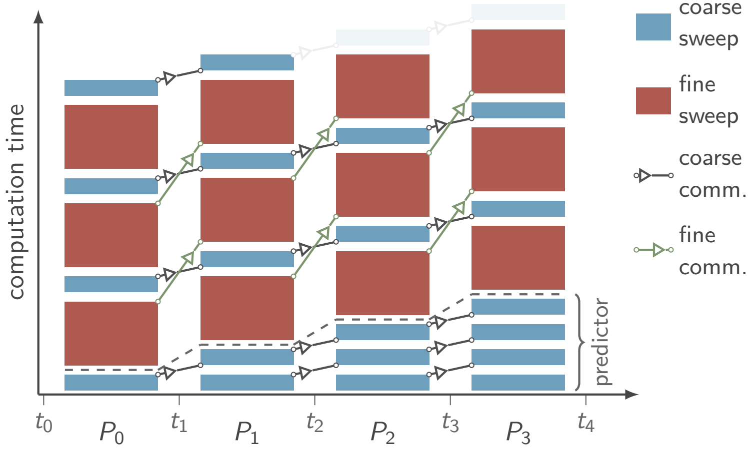
A quick visual introduction to PFASST



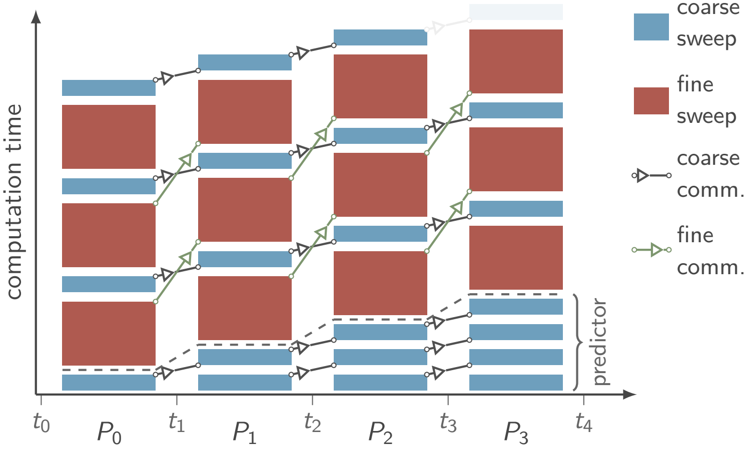
A quick visual introduction to PFASST



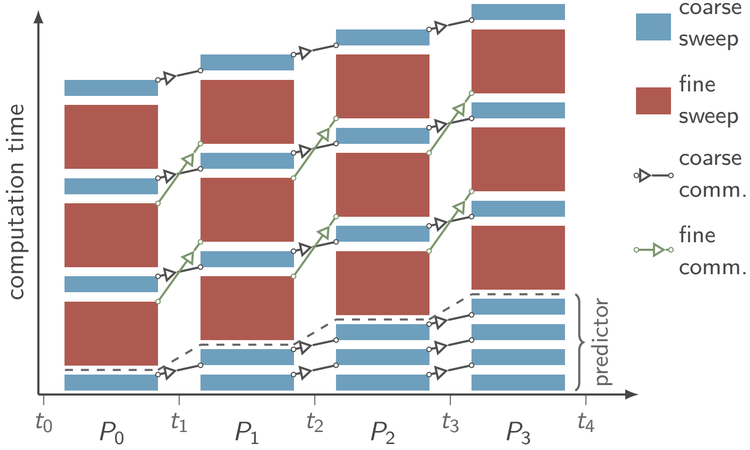
A quick visual introduction to PFASST



A quick visual introduction to PFASST

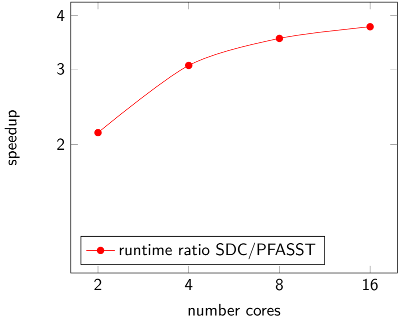


A quick visual introduction to PFASST



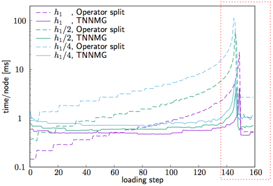
Gray-Scott simulation with DUNE and PFASST

$$u_t = D_u \Delta u - uv^2 + F(1-u)$$
$$v_t = D_v \Delta v + uv^2 - (F+K)v$$



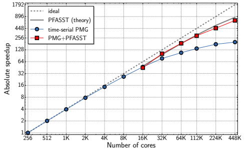
512^2 elements, $dt = 2$, tol $= 10^{-14}$, $D_u = 10^{-4}$, $D_v = 10^{-5}$, $F = 0.0367$, $K = 0.0649$

Three takeaways



Parallel TNNMG method provides reliable and efficient solver for phase-field problems

Parallel-in-Time integration (PinT) can help to extend prevailing scaling limits



Space-time parallelization within **DUNE** leaves us with a ton of things to play with

The PinT Community

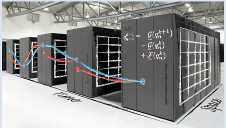
To learn more about PinT check out the website

`www.parallelintime.org`

and/or join one of the PinT Workshops, e.g.

8th Workshop on Parallel-in-Time Integration

- May 20-24, 2019
- ZIF Bielefeld, Germany
- organized by Daniel Ruprecht, Sebastian Schöps and Robert Speck



Also, there is a mailing list, join by writing to

`parallelintime+subscribe@googlegroups.com`

Toward space-time parallel simulations of phase-field models

Oktober 9, 2018 | Ruth Schöbel & Robert Speck
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH