SIAM J. SCI. COMPUT.
Vol. 46, No. 3, pp. A1690–A1713

SPECTRAL DEFERRED CORRECTION METHODS FOR SECOND-ORDER PROBLEMS[*]

IKROM AKRAMOV[†], SEBASTIAN GÖTSCHEL[†], MICHAEL MINION[‡], DANIEL RUPRECHT[†], AND ROBERT SPECK[§]

Abstract. Spectral deferred corrections (SDC) are a class of iterative methods for the numerical solution of ordinary differential equations. SDC can be interpreted as a Picard iteration to solve a fully implicit collocation problem, preconditioned with a low-order method. It has been widely studied for first-order problems, using explicit, implicit, or implicit-explicit Euler and other low-order methods as preconditioner. For first-order problems, SDC achieves arbitrary order of accuracy and possesses good stability properties. While numerical results for SDC applied to the second-order Lorentz equations exist, no theoretical results are available for SDC applied to second-order problems. We present an analysis of the convergence and stability properties of SDC using velocity-Verlet as the base method for general second-order initial value problems. Our analysis proves that the order of convergence depends on whether the force in the system depends on the velocity. We also demonstrate that the SDC iteration is stable under certain conditions. Finally, we show that SDC can be computationally more efficient than a simple Picard iteration or a fourth-order Runge–Kutta–Nyström method.

Key words. spectral deferred corrections (SDC), Picard iteration, collocation method, velocity-Verlet, stability, convergence

MSC codes. 65L07, 65L05, 65L70

DOI. 10.1137/23M1592596

See reproducibility of computational results at end of the article.

1. Introduction.

Many problems in science and engineering can be modeled using Newton's second law, giving rise to initial value problems of the form

$$\ddot{x} = f(t, x(t), \dot{x}(t)),\ x(t_0) = x_0,\ \dot{x}(t_0) = \dot{x}_0, \tag{1.1}$$

where $x : \mathbb{R} \to \mathbb{R}^d$, $f : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, and $t_0 \le t \le t_{\text{end}}$. Only for very simple problems is it possible to find solutions analytically. In most cases, numerical time stepping algorithms must be used to generate approximate solutions. A straightforward approach is to rewrite (1.1) as

$$\dot{x}(t) = v(t), \tag{1.2a}$$

$$\dot{v}(t) = f(t, x(t), v(t)) \tag{1.2b}$$

and apply standard methods for first-order ODEs like Runge–Kutta or multistep methods. This, however, means treating both the equation for the position $x(t)$ and $v(t)$ in the same way and can forfeit opportunities to improve method performance.

For the harmonic oscillator with $f(t, x(t), v(t)) = -x(t)$, for example, explicit Euler is unconditionally unstable while implicit Euler leads to heavy numerical

[*] Submitted to the journal's Numerical Algorithms for Scientific Computing section August 8, 2023; accepted for publication (in revised form) February 8, 2024; published electronically May 14, 2024.
https://doi.org/10.1137/23M1592596

[†] Lehrstuhl Computational Mathematics, Technische Universität Hamburg, Hamburg, Germany (ikrom.akramov@tuhh.de, sebastian.goetschel@tuhh.de, ruprecht@tuhh.de).

[‡] Lawrence Berkeley National Lab, Berkeley, CA 94720 USA (mlminion@lbl.gov).

[§] Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Jülich, Germany (r.speck@fz-juelich.de).

damping. By contrast, the symplectic Euler method, which integrates (1.2a) explicitly and (1.2b) implicitly (although no implicit solver is required if f does not depend on v), is conditionally stable and energy conserving [15]. A generalization of this approach are Runge–Kutta–Nystrom (RKN) methods that use different Butcher tables for position and velocity [14, sect. II.2]. Derivation of higher-order RKN methods leads to a quickly growing number of order conditions [14, sect. III.3.2]. A widely studied special case [3, 19, 34, 35] is where $f(t,x(t),v(t)) = f(x(t))$, that is, the right hand side depends only on the position but not on the velocity. This greatly simplifies order conditions, allowing for easy construction of high-order methods with favorable properties [14, sect. III.2.3]. For the general case, however, constructing good high-order methods remains a challenge and RKN often struggle to outperform standard Runge-Kutta methods designed for first-order problems [2].

Spectral deferred corrections (SDC), introduced in 2000 by Dutt, Greengard, and Rokhlin [11], provide an easy way to construct high-order methods for first-order problems. There is a significant amount of theory [10, 13, 17], a number of algorithmic improvements [18, 22, 23], and studies of its performance in complex applications [5, 26]. For second-order problems, only a special variant based on the Boris integrator [4] has been proposed, which is specifically tailored to the Lorentz equations modeling trajectories of charged particles in electromagnetic fields [37]. This Boris-SDC method has been improved [33], used to compute fast ion trajectories in fusion reactors [32], and studied for other plasma physics problems [27]. However, no attempts have been made to adopt SDC for second-order problems other than the Lorentz system and, unlike the first-order case, no theoretical foundation exists.

