From Spectral Deferred Corrections to the Parallel Full Approximation Scheme in Space and Time

(A summary of recent work)

R. Speck*†, D. Ruprecht†, M. Emmett‡, M. Minion§, M. Bolten¶ R. Krause†,

*Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany.

†Institute of Computational Science, Università della Svizzera italiana, Switzerland.

‡Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, USA.

§Institute for Computational and Mathematical Engineering, Stanford University, USA.

¶Department of Mathematics, Bergische Universität Wuppertal, Germany.

I. Introduction

A recently developed time-parallel method is the "parallel full approximation scheme in space and time" (PFASST) introduced in [1], [2]. It is based on spectral deferred correction methods (SDC) [3], a class of methods that iteratively uses low order methods to obtain an overall method of high order of accuracy. By intertwining the SDC iterations with a Parareal-like iteration (see [4] for Parareal), PFASST features an improved bound on parallel efficiency. The efficacy of PFASST in extreme-scale parallel simulations on a BlueGene/P has been demonstrated in [5].

II. SDC, multi-level SDC and PFASST

Below, first the time-serial single level spectral deferred corrections method is described briefly. Then, the time-serial, multi-level SDC (MLSDC) approach is discussed. Finally, the time-parallel, multi-level PFASST algorithm is sketched.

A. Spectral deferred corrections (SDC)

The SDC method introduced in [3] is an iterative approach to compute a solution of a collocation formula. Given a time-step $[T_n, T_{n+1}]$, denote by $T_n \leq t_0 < \ldots < t_M \leq T_{n+1}$ a set of intermediate Gauß collocation points. Typically, Gauß-Lobatto nodes are used, so that $T_n = t_0$ and $T_{N+1} = t_M$. Integrating an initial value problem from T_n to T_{n+1} is then equivalent to solving the Picard formulation

$$u(t) = u_n + \int_{T_n}^{T_{n+1}} f(u(\tau), \tau)\ d\tau. \qquad (1)$$

Approximating (1) with a quadrature rule with nodes t_m results in a linear or nonlinear system of equations (depending on the problem) to be solved for the coefficients of the collocation polynomial. Instead of solving the full system directly, SDC proceeds iteratively using so-called "sweeps" of a low order integration method, typically forward or backward Euler. For a backward Euler, the sweeps are of the form

$$U_{m+1}^{k+1} = U_m^{k+1} + \Delta t_m \left(f(U_{m+1}^{k+1}) - f(U_{m+1}^{k}) \right) + S_m^k. \qquad (2)$$

Here, the operator S_m^k approximates the Picard integral from t_m to t_{m+1}, that is

$$S_m^k \approx \int_{t_m}^{t_{m+1}} f(u^k(\tau), \tau)\ d\tau. \qquad (3)$$

If the iteration converges, the term $f(U_{m+1}^{k+1}) - f(U_{m+1}^{k})$ vanishes and (2) for $m = 0, \ldots, M-1$ can be combined into

$$U_M^{k+1} = U_0^{k+1} + \sum_{m=0}^{M-1} S_m^k \qquad (4)$$

which is precisely the collocation approximation of (1).

B. Multi-level spectral deferred corrections

In [6], a multi-level SDC method (MLSDC) is presented, that in a certain sense provides the "missing link" between single-level, time-serial SDC and multi-level, time-parallel PFASST. In contrast to SDC, MLSDC performs sweeps not on a single level but on a hierarchy of levels, where higher levels use fewer collocations nodes and therefore a coarsened temporal discretization. A FAS correction is employed in order to ensure information is properly transferred between levels. It is shown in [6] that MLSDC provides the same accuracy as SDC, minimally improved stability and that it can reduce the number of iterations required for convergence. Also, the incorporation of weighting matrices required e.g. for the use of compact finite difference stencils is explained. Here, the SDC sweep equation as well as the FAS correction need to be modified to achieve higher-order discretizations in space.

