Metadata of the chapter that will be visualized online

Chapter Title	Inexact Spectral Deferred Corrections	
Copyright Year	2015	
Copyright Holder	Springer International Publishing Switzerland	
Author	Family Name	**Speck**
	Particle	
	Given Name	**Robert**
	Suffix	
	Email	r.speck@fz-juelich.de
Author	Family Name	**Ruprecht**
	Particle	
	Given Name	**Daniel**
	Suffix	
	Email	daniel.ruprecht@usi.ch
Author	Family Name	**Krause**
	Particle	
	Given Name	**Rolf**
	Suffix	
	Email	rolf.krause@usi.ch
Author	Family Name	**Minion**
	Particle	
	Given Name	**Michael**
	Suffix	
	Email	mlminion@gmail.com
Author	Family Name	**Emmett**
	Particle	
	Given Name	**Matthew**
	Suffix	
	Email	mwemmett@lbl.gov

Inexact Spectral Deferred Corrections

Robert Speck, Daniel Ruprecht, Michael Minion, Matthew Emmett, and Rolf Krause

1 Introduction

Implicit integration methods based on collocation are attractive for a number of reasons, e.g. their ideal (for Gauss-Legendre nodes) or near ideal (Gauss-Radau or Gauss-Lobatto nodes) order and stability properties. However, straightforward application of a collocation formula with M nodes to an initial value problem with dimension d requires the solution of one large $Md \times Md$ system of nonlinear equations.

Spectral deferred correction (SDC) methods, introduced by Dutt et al. (2000), are an attractive approach for iteratively computing the solution to the collocation problem using a low-order method (like implicit or IMEX Euler) as a building block. Instead of solving one huge system of size $Md \times Md$, SDC iteratively solves M smaller $d \times d$ systems to approximate the solution of the full system (see also the

R. Speck (✉)
Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany
e-mail: r.speck@fz-juelich.de

R. Speck • D. Ruprecht • R. Krause
Institute of Computational Science, Università della Svizzera italiana, Lugano, Switzerland
e-mail: daniel.ruprecht@usi.ch; rolf.krause@usi.ch

M. Minion
Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA
e-mail: ~~mlminion@gmail.com~~

M. Emmett
Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, CA 94720, USA
e-mail: mwemmett@lbl.gov

R. Krause et al. (eds.), *Domain Decomposition Methods in Science and Engineering XXII*, Lecture Notes in Computational Science and Engineering 104, DOI 10.1007/978-3-319-18827-0_66

discussion in Huang et al. 2006). It has been shown e.g. by Xia et al. (2007) that each iteration/sweep of SDC raises the order by one, so that SDC with k iterations and a first-order base method is of order k, up to the order of the underlying collocation formula. Therefore, to achieve formal order p, SDC requires $p/2$ nodes and p iterations and thus $p^2/2$ solves of a $d \times d$ system (for Gauss-Legendre nodes).

Considering the number of solves required to achieve a certain order, one might conclude that, notwithstanding the results presented here, SDC is less efficient than e.g. diagonally implicit Runge Kutta (DIRK) methods, see e.g. Alexander (1977), which only require $p-1$ solves. However, the flexibility of the choice of the base propagator in SDC and the very favorable stability properties make it an attractive method nevertheless. In particular, semi-implicit methods of high order can easily be constructed with SDC which make it competitive for complex applications, see Minion (2003) and Bourlioux et al. (2003). Further extensions to SDC allow it to integrate processes with different time scales, see Layton and Minion (2004); Bouzarth and Minion (2010) efficiently; and the iterative nature of SDC also allows it to be extended to a multigrid-like multi-level algorithm, where work is shifted to coarser, computationally cheaper levels, see Speck et al. (2014b).

In the present paper, we introduce another strategy to improve the efficiency of SDC, which is similar to ideas from Oosterlee and Washio (1996) where a single V-cycle of a multigrid method is used as a preconditioner. We show here that the iterative nature of SDC allows us to use incomplete solves of the linear systems arising in each sweep. In the resulting *inexact spectral deferred corrections* (ISDC), the linear problem in each Euler step is solved only approximately using a small number (two in the examples presented here) of multigrid V-cycles. It is numerically shown that this strategy results in only a small increase of the number of required sweeps while reducing the cost for each sweep. We demonstrate that ISDC can provide a significant reduction of the overall number of multigrid V-cycles required to complete an SDC time step.