This paper fills this gap by providing a systematic study of the mathematical properties of second-order SDC, including a proof of consistency and an assessment of stability. It studies convergence and demonstrates that SDC can compete with an RKN-4 method in terms of computational efficiency. Section 2 describes the SDC method for second-order IVPs using a velocity-Verlet integrator as base method. Section 3 investigates stability, using the damped harmonic oscillator as a test problem. The related issues of stability and convergence of the SDC iteration are discussed and stability domains of SDC are compared against an RKN-4 method and a collocation method using Picard iterations. Section 4 proves consistency and that each iteration increases the order by two in the case where f does not depend on v but only by one if it does. The theoretical statements on convergence order are validated against numerical examples. Finally, section 5 compares the computational efficiency of SDC against Picard iteration and RKN-4. All the numerical examples were produced with the project `Second_orderSDC` in the pySDC software [29], which is publicly available [30].

2. Spectral deferred corrections for second-order problems.

For the sake of notational simplicity, we focus on the autonomous case of (1.2) since any nonautonomous problem can be transformed into an equivalent autonomous problem [9, pp. 6–7]. Formulation of second-order SDC as well as notation are based on the description of the Boris-SDC algorithm by Winkel, Speck, and Ruprecht [37].

2.1. Collocation formulation.

Consider (1.2) in integral form,

$$x(t) = x_0 + \int_{t_n}^{t} v(s)ds, \tag{2.1a}$$

$$v(t) = v_0 + \int_{t_n}^{t} f(x(s), v(s))ds, \tag{2.1b}$$

over a time step $[t_n, t_{n+1}]$ with starting values $x_0 \approx x(t_n)$ and $v_0 \approx v(t_n)$. Then, define a set of quadrature nodes

$$t_n \leq \tau_1 < \ldots < \tau_M \leq t_{n+1},$$

with associated weights

$$\Delta t q_{m,j} = \Delta t \int_0^{e_m} l_j(s) ds = \int_{t_n}^{\tau_m} \bar{l}_j(s) ds, \ m, j = 1, \ldots, M,$$

where $\Delta t = t_{n+1} - t_n$, $l_j(s)$, and $\bar{l}_j(s)$, $j = 1, \ldots, M$ are Lagrange polynomials corresponding to the quadrature nodes on the intervals $[0,1]$ and $[t_n, t_{n+1}]$, respectively. By x_j, v_j, f_j we denote numerical approximations to $x(\tau_j)$, $v(\tau_j)$, and $f(x(\tau_j), v(\tau_j))$ [16, pp. 211–214]. Approximating the integrals in (2.1) using quadrature we obtain

$$x_m = x_0 + \Delta t \sum_{j=1}^{M} q_{m,j} v_j, \tag{2.2a}$$

$$v_m = v_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f_j \tag{2.2b}$$

for $m = 1, \ldots, M$. Next, substitute the second equation in (2.2) into the first so that

$$x_m = x_0 + \Delta t \sum_{j=1}^{M} q_{m,j} v_0 + \Delta t^2 \sum_{j=1}^{M} qq_{m,j} f_j, \tag{2.3a}$$

$$v_m = v_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f_j \tag{2.3b}$$

with $qq_{m,j} = \sum_{i=1}^{M} q_{m,i} q_{i,j}$ and $m = 1, \ldots, M$. The x_m, v_m correspond to the stages of a fully implicit RKN method [16, pp. 283–300]. We use Gauss–Legendre nodes throughout this paper, making the collocation method symplectic [14, Theorem 4.2].

Collocation in matrix form. For the purpose of analysis, we will write the M coupled equations (2.3) as a single system. Let $\bar{Q} \in \mathbb{R}^{M \times M}$ have entries $q_{m,j}$ and let

$$\mathbf{V} = (v_0, v_1, \ldots, v_M)^T, \ \mathbf{X} = (x_0, x_1, \ldots, x_M)^T \in \mathbb{R}^{d(M+1)} \tag{2.4}$$

be vectors that contain the approximations at all nodes.[1] With initial conditions

$$\mathbf{X}_0 := (x_0, x_0, \ldots, x_0)^T, \ \mathbf{V}_0 := (v_0, v_0, \ldots, v_0)^T \in \mathbb{R}^{d(M+1)}$$

and $F(\mathbf{X}, \mathbf{V}) = (f_0, f_1, \ldots, f_M)^T \in \mathbb{R}^{d(M+1)}$ denoting the vector that contains the forces at each node, (2.3) can be written compactly as

$$\mathbf{X} = \mathbf{X}_0 + \Delta t \mathbf{Q} \mathbf{V}_0 + \Delta t^2 \mathbf{Q}\mathbf{Q} F(\mathbf{X}, \mathbf{V}), \tag{2.5a}$$

$$\mathbf{V} = \mathbf{V_0} + \Delta t \mathbf{Q} F(\mathbf{X}, \mathbf{V}). \tag{2.5b}$$