In order to reduce the computational cost of MLSDC at coarser levels, multiple strategies are presented to also coarsen the spatial discretization on the higher levels of the hierarchy:

- Reduced spatial resolution
- Reduced order discretization
- Reduced implicit solve
- Reduced physics

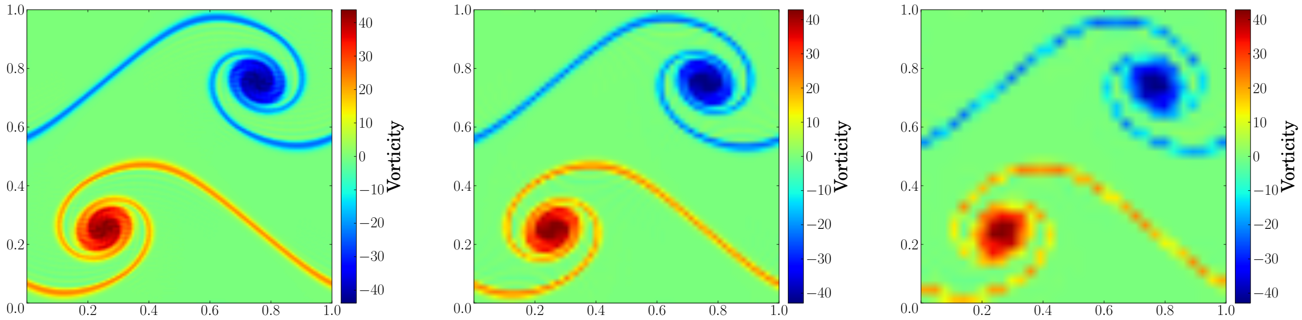
The first two strategies are subsequently investigated in detail for a linear advection-diffusion problem, nonlinear viscous Burgers' equation and a shear layer instability described by the Navier-Stokes equations in vorticity-velocity formulation, see Section III for a tentative summary of the last problem.

One key advantage of MLSDC is that it can be parallelized in time, leading to the "parallel full approximation scheme in space and time" (PFASST) described below in Subsection II-C. Besides providing the possibility to parallelize in time, however, MLSDC is also of interest in its own right: It provides a starting point to extend it to a full space-time multi-grid method and thus enables a completely new approach to the field of space-time parallel multi-grid methods as studied e.g. in [7], [8].

C. PFASST

The PFASST method is pioneered in [1]. In [2], it is introduced in its ultimate form and its performance is studied for viscous Burgers' equation and the Kuramoto-Silvashinsky equation. PFASST basically corresponds to a number of concurrent MLSDC iterations running for multiple time intervals assigned to different processors plus a frequent exchange of updated values. PFASST employs SDC sweeps on multiple levels and uses a FAS correction as well to efficiently transfer information between levels. In particular, the FAS correction also allows for efficient use of different spatial coarsening strategies very similar to the MLSDC approach. Furthermore, to optimize efficiency and reduce overhead from the coarse level sweeps, PFASST features a pipelining strategy, see [1].

Workshop "Innovative space-time-parallel methods: Analysis and Applications"
Manchester, UK, June 18-19, 2013.



(a) Fine level: 128×128 grid points, $M = 9$ collocation points and fourth order spatial discretization.

(b) Medium level: 64×64 grid points, $M = 5$ collocation points and second order spatial discretization.

(c) Coarse level: 32×32 grid points, $M = 3$ collocation points and second order spatial discretization.

Fig. 1: Vorticity ω at $t = 1.0$ in the shear layer instability example on three levels of a MLSDC hierarchy. The coarse level shows clear signs of underresolution like oscillations in the trailing tails of the vortical structures. MLSDC performs 256 time-steps in this example and requires on average 6.9 iterations on each time-step to converge to a tolerance of 10^{-12}.

By intertwining the SDC sweeps on the different levels with the outer iteration, PFASST manages to achieve an improved bound on parallel efficiency compared to plain Parareal. Denote by P_{T} the number of processors, by K_s the number of sweeps of the underlying SDC scheme, by α the ratio of the execution time of one coarse to one fine sweep and by K_{p} the number of PFASST iterations. The speedup provided by PFASST then reads

$$S(P_{\mathrm{T}}, \alpha) = \frac{P_{\mathrm{T}} K_{\mathrm{s}}}{P_{\mathrm{T}} \alpha + K_{\mathrm{p}} (1 + \alpha)} \leq \frac{K_{\mathrm{s}}}{K_{\mathrm{p}}} P_{\mathrm{T}}. \tag{5}$$

