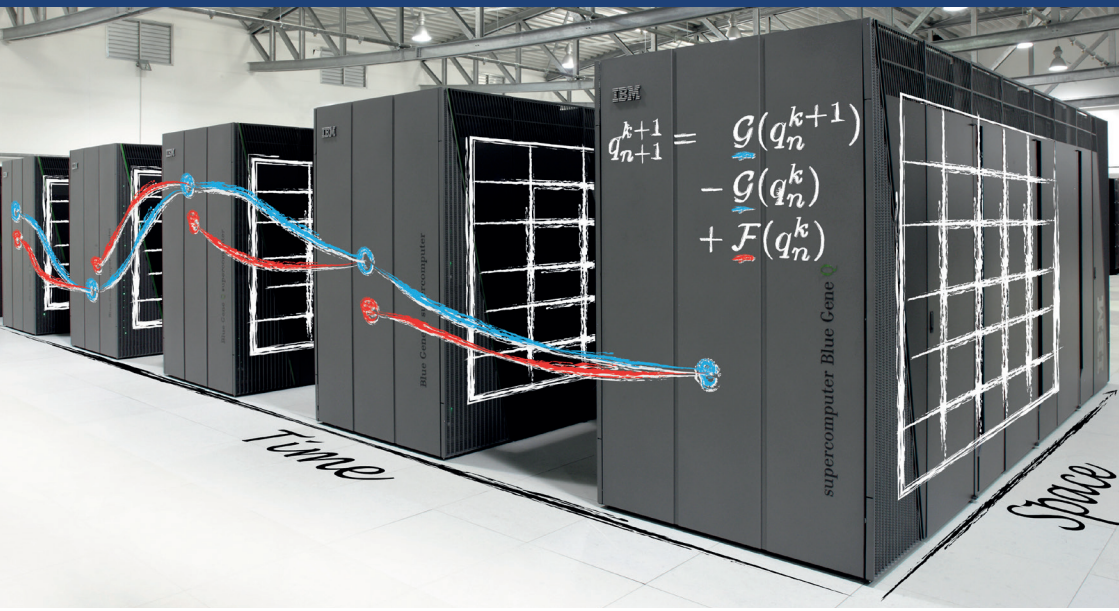
3rd Workshop on Parallel-in-Time Integration

Parallel multi-level methods in space and time

May 26 - 28, 2014

Jülich Supercomputing Centre, Forschungszentrum Jülich



BOOK OF ABSTRACTS

Dear participants,

It is our distinct pleasure to welcome you at Jülich Supercomputing Centre to the **3rd Workshop on Parallel-in-Time Integration.** With 43 registered participants coming from 12 different countries this event gives an impressive illustration of the enormous interest that the field of time-parallel methods attracts today. This meeting is the third in a series that started in 2011 at the Institute of Computational Science in Lugano and was continued by a workshop at the School of Mathematics at the University of Manchester in 2013. We are confident that this series will go on and continue to provide people from all disciplines interested in time parallelization with a forum to exchange ideas, have exciting discussions and pleasant conversations.

Although ideas for algorithms providing concurrency in the time dimension have been around for 50 years now, in recent years the field has attracted more and more attention and interest as scientists from many disciplines are meeting the challenges from the extreme degree of concurrency required by today's and future supercomputers. It is our sincere hope that this workshop will support researchers not only in the development, improvement and analysis of novel and innovative numerical methods but also in their efficient implementation and ultimately their application. Hopefully, this series of workshops can in this way contribute to continue advancing timeparallel methods as an efficient and powerful tool in modern parallel computing.

We wish you a stimulating and interesting workshop and a pleasant time at JSC. We also would like to thankfully acknowledge support from our sponsors, the German Research Foundation, particularly the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA), and the Helmholtz Association via Forschungszentrum Jülich GmbH.

Your organizing committee,

Matthias Bolten

University of Wuppertal
Mathematics and Natural Sciences
Department of Mathematics
D-42097 Wuppertal

Daniel Ruprecht

Università della Svizzera italiana
Institute of Computational Science
Via Giuseppe Buffi 13
CH-6900 Lugano

Rolf Krause

Università della Svizzera italiana
Institute of Computational Science
Via Giuseppe Buffi 13
CH-6900 Lugano

Robert Speck

Forschungszentrum Jülich GmbH
Institute for Advanced Simulation
Jülich Supercomputing Centre
D-52425 Jülich

Welcome to Jülich Supercomputing Centre

The Jülich Supercomputing Centre (JSC) is all about simulations using supercomputers of the highest performance class. Not only do the experts at JSC operate leading computer systems around the clock, they also support scientific users in terms of methods and content. At the same time, they perform intensive research on solutions for the grand challenges facing modern society and are involved in the development of state-of-the-art computer architectures.

Scientific challenges are as diverse as the requirements on the computers used to address them. While simulations in brain research, such as the Human Brain Project, or from the area of climate research primarily require high main memory capacity, quantum calculations in nanoelectronics need extremely high computing power. Data analysis problems are best solved with cluster computers, some of which are accelerated with graphics chips. One example is the analysis at JSC of data from the Alpha Magnetic Spectrometer (AMS) experiment on the International Space Station, which is designed to detect antimatter in space.

Tailor-made solutions must be developed for use on all these systems. Only then is it possible to guarantee the most efficient exploitation of these expensive resources. This is made possible on site by the expertise of the simulation laboratories specializing in individual subject areas, and interdisciplinary teams focusing on methodology, algorithmics, optimization, analysis, and visualization. Outside of Forschungszentrum Jülich, JSC represents user needs in national and international committees and organizations, such as the Gauss Centre for Supercomputing (GCS) and the Partnership for Advanced Computing in Europe (PRACE). It is precisely this comprehensive user support and the simultaneous focus on thematic priorities that have earned the Jülich Supercomputing Centre its international reputation.

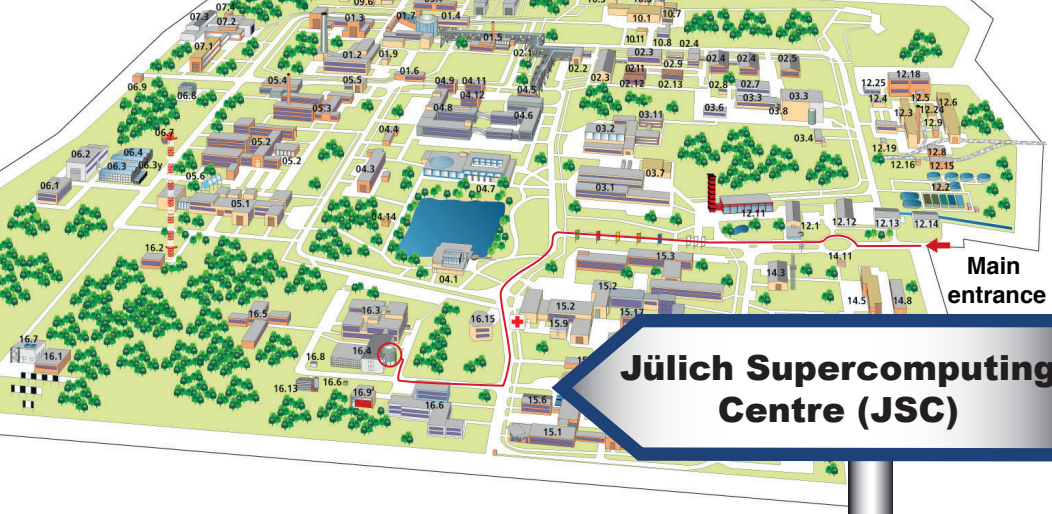
About SPPEXA

The Priority Programme "Software for Exascale Computing" (SPPEXA) addressesfundamental research on the various aspects of HPC software, which is particularly urgent against the background that we are currently entering the era of ubiquitous massive parallelism. This massive parallelism only, subsumed to the notion of manycore processors and their assembly to systems beyond 10^7 processing units, will smooth the way for extreme computing up to exascale, i.e. computations with 10^{18} floating point operations per second and beyond, and the insight resulting from those simulations. Mastering the various challenges related to this paradigm shift from sequential or just moderately parallel to massively parallel processing will be the key to any future capability computing application at exascale, but it will also be crucial for learning how to effectively and efficiently deal with commodity systems of the day after tomorrow for smallerscale or capacity computing tasks and it is the overall scientific objective of SPPEXA. To this end, SPPEXA reconnects several relevant subfields of computer science with the needs of Computational Science and Engineering (CSE) and HighPerformance Computing (HPC). SPPEXA provides the framework for a much closer cooperation and a much more codesigndriven approach instead of a merely servicedriven collaboration of groups focusing on fundamental HPC methodology (computer science or mathematics) on the one hand with those working on science applications and providing the large codes (science and engineering) on the other hand.

Topically, SPPEXA will drive research towards extremescale computing in six areas or research directions:

- computational algorithms,
- system software,
- application software,
- data management and exploration,
- programming,
- software tools.

Hardware peak performance is ever increasing, exascale systems are currently predicted for around 2018, and insight is growing worldwide that a "racks without brains" strategy will not allow the science communities to exploit the huge potential of the computational approach in a massively parallel world. Against this background, SPPEXA provides an ideal framework for bundling research activities nationwide and enabling the participating groups to significantly advance the state of the art in HPC software technology at an international scale.



Transfer

Monday, May 26, 2014

13:00* Hotel Kaiserhof → JSC
17:15 JSC → Hotel Kaiserhof

Tuesday, May 27, 2014

08:20 Hotel Kaiserhof → JSC
17:45 JSC → Hotel Kaiserhof
18:30 Hotel Kaiserhof → Burg Obbendorf
22:00 Burg Obbendorf → Hotel Kaiserhof

Wednesday, May 28, 2014

08:20 Hotel Kaiserhof → JSC
17:00* JSC → Hotel Kaiserhof

**registration only*

Participants

Teresa Beck	EMCL, Uni Heidelberg
Marek Behr	RWTH Aachen, CATS
Matthias Bolten	Bergische Universität Wuppertal
Matthew Emmett	Lawrence Berkeley National Laboratory
Stefan Findeisen	Karlsruhe Institute of Technology (KIT)
Martin J. Gander	University of Geneva
Johannes Grotendorst	Jülich Supercomputing Centre
Stefanie Günther	RWTH Aachen University
Yousaf Habib	National University of Science and Technology, Pakistan
Ronald Haynes	Memorial University of Newfoundland
Konstantinos Ioakimidis	Institute of Fluid Mechanics and Hydraulic Machinery, University of Stuttgart
Timon Kirschke	Jülich Supercomputing Centre
Torbjörn Klatt	Jülich Supercomputing Centre
Harald Klimach	University of Siegen, Simulation Techniques and Scientific Computing
Michael Klöppel	Institute of Scientific Computing , TU Dresden
Uwe Köcher	Helmut Schmidt University, Univ. of the German Federal Armed Forces Hamburg
Gijs Kooy	University of Twente
Rolf Krause	USI Lugano, Switzerland
Felix Kwok	University of Geneva
Andreas Kreienbuehl	USI Lugano, Switzerland
Vadim Lisitsa	Institute of Petroleum Geology and Geophysics SB RAS
Thomas Loderer	Robert Bosch GmbH
Scott MacLachlan	Tufts University

Dieter Moser	Jülich Supercomputing Centre
Benson Muite	Institute of Computer Science, University of Tartu
Nabil Nassif	American University of Beirut
Kees Oosterlee	CWI center for mathematics and computer science
Hannah Rittich	Bergische Universität Wuppertal
Christoph Riesinger	Technische Universität München, Lehrstuhl für Wissenschaftliches Rechnen
Albert Ruprecht	Institute of Fluid Mechanics and Hydraulic Machinery, University of Stuttgart
Daniel Ruprecht	USI Lugano, Switzerland
Giovanni Samaey	KU Leuven
Achim Schädle	Heinrich Heine Universität
Jacob Schroder	Lawrence Livermore National Laboratory
Thomas Slawig	Christian-Albrechts-Universität Kiel
Robert Speck	Jülich Supercomputing Centre
Sven Strohmer	Jülich Supercomputing Centre
Godehard Sutmann	Jülich Supercomputing Centre
Toshiya Takami	Kyushu University
Rasmus Tamstorf	Walt Disney Animation Studios
Xavier Vasseur	CERFACS
Marina Weingartz	Jülich Supercomputing Centre
Jörg Wensch	TU Dresden, Institut für wissenschaftliches Rechnen