2 Semi-Implicit Spectral Deferred Corrections

We consider an initial value problem in Picard form

$$u(t) = u_0 + \int_{T_0}^{t} f(u(s))\, \mathrm{d}s \tag{1}$$

where $t \in [T_0, T]$ and $u, f(u) \in \mathbb{R}^N$. Subdividing a time interval $[T_n, T_{n+1}]$ into M intermediate substeps $T_n = t_0 \leq t_1 < \ldots < t_M \leq T_{n+1}$, the integrals from t_m to t_{m+1} can be approximated by

$$I_m^{m+1} = \int_{t_m}^{t_{m+1}} f(u(s))\, \mathrm{d}s \approx \Delta t \sum_{j=0}^{M} s_{m,j} f(u_j) = S_m^{m+1} F(u) \tag{2}$$

where $u_m \approx u(t_m)$, $m = 0, 1, \ldots, M$, $F(u) = (f(u_1), \ldots, f(u_M))^T$, $\Delta t = T_{n+1} - T_n$, and $s_{m,j}$ are quadrature weights. The nodes t_m correspond to quadrature nodes of a spectral collocation rule like Gauss-Legendre or Gauss-Lobatto quadrature rule. The basic implicit SDC update formula at node $m + 1$ in iteration $k + 1$ can be written as

$$u_{m+1}^{k+1} = u_m^{k+1} + \Delta t_m \left[f(u_{m+1}^{k+1}) - f(u_{m+1}^k) \right] + S_m^{m+1} F(u^k), \tag{3}$$

where $\Delta t_m = t_{m+1} - t_m$, for $m = 0, \ldots, M - 1$. Alternatively, if f can be split into a stiff part f^I and a non-stiff part f^E, a semi-implicit update is easily constructed for SDC using

$$\begin{aligned} u_{m+1}^{k+1} = u_m^{k+1} &+ \Delta t_m \left[f^I(u_{m+1}^{k+1}) - f^I(u_{m+1}^k) \right] \\ &+ \Delta t_m \left[f^E(u_m^{k+1}) - f^E(u_m^k) \right] + S_m^{m+1} F(u^k). \end{aligned} \tag{4}$$

Here, only the f^I-part is treated implicitly, while f^E is explicit. We refer to Minion (2003) for the details on semi-implicit spectral deferred corrections.

3 Inexact Spectral Deferred Corrections

In the following, we consider the linearly implicit case $f^I(u) = Au$, where A is a discretization of the Laplacian operator. Here, spatial multigrid is a natural choice for solving the implicit part in (4). As in Speck et al. (2014b), we use a high-order compact finite difference stencil to discretize the Laplacian (see e.g. Spotz and Carey 1996). This results in a weighting matrix W for the right-hand side of the implicit system and, with the notation $\tilde{f}^I(u) = Wf^I(u)$, the semi-implicit SDC update (4) becomes

$$\begin{aligned} (W - \Delta t_m A) u_{m+1}^{k+1} = W u_m^{k+1} &+ \Delta t_m W \left[f^E(u_m^{k+1}) - f^E(u_m^k) \right] \\ &- \Delta t_m \tilde{f}^I(u_{m+1}^k) + S_m^{m+1} \tilde{F}(u^k), \end{aligned} \tag{5}$$

where $\tilde{f} = Wf^E + \tilde{f}^I$ and $\tilde{F}(u^k) = (\tilde{f}(u_1^k), \ldots, \tilde{f}(u_M^k))^T$. Thus, instead of inverting the operator $I - \Delta t_m A$ in (4), the right-hand side of (4) is modified by W and the operator $W - \Delta t_m A$ needs to be inverted. We note that for calculating the residual during the SDC iteration, the weighting matrix needs to be inverted once per node, which can be done using multigrid as well.

For classical SDC, each computation of u_{m+1}^{k+1} includes a full inversion of $W - \Delta t_m A$ using e.g. a multigrid solver in space. For K iterations and M nodes, the multigrid solver is executed $K(M - 1)$ times, each time until a predefined tolerance is reached. In order to reduce the overall number of required multigrid V-cycles, ISDC replaces this full solve with a small fixed number L of V-cycles, leading to an accumulated number of $\tilde{K}(M-1)L$ V-cycles in total. Naturally, the number of iterations in ISDC will be larger than the number of SDC, that is $K \leq \tilde{K}$. However, if $\tilde{K}$ is small enough so that $\tilde{K}(M-1)L$ is below the total number of multigrid V-cycles required for $K(M-1)$ full multigrid solves, inexact SDC will be more efficient than classical SDC.