[1] We use boldface variables to indicate values that have been aggregated over multiple quadrature nodes. However, note that nonboldface variables can be vectors, too. For example, $v_1 \in \mathbb{R}^d$ is the velocity at the first quadrature node whereas $\mathbf{V} \in \mathbb{R}^{d(M+1)}$ are the velocities at all quadrature nodes.

Here,

$$Q := \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \bar{Q} \end{pmatrix} \in \mathbb{R}^{(M+1)\times(M+1)}$$

with $\mathbf{0}$ being the M-dimensional zero vector, and

$$\mathbf{Q} = Q \otimes \mathbf{I}_d,\ \mathbf{QQ} = (Q \otimes \mathbf{I}_d) \otimes (Q \otimes \mathbf{I}_d) = QQ \otimes \mathbf{I}_d \tag{2.6}$$

with $\mathbf{I}_d$ being the identity matrix of dimension d. Finally, let

$$\mathbf{U} = (\mathbf{X}, \mathbf{V}) = (x_0, \dots, x_M, v_0, \dots, v_M)^T \in \mathbb{R}^{2d(M+1)}.$$

Then, (2.5) can be written as

$$\mathbf{C}_{\text{coll}}\mathbf{U}_0 = \mathbf{U} - \Delta t \mathbf{Q}_{\text{coll}}\mathbf{F}(\mathbf{U}) =: \mathbf{M}_{\text{coll}}(\mathbf{U}) \tag{2.7}$$

with $\mathbf{U}_0 = (x_0, \dots, x_0, v_0, \dots, v_0)^T,\ \mathbf{F}(\mathbf{U}) = (f_0, \dots, f_M, f_0, \dots, f_M)^T \in \mathbb{R}^{2d(M+1)}$, and

$$\mathbf{Q}_{\text{coll}} = \begin{pmatrix} \Delta t \mathbf{QQ} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q} \end{pmatrix},\ \mathbf{C}_{\text{coll}} = \begin{pmatrix} \mathbf{I}_{d(M+1)} & \Delta t \mathbf{Q} \\ \mathbf{O} & \mathbf{I}_{d(M+1)} \end{pmatrix} \in \mathbb{R}^{2d(M+1)\times 2d(M+1)}, \tag{2.8}$$

where $\mathbf{O}$ denotes a $d(M+1) \times d(M+1)$-dimensional matrix with zero entries. Using (2.7) we obtain the collocation problem in operator form,

$$\mathbf{M}_{\text{coll}}(\mathbf{U}) = \mathbf{C}_{\text{coll}}\mathbf{U}_0. \tag{2.9}$$

Once the stages x_m, v_m are known, the approximations at the end of the time step can be computed via

$$x(t_{n+1}) \approx x_{n+1} = x_0 + \Delta t \sum_{m=1}^{M} q_m v_m, \tag{2.10a}$$

$$v(t_{n+1}) \approx v_{n+1} = v_0 + \Delta t \sum_{m=1}^{M} q_m f_m, \tag{2.10b}$$

where

$$q_j = \int_0^1 l_j(s) ds,\ j = 1, \dots, M.$$

Insert (2.3b) into (2.10a) to obtain

$$x_{n+1} = x_0 + \Delta t v_0 + \Delta t^2 \sum_{m=1}^{M} \sum_{i=1}^{M} q_i q_{i,m} f_m, \tag{2.11a}$$

$$v_{n+1} = v_0 + \Delta t \sum_{m=1}^{M} q_m f_m, \tag{2.11b}$$

where we use that $\sum_{m=1}^{M} q_m = 1$ by consistency of the quadrature rule. Equations (2.11) can again be written in vector form

$$x_{n+1} = x_0 + \Delta t \mathbf{q} \mathbf{V}_0 + \Delta t^2 \mathbf{q}\mathbf{Q} F(\mathbf{X}, \mathbf{V}), \tag{2.12a}$$

$$v_{n+1} = v_0 + \Delta t \mathbf{q} F(\mathbf{X}, \mathbf{V}), \tag{2.12b}$$

where $q := (0, q_1, \dots, q_M) \in \mathbb{R}^{1\times(M+1)}$ and $\mathbf{q} := q \otimes \mathbf{I}_d \in \mathbb{R}^{d\times d(M+1)}$. This is the collocation problem for second-order IVPs that our SDC method will solve iteratively.