Monday, May 26, 2014	
12:00 - 13:45	Lunch + Registration
13:45 - 14:00	Opening + Welcome
14:00 - 15:00	**Martin Gander,** University of Geneva (Switzerland) *The History of Time Parallel Methods*
15:00 - 15:30	Coffee break
15:30 - 16:00	**Giovanni Samaey,** KU Leuven (Belgium) *A Micro-Macro Parareal Algorithm: Application to Singularly Perturbed Ordinary Differential Equations*
16:00 - 16:30	**Toshiya Takami,** Kyushu University (Japan) *Identity Parareal Method and Its Performance*

Tuesday, May 27, 2014	
09:00 - 10:00	**Scott MacLachlan,** Tufts University (USA) *Multigrid Reduction Techniques For Parallel-In-Time Integration*
10:00 - 10:30	Coffee break
10:30 - 11:00	**Jacob Schroder**, Lawrence Livermore National Laboratory (USA) *Multigrid Reduction in Time: A Flexible and Non-Intrusive Method*
11:00 - 11:30	**Felix Kwok,** University of Geneva (Switzerland) *Schwarz Methods for the Time-Parallel Solution of Parabolic Control Problems*
11:30 - 12:00	**Nabil Nassif,** American University of Beirut (Lebanon) *Comparing the Parareal and the Adaptive Parallel Time Integration (APTI) methods for a Satellite Problem*
12:00 - 14:00	Lunch + Photo
14:00 - 15:00	**Open Questions Session**
15:00 - 15:30	Coffee break
15:30 - 16:00	**Vadim Lisitsa,** IPGG SB RAS (Russia) *Combining Finite Differences and Discontinuous Galerkin method for simulation of seismic waves propagation*
16:00 - 16:30	**Marek Behr,** RWTH Aachen (Germany) *Space-Time Finite Elements and Non-Uniform Temporal Refinement*
16:30 - 17:45	Individual Meetings
19:00 - 22:00	Workshop Dinner, Burg Obbendorf, Hambach *www.burgobbendorf.de*

	Wednesday, May 28, 2014
09:00 - 10:00	**Kees Oosterlee,** CWI Amsterdam, (The Netherlands) *Handling with the time-wise dimension by Fourier techniques, Backward Stochastic Differential Equations and Graphics Processing Unit*
10:00 - 10:30	Coffee break
10:30 - 11:00	**Stefan Findeisen,** Karlsruhe Institute of Technology (Germany) *First step towards Parallel and Adaptive space-time Computation of Maxwell's Equations*
11:00 - 11:30	**Uwe Köcher,** Helmut Schmidt University (Germany) *Variational Space-Time Methods for the Elastic Wave Equation*
11:30 - 12:00	**Konstantinos Ioakimidis,** University of Stuttgart (Germany) *Parallel in Time simulation of the Navier-Stokes equations using the Finite Finite Element Method*
12:00 - 14:00	Lunch + Tours
14:00 - 15:00	**Matt Emmett,** Lawrence Berkeley National Laboratory (USA) *Timing and performance of PFASST*
15:00 - 15:30	Coffee break
15:30 - 16:00	**Dieter Moser,** Forschungszentrum Jülich (Germany) *PyPinT-Towards a framework for rapid prototyping of iterative parallelintime algorithms*
16:00 - 16:30	**Michael Klöppel,** TU Dresden (Germany) *Using time-parallel methods for the simulation of a machine tool*
16:30 - 16:45	Closing + Farewell

ABSTRACTS

50 Years of Time Parallel Time Integration

Martin J. Gander *Section of Mathematics, University of Geneva*

Time domain decomposition methods have received renewed interest over the last decade because of the advent of massively parallel computers. When solving time dependent partial differential equations, the time direction is usually not used for parallelization. When parallelization in space saturates however, the time direction offers itself as a further direction for parallelization. The time direction is however special, and for evolution problems there is a causality principle: the solution later in time is affected (it is even determined) by the solution earlier in time, but not the other way round. Algorithms trying to use the time direction for parallelization must therefore be special, and take this very different property of the time dimension into account. The development of time parallel time integration methods spans now half a century, and various methods have been invented and reinvented over this period. Here is a brief account of major contributions:

Nievergelt 1964: precisely 50 years ago, Nievergelt started the era of time parallel methods in the visionary paper [1]. To solve an ordinary differential equation (ODE) in parallel, a process which "by all standard methods, is entirely serial", he first partitions the time interval into subintervals. He then computes a rough initial guess at the beginning of each subinterval, for example using a coarse integrator. Starting with many initial guesses in the neighborhood of the rough initial guess, he then computes in parallel many accurate trajectories across each time interval. Finally, an accurate overall trajectory is obtained by interpolation sequentially time interval after time interval. For linear ODEs, this leads to the exact solution, and only two solutions over each time interval would suffice. Otherwise, and approximate solution is obtained.

Miranker and Liniger 1967: a very different approach was proposed by Miranker and Liniger [2]. They first note that classical predictor corrector methods are completely sequential: one first has to compute the predicted value, before one can compute the corrected value. To "widen this narrow computation front", they propose to compute the prediction step using the result of the previous prediction step, instead of the corrected value, and thus the prediction and correction steps can now be performed in parallel using two processors. They then generalize this idea to methods which can use $2n$ processors. In contrast to Nievergelt's idea, this leads to small scale parallelism.

Lelarasmee, Ruehli and Sangiovanni-Vincentelli 1982: in the circuit community, a further very different time parallel method was developed based on a decomposition of a large scale circuit into sub-circuits, and an iteration. Each subcircuit is solved independently over a so called time window, using along cables which were cut to partition the circuit signals from the previous iteration. Since such signals are called waveforms, the method was baptized waveform relaxation method. On bounded time windows, these methods converge typically superlinearly.

Hackbusch 1984: the parabolic multigrid method developed in [3] works as follows: the parabolic problem is first discretized in space time using an implicit scheme. Then on n time levels, one applies a few smoothing iterations to the systems that have to be solved, sequentially one time level after the other. Then, as in multigrid, one computes the residual, restricts it to a coarse grid, and does again the same thing recursively. If coarsening is only done in space, this leads to multigrid performance. If one also coarsens in time, the method in general can not deliver multigrid performance. Parabolic multigrid is not naturally parallel, but like other multigrid methods, the smoother can be naturally parallelized, and the restrictions and extensions over many time steps are also naturally parallel.

Axelsson and Verwer 1985: boundary value methods are also a technique to solve evolution problems in parallel [4]. The idea of boundary value methods is best described as a discretization of an initial value problem on a given time grid using a linear multistep method, and then, instead of a starting procedure for the first few unknown values, one imposes the discretized differential equation as an 'ending procedure' at the other end, thus obtaining a discretized system that looks like a boundary value problem and can be solved simultaneously by iteration. Stability properties of boundary value methods are however very different from time stepping methods, and care must be taken to obtain convergent schemes.

Lubich and Ostermann 1987: a new idea to use waveform relaxation directly as the smoothing iteration was proposed in [5]. The algorithm is formulated at the continuous level in time, and the easiest way to understand it is to imagine to take a Laplace transform in time to obtain a steady problem depending on the Laplace parameter, and then formally applying a standard multigrid method to this steady problem. Once the multigrid algorithm is applied, one simply backtransforms the algorithm to get the so called multigrid waveform relaxation algorithm. Since the algorithm is formulated and analyzed at the continuous level in time, it does not consider time coarsening in its basic variant, and thus delivers multigrid performance naturally.

Gear 1988: a first review paper on parallel methods for evolution problems is [6]. Gear divides these methods into two classes: parallelism across space, and parallelism across time. For parallelism across space, all methods developed for steady problems can be used. For parallelism across time, Gear distinguishes small scale parallelism along the lines of Miranker and Liniger, and also small scale block methods, which compute a few consecutive values in time approximately by iteration, following Milne's starting procedure for multistep methods in [7]. For parallelism across time, after reviewing Nievergelt's method, he proposes for linear problems to use quadrature and matrix exponential evaluations, with an interesting speculation for a new method at the end.

Bellen and Zennaro 1989: in [8], the original idea of Nievergelt is picked up again, but now to formally develop an iterative method to connect trajectories, and with the concrete idea of multiple shooting. The method is formulated directly for a discrete problem, i.e. a recurrence relation. To obtain parallelism across time, the recurrence relation is directly formulated over many time steps as a

fixed point problem, and the Steffensen method, which is a variant of Newton's method with an approximate Jacobian but still quadratic convergence, is applied to solve the fixed point equation. They discover already the property that each iteration leads to one more converged value in time. They also state quadratic convergence locally, but the proof is given in an internal report.

Womble 1990: a different direction investigated in [9] is to use the fact that implicit time discretizations require the solution of non-linear systems at each time step. At each time step, the non-linear system to be solved depends on the converged result of the previous time step. Womble now suggests to start the iteration already without having the converged result of the previous time step, and to simply use the most accurate approximation currently available. This way, the non-linear solver can iterate simultaneously over several time steps. A convergence result for this iteration stating that this approach has the same asymptotic convergence factor as the underlying fixed point iteration is however misleading, since this asymptotic regime is only reached after many iterations, and the initial convergence factor is not good, and gets worse, the more time steps iterate simultaneously.

Chartier and Philippe 1993: in [10], the authors propose a time parallel method very much related to the method of Bellen and Zennaro, but at the continuous level. They introduce Newton's method in order to solve the continuity equations arising from a multiple shooting formulation, and prove locally quadratic convergence. They also show already that this method is not necessarily effective on general problems, and restrict their later analysis to dissipative right hand sides, for which they prove a global convergence result. Finally, also discrete versions of the algorithm are studied.

Horton and Vandewalle 1995: the difficulty of parabolic multigrid with time coarsening is due to the strong anisotropy introduced in the fully discrete problem because of the time direction [11]. This can be overcome using a special coarsening strategy, and adapted prolongation operators in time, which do not transfer information backward in time. Using F-cycles (also called full multigrid schemes), a fully mesh independent space-time parallel multigrid solver is obtained for the heat equation.

Saha, Stadel and Tremaine 1996: the work of Saha, Stadel and Tremaine is on the integration of the solar system over very long time [12]. They formulate the differential equations again as quadrature problems, and solve the associated fixed point problem using Newton's method. They mention that the general idea is related to waveform relaxation, and cite the paper by Bellen and Zennaro. They show simulations using several thousands of processors for simulating planetary orbits with 9 planets, and need about 10 iterations for convergence.

Lions, Maday and Turinici 2001: the parareal algorithm invented in [13] is also based on a decomposition of the time interval into subintervals, and an iterative method to improve approximations at the beginning of each subinterval. It was first presented as a method that propagates jumps over a coarse grid, but later identified as a multiple shooting method with a coarse approximation of the

Jacobian in the Newton solver. The parareal algorithm sparked a wave of new research in time parallel methods.

Direct time parallel solvers: most of the time parallel solvers are iterative, but there have also been attempts to devise direct solvers. One idea described in [14] is to diagonalize the time stepping matrix. This is not an easy task, since for example for Forward Euler with equal time steps, this matrix is a Jordan block and thus not diagonalizable. If one uses different time steps however, diagonalization is possible, and one has to trade off time stepping and conditioning to obtain a direct time parallel solver. A different approach is the ParaExp algorithm [15], which is based on a completely overlapping time decomposition, and uses a Krylov method to propagate the solution of linear homogeneous problems accurately over long time, while incorporating source term influences by fine and accurate computations in parallel. For wave propagation, this is currently the most promising approach for time parallelization.