Note how strategies to reduce the overhead of coarse level evaluations in MLSDC directly translate into strategies to reduce α and thus improve the speedup provided by PFASST. The potential of PFASST to extend the strong scaling limit of the N-body tree-code PEPC [9] in extreme-scale parallel simulations has been demonstrated in [5], where timing result from runs on up to 262,144 processors on a BlueGene/P system are presented. There, in order to optimize α, a tree-code specific coarsening criterion was developed that roughly corresponds to "reduced order in space". An accuracy study accompanying the performance study and discussing the accuracy of SDC and PFASST in combination with an N-body solver can be found in [10].

III. Example

One example studied in [6] is the performance of MLSDC for a 2D shear layer instability. A tentative summary is given here. The problem is described by the Navier-Stokes equations in vorticity-velocity formulation

$$\omega_t + u \cdot \nabla \omega = \nu \Delta \omega \tag{6}$$

with velocity u and vorticity $\omega = \nabla \times u$. The computational domain $[0,1]^2$ is assumed to be periodic in both directions and the initial velocity field is given by

$$u_1^0(x,y) = -1.0 + \tanh(\rho(0.75 - y)) + \tanh(\rho(y - 0.25)) \tag{7}$$

$$u_2^0(x,y) = -\delta \sin(2\pi(x + 0.25)), \tag{8}$$

corresponding to two shear layers at $y = 0.25$ and $y = 0.75$ with a thickness parameter $\rho = 50$ and an initial disturbance in velocity of amplitude $\delta = 0.05$. MLSDC uses an IMEX-type sweep, where the advection of vorticity is treated explicitly while the diffusive term is treated implicitly. Three levels are used and both the medium and the coarse level feature a reduced resolution in space and reduced order of the spatial discretization in addition to the reduced number of collocation points, see Figure 1 for the exact values. Despite the fact that clear signs of under-resolution are present on the coarse level, the MLSDC iteration converges quickly and robustly and also conserves total vorticity. The Poisson problems arising from the implicit part of the IMEX scheme and during the recovery of the vorticity field from a solenoidal stream function are solved using a the multi-grid method PMG [11]. In the current version, the linear problems are always solved to full accuracy and the strategy of using a "reduced implicit solve" to coarsen in space has not yet been investigated.

IV. Outlook

In the talk, we will provide an overview of the results on MLSDC in [6]. We will also present first results of incorporating the third coarsening strategy, using a "reduced implicit solve" for the implicit part on coarser levels. Further, the connection of MLSDC and PFASST will be illustrated. Studies comparable to the ones conducted for MLSDC will be conducted with PFASST, in order to provide a detailed assessment of the differences between time-serial MLSDC and time-parallel PFASST.

References

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

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

[3] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations," *BIT Numerical Mathematics*, vol. 40, no. 2, pp. 241–266, 2000.

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

[5] R. Speck, D. Ruprecht, R. Krause, M. Emmett, M. Minion, M. Winkel, and P. Gibbon, "A massively space-time parallel N-body solver," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. CA, USA: IEEE Computer Society Press, 2012, pp. 92:1–92:11.

[6] M. Bolten, M. Emmett, R. Krause, M. Minion, D. Ruprecht, and R. Speck, "A multi-level spectral deferred correction method," (In preparation).

[7] W. Hackbusch, "Parabolic multi-grid methods," *Computing Methods in Applied Sciences and Engineering, VI*, pp. 189–197, 1984.

[8] 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.

[9] M. Winkel, R. Speck, H. Hübner, L. Arnold, R. Krause, and P. Gibbon, "A massively parallel, multi-disciplinary Barnes-Hut tree code for extreme-scale N-body simulations," *Comp. Phys. Comm.*, vol. 183, no. 4, pp. 880–889, 2012.

[10] 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 *Proceedings of the 21st International Conference on Domain Decomposition Methods*, ser. Lecture Notes in Computational Science and Engineering, 2012, (Accepted).

[11] M. Bolten, "Highly scalable multigrid solver for 3-level Toeplitz and circulant matrices," *Parallel Computing*, (Submitted).