2.2. Velocity-Verlet scheme. We use velocity-Verlet integration [36] as the low-order base method for the SDC iteration for a second-order IVP. Applying velocity-Verlet to (1.2) with time steps $\tau_0, \dots, \tau_M$ gives

$$x_{m+1} = x_m + \Delta\tau_{m+1}\left(v_m + \frac{\Delta\tau_{m+1}}{2} f_m\right), \tag{2.13a}$$

$$v_{m+1} = v_m + \frac{\Delta\tau_{m+1}}{2}(f_m + f_{m+1}), \tag{2.13b}$$

where $\Delta\tau_{m+1} = \tau_{m+1} - \tau_m,\ m = 0, \dots, M-1$. To obtain a matrix formulation for (2.13a), we convert it into

$$x_{m+1} = x_0 + \sum_{l=1}^{m+1} \Delta\tau_l v_{l-1} + \frac{1}{2}\sum_{l=1}^{m+1} (\Delta\tau_l)^2 f_{l-1}, \tag{2.14a}$$

$$v_{m+1} = v_0 + \frac{1}{2}\sum_{l=1}^{m+1} \Delta\tau_l (f_{l-1} + f_l). \tag{2.14b}$$

These equations can be rearranged into vector form by defining

$$Q_{\mathrm{E}} := \frac{1}{\Delta t}\begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ \Delta\tau_1 & 0 & 0 & \dots & 0 \\ \Delta\tau_1 & \Delta\tau_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \Delta\tau_1 & \Delta\tau_2 & \dots & \Delta\tau_M & 0 \end{pmatrix},\ Q_{\mathrm{I}} := \frac{1}{\Delta t}\begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \Delta\tau_1 & 0 & \dots & 0 \\ 0 & \Delta\tau_1 & \Delta\tau_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \Delta\tau_1 & \Delta\tau_2 & \dots & \Delta\tau_M \end{pmatrix},$$

and

$$Q_{\mathrm{T}} := \frac{1}{2}(Q_{\mathrm{E}} + Q_{\mathrm{I}}) \in \mathbb{R}^{(M+1)\times(M+1)}. \tag{2.15}$$

Then, (2.14) becomes

$$\mathbf{X} = \mathbf{X}_0 + \Delta t \mathbf{Q}_{\mathrm{E}}\mathbf{V} + \frac{\Delta t^2}{2}(\mathbf{Q}_{\mathrm{E}} \circ \mathbf{Q}_{\mathrm{E}})F(\mathbf{X}, \mathbf{V}), \tag{2.16a}$$

$$\mathbf{V} = \mathbf{V}_0 + \Delta t \mathbf{Q}_{\mathrm{T}} F(\mathbf{X}, \mathbf{V}) \tag{2.16b}$$

with $\circ$ denoting the Hadamard product (elementwise product of two matrices). We substitute the expression for $\mathbf{V}$ from (2.16b) into (2.16a) so that

$$\mathbf{X} = \mathbf{X}_0 + \Delta t \mathbf{Q}_{\mathrm{E}}\mathbf{V}_0 + \Delta t^2 \mathbf{Q}_{\mathrm{x}} F(\mathbf{X}, \mathbf{V}),$$

where

$$\mathbf{Q}_{\mathrm{x}} := Q_{\mathrm{x}} \otimes \mathbf{I}_d \text{ with } Q_{\mathrm{x}} := Q_{\mathrm{E}} Q_{\mathrm{T}} + \frac{1}{2}(Q_{\mathrm{E}} \circ Q_{\mathrm{E}}). \tag{2.17}$$

In order to combine both equations into a compact form based on $\mathbf{U}$, we set

$$\mathbf{C}_{\mathrm{vv}} := \begin{pmatrix} \mathbf{I}_{d(M+1)} & \Delta t \mathbf{Q}_{\mathrm{E}} \\ \mathbf{O} & \mathbf{I}_{d(M+1)} \end{pmatrix},\ \mathbf{Q}_{\mathrm{vv}} := \begin{pmatrix} \Delta t \mathbf{Q}_{\mathrm{x}} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}_{\mathrm{T}} \end{pmatrix}.$$

Finally, the matrix representation of the velocity-Verlet scheme is

$$\mathbf{U} = \mathbf{C}_{\mathrm{vv}}\mathbf{U}_0 + \Delta t \mathbf{Q}_{\mathrm{vv}}\mathbf{F}(\mathbf{U})$$

or

$$\mathbf{M}_{\mathrm{vv}}(\mathbf{U}) := \mathbf{U} - \Delta t \mathbf{Q}_{\mathrm{vv}}\mathbf{F}(\mathbf{U}) = \mathbf{C}_{\mathrm{vv}}\mathbf{U}_0. \tag{2.18}$$

We will use $\mathbf{M}_{\mathrm{vv}}(\cdot)$ as preconditioner for the Picard iteration to solve (2.9). As shown below, $\mathbf{M}_{\mathrm{vv}}(\cdot)$ is easy to invert by a "sweep" with velocity-Verlet through all the nodes.