REFERENCES

[1] J. Nievergelt, "Parallel methods for integrating ordinary differential equations," *Comm. ACM*, vol. 7, pp. 731–733, 1964.

[2] W. L. Miranker and W. Liniger, "Parallel methods for the numerical integration of ordinary differential equations," *Math. Comp.*, vol. 91, pp. 303–320, 1967.

[3] W. Hackbusch, "Parabolic multi-grid methods," in *Computing Methods in Applied Sciences and Engineering, VI*, R. Glowinski and J.-L. Lions, Eds. North-Holland, 1984, pp. 189–197.

[4] A. Axelsson and J. Verwer, "Boundary value techniques for initial value problems in ordinary differential equations," *Mathematics of Computation*, vol. 45, no. 171, pp. 153–171, 1985.

[5] C. Lubich and A. Ostermann, "Multi-grid dynamic iteration for parabolic equations," *BIT*, vol. 27, no. 2, pp. 216–234, 1987.

[6] C. W. Gear, "Parallel methods for ordinary differential equations," *Calcolo*, vol. 25, no. 1-2, pp. 1–20, 1988.

[7] W. E. Milne and J. Wiley, *Numerical solution of differential equations*. Wiley New York, 1953, vol. 19.

[8] A. Bellen and M. Zennaro, "Parallel algorithms for initial-value problems for difference and differential equations," *J. Comput. Appl. Math.*, vol. 25, pp. 341–350, 1989.

[9] D. E. Womble, "A time-stepping algorithm for parallel computers," *SIAM J. Sci. Stat. Comput.*, vol. 11, no. 5, pp. 824–837, 1990.

[10] P. Chartier and B. Philippe, "A parallel shooting technique for solving dissipative ODEs," *Computing*, vol. 51, pp. 209–236, 1993.

[11] G. Horton and S. Vandewalle, "A space-time multigrid method for parabolic partial differential equations," *SIAM Journal on Scientific Computing*, vol. 16, no. 4, pp. 848–864, 1995.

[12] P. Saha, J. Stadel, and S. Tremaine, "A parallel integration method for solar system dynamics," *The Astronomical Journal*, 1997, arXiv:astro-ph/9605016.

[13] J. Lions, Y. Maday, and G. Turinici, "A 'parareal' in time discretization of PDEs," *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, vol. 332, no. 7, pp. 661–668, 2001.

[14] Y. Maday and E. M. Rønquist, "Parallelization in time through tensor-product space–time solvers," *Comptes Rendus Mathematique*, vol. 346, no. 1, pp. 113–118, 2008.

[15] M. J. Gander and S. Güttel, "PARAEXP: a parallel integrator for linear initial-value problems," *SIAM Journal on Scientific Computing*, vol. 35, no. 2, pp. C123–C142, 2013.

[16] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time-domain analysis of large scale integrated circuits," *IEEE Trans. on CAD of IC and Syst.*, vol. 1, pp. 131–145, 1982.

Multigrid Reduction Techniques For Parallel-In-Time Integration

Scott MacLachlan *Department of Mathematics, Tufts University, USA*
Stephanie Friedhoff *Department of Mathematics, Tufts University, USA*
Robert Falgout *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, USA*
Tzanio Kolev *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, USA*
Jacob Schroder *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, USA*

I. Summary

We present a family of truly multilevel approaches to parallel time integration based on multigrid reduction principles. The resulting multigrid-reduction-in-time (MGRIT) algorithms are non-intrusive approaches, which directly use an existing time propagator and, thus, can easily exploit substantially more computational resources than standard sequential time stepping. Furthermore, we demonstrate that MGRIT offers excellent strong and weak parallel scaling up to thousands of processors for solving diffusion equations in two and three space dimensions. The MGRIT approaches are natural multilevel extensions of the parareal algorithm; thus, they provide techniques that offer parallel scalability for cases where the "coarse-in-time" grid is still too large to be treated sequentially.

II. The MGRIT Methodology

While we typically think of the solution to a linear, time-dependent PDE to be defined via time-stepping, we can also represent it as the solution of a linear system, written in block form as

$$\begin{bmatrix} I & & & & \\ -\Phi_{\Delta t} & I & & & \\ & -\Phi_{\Delta t} & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi_{\Delta t} & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N_t} \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{N_t} \end{bmatrix},$$

where $\Phi_{\Delta t}$ represents the time-stepping operator that takes a solution at time $k\Delta t$ to that at time $(k+1)\Delta t$, along with a time-dependent forcing term g_k. For such bidiagonal systems, cyclic reduction is a natural and tempting approach to fast solution, whereby we first solve the Schur complement system,

$$\begin{bmatrix} I & & & & \\ -\Phi_{\Delta t}^m & I & & & \\ & -\Phi_{\Delta t}^m & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi_{\Delta t}^m & I \end{bmatrix} \begin{bmatrix} u_0 \\ u_m \\ u_{2m} \\ \vdots \\ u_{N_t} \end{bmatrix} = \begin{bmatrix} g_0 \\ \hat{g}_m \\ \hat{g}_{2m} \\ \vdots \\ \hat{g}_{N_t} \end{bmatrix},$$

for the value of the solution at every m^{th} temporal point, with consistently restricted forcing terms, then define the solution at the remaining temporal points by local (and parallel) time-stepping between those points defined from the Schur complement.

Interpreting this as a multigrid reduction algorithm, we can define the coarse temporal mesh, or C-points, to be those points included in the Schur complement system, with the remaining temporal points as F-points. We can further define "ideal" interpolation as the map which takes a solution at the C-points and yields a zero residual at the F-points, with a similar definition for "ideal" restriction. The Schur complement then arises as the standard Petrov-Galerkin coarse-grid operator with these definitions of restriction and interpolation. As is typical in the multigrid reduction setting, the MGRIT approaches replace the true Schur complement with a simpler operator (typically of the same form as the original bidiagonal system, but with time-step $m\Delta t$), replace ideal restriction with simple injection, and compensate by adding relaxation. Furthermore, the two-level method can be extended to multiple levels in a simple recursive manner.

III. Results

Figure 1 shows weak parallel scaling results for several MGRIT variants applied to the discretization of the heat equation in two space dimensions, $u_t = \Delta u$, with an implicit Euler discretization for the time derivative and central finite-difference discretization of the spatial derivatives. We consider the spatial domain $[0, \pi]^2$ and the time interval $0 < t < \pi^2/64$, with specified initial condition $u(x, y, 0) = \sin(x)\sin(y)$, and fix the time-step, $\delta t = (\delta x)^2 = (\delta y)^2$ for δx, δy determined by the spatial grid. The problem size per processor is fixed at (roughly) 2^7 points in each spatial direction and 2^8 points in the temporal direction. Thus, on 1 processor, we use a uniform grid of $\delta x = \delta y = \pi/128$ and 257 points in time while, on 4096 processors, we use a uniform grid of $\delta x = \delta y = \pi/1024$ and 16,385 points in time. Shown are results for three different coarsening schemes: two coarsen uniformly across all grids, with factors $m = 2$ or $m = 16$, while the third, denoted $m = 16/2$ in the figure, coarsens by factors of 16 until fewer than 16 temporal points are left on each processor, then coarsens by factors of 2. For each coarsening scheme, the solid line shows results for standard FCF-relaxation, while the dashed lines correspond to multigrid schemes that use F-relaxation on the finest grid, and FCF-relaxation on all coarse grids. Spatial problems are solved using a parallel spatial multigrid method with a heuristic stopping tolerance. Here, we see excellent weak scaling, with roughly 50% efficiency at 4096 processors.

Strong scaling results for the heat equation in two space dimensions are shown in Figure 2. Here, we consider a problem on the spatial domain $[0, \pi]^2$ and the time interval $0 < t < \pi^2$. The problem is discretized on a 129^2 mesh in space, with 16,385 temporal points. For the time-stepping approach, we parallelize only in space and use sequential time-stepping. For all three MGRIT variants, we parallelize over 16 processors in the spatial dimensions, with increasing numbers of processors in the temporal dimension. Here, we again see slight improvement

from the MGRIT variant using V-cycles with F-relaxation on the finest grid and FCF-relaxation on all coarse grids (denoted F-FCF in the figure) over that with FCF-relaxation on all grids. While an F-cycle variant using F-relaxation on all grids offers some improvement on smaller numbers of processors, it shows somewhat poorer parallel scalability. Overall, we see excellent speedup for the MGRIT results at high processor counts. We note that these results highlight the fact that the MGRIT framework is largely intended for the case where many more processors are available than can be effectively utilized by sequential timestepping. While there is substantial overhead in MGRIT algorithms over the optimal algorithmic scaling of time-stepping, this extra work can be effectively parallelized at very large scales.

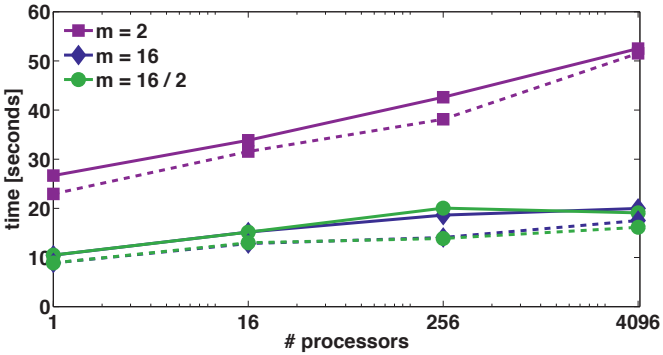


Fig. 1: *Weak scaling results for MGRIT variants applied to the heat equation in two space dimensions.*

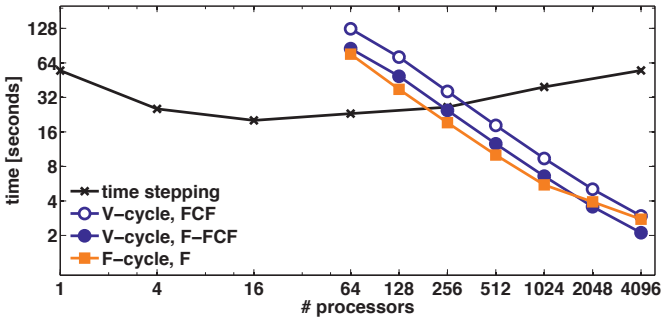


Fig. 2: *Strong scaling results for MGRIT variants applied to the heat equation in two space dimensions.*

Handling with the time-wise dimension by Fourier techniques, Backward Stochastic Differential Equations and Graphics Processing Units

Cornelis W. Oosterlee *CWI – Center for Mathematics and Computer Science – Amsterdam, and Delft University of Technology, the Netherlands*

Joint work: This presentation is based on various pieces of research in our group. Co-authors of this work include: H. Knibbe, C. Vuik (Helmholtz equation), M. Ruijter (BCOS method), S. Jain, A. Leitao (parallel SGBM).

I. Abstract

In this presentation with its lengthy title we will discuss ”parallel in time” in a slightly different fashion. We will discuss in detail the latest advancements regarding Fourier transformation of time-dependent partial differential equation (PDE) problems. Next to a discussion about efficient solvers for the Helmholtz equation (ie. the Fourier transform of the wave equation), we will discuss stochastic approaches for solving time-dependent problems.