Convergence is monitored using the maximum norm of the SDC residual, a discrete analogue of $u^k(t) - u_0 - \int_{T_0}^{t} f(u^k(s))\,\mathrm{d}s$, that measures how well our iterative solution satisfies the discrete collocation problem. See Speck et al. (2014b) for definition and details. In the tests below, sweeps are performed until the SDC residual is below a set threshold.

4 Numerical Tests

In order to illustrate the performance of ISDC, we consider two different numerical examples, the 2D diffusion equation and 2D viscous Burgers' equation. As described above, in both cases the diffusion term is discretized using a 4th-order compact stencil with weighting matrix and a spatial mesh with 64 points. For Burgers' equation, the advection term is discretized using a fifth order WENO scheme.

4.1 Setups

The first test problem is the 2D heat equation on the unit square, namely

$$u_t(\mathbf{x},t) = \nu \Delta u(\mathbf{x},t), \quad \mathbf{x} \in \Omega = (0,1)^2 \tag{6}$$

$$u(\mathbf{x},0) = \sin(\pi x)\sin(\pi y) \tag{7}$$

$$u(\mathbf{x},t) = 0 \text{ on } \partial\Omega \tag{8}$$

with $\mathbf{x} = (x,y)$. The exact solution is $u(x,t) = \exp(-2\pi^2 \nu t)\sin(\pi x)\sin(\pi y)$. An implicit Euler is used here as base method in SDC.

The second test problem is the nonlinear viscous Burgers' equation

$$u_t(\mathbf{x},t) + u(\mathbf{x},t)u_x(\mathbf{x},t) + u(\mathbf{x},t)u_y(\mathbf{x},t) = \nu \Delta u(\mathbf{x},t), \mathbf{x} \in (-1,1)^2, \tag{9}$$

$$u(x,0) = \exp\left(-\frac{x^2}{\sigma^2}\right), \quad \sigma = 0.1 \tag{10}$$

with periodic boundary conditions. Here, an IMEX Euler is used as base method, i.e. the Laplacian is treated implicitly, while the advection term is integrated explicitly.

Table 1 Accumulated multigrid V-cycles for (a) the heat equation and (b) the viscous Burgers' equation with different values for the diffusion coefficients ν and the number of quadrature nodes M

(a) Heat equation					(b) Viscous Burgers' equation				
ν	M	SDC	ISDC	Savings (%)	ν	M	SDC	ISDC	Savings (%)
1	3	16(4)	12(4)	25	10^{-1}	3	21(8)	21(8)	0
	5	23(3)	20(3)	13		5	26(6)	26(6)	0
	7	32(3)	28(3)	13		7	33(5)	33(5)	0
10	3	36(5)	20(5)	44	1.0	3	97(17)	66(17)	32
	5	61(5)	40(5)	34		5	140(17)	117(17)	16
	7	79(4)	47(4)	41		7	160(15)	143(15)	11
100	3	106(13)	52(13)	51	10	3	207(25)	100(25)	52
	5	150(10)	104(13)	31		5	523(38)	298(38)	43
	7	187(9)	167(14)	11		7	902(50)	578(50)	36

Cycles are accumulated over all sweeps required to reduce the SDC or ISDC residual below 5×10^{-8}. The number of deferred correction sweeps is shown in parentheses. Saving indicates the amount of V-cycles saved by ISDC in percent of the cycles required by SDC

In both examples the diffusion parameter ν controls the stiffness of the term f^I: for a given spatial resolution, the shifted Laplacian $W - \nu \Delta t A$, and therefore the performance of the multigrid solver, depends critically on ν. We choose three different values of ν for each example to measure the impact of stiffness on the performance of ISDC: $\nu = 1, 10, 100$ for the heat equation and $\nu = 0.1, 1, 10$ for Burgers' equation. For ISDC, each implicit solve is approximated using $L = 2$ V-cycles. A single time-step of length $\Delta t = 0.001$ is analyzed for a spatial discretization with $\Delta x = \Delta y = 1/64$ in both cases, leading to CFL numbers for the diffusive term of approximately 4.1, 41 and 410 for the heat equation and 0.41, 4.1 and 41 for Burgers' equation.

4.2 *Results*

Table 1 shows the total number of multigrid V-cycles for the heat equation (left) and for Burgers' equation (right) for three different numbers of collocation nodes M and different values of ν. The number of SDC or ISDC sweeps performed is shown in parentheses. In each case, sweeps are performed until the SDC or ISDC residual is below 5×10^{-8}. To simplify the analysis in the presence of the weighting matrix, the V-cycles required to invert the weighting matrix are not counted here. In the last row, the amount of V-cycles saved by ISDC is given in percent of the required SDC cycles.