2.3. Spectral deferred corrections. Applying a Richardson iteration [20] to (2.9) gives

$$\mathbf{U}^{k+1} = (\mathbf{I}_{2d(M+1)} - \mathbf{M}_{\mathrm{coll}})(\mathbf{U}^k) + \mathbf{C}_{\mathrm{coll}}\mathbf{U}_0 = \mathbf{C}_{\mathrm{coll}}\mathbf{U}_0 + \Delta t\mathbf{Q}_{\mathrm{coll}}\mathbf{F}(\mathbf{U}^k), \tag{2.19}$$

where $k = 0, \dots, K$ is the iteration index and we use $\mathbf{U}^0$ to start the iteration.

PROPOSITION 2.1. *Let f be a Lipschitz continuous function with Lipschitz constant L and Δt sufficiently small so that $\Delta t L \|\mathbf{Q}_{coll}\| < 1$. Then,* (2.19) *converges to the collocation solution for all starting values $\mathbf{U}_0$.*

Proof. Subtracting (2.19) for $k+1$ and k yields

$$\mathbf{U}^{k+1} - \mathbf{U}^k = \Delta t\mathbf{Q}_{\mathrm{coll}}(\mathbf{F}(\mathbf{U}^k) - \mathbf{F}(\mathbf{U}^{k-1})).$$

Applying a norm and using Lipschitz continuity gives us

$$\|\mathbf{U}^{k+1} - \mathbf{U}^k\| \le \Delta t L\|\mathbf{Q}_{\mathrm{coll}}\|\|\mathbf{U}^k - \mathbf{U}^{k-1}\|.$$

Since $\Delta t L\|\mathbf{Q}_{\mathrm{coll}}\| < 1$, the iteration converges [1, pp. 1–10]. □

Often, the Picard iteration converges only for an impractically small time step. To improve convergence, we use $\mathbf{M}_{\mathrm{vv}}$ as a preconditoner [20], leading to

$$\mathbf{M}_{\mathrm{vv}}(\mathbf{U}^{k+1}) = (\mathbf{M}_{\mathrm{vv}} - \mathbf{M}_{\mathrm{coll}})(\mathbf{U}^k) + \mathbf{C}_{\mathrm{coll}}\mathbf{U}_0. \tag{2.20}$$

Each iteration requires solving a linear or nonlinear system of equations, depending on the right–hand function f in (1.2), to invert $\mathbf{M}_{\mathrm{vv}}$. However, the structure of $\mathbf{M}_{\mathrm{vv}}$ allows it to be done by sweeping through the quadrature nodes seqentially. Using (2.18) and (2.7) we obtain the operator form of the SDC iteration for second-order equations

$$(\mathbf{I}_{2d(M+1)} - \Delta t\mathbf{Q}_{\mathrm{vv}}\mathbf{F})(\mathbf{U}^{k+1}) = \Delta t(\mathbf{Q}_{\mathrm{coll}} - \mathbf{Q}_{\mathrm{vv}})\mathbf{F}(\mathbf{U}^k) + \mathbf{C}_{\mathrm{coll}}\mathbf{U}_0 \tag{2.21}$$

for $k = 0, \dots, K$.

Remark 2.2. Typically, the starting value $\mathbf{U}^0$ for iteration (2.21) will be generated from the initial value $\mathbf{U}_0$ at the beginning of the time step, either by setting $\mathbf{U}^0 = \mathbf{U}_0$ or by an initial sweep of the velocity-Verlet base method to solve

$$\mathbf{M}_{\mathrm{vv}}(\mathbf{U}^0) = \mathbf{C}_{\mathrm{vv}}\mathbf{U}_0. \tag{2.22}$$

Our theoretical analysis in subsection 4.1 does not make any assumptions about how $\mathbf{U}^0$ is generated. Since the aim of the numerical examples in subsection 4.2 is to validate the theory, we always initialize $\mathbf{U}^0$ with random values. Even a simple copy of $\mathbf{U}_0$ was found to lead to convergence orders that are better than what the theory guarantees in some cases. For the comparison of computational efficiency in section 5 we use $\mathbf{U}^0 = \mathbf{U}_0$.