This latter issue is closely connected to the Feynman-Kac theorem which states that we can find the solution of a time-dependent PDE by means of the computation of a conditional expectation. This theorem has been generalized to semi-linear and even nonlinear PDEs, where so-called Backward Stochastic Differential Equations (BSDEs) play an important role for their numerical solution. We will explain the concept and its applicability. At the same time we have developed a Fourier technique to deal with these BSDEs highly efficiently called the BCOS method (is ”BSDE COS method”, based on Fourier cosine series expansions). So, from a stochastics point-of-view, highly efficient and quite general Fourier-based solution techniques have recently been developed for time-dependent, but also for nonlinear, PDEs. These discrete schemes have a close connection to fundamental solutions of PDEs, and their Fourier transforms, and also to the characteristic function, which is the Fourier transform of the probability density function, well-known in probability theory. Next to these Fourier cosine techniques, we will briefly explain a specific Monte Carlo technique which can also be used for solving essentially the same PDE problems, and which has been implemented on Graphics Processing Unit (GPU) parallel hardware. Speed up numbers are reported.

A. Helmholtz equation

The acceleration of computing time with the help of GPUs also plays a role in the first part of the presentation which deals with the Helmholtz equation (the

Fourier transform of the time-dependent wave equation) and efficient solution methods for this type of equation. This work is based on the shifted Laplacian preconditioner, and we aim to show that three-dimensional reverse-time migration, a well-known application in seismic imaging, can be efficiently performed on the basis of the Helmholtz equation on parallel hardware.

Three-dimensional reverse-time migration with the constant-density acoustic wave equation requires an efficient numerical scheme for the computation of wave-fields. An explicit finite-difference scheme in the time domain is a common choice. However, it requires a significant amount of disk space for the imaging condition. The frequency-domain approach simplifies the correlation of the source and receiver wave-fields, but requires the solution of a large sparse linear system of equations. For the latter, we use an iterative Krylov solver based on a shifted Laplace multigrid preconditioner with matrix-dependent prolongation. The question is whether migration in the frequency domain can compete with a time-domain implementation when both are performed on a parallel architecture. Both methods are naturally parallel over shots, but the frequency-domain method is also parallel over frequencies. If we have a sufficiently large number of compute nodes, we can compute the result for each frequency in parallel and the required time is dominated by the number of iterations for the highest frequency. As a parallel architecture, we consider a commodity hardware cluster that consists of multi-core central processing units (CPUs), each of them connected to two graphics processing units (GPUs). Here, GPUs are used as accelerators and not as an independent compute node. The parallel implementation of the 3D migration in frequency domain is compared to a time-domain implementation.

The choice of the numerical scheme is motivated by complexity analysis. Consider a 3-D problem of size $N = n^3$, with n_s shots, n_t time steps, n_f frequencies, n_{it} iterations. The number of shots is usually $n_s \sim n^2$, the number of time steps $n_t \sim n$, the number of frequencies is $n_f \sim n$ at most, and the number of iterations for the iterative frequency-domain method is $n_{\text{it}} \sim n_f$.

B. Feynman-Kac Theorem

One of the targets of our research is to combine the different research lines (solvers for Seismics and solvers for Computational Finance problems) in our group. In other words, it should be possible to also solve the Helmholtz equation, with spatially-dependent wave numbers, by means of fundamental solutions, Fourier cosine expansions, a discrete version of the characteristic function, and the Feynman-Kac relation or the connection with BSDEs.

The Feynman-Kac theorem states that the solution of the linear partial differential equation:

$$v_t(t,x) + \mathcal{L}v(t,x) + g(t,x) = 0,\ v(T,x) = h(x),$$

with operator

$$\mathcal{L}v(t,x) = \mu(x)v_x(t,x) + \frac{1}{2}\sigma^2(x)v_{xx}(t,x),$$

can be obtained as a conditional expectation, ie.

$$v(t,x) = \mathbb{E}\left[\int_t^T g(s,X_s)ds + h(X_T)\right],$$

where X_s is the solution to the forward stochastic differential equation (FSDE):

$$dX_s = \mu(X_s)ds + \sigma(X_s)d\omega_s,\ X_t = x.$$

A pricing approach originating from this relation is based on quadrature,

$$v(t_0,x_0) = e^{-r(T-t_0)}\int_{\mathbb{R}} v(T,x_T)f(x_T,x_0)dx_T$$

The transitional probability density function, $f(x_T,x_0)$, is typically not available, but the characteristic function often is. The COS method has been developed (targeting applications in Computational Finance) based on the availability of the characteristic function. The commonly known *continuous* characteristic function is however not available in the case of spatially-dependent wave numbers, but the COS method has recently been generalized to work with characteristic functions for *discrete* SDE schemes. We will report upon this development, as well as on the generalization of these concepts towards BSDEs.

In recent years, also different Monte Carlo simulation techniques for valuation of high-dimensional conditional expectations were developed in order to deal with this challenging problem. In principle, these techniques can also be used for our purpose here. One of these Monte Carlo pricing techniques is the Stochastic Bundling Grid Method (SGBM), proposed by Jain and Oosterlee for pricing Bermudan financial options with several underlying assets. The method is a hybrid of *regression*- and *bundling*- based approaches, and uses regressed value functions, together with bundling of the state space to approximate numerical solutions at different time steps. The method's applicability has been extended by increasing the number of bundles and the problem dimensionality, which, together also imply a drastic increase of the number of Monte Carlo paths. As the method becomes much more time-consuming then, we propose a parallel SGBM method taking advantage of the General-Purpose computing on Graphics Processing Units (GPGPU) paradigm.

Timing and Performance of PFASST

Matthew Emmett *Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley USA*

I. PFASST

The *parallel full approximation scheme in space and time* (PFASST) algorithm was first introduced in [1], and is an extension of earlier work that combines the *spectral deferred correction* (SDC) algorithm with the *Parareal* algorithm (see [2], [3] and references therein). I will present a brief overview of PFASST and how it operates.

PFASST has been used to solve a growing number of problems on a variety of machines, including various PDEs and particle systems (please see, e.g., [1], [4], [5], [6], [7] and references therein). Modest speedups have been attained on large machines, perhaps most notably a 50% speedup across 32 processors for a pseudo-spectral implementation of the 3d Navier-Stokes equation (using parallel FFTW operators); and the now quintessential PFASST+PMG run of the 3d heat equation on JUQUEEN using all 458,752 of its cores, where a speedup of 15.12 was achieved across 28 time ranks. Most of these applications are implemented using the LIBPFASST library, which is a pure Fortran 90 implementation of PFASST. I will present some scaling and timing results obtained using LIBPFASST and highlight how these results should be interpreted with respect to PFASSTs critical path [8].

II. Implementation

Early versions of LIBPFASST were relatively concise and implemented the basic PFASST algorithm. As new features have been added, LIBPFASST has become increasingly complex especially with regards to: stopping criteria, looping strategies, and how status information is communicated to other processors. I will present how LIBPFASST handles these complexities, and contend that some of these complexities are best addressed outside of the MPI framework.

III. Benchmarks

Various time-parallel algorithms have begun to mature and it is time for our community to build a set of common benchmarks. This is a tricky endeavour, and should be done collaboratively with the understanding that particular techniques are (most likely) better suited for different classes of problems. If our field is to move beyond a proof-of-concept stage and into an export stage, where a broader set of scientists begin leveraging and using our algorithms, we must provide compelling evidence to show that our techniques will provide (additional) speedup; and we must be able to guide informed decisions on which algorithms

are appropriate for a particular problem. I contend that some set of benchmarks are necessary for this to happen.

References

[1] M. Emmett and M. L. Minion, "Toward an efficient parallel in time method for partial differential equations," *Comm. Applied Math. and Comput. Sci.*, vol. 7, pp. 105–132, 2012.

[2] M. L. Minion, "A hybrid parareal spectral deferred corrections method," *Comm. Applied Math. and Comput. Sci.*, vol. 5, no. 2, pp. 265–301, 2010.

[3] M. L. Minion and S. A. Williams, "Parareal and spectral deferred corrections," in *AIP Conf. Proc.*, vol. 1048, 2008, p. 388.

[4] R. Speck, D. Ruprecht, R. Krause, M. Emmett, M. Minion, M. Winkel, and P. Gibbon, "Integrating an N-body problem with SDC and PFASST," in *Proc. 21st Int. Conf. Domain Decomposition Methods*, 2012.

[5] ——, "A massively space-time parallel N-body solver," in *Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis*, ser. SC '12, 2012, pp. 92:1–92:11.

[6] D. Ruprecht, R. Speck, M. Emmett, M. Bolten, and R. Krause, "Poster: Extreme-scale space-time parallelism," in *Proc. 2013 Conf. on High Performance Computing Networking, Storage and Analysis Companion*, ser. SC '13 Companion, 2013.

[7] R. Speck, D. Ruprecht, M. Emmett, M. Bolten, and R. Krause, "A space-time parallel solver for the three-dimensional heat equation," in *Proc. ParCo: Int. Conf. Parallel Computing*, 2013.

[8] M. Emmett and M. L. Minion, "Efficient implementation of a multi-level parallel in time algorithm," in *Proc. 21st Int. Conf. Domain Decomposition Methods*, 2012.

[9] M. L. Minion, "Semi-implicit spectral deferred correction methods for ordinary differential equations," *Comm. Math. Sci.*, vol. 1, no. 3, pp. 471–500, 2003.

Parallel-in-Time Integration for Atmospheric Gas-Phase Chemistry

Teresa Beck *Engineering Mathematics and Computing Lab, University of Heidelberg, Germany*

High performance computing has become an indispensable tool for numerical weather prediction. To employ parallel architectures, adequate parallelization strategies are required, that optimally utilize the existing hardware architectures for the scenario considered. Most of the approaches in use address a parallelization in space only. In view of future computing architectures with extreme scales of processing units, I investigate the potential of additionally considering a parallelization in time. By means of the regional climate and air quality prediction code COSMO-ART [1], I examine a parallelization in time for its most time-consuming sub-model in combination with a model reduction technique of intrinsic low dimensional manifolds.

I. Motivation for a parallelization in time

COSMO-ART is a coupled climate model, that bases on the numerical weather prediction model COSMO and is being supplemented by an Aerosols and Reactive Trace gases (ART) extension, developed by Institute of Meteorology and Climate Research at Karlsruhe Institute of Technology. COSMO-ART allows for a calculation of the interdependency between atmospheric gases and aerosols with the state of the atmosphere. The current approach to parallelize the meteorological core COSMO is based on a two dimensional domain decomposition method. Within the ART extension, the chemical kinetics of the gas phase chemistry is decoupled from the meteorological core, advection and diffusion. Conceptually, chemistry is considered to take place in isolated boxes, which are coupled to the meteorological core by mutual updates at distinct time intervals. As individual chemistry boxes do not couple with each other during one such integration interval, the decoupling allows for a trivial parallelization: Each chemistry box can be solved individually on one processing unit.

Gas phase chemistry is being modeled by a nonlinear system of ordinary differential equations. As processes on a wide range of temporal scales are being considered, the required time step size to solve these systems numerically is very small. Despite the trivial parallelization, this sub-model accounts for approximately 33% of the total computing time of COSMO-ART. The level of parallelization for this approach however is constrained by the number of chemistry boxes to be solved in parallel, as increasing the number of processing units above the number of chemistry boxes, will leave performance unaffected.

Even though the currently employed computing resources do not yet support an exploitation of the full potential of the trivial parallelism, I take one step ahead exascale computing and investigate the additional potential of a supplementary parallelization in time. Employing such approaches will be related to a higher

total process time compated to a serial model, still, it may be very promising for certain scenarios for which an extremely fast prediction is in pursuit. This could be the case for the forecast of the air quality after a massive pollution of the atmosphere induced by a volcanic eruption or a chemical disaster. In such cases, fast numerical forecasts can save both money and lives and therefore justify the usage of extreme computing power. Heading for an optimal benefit from future extreme-scale architectures, I examine the parareal algorithm for atmospheric chemistry in combination with a reduced model composed of intrinsic low dimensional manifolds.