In most cases, ISDC provides a substantial reduction of the total number of required multigrid V-cycles and requires only slightly more sweeps to converge than SDC. The most savings can be obtained if the number of multigrid V-cycles in SDC is high but ISDC does not lead to a significant increase in sweeps, which is the case for mildly stiff problems (e.g. $\nu = 10$ for heat equation and $\nu = 1$ and $\nu = 10$ for Burgers) or stiff problems with small values for M. For stiff problems with large M (e.g. heat equation with $\nu = 100$ and $M = 7$), however, ISDC leads to a more significant increase in required sweeps, therefore only resulting in small savings. For the non-stiff cases, particularly for Burger's equation, ISDC does not provide much benefit, but also does no harm: the multigrid solves in SDC take only very few V-cycles to converge, so that SDC and ISDC are almost identical (for Burgers with $\nu = 0.1$, SDC and ISDC are actually identical). In a sense, for simple problems where the stopping criterion of the multigrid solver is reached after one or two V-cycle anyhow, SDC automatically reduces to ISDC.

In summary, the tests presented here suggest that replacing full multigrid solves by a small number of V-cycles in SDC only leads to a small increase in the total number of SDC sweeps required for convergence but can significantly reduce the computational cost of each sweep. The savings in the overall number of multigrid V-cycles of ISDC directly translates into faster run times of ISDC runs compared to classical SDC. Preliminary numerical tests not document here suggest that, as long as the approximate solution of the linear system is sufficiently accurate, the order of ISDC still increases by each iteration, as shown for SDC in Xia et al. (2007). A detailed study confirming this, including a possible extension of the proof, is left for future work.

4.3 Interpretation

The good performance of ISDC in the examples presented above is mainly due to the choice of the starting values for the multigrid solver. When performing the implicit Euler step to compute u_{m+1}^{k+1}, the value u_{m+1}^{k} from the previous SDC sweep gives a very good starting value, particularly in later sweeps. Therefore, even two multigrid V-cycles are sufficient to approximate the real solution of the linear system of equations reasonably well. This effect can be observed by monitoring the number of V-cycles in classical SDC. During the first sweep, many more V-cycles are typically required for multigrid to converge than in later sweeps where the initial guess becomes very accurate as the SDC iterations converge. In fact, during the last sweeps of SDC, a single V-cycle is often sufficient for solving the implicit system. Hence, the additional sweeps required by ISDC are mainly due to the less accurate approximations during the first sweeps. As soon as the initial guess u_{m+1}^{k} for u_{m+1}^{k+1} is good enough, ISDC basically proceeds like SDC. A computational experiment that confirms this is as follows: if, when solving for u_{m+1}^{k+1}, we replace the initial guess with the zero vector, or even u_{m}^{k+1}, then ISDC fails to converge altogether. On the other hand, SDC still convergences in this scenario, but the number of required multigrid V-cycles increases dramatically.

It is important to contrast this behavior to non-iterative schemes like diagonally implicit Runge-Kutta, where usually only the value from the previous time step or stage is available to be used as starting value. Our experience with SDC methods suggests that more multigrid V-cycles would be required to solve each stage in a DIRK scheme than in later SDC iterations. Hence, simply counting the number of implicit function evaluations required could be a misleading way to compare the cost of SDC and DIRK schemes.

5 Conclusion and Outlook

The paper presents a variant of spectral deferred corrections called *inexact spectral deferred corrections* that can significantly reduce the computational cost of SDC. In ISDC, full spatial solves within SDC sweeps with an implicit or semi-implicit Euler are replaced by only a few V-cycles of a multigrid. In the two investigated examples, ISDC saves up to 52 % of the total multigrid V-cycles required by SDC with full linear solves in each step, while only minimally increasing the number of sweeps required to reduce the SDC residual below some set tolerance. The main reason for the good performance of ISDC is that the iterative nature of SDC provides very accurate initial guesses for the multigrid solver. Besides providing significant speedup, ISDC essentially removes the need to define a tolerance or maximum number of iterations for the spatial solver.