For analysis, it will be helpful to split the equations for position and velocity

$$\mathbf{X}^{k+1} - \Delta t^2\mathbf{Q}_{\mathrm{x}}F(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = \mathbf{X}_0 + \Delta t\mathbf{Q}\mathbf{V}_0 + \Delta t^2(\mathbf{Q}\mathbf{Q} - \mathbf{Q}_{\mathrm{x}})F(\mathbf{X}^k, \mathbf{V}^k), \tag{2.23a}$$

$$\mathbf{V}^{k+1} - \Delta t\mathbf{Q}_{\mathrm{T}}F(\mathbf{X}^{k+1}, \mathbf{V}^{k+1}) = \mathbf{V}_0 + \Delta t(\mathbf{Q} - \mathbf{Q}_{\mathrm{T}})F(\mathbf{X}^k, \mathbf{V}^k). \tag{2.23b}$$

Using definitions (2.15) and (2.17) we obtain the sweep formulation

(2.24a)
$$x_{m+1}^{k+1} = x_0 + \Delta t \sum_{l=0}^{M} q_{m+1,l} v_0 + \Delta t^2 \sum_{l=0}^{M} q^{\mathrm{x}}_{m+1,l} \left(f_l^{k+1} - f_l^k \right) + \Delta t^2 \sum_{l=0}^{M} qq_{m+1,l} f_l^k,$$

(2.24b)
$$v_{m+1}^{k+1} = v_0 + \Delta t \sum_{l=0}^{M} q^{\mathrm{T}}_{m+1,l} \left(f_l^{k+1} - f_l^k \right) + \Delta t \sum_{l=0}^{M} q_{m+1,l} f_l^k,$$

where $m = 0, \ldots, M$ and $k = 0, \ldots, K$, $f_l^k := f(x_l^k, v_l^k)$ and $(q^{\mathrm{x}}_{m,l})_{m,l=0,\ldots,M}$ and $(q^{\mathrm{T}}_{m,l})_{m,l=0,\ldots,M}$ are the entries of Q_{x} and Q_{T}. By taking the difference between (2.24) for $m+1$ and m and exploiting that Q_{T} is lower diagonal and Q_{x} strictly lower diagonal, we get

(2.25a)
$$x_{m+1}^{k+1} = x_m^{k+1} + \Delta\tau_{m+1} v_0 + \Delta t^2 \sum_{l=0}^{m} s^{\mathrm{x}}_{m+1,l} \left(f_l^{k+1} - f_l^k \right) + \Delta t^2 \sum_{l=0}^{M} sq_{m+1,l} f_l^k,$$

(2.25b)
$$v_{m+1}^{k+1} = v_m^{k+1} + \frac{\Delta\tau_{m+1}}{2} \left(f_{m+1}^{k+1} - f_{m+1}^k \right) + \frac{\Delta\tau_{m+1}}{2} \left(f_m^{k+1} - f_m^k \right) + \Delta t \sum_{l=0}^{M} s_{m+1,l} f_l^k,$$

for $m = 0, \ldots, M-1$ and $k = 0, \ldots, K$. Here,

$$s_{m,j} := q_{m,j} - q_{m-1,j}, \; s^{\mathrm{x}}_{m,j} := q^{\mathrm{x}}_{m,l} - q^{\mathrm{x}}_{m-1,l}, \; sq_{m,l} := qq_{m,l} - qq_{m-1,l}$$

with $m, j = 1, \ldots, M$. The factor in front of v_0 is due to $\Delta t \sum_{j=0}^{M} s_{m,j} = \Delta\tau_m$. Since (2.25) is a sweep through the quadrature nodes using a velocity-Verlet method with some additional terms on the right-hand side, implementation is straightforward.

Remark 2.3. If f does not depend on v, (2.25) is a fully explicit SDC iteration.

3. Stability.

Similarly to how the Dahlquist equation is used to study stability for first-order problems, we use the damped harmonic oscillator with unit mass

(3.1a) $$\dot{x}(t) = v(t),$$
(3.1b) $$\dot{v}(t) = f(x(t), v(t)) := -\kappa x(t) - \mu v(t)$$

as the test problem to study the stability of second-order SDC. Here, κ is the spring constant and μ the friction coefficient. Assuming that $\mathbf{M}_{\mathrm{vv}} = \mathbf{I}_{2(M+1)} - \mathbf{Q}_{\mathrm{vv}}\mathbf{F}$ is invertible, iteration (2.21) becomes

(3.2) $$\mathbf{U}^{k+1} = \mathbf{K}_{\mathrm{sdc}} \mathbf{U}^k + (\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\mathrm{vv}} \mathbf{F})^{-1} \mathbf{C}_{\mathrm{coll}} \mathbf{U}_0,$$

where $\mathbf{K}_{\mathrm{sdc}} := (\mathbf{I}_{2(M+1)} - \Delta t \mathbf{Q}_{\mathrm{vv}} \mathbf{F})^{-1} (\Delta t \mathbf{Q}_{\mathrm{coll}} - \Delta t \mathbf{Q}_{\mathrm{vv}}) \mathbf{F}$. For fixed M and choice of quadrature nodes, the iteration matrix $\mathbf{K}_{\mathrm{sdc}}$ depends only on $\Delta t\kappa$, $\Delta t\mu$. The iteration matrix is similar to the one in first-order problems [18].