The parareal algorithm has already been applied to chemical kinetic systems by e.g. Maday [2] and Blouza et al. [3]. Most of these investigations employed reduced models based on quasi-steady states assumptions for certain species, or a classification of slow or fast reactions according to the magnitude of the respective rate constants. Although these are widely used approaches, they have certain drawbacks: A chemical species can for example contribute to both fast and slow reactions and a *fast* reaction may have no turnover due to missing species contributions, same as a *slow* one. Differently than classifying species or reactions, I consider an approach, that relies on a classification of modes, where each mode is associated to one single time scale amongst the multiplicity of time scales present in such a system.

II. Planned Work

Within my PhD, I want to design and test an efficient parallel-in-time algorithm for atmospheric gas-phase chemistry by means of a cascade of scenarios of increasing complexity. The scenarios being considered range from both one very simple linear and one more complex nonlinear three-variable system, a six-variable nonlinear system, and finally a highly non-linear system as it arises in COSMO-ART, composed of 80 chemical species that interact in approx. 200 reactions. In the first phase of my PhD, I investigated model reduction techniques for atmospheric chemistry, especially focusing on the idea of intrinsic low dimensional manifolds. In the following, I want to give you a short idea of what I plan to do next.

A. Draft of an Algorithm

I plan to follow the basic idea of a parallel-in-time integration and start with a first approximation over an interval using a coarse model. Consequently, the interval is partitioned into several timeslabs, and the coarsly predicted approximation can be used as seed values for each of the timeslabs. Then, the problem on the fine timeslabs is solved locally, in parallel and using a fine model, while iteratively updating the seed values for each timeslab.

B. The Reduced Model

As a coarse model, I intend to use a reduced model adopting the idea of describing the long term behaviour of a kinetic system in terms of intrinsic low

dimensional manifolds (ILDM), as introduced by [4]. The approach bases on describing the long term development of the chemical system in terms of slow modes only, which can be identified via an eigenvalue analysis. Fast modes are treated as functions of the slow modes. A description in terms of slow manifolds decreases the numbers of unkowns in the system of ODEs to be solved, as well as it relaxes its stiffness. With the fast timescales completely being removed from the system, one can use larger timesteps for its numerical approximation. Once the integration step of the low-dimensional manifold has been carried out, the contributions of fast modes, that had been neglected so far, are being updated in terms of the slow ones. The approach assumes that fast modes are in a local equilibrium on a long term view. By assuming their temporal change to be negligible, the slow modes implicitely define the fast ones in terms of a nonlinear problem. The update step therefore mainly consists of solving a nonlinear problem, which in practice and especially for the scenario under investigation, turns out to be time consuming.

The idea of the ILDM goes back to Maas et al. [4] and has been successfully applied to the simulation of combustion proccesses for more than two decades. In combustion simulation, kinetic systems can be reduced to one- or two-dimensional manifolds, which enables the usage of tabulation strategies for pre-calculated updates, instead of directly computing the update steps. The existence of such low-dimensional manifolds in tropospheric chemical systems, with 80 chemical species reacting in approx. 200 chemical reactions, has been studied by e.g. Tomlin et al. [5]. In general, such systems seem to inhere low-dimensional manifolds with sizes significantly below the original problem size. As most of the reactions are highly sensitive to the radiation-depending photolysis rate, these low-dimensional manifolds show a diurnal variation. So far, my investigation for the gas-phase chemistry within COSMO-ART shows a minimal dimension of the ILDM ranging from 2 (at night) up to 16 (at day), which technically disables the usage of a tabulation strategy instead of an on-the-fly computation of the ILDM.

References

[1] C. Knote, D. Brunner, H. Vogel, J. Allan, A. Asmi, M. Äijälä, S. Carbone, H. D. van der Gon, J. L. Jimenez, A. Kiendler-Scharr, C. Mohr, L. Poulain, A. S. H. Prévôt, E. Swietlicki, and B. Vogel, "Towards an online-coupled chemistry-climate model: evaluation of trace gases and aerosols in cosmo-art," *Geoscientific Model Development*, vol. 4, no. 4, pp. 1077–1102, 2011. [Online]. Available: http://www.geosci-model-dev.net/4/1077/2011/

[2] Y. Maday, "Parareal in time algorithm for kinetic systems based on model reduction," *Amer. Math. Soc.*, pp. 183–194, 2007.

[3] A. Blouza, L. Boudin, and S. M. Kaber, "Parallel in time algorithms with reduction methods for solving chemical kinetics," *Communications in Applied Mathematics & Computational Science*, vol. 5, no. 2, pp. 241–263, Sep. 2010.

[4] U. Maas and S. Pope, "Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space," *Combustion and Flame*, vol. 88, no. 3-4, pp. 239–264, Mar. 1992.

[5] A. S. Tomlin, L. Whitehouse, R. Lowe, and M. J. Pilling, "Low-dimensional manifolds in tropospheric chemical systems," *Faraday Discuss.*, vol. 120, pp. 125–146, 2002.

First step towards Parallel and Adaptive space-time Computation of Maxwell's Equations

Stefan Findeisen *Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology (KIT), Germany*
Christian Wieners *Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology (KIT), Germany*

I. Maxwell Problem

An electromagnetic wave consists of two fields, the electric $\mathbf{E}$ and the magnetic $\mathbf{H}$. In a linear material they can be computed by the first–order Maxwell system

$$\begin{aligned} \mu\partial_t\mathbf{H} + \nabla\times\mathbf{E} = 0, \qquad & \varepsilon\partial_t\mathbf{E} - \nabla\times\mathbf{H} = 0, \\ \nabla\cdot(\mu\mathbf{H}) = 0, \qquad & \nabla\cdot(\varepsilon\mathbf{E}) = 0, \end{aligned}$$

with permeability μ and permittivity ε for all $t \in [0,T]$. For a given initial condition $\mathbf{u}_0$ this can be written as an evolution equation of the form

$$L\mathbf{u}(t) := M\partial_t\mathbf{u}(t) + A\mathbf{u}(t) = \mathbf{0} \qquad t\in[0,T]\,, \qquad \mathbf{u}(0) = \mathbf{u}_0\,.$$

Here, we present a fully implicit parallel space–time approach in $Q := \Omega\times[0,T]$ as an alternative to estabilshed time stepping methods (see [1]). In the examples in IV we considered a 2D reduction of Maxwell's equations for the TM mode.

II. Discretization

We use discontinuous Galerkin finite elements with upwind flux for the spatial discretization (see [2, Ch. 6.5]) and a Petrov-Galerkin discretization in time with continuous ansatz space and discontinuous test space. For discretization we define space–time cells $\tau := K_\tau \times I_\tau$ which consists of a spatial element K_τ and a local time interval $I_\tau = (t_\tau^{\min}, t_\tau^{\max})$. Hence, the space–time cylinder can be decomposed into a finite set $\mathcal{T}$ of open space–time elements $\tau \subset Q$ such that $\overline{Q} = \bigcup_{\tau\in\mathcal{T}}\overline{\tau}$. Hence our ansatz, test and local test spaces U_h, H_h and $H_{h,\tau}$ are given as

$$\begin{aligned} U_h = \Big\{ & \mathbf{u}_h \in \mathrm{H}^1(0,T;\mathrm{L}_2(\Omega)^3) : \mathbf{u}_h(0) = \mathbf{u}_0 \text{ and for all } \mathbf{x}\in\Omega \\ & \mathbf{u}_h(\mathbf{x},t) = \frac{t_\tau^{\max}-t}{t_\tau^{\max}-t_\tau^{\min}}\mathbf{w}_{\tau,h}(\mathbf{x},t_\tau^{\min}) + \frac{t-t_\tau^{\min}}{t_\tau^{\max}-t_\tau^{\min}}\mathbf{v}_{\tau,h}(\mathbf{x}) \\ & \text{where } (\mathbf{x},t)\in\tau \text{ and } \mathbf{w}_{\tau,h}\in U_h|_{[0,t_\tau^{\min}]} \text{ and } \mathbf{v}_{\tau,h}\in H_{\tau,h}\Big\}\,, \\ H_h = \big\{ & \mathbf{v}_h \in \mathrm{L}_2(Q)^3 : \mathbf{v}_h|_\tau \in H_{\tau,h}\big\} \text{ and } H_{\tau,h} = (\mathbb{P}_{p_\tau}(K_\tau)\times\mathbb{P}_{q_\tau-1}(I_\tau))^3. \end{aligned}$$

Lemma 1. *Let A_h be the discontinuous Galerkin operator with upwind flux approximating A and $L_h := M_h\partial_t + A_h$ and $\mathbf{f}\in\mathrm{L}_2(Q)^3$. Then a unique*

discrete solution $\mathbf{u}_h \in U_h$ *exists and is characterized by the variational equation*

$$(L_h \mathbf{u}_h, \mathbf{v}_h)_Q = (\mathbf{f}, \mathbf{v}_h)_Q \,, \qquad \mathbf{v}_h \in H_h \,.$$

By using hash map containers to store the space–time cells, it is easy to distribute the space–time cells among the different processes and solve the complete problem in parallel. Since this discretization is implicit in time, no CFL limitation applies. The arising linear system is solved with a parallel multigrid method on a space-time mesh hierarchy. It consists of a GMRES solver and a multigrid preconditioner. The preconditioner uses l meshes $Q_l, \ldots, Q_0$ and different polynomial degrees in space and time. Furthermore we use a few steps of the Gauß–Seidel method as a pre– and postsmoother (see [3]).

III. Adaptivity

In every space–time cell τ we can select a polynomial degree p_τ in time and q_τ in space. We use a so called dual error estimator to estimate the error $\mathbf{e} := \mathbf{u} - \mathbf{u}_h$ with respect to a given linear error functional $\mathcal{J}(\mathbf{v}) := (\mathbf{j}, \mathbf{v})_Q$ and density function $\mathbf{j}(\mathbf{x}, t)$ (see [4, Ch. 9.3]). Now, the aim is to minimize $\mathcal{J}(\mathbf{e})$. We use the duality argument

$$(L\mathbf{v} - \mathbf{f}, \mathbf{u}^*)_Q = \mathcal{J}(\mathbf{v}) \qquad \forall \mathbf{v} \in \mathrm{L}_2(Q)^3$$

with dual solution $\mathbf{u}^* \in \mathrm{L}_2(Q)^3$ to get the error representation

$$\mathcal{E} = \mathcal{J}(\mathbf{e}) = (L\mathbf{e} - \mathbf{f}, \mathbf{u}^* - \mathbf{u}_h^*)_Q = (L(\mathbf{u} - \mathbf{u}_h) - \mathbf{f}, \mathbf{u}^* - \mathbf{u}_h^*)_Q .$$

The error can be estimated as

$$|\mathcal{E}| = |\mathcal{J}(\mathbf{e})| \leq |\mathcal{E}_0| + \sum_{\tau \in \mathcal{T}} \eta_\tau = |\mathcal{E}_0| + \sum_{\tau \in \mathcal{T}} \left(\rho_\tau^{(1)} \omega_\tau^{(1)} + \rho_\tau^{(2)} \omega_\tau^{(2)} \right)$$

with initial discretization error $\mathcal{E}_0$ and residuals $\rho_\tau^{(1)}(\mathbf{u}_h)$, $\rho_\tau^{(2)}(\mathbf{u}_h)$ and weights $\omega_\tau^{(1)}(\mathbf{u}^* - \mathbf{u}_h^*)$, $\omega_\tau^{(2)}(\mathbf{u}^* - \mathbf{u}_h^*)$. The unknown exact dual solution $\mathbf{u}^*$ is approximated from the computed solution $\mathbf{u}_h^*$ via a polynomial recovery of higher order in space and time. The estimated error on each cell η_τ is used in a marking strategy, where for example 25% of the cells with the highest estimated error are marked. In a second step the polynomial degrees on these cells are increased. Hence high polynomial degrees are used in areas where it is necessary to minimize the error with respect to $\mathcal{J}$. Whereas lowest polynomial degrees are used in areas which do not effect $\mathcal{J}(\mathbf{e})$.