A natural extension of the work presented in this paper is the application of ISDC sweeps in MLSDC, the multi-level version of SDC. MLSDC performs SDC sweeps in a multigrid-like way on multiple levels. The levels are connected through an FAS correction term in forming the coarsened spatial representation of the problem on upper levels of the hierarchy. Using ISDC corresponds to the "reduced implicit solve" strategy mentioned in Speck et al. (2014b) and incorporating it into MLSDC could further improve its performance. Finally, the "parallel full approximation scheme in space and time" (PFASST, see Minion 2010; Emmett and Minion 2012, 2014 for details) performs SDC sweeps on multiple levels combined with a forward transfer of updated initial values in a manner similar to Parareal (see Lions et al. 2001). Instead of performing a full time integration as done in Parareal, PFASST interweaves SDC sweeps with Parareal iterations so that on each time level, only a single SDC sweep is performed (i.e. an inexact time integrator is applied), leading to a time-parallel method with good parallel efficiency (see e.g. Speck et al. 2014a and Ruprecht et al. 2013). Integrating ISDC into PFASST could further improve its parallel efficiency.

Acknowledgement This work was supported by the Swiss National Science Foundation (SNSF) under the lead agency agreement through the project "ExaSolvers" within the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA) of the Deutsche Forschungsgemeinschaft (DFG). Matthew Emmett and Michael Minion were supported by the Applied Mathematics Program of the DOE Office of Advanced Scientific Computing Research under the U.S. Department of Energy under contract DE-AC02-05CH11231. Michael Minion was also supported by the U.S. National Science Foundation grant DMS-1217080. The authors acknowledge support from Matthias Bolten, who provided the employed multigrid solver.

References

R. Alexander, Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. SIAM J. Numer. Anal. **14**(6), 1006–1021 (1977)

A. Bourlioux, A. Layton, M. Minion, High-order multi-implicit spectral deferred correction methods for problems of reactive flow. J. Comput. Phys. **189**(2), 651–675 (2003)

E. Bouzarth, M. Minion, A multirate time integrator for regularized stokeslets. J. Comput. Phys. **229**(11), 4208–4224 (2010)

A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations. BIT Numer. Math. **40**(2), 241–266 (2000)

M. Emmett, M. Minion, Toward an efficient parallel in time method for partial differential equations. Commun. Appl. Math. Comput. Sci. **7**, 105–132 (2012)

M. Emmett, M.L. Minion, *Efficient Implementation of a Multi-Level Parallel in Time Algorithm*. Lecture Notes in Computational Science and Engineering, vol. 98 (Springer, Switzerland, 2014), pp. 359–366

J. Huang, J. Jia, M. Minion, Accelerating the convergence of spectral deferred correction methods. J. Comput. Phys. **214**(2), 633–656 (2006)

A. Layton, M. Minion, Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics. J. Comput. Phys. **194**(2), 697–715 (2004)

J.-L. Lions, Y. Maday, G. Turinici, A "parareal" in time discretization of PDE's. C. R. l'Académie Sci. Math. **332**, 661–668 (2001)

M. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations. Commun. Math. Sci. **1**(3), 471–500 (2003)

M. Minion, A hybrid parareal spectral deferred corrections method. Commun. Appl. Math. Comput. Sci. **5**(2), 265–301 (2010)

C. Oosterlee, T. Washio, On the use of multigrid as a preconditioner, in *Proceedings of 9th International Conference on Domain Decomposition Methods*, pp. 441–448 (1996)

D. Ruprecht, R. Speck, M. Emmett, M. Bolten, R. Krause, Poster: Extreme-scale space-time parallelism, in *Proceedings of the 2013 Supercomuting Companion, SC '13 Companion, 2013*

R. Speck, D. Ruprecht, M. Emmett, M. Bolten, R. Krause. A space-time parallel solver for the three-dimensional heat equation. in *Parallel Computing: Accelerating Computational Science and Engineering (CSE)*, vol. 25 (2014a), pp. 263–272

R. Speck, D. Ruprecht, M. Emmett, M. Minion, M. Bolten, R. Krause, A multi-level spectral deferred correction method. BIT Numer. Math. (2014b)

W. Spotz, G. Carey, A high-order compact formulation for the 3D Poisson equation. Numer. Meth. PDEs **12**(2), 235–243 (1996)

Y. Xia, Y. Xu, C.-W. Shu, Efficient time discretization for local discontinuous Galerkin methods. Disc. Cont. Dyn. Syst. **8**(3), 677–693 (2007)

AUTHOR QUERIES

AQ1. Please check the presentation of Table 1.
AQ2. Please check the publisher details are appropriate for Ref. Emmett and Minion (2014).
AQ3. Please provide Publisher Details for Ref.‘Speck et al. (2014a).