There are two different but related issues regarding stability of SDC convergence of the SDC iteration for a single time step as $K \to \infty$ and boundedness of the sequence of approximations x_n, v_n generated by subsequent applications of SDC as $n \to \infty$.

Proposition 3.1. *The sequence* $\{\mathbf{U}^k\}$ *generated by* (3.2) *converges for any* $\mathbf{U}^{\mathbf{0}}$ *and starting values* $\mathbf{U}_0$ *if and only if*

$$\rho(\mathbf{K}_{sdc}) := \max_{\lambda\in\operatorname{spec}(\mathbf{K}_{sdc})} |\lambda| < 1.$$

Proof. The proof works along the lines of the proof of [12, Theorem 2.16]. □

If we identify a pair of positive parameters $\Delta t(\kappa,\mu)$ with a point in the positive quadrant $\mathbb{R}^2_+$, we can define the *convergence domain* of SDC as

$$\Omega_{\text{conv}} := \left\{(\Delta t\kappa, \Delta t\mu) \in \mathbb{R}^2_+ : \rho(\mathbf{K}_{\text{sdc}}) < 1\right\}. \tag{3.3}$$

For a set of parameters inside Ω_{conv}, SDC will converge to the solution of the collocation problem (2.2) as $k\to\infty$.

To assess stability as $n\to\infty$, we derive the stability function of SDC. Using induction and (3.2) we find that

$$\mathbf{U}^{k+1} = \mathbf{K}_{\text{sdc}}^{k+1}\mathbf{U}_0 + \sum_{j=0}^{k}\mathbf{K}_{\text{sdc}}^{j}\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}\mathbf{U}_0.$$

Using the geometric series formula we obtain

$$\mathbf{U}^{k+1} = \mathbf{K}_{\text{sdc}}^{k+1}\mathbf{U}_0 + \left(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}^{k+1}\right)\left(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}\right)^{-1}\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}\mathbf{U}_0.$$

This can be slightly simplified to

$$\mathbf{U}^{k+1} = \mathbf{P}_{\text{sdc}}^{k+1}\mathbf{U}_0, \tag{3.4}$$

where

$$\mathbf{P}_{\text{sdc}}^{k} := \mathbf{K}_{\text{sdc}}^{k} + \left(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}^{k}\right)\left(\mathbf{I}_{2(M+1)} - \mathbf{K}_{\text{sdc}}\right)^{-1}\mathbf{M}_{\text{vv}}^{-1}\mathbf{C}_{\text{coll}}.$$

The final quadrature step (2.12) can also be written in matrix form

$$\begin{pmatrix} x_{n+1} \\ v_{n+1}\end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1\end{pmatrix}\begin{pmatrix} x_0 \\ v_0\end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q}\end{pmatrix}\mathbf{F}\mathbf{U}^{k+1}. \tag{3.5}$$

Inserting the expression for $\mathbf{U}^{k+1}$ from (3.4) into (3.5) yields

$$\begin{pmatrix} x_{n+1} \\ v_{n+1}\end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1\end{pmatrix}\begin{pmatrix} x_0 \\ v_0\end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q}\end{pmatrix}\mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\mathbf{U}_0 \tag{3.6}$$

or

$$\begin{pmatrix} x_{n+1} \\ v_{n+1}\end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1\end{pmatrix}\begin{pmatrix} x_0 \\ v_0\end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q}\end{pmatrix}\mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\bar{\mathbf{1}}\begin{pmatrix} x_0 \\ v_0\end{pmatrix}, \tag{3.7}$$

where $\bar{\mathbf{1}} = \left(\begin{smallmatrix}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{smallmatrix}\right) \in \mathbb{R}^{2(M+1)\times 2}$ with $\mathbf{1} = (1,1,\ldots,1)^T \in \mathbb{R}^{M+1}$. A full-step of SDC from t_n to t_{n+1} for the damped harmonic oscillator therefore becomes

$$\begin{pmatrix} x_{n+1} \\ v_{n+1}\end{pmatrix} = \left(\begin{pmatrix} 1 & \Delta t \\ 0 & 1\end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q}\end{pmatrix}\mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\bar{\mathbf{1}}\right)\begin{pmatrix} x_0 \\ v_0\end{pmatrix} \tag{3.8}$$

and the stability function of SDC iteration is

$$R(\Delta t\kappa, \Delta t\mu) = \begin{pmatrix} 1 & \Delta t \\ 0 & 1\end{pmatrix} + \begin{pmatrix} \Delta t^2\mathbf{q}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \Delta t\mathbf{q}\end{pmatrix}\mathbf{F}\mathbf{P}_{\text{sdc}}^{k+1}\bar{\mathbf{1}}. \tag{3.9}$$