IV. Numerical Results

A. Travelling Wave

We test the method for a $\mathbf{u}(\mathbf{x}, t) = \mathbf{a}(\mathbf{x} \cdot \mathbf{k} - ct)$ with an amplidude profile $\mathbf{a} \in \mathcal{C}_2(Q)^3$ in $Q = [0, 2] \times [0.5, 0.5] \times [0, 2]$. The linear error functional is given as $\mathcal{J}(\mathbf{e}) := |S|^{-1} \int_S \mathbf{e}_y \, \mathrm{d}(x, t)$ with region of interest $S := [1.75, 2] \times [0.5, 0.5] \times [1.75, 2]$. In the adaptive case we achieve the same accuracy as in the uniform case, but we saved over 50% of the degrees of freedom (see table I).

level of refinement	adaptive #DoFs	adaptive $\lvert\mathcal{J}(\mathbf{e})\rvert$	uniform #DoFs	uniform $\lvert\mathcal{J}(\mathbf{e})\rvert$
0	50.688	1.9359e-1	50.688	1.9359e-1
1	178.518	6.2293e-3	304.128	6.2307e-3
2	469.308	1.6033e-4	912.584	1.4979e-4
3	978.459	7.3783e-5	2.027.520	7.3686e-5

TABLE I: *Accuracy of the adaptive and the uniform case*

B. Double-slit

In this example we consider the well known double-slit experiment, where we are only interested in the location of the first minimum on the right boundary of our computational domain (see 1a). Only parts which take influence to the minimum are refined (see 1b)–d)).

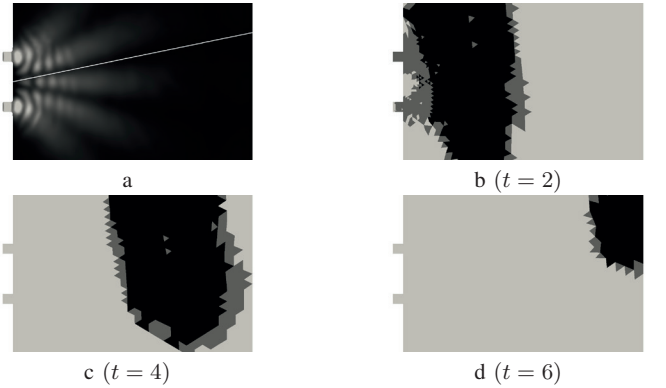


Fig. 1: *Double-slit experiment: a) location of the first minimum, b)–d) distribution of the polynomial degrees (bright: lowest, dark: highest)*

REFERENCES

[1] M. Hochbruck, T. Pažur, A. Schulz, E. Thawinan, and C. Wieners, "Efficient time integration for discontinuous Galerkin approximations of linear wave equations," Karlsruhe Institute of Technology, Tech. Rep., 2013. [Online]. Available: http://na.math.kit.edu/download/papers/TI-DG-Wave.pdf

[2] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods : algorithms, analysis, and applications*. New York, NY: Springer, 2008.

[3] C. Wieners, "A geometric data structure for parallel finite elements and the application to multigrid methods with block smoothing," *Comput. Vis. Sci.*, vol. 13, no. 4, pp. 161–175, 2010.

[4] W. Bangerth and R. Rannacher, *Adaptive finite element methods for differential equations*. Basel: Birkhäuser, 2003.

Using time-parallel methods for the simulation of a machine tool

Michael Klöppel *TU Dresden, Institute of Scientific Computing*
Andreas Naumann *TU Dresden, Institute of Scientific Computing*
Jörg Wensch *TU Dresden, Institute of Scientific Computing*

I. Introduction

The last decade has seen an ever increasing availability of processor cores even on consumer hardware. On the other hand, the computation power of single cores seems to have hit a limit. This development leads to new challenges when implementing numerical software, where one of the major challenges is the invention and implementation of efficient parallel algorithms. For the solution of ODEs and PDEs several approaches exist. Whereas parallelization in the spatial domain is rather common, parallization in the time domain is a relatively new development. Our work concentrates on how these latter ones can be used to further speed up our computations.

We currently work on two different projects, which both could benefit from parallel-in-time integration. The first project is the DFG-founded SFB/Transregio 96[1], which deals with the thermo-elastic simulation of certain machine tools (see Fig. 1 for an example). The second project is the BMBF-founded High Performance Computing Open Modelica[2] (HPCOM). Goal of this project is to enable the simulation of multibody systems on HPC infrastructure using the OpenModelica[3] simulation environment.

II. Findings and problems

To familiarize ourselves with parallel-in-time integrators we implemented our own versions of Parareal [1] and ParaEXP [2]. One especially noteworthy problem we met was calculating the matrix exponential in the ParaEXP method. According to Moler and Van Loan [3], expokit [4] is the most extensive software for evaluating this quantity. Nevertheless, it turned out that expokit is actually too computationally demanding to be used within ParaEXP. In our talk we give an account of this and further difficulties we encountered in the course of implementation.

A. Machine tool

The underlying problem in the machine tool simulation is linear, allowing the usage of Parareal as well as ParaEXP. With both methods we could observe

[1] transregio96.de
[2] hpc-om.de
[3] openmodelica.org

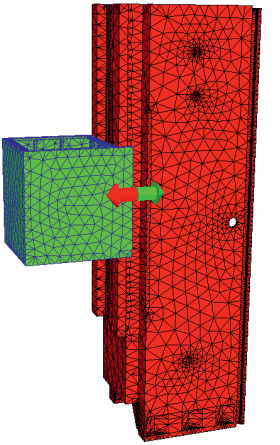


Fig. 1: *Machine tool with FEM mesh*

moderate speed ups ranging from 1.6 to 1.7 using 4 cores. Achieving higher speed ups in this example was limited by the given time scale, i.e., using more processors would have meant to use smaller time intervalls than in a serial solution approach.

B. Multibody Systems

Our standard test case for multibody systems is an N-pendulum, consisting of 50 linked parts. Using ParaReal, we simulated the motion of the pendulum and observed speed ups of about 1.5 using 4 cores. We discuss the influence of several different parameters (e.g., number of time intervals, coarse/fine integrator steps, ParaReal iterations, etc.) on the speed up and stability in our talk. Furthermore, we present results for more difficult test cases.

REFERENCES

[1] J.-L. Lions, Y. Maday, and G. Turinici, "Résolution d'edp par un schéma en temps pararéel," *C.R.Acad Sci. Paris Sér. I Math*, vol. 332, pp. 661–668, 2001.

[2] M. Gander and S. Güttel, "Paraexp: A parallel integrator for linear initial-value problems," *SIAM Journal on Scientific Computing*, vol. 35, no. 2, pp. C123–C142, 2013. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/110856137

[3] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003.

[4] R. B. Sidje, "Expokit: A software package for computing matrix exponentials," *ACM Trans. Math. Softw.*, vol. 24, no. 1, pp. 130–156, Mar. 1998. [Online]. Available: http://doi.acm.org/10.1145/285861.285868

Variational space-time methods for the elastic wave equation

M. Bause *Num. Meth. Comput. Eng., Helmut Schmidt Univ., Hamburg*
U. Köcher *Num. Meth. Comput. Eng., Helmut Schmidt Univ., Hamburg*

The accurate and reliable numerical approximation of the hyperbolic wave equation is of fundamental importance to the simulation of acoustic, electromagnetic and elastic wave propagation phenomena. Here, we study the elastic wave equation

$$\begin{aligned} \rho(\boldsymbol{x})\,\partial_t \boldsymbol{v}(\boldsymbol{x},t) - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}(\boldsymbol{u}(\boldsymbol{x},t)) &= \boldsymbol{f}(\boldsymbol{x},t) \quad \text{in } \Omega\times I\,, \\ \rho(\boldsymbol{x})\,\partial_t \boldsymbol{u}(\boldsymbol{x},t) - \rho(\boldsymbol{x})\,\boldsymbol{v}(\boldsymbol{x},t) &= \boldsymbol{0} \qquad\quad \text{in } \Omega\times I\,, \end{aligned} \tag{1}$$

written as velocity-displacement formulation and equipped with appropriate initial conditions and boundary conditions, where we denote by $\boldsymbol{v}$ the velocity, by $\boldsymbol{u}$ the displacement, by ρ the mass density, by $\boldsymbol{f}$ the body forces, by $\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}$ the stress and finally by $\boldsymbol{\varepsilon}$ the strain. Elastic waves appear in the design of integrated structural health monitoring systems for composites. For this, it is strictly necessary to understand phenomenologically and quantitatively wave propagation in layered fibre reinforced composites and the influene of the geometrical and mechanical properties of the system structure. Therefore, the ability to solve numerically the wave equation in three space dimensions is particularly important from the point of view of physical realism.

Recently, variational space-time discretisation schemes were proposed and studied for challenging problems, such as the nonstationary incompressible flow; cf. [3].

In this contribution we will focus on the presentation of variational time integration methods from the variational space-time approach for the hyperbolic elastic wave equation. For the spatial discretisation a symmetric interior penalty discontinuous Galerkin method for anisotropic media is used; cf. [2], [4].

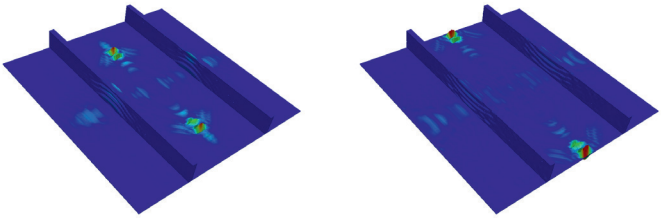


Fig. 1: *Guided ultrasonic waves in carbon fibre composite.*

As common starting point, we choose the weak formulation of Eq. (1), yielding

$$\begin{aligned} \int_I (\rho\, \partial_t \boldsymbol{v}, \boldsymbol{w}) + (\boldsymbol{\sigma}(\boldsymbol{u}), \boldsymbol{\varepsilon}(\boldsymbol{w}))\, \mathrm{d}t &= \int_I (\boldsymbol{f}, \boldsymbol{w})\, \mathrm{d}t\,, \quad \forall \boldsymbol{w} \in L^2(I, \boldsymbol{V})\,, \\ \int_I (\rho\, \partial_t \boldsymbol{u}, \boldsymbol{w}) - (\rho\, \boldsymbol{v}, \boldsymbol{w})\, \mathrm{d}t &= 0\,, \qquad \forall \boldsymbol{w} \in L^2(I, \boldsymbol{V})\,, \end{aligned} \tag{2}$$

with $\boldsymbol{V} = H_0^1(\partial\Omega_D; \Omega)^3$. We denote by $(\cdot, \cdot)$ the vector-valued L^2 inner product in space. Next, we discretise the time interval I into N disjoint elements $I_n = (t_{n-1}, t_n]$. Finally, we derive variational time integration methods by choosing boundary conditions in time, numerical quadrature in time and the test function space. By choosing a discontinuous test space in time, we can rewrite the resulting finite element in time method as time marching scheme over one or several elements I_n. Doing this, we can easily derive numerous well known schemes, such as the second order in time Crank-Nicolson scheme, which is equivalent to the unconditionally stable second order in time Newmark scheme (cf. [5]), as well as various new higher order continuous and discontinuous Galerkin schemes in time; cf. [1], [2].

From these classes of uniform Galerkin discretisations in space and time an approach of fourth-order accuracy is analysed carefully. More precisely, we use a continuous Petrov-Galerkin method of third order accuracy in time and apply an inexpensive post-processing step, which makes the numerical solution continuously differentiable in time. Further, the efficient solution of the resulting block-matrix system and inherently parallel numerical simulation through domain partioning is adressed. The performance properties of the schemes are illustrated by sophisticated and challenging numerical experiments with complex wave propagation phenomena in heterogeneous and anisotropic media.

References

[1] U. Köcher, M. Bause: Variational space-time methods for the wave equation, J. Sci. Comput., to appear, DOI:10.1007/s10915-014-9831-3, 2014.

[2] U. Köcher: Variational space-time methods for the elastic wave equation, Doctoral thesis, in preparation, 2014.

[3] S. Hussain, F. Schieweck, S. Turek: An efficient and stable finite element solver of higher order in space and time for nonstationary incompressible flow, Int. J. Numer. Meth. Fl., 73(11):927–952, 2013.

[4] J. De Basabe, M. Sen, M. Wheeler: The interior penalty discontinuous Galerkin method for elastic wave propagation: grid dispersion, Geophys. J. Int., 175:83–93, 2008.

[5] W. Bangerth, M. Geiger, R. Rannacher: Adaptive Galerkin finite element methods for the wave equation, Comput. Meth. Appl. Math., 10(1):3–48, 2010.

PyPinT — Towards a framework for rapid prototyping of iterative parallel-in-time algorithms

Torbjörn Klatt, Dieter Moser, Robert Speck *Juelich Supercomputing Centre, Juelich, Germany*

I. Motivation

With growing interest in parallel-in-time methods many different and new solvers for ordinary differential equations have gained the attention of researchers from various fields. In order to clearly estimate the potential and limitations of these mostly iterative solvers, a modular prototyping framework not only helps to understand their properties and various facets but also allows to easily implement and test new ideas.

As an example, the "parallel full approximation scheme in space and time" (*PFASST* [1]) and its serial counterpart, multi-level spectral deferred corrections (*MLSDC* [2]) are composed of multiple levels and even types of spectral deferred correction sweeps which are coupled by space-time restriction and interpolation operators. These modular and interchangeable combinations of different techniques already generate a vast amount of variations with different effects on solvability and efficiency towards a diverse set of problems.

II. PyPinT

For a thorough and systematic analysis of methods like *PFASST* or *Parareal* [3] we take the path of a well-planned and fully modular implementation of these algorithms. By following the object-oriented programming paradigm we create an abstract decomposition of the methods' functional components combined in a framework for parallel-in-time algorithms. Different methods implemented in a single framework using a unified base functionality enables detailed qualitative and quantitative analysis without paying too much attention to underlying implementation details.
As a proof of concept and intermediate step, we show results for a two-dimensional parabolic test equation solved with a *MLSDC* solver coupled with a multigrid algorithm in space.

Due to its flexibility, extensibility and rather comfortable learning curve Python has an ever growing world-wide community within science, academia and industry. For *PyPinT*, it provides the building block for a flexible and unified framework, allowing fast prototyping of iterative parallel-in-time algorithms. Well-maintained and open-source third party modules such as *NumPy* and *SciPy* offer high-level interfaces to performant low-level functionalities for matrix and vector arithmetics, common mathematical methods and plotting capabilities. In addition, current efforts leave the door open for enabling *PyPinT* to be applied on HPC clusters.

III. Goals

Accompanying the development of parallel-in-time algorithms, new ideas can be implemented in *PyPinT* immediately. Utilizing the framework's analysis tools such as calculation and plotting of stability regions, runtime and characteristic values (e.g. residuals), new algorithms can easily be studied in detail. Clearly defined interfaces, a strictly modular concept and different levels of abstraction enable the user to exchange certain parts of the algorithms and add his or her own methods to enrich the whole framework.

Students, undergraduate and graduate, with a basic knowledge of iterative solvers and some programming skills will be able to use and extend *PyPinT* and discover, learn and understand the mechanics of parallel-in-time methods. *PyPinT* is open-source licensed and available on GitHub [4], thus fostering collaboration and ease contribution of amplifications by interested people.
Ultimately, *PyPinT* should not only represent a package for applying and studying parallel-in-time methods but also provide a development environment for enhancing existing and inventing new methods. Finally, *PyPinT* can also provide valuable insight and guidance for future, performance-oriented implementations in other programming languages.

References

[1] M. Emmett and M. L. Minion, "Toward an efficient parallel in time method for partial differential equations," *Communications in Applied Mathematics and Computational Science*, pp. 1–28, 2012. [Online]. Available: http://msp.org/camcos/2012/7-1/camcos-v7-n1-p.pdf#page=107

[2] R. Speck, D. Ruprecht, M. Emmett, M. L. Minion, M. Bolten, and R. Krause, "A multi-level spectral deferred correction method," *arXiv Mathematics Numerical Analysis*, p. 30, 2013. [Online]. Available: http://arxiv.org/abs/1307.1312

[3] J.-L. Lions, Y. Maday, and G. Turinici, "A "Parareal" in time discretization of PDE's," *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, vol. 332, no. 7, pp. 661–668, Apr. 2001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0764444200017936

[4] T. Klatt, D. Moser, and R. Speck, "PyPinT — a Python framework for parallel-in-time algorithms," 2014. [Online]. Available: https://github.com/Parallel-in-Time/PyPinT

A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations

Giovanni Samaey *Department of Computer Science, KU Leuven, Belgium*
Frédéric Legoll *Laboratoire Navier, École Nationale des Ponts et Chaussées, Université Paris-Est and INRIA Rocquencourt, MICMAC team-project, France*
Tony Lelièvre *CERMICS, École Nationale des Ponts et Chaussées, Université Paris-Est, and INRIA Rocquencourt, MICMAC team-project, France*

We introduce a micro-macro parareal algorithm for the time-parallel integration of multiscale-in-time systems. The algorithm first computes a cheap, but inaccurate, solution using a coarse propagator (simulating an approximate slow macroscopic model), which is iteratively corrected using a fine-scale propagator (accurately simulating the full microscopic dynamics). This correction is done in parallel over many subintervals, thereby reducing the wall-clock time needed to obtain the solution, compared to the integration of the full microscopic model. We provide a numerical analysis of the algorithm for a prototypical example of a micro-macro model, namely singularly perturbed ordinary differential equations. We show that the computed solution converges to the full microscopic solution (when the parareal iterations proceed) only if special care is taken during the coupling of the microscopic and macroscopic levels of description. The convergence rate depends on the modeling error of the approximate macroscopic model. We illustrate these results with numerical experiments. These results have been published in [1].

References

[1] F. Legoll, T. Lelievre, and G. Samaey, "A micro-macro parareal algorithm: Application to singularly perturbed ordinary differential equations," *SIAM Journal on Scientific Computing*, vol. 35, no. 4, pp. A1951–A1986, 2013.

Multigrid Reduction in Time

J. B. Schroder *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory*† *(LLNL)*
R. D. Falgout *Center for Applied Scientific Computing, LLNL*†
Tz. Kolev *Center for Applied Scientific Computing, LLNL*†
V. A. Dobrev *Center for Applied Scientific Computing, LLNL*†
S. Friedhoff *Department of Mathematics, Tufts University*
S. P. MacLachlan *Department of Mathematics, Tufts University*
U. M. Yang *Center for Applied Scientific Computing, LLNL*†
N. A. Petersson *Center for Applied Scientific Computing, LLNL*†

I. Overview

The need for parallel-in-time approaches is being driven by current trends in computer architectures where performance improvements are coming from greater parallelism, not faster clock speeds. This leads to a bottleneck for sequential time integration methods because they lack parallelism in the time dimension. To address this issue, we examine an optimal-scaling multigrid method applied to the time dimension by solving the (non)linear systems that arise when considering the solution to multiple time steps simultaneously. Targeting an early impact on existing simulation codes (which can be extremely complex), we consider an approach based on multigrid reduction methods (MGR) that is designed to be as non-intrusive as possible, while maintaining a high degree of versatility. From one perspective, this approach is a generalization of the popular parareal algorithm [1] from a two-level setting to a multilevel setting with additional options relating to choices of relaxation and interpolation. A critical facet of the approach is the extra parallelism available through the addition of coarse grids.

In this talk, we will focus on practical aspects of our multigrid-reduction-in-time algorithm (MGRIT) [2]. The non-intrusiveness allows users to wrap existing time stepping codes into our code and the versatility allows the user to drive adaptive refinement and coarsening in both time and space. Applicability to nonlinear problems is achieved through a full approximation scheme (FAS) multigrid approach.

II. Application

To highlight the non-intrusiveness of MGRIT, we will present some practical experience and results, with the focus being on wrapping a serial nonlinear 2D Euler flow simulation code [3] inside of our MGRIT code. The target problem exhibits an unsteady vortex shedding as a fluid flows around a cylinder in the

†This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-ABS-653803)

center of the domain. While our work is in a preliminary state, MGRIT converges and in some settings can exhibit a speedup over sequential time stepping. MGRIT convergence is depicted in Figure 1, where the magnitude of the velocity is plotted for the region surrounding the cylinder. The four plots depict the final time value $t = 2.56s$ (the 1280^{th} time step) over various MGRIT iterations, in order to show how the method quickly captures the nature of the flow at the end of the simulation domain.

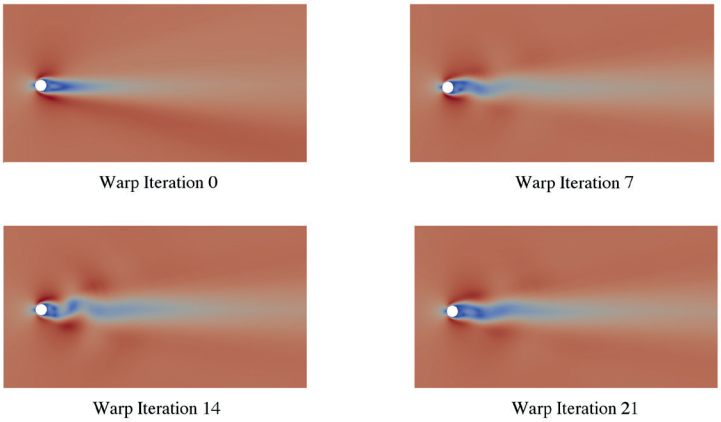


Fig. 1: *Zoomed plot of velocity magnitude at final time value as MGRIT converges, $t = 2.56s$, 1280^{th} time step, Unsteady vortex shedding problem around cylinder.*

However, significant challenges remain for this problem. In particular, the MGRIT iteration count can grow as the time domain is increased (i.e., Δt is fixed and the number of time steps increases). One approach under consideration is to leverage the periodic nature of the vortex shedding to provide improved residual corrections at later time steps.

References

[1] J. L. Lions, Y. Maday, and G. Turinici, "Résolution d'EDP par un schéma en temps pararéel," *C.R.Acad Sci. Paris Sér. I Math*, vol. 332, pp. 661–668, 2001.

[2] R. Falgout, S. Friedhoff, T. Kolev, S. MacLachlan, and J. Schroder, "Parallel time integration with multigrid," *SIAM J. Sci. Comput.*, (submitted 2013). LLNL-JRNL-645325. [Online]. Available: http://computation.llnl.gov/casc/linear_solvers/pubs/mgritPaper-2013.pdf

[3] A. J. Katz and D. Work, "High-order flux correction/finite difference schemes for strand grids," *52nd Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics*, Jan. 2014. [Online]. Available: http://dx.doi.org/10.2514/6.2014-0937

Potential for Parallel-in-time Computations in Climate Research

Thomas Slawig *Research Group Algorithmic Optimal Control, Christian-Albrechts-Universität Kiel, 24098 Kiel, Germany*

I. Motivation

Since some decades, the discussion about the observable climate change and its reasons is a main topic in scientific and political discussions in our society. Since it is not possible to conduct experiments with the Earth's climate all investigations and prognoses have to rely on models and their evaluation, i.e., numerical simulations.

II. The Climate System

In order to conduct numerical simulations to predict climate change scenarios, the rather complex climate system has to be understood and its numerous internal interactions has to be analyzed and modeled. The climate climate system usually is separated into the following five components (see e.g. [1]):

- The hydrosphere (ocean and all water on land),
- the atmosphere,
- the cryosphere, separated in land and sea ice,
- the land surface,
- and the biosphere, basically incorporating marine biota and land vegetation.