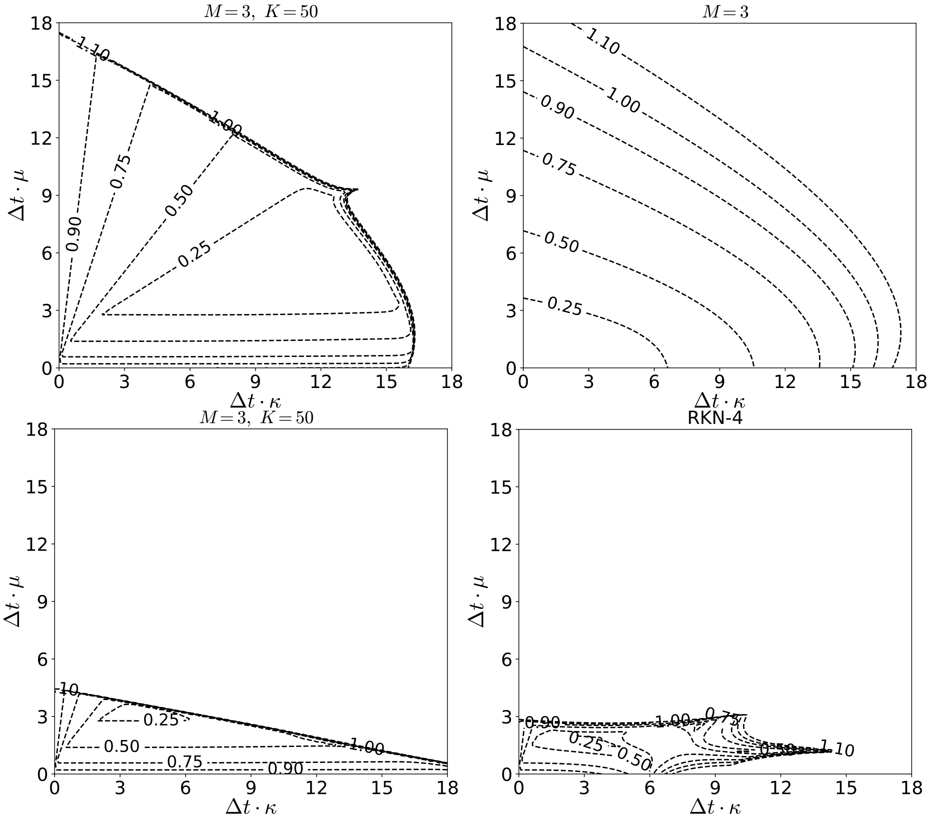

FIG. 1. *Stability domain for $K = 50$ iterations (upper left) and convergence domain (upper right) of SDC with $M = 3$ Gauss–Legendre quadrature nodes. Stability domain of the Picard iteration with $K = 50$ iterations and $M = 3$ nodes (lower left) and stability domain of RKN-4 (lower right).*

Stability in the sense that x_n and v_n remain bounded as $n \to \infty$ is then ensured if $(\Delta t\kappa, \Delta t\mu)$ is an element of the stability domain

$$\Omega_{\text{stab}} := \left\{ (\Delta t\kappa, \Delta t\mu) \in \mathbb{R}^2_+ : \rho(R(\Delta t\kappa, \Delta t\mu)) < 1 \right\}. \tag{3.10}$$

Figure 1 shows the stability domain after $K = 50$ iterations (upper left) and the convergence domain (upper right) for $M = 3$ Gauss–Legendre nodes. Note how the boundaries of the convergence and stability domain coincide. In general, the stability domains grows as M increases. While this is not documented here, readers can generate stability domains for larger values of M using the provided code.

Figure 1 also compares the stability/convergence domains of the Picard iteration (2.19) (lower left) and RKN-4 (lower right) with the SDC iteration (3.2) (upper left) for $K = 50$. For the undamped system with $\mu = 0$, the Picard iteration converges up to around $\Delta t\kappa = 18$ while SDC only converges until $\Delta t\kappa = 16$, although neither method will provide accuracy for such a low resolution. However, once damping is added to the system SDC converges for a much larger range of parameters. In particular, SDC converges for the stiff case with very strong damping while Picard does not. The stability domains changes when the number or type of quadrature nodes changes. The reader can use the provided code to generate stability domains for other choices.