The so-called anthroposhere can be added to take account for human influence and/or interactions, for example when considering economical questions or climate (change) impact. These parts of the climate system have to be represented in a more or less detailed way in climate models, where the complexity level of the models can vary in

- spatial and temporal resolution,
- the number of represented processes and
- the complexity of the coupling between them and the different components.

III. Climate Models

Climate models usable to perform climate predictions or so called "hind-casts" of climate variations or change in the past are nowadays called *Earth System Models (ESMs)*. They consist of coupled systems of partial differential(-algebraic) equations. From the history of the climate model development, a typical ESM has the components

- ocean component or model with coupled marine biogeochemical (or marine ecosystem) and sea-ice models

- atmosphere model with coupled vegetation and land ice models.

Both components are coupled

- physically by the ocean surface where many exchange processes take place
- in the software by a special coupling software.

Since climate time scales for the Earth system are ranging to 100'000 years (e.g. for the simulation of the past glacial cycle) and more, the simulation of long time intervals is often performed using so-called *Earth System Models of Intermediate Complexity* (EMICs, see [2]), where many processes are parametrized, often using statistical assumptions and quantities. As a result, the computational time is significantly reduced, but spatial and temporal resolution is coarsened.

IV. Climate Simulations

Model runs with ESMs are performed on massive-parallel hardware, whereas EMIC runs often can be done on smaller machines in reasonable time. The parallelization in ESMs is only performed w.r.t. space, i.e., by some domain decomposition ansatz. It is a rather new approach to use a paralyzation also in time. It is motivated by

- the aim to perform very long runs (100'000 years and more model time) with a finer resolution than EMICs provide,
- and the limit in the expected gain of performance of today's CPU kernels.

Another way to gain performance is usage of hardware accelerators as GPGPUs.

V. Parallel-in-time Strategy using Reduced Models of Intermediate Complexity

Since there are some EMICs that have been proven to perform quite well in the simulation of long time periods (for example the *Climber* model [3]), one promising approach for a parallel-in-time scheme is to use of of these EMICs as "coarse" models and finer resolving ESMs parallel on shorter time slices, following the so-called "para-real" method presented by Lions et al., see [4].

References

[1] K. McGuffie and A. Henderson-Sellers, *A Climate Modelling Primer*, 3rd ed. Chichester: Wiley, 2005.

[2] M. Claussen, L. A. Mysak, A. J. Weaver, M. Crucifix, T. Fichefet, M.-F. Loutre, S. L. Weber, J. Alcamo, V. A. Alexeev, A. Berger, R. Calov, A. Ganopolski, H. Goosse, G. Lohmann, F. Lunkeit, I. I. Mokhov, V. Petoukhov, P. Stone, and Z. Wang, "Earth system models of intermediate complexity: closing the gap in the spectrum of climate models," *Climate Dynamics*, vol. 18, pp. 579–586, 2002.

[3] S. Rahmstorf, V. Brovkin, M. Claussen, and C. Kubatzki, "Climber-2: A climate system model of intermediate complexity. part ii: Model sensitivity," *Climate Dynamics*, vol. 17, pp. 735–751, 2001.

[4] J.-L. Lions, Y. Maday, and G. Turinici, "Résolution d'EDP par un schéma en temps "pararéel"." *C. R. Acad. Sci., Paris, Sér. I, Math.*, vol. 332, no. 7, pp. 661–668, 2001.

Identity Parareal Method and its Performance

Toshiya Takami *Research Institute for Information Technology, Kyushu University, Japan*

I. Identity Parareal Method

The standard implementation of a parareal-in-time algorithm [1] is usually based on the use of a coarse integrator $\mathcal{G}_k$ as an approximation of an original fine integrator $\mathcal{F}_k$, i.e., the original sequence $\{x_0, x_1, ..., x_k, ...\}$ is defined by $x_{k+1} = \mathcal{F}_k(x_k)$. By these integrators, we can introduce a parareal iteration,

$$x_{k+1}^{(r+1)} = \mathcal{G}(x_k^{(r+1)}) + \mathcal{F}(x_k^{(r)}) - \mathcal{G}(x_k^{(r)}). \tag{1}$$

This procedure is often explained as a predictor-corrector scheme, and an approximate sequence $\{x_k^{(r)}\}$ is expected to converge to the exact one for a sufficiently large r. However, it is known that there are several drawbacks in this algorithm: a tailor-made approximate solver is necessary, and its property affects convergence and speed-up ratio, i.e., total performance of the time-parallel method.

We showed that a simplified implementation called 'Identity Parareal' is available for time-evolution problems [2], [3], where the iteration is given by

$$x_{k+1}^{(r+1)} = x_k^{(r+1)} + \mathcal{F}(x_k^{(r)}) - x_k^{(r)}. \tag{2}$$

An identity transformation, i.e., nothing is changed, is used as a coarse integrator. Formal validation of this formula is given as follows. Suppose $x(t)$ is a vector of dynamical variables and is differentiable by t. Then, we can write

$$x(t + \delta t) = x(t) + \frac{\partial x}{\partial t}\delta t + \cdots, \tag{3}$$

which means that the integrator $\mathcal{F}(x(t))$ is approximated by an identity transformation. It is also seen that the leading error by this approximation is linear to δt when we use sufficiently small δt.

II. Properties of the Identity Parareal and Speed-up

Since the formal validity does not mean general applicability of this method to actual problems, we must analyze convergence properties numerically. We have shown that our implementation shows good convergence properties for those problems described by ordinary differential equations [2], [3]. We will present numerical convergence properties for molecular dynamics simulations and quantum time-evolutions.

Our simple implementation also shows effective performance if we parallelize the program with bucket-brigade communications [4]. Then, we have no limitations of speed-up ratio coming from a rate of computational times, T_f/T_g. In usual

implementations, a speed-up ratio by the parareal-in-time method is described by the formula,

$$S_{\mathrm{para}}(P,R) = \frac{(P+R-1)T_f}{(P-1)(T_g+T_c)+R(T_g+T_f+2T_c)}, \tag{4}$$

where P is the number of parallel resources and R is the number of iterations. T_f, T_g, and T_c are costs for a fine computation, a coarse computation, and communications, respectively. On the other hand, the speed-up of our implementation,

$$S_{\mathrm{ipara}}(P,R) = \frac{(P+R-1)T_f}{(P-1)T_c+R(T_f+2T_c)}, \tag{5}$$

can be higher than $S_{\mathrm{para}}(P,R)$, when we use the same parameters P and R.

In addition to this, communication performance can be optimized by segmenting transferred data into small pieces and overlapping with computations. Thus, the new implementation with an identity approximation is efficiently used in time-parallel computations on massively parallel computers [4].

III. Discussion

The 'identity parareal' is aimed at high-performance in large parallel computers. The strategy we used was different from the standard one while convergence properties are considered the most important. Another important property is stability in systems described by partial differential equations. We will discuss these properties including stability analysis of the 'identity parareal'.

References

[1] J.-L. Lions, Y. Maday, and G. Trinici, "A "parareal" in time discretization of pde's," *C. R. Acad. Sci. Ser. I Math.*, vol. 332, pp. 661–668, 2001.

[2] T. Takami and D. Fukudome, "An efficient pipelined implementation of space-time parallel applications," *Adv. Par. Comp.*, vol. 25, pp. 273–281, 2014.

[3] ——, "An identity parareal method for temporal parallel computations," *LNCS*, vol. 8384, 2014, in press.

[4] D. Fukudome and T. Takami, "Parallel bucket-brigade communication interface for scientific applications," in *Proceedings of EuroMPI'13*, 2013, pp. 135–136.

Comparing the Parareal and the Adaptive Parallel Time Integration (APTI) methods for a Satellite Problem

Nabil Nassif *Mathematics Department, American University of Beirut, Lebanon*

In this work, we tackle the orbit's computations of a satellite using 2 parallel approaches on a system of Ordinary Differential Equations modeling the J_2 perturbed Keplerian motion. The main points of the presentation consist of:

1. Implementing Lions Parareal Algorithm to the satellite problem.
2. Presenting the main features of the APTI method and its implementation for satellite computations.
3. Evaluating and comparing speed-ups through tests conducted on specific orbits cases

Space-Time Finite Elements and Non-Uniform Temporal Refinement

Marek Behr *RWTH Aachen, CATS, Germany*

Space-time approaches offer some not-yet-fully-exploited advantages when compared to standard discretizations (finite-difference in time and finite-element in space, using either method of Rothe or method of lines); among them, the potential to allow some degree of unstructured space-time meshing. A method for generating simplex space-time meshes is presented, allowing arbitrary temporal refinement in selected portions of space-time slabs. The method increases the flexibility of space-time discretizations. The resulting tetrahedral (for 2D problems) and pentatope (for 3D problems) meshes are tested in the context of advection-diffusion equation, and are shown to provide reasonable solutions, while enabling varying time refinement in portions of the domain.

Combining Finite Differences and Discontinuous Galerkin method for simulation of seismic waves propagation

Vadim Lisitsa *Institute of Petroleum Geology and Geophysics SB RAS, Russia*

Presence of the complex near-surface parts in geological modes, especially irregular high-contrast interfaces, such as free-surface and sea-bed, makes it troublesome to simulate seismic wave propagation with the use of finite differences where a stair-step approximation of the interfaces is typically applied. To overcome this effect an irregular tetrahedral (triangular) mesh is suggested to be used with further discontinuous Galerkin (DG) approximation of the elastic wave equation. However, discontinuous Galerkin method is highly computationally intense and memory consuming compared to finite differences on a staggered grid. What is suggested in this paper is the coupling of the two methods, so that DG is used in the near-surface part of the model, thus ensuring high accuracy of the high-contracts interfaces treatment. Whereas, the standard staggered grid finite difference scheme is used elsewhere, making the overall algorithm efficient.

Schwarz Methods for the Time-Parallel Solution of Parabolic Control Problems

Felix Kwok *University of Geneva, Section de mathematiques, Switzerland*

Discretized parabolic control problems lead to very large systems of equations, because trajectories must be approximated forward and backward in time. It is therefore of interest to devise parallel solvers for such systems, and a natural idea is to apply Schwarz preconditioners to the large space-time discretized problem. The performance of Schwarz preconditioners for elliptic problems is well understood, but how do such preconditioners perform on discretized parabolic control problems ? We present a convergence analysis for a class of Schwarz methods applied to a model parabolic optimal control problem. We show that just applying a classical Schwarz method in time already implies better transmission conditions than the ones usually used in the elliptic case, and we propose an even better variant based on optimized Schwarz theory.

Parallel in Time simulation of the Navier-Stokes equations using the Finite Finite Element Method

Konstantinos Ioakimidis *Institute of Fluid Mechanics and Hydraulic Machinery, University of Stuttgart, Germany*

The numerical solution of the Navier Stokes Equations (NSE) is an important workload in the present high performance computing systems. It is well known that parallelization by domain decomposition is by now a standard technique in engineering applications. While this approach is able to issue very good parallel scale to a large numbers of cores, it nevertheless saturates at some critical number of processors beyond which the subdomains become too small. Besides that, engineering applications encounter complex fluid flow phenomena like turbulence. For such flows very small time steps and very long simulation times are needed in order to resolve them accurately and to get accurate statistical turbulence quantities. It seems that the Parareal algorithm for parallel-in-time integration is a promising approach to overcome the problems mentioned above. In this work it is attempted to combine the Parareal algorithm with the Finite Element Characteristic Based Split method, which has been first introduced by Zienkiewicz, for solving practical applications in the field of hydropower engineering.

Università della Svizzera italiana
Faculty of Informatics
Institute of Computational Science ICS

JÜLICH
FORSCHUNGSZENTRUM

BERGISCHE UNIVERSITÄT WUPPERTAL

www.fz-juelich.de/ias/jsc/pintws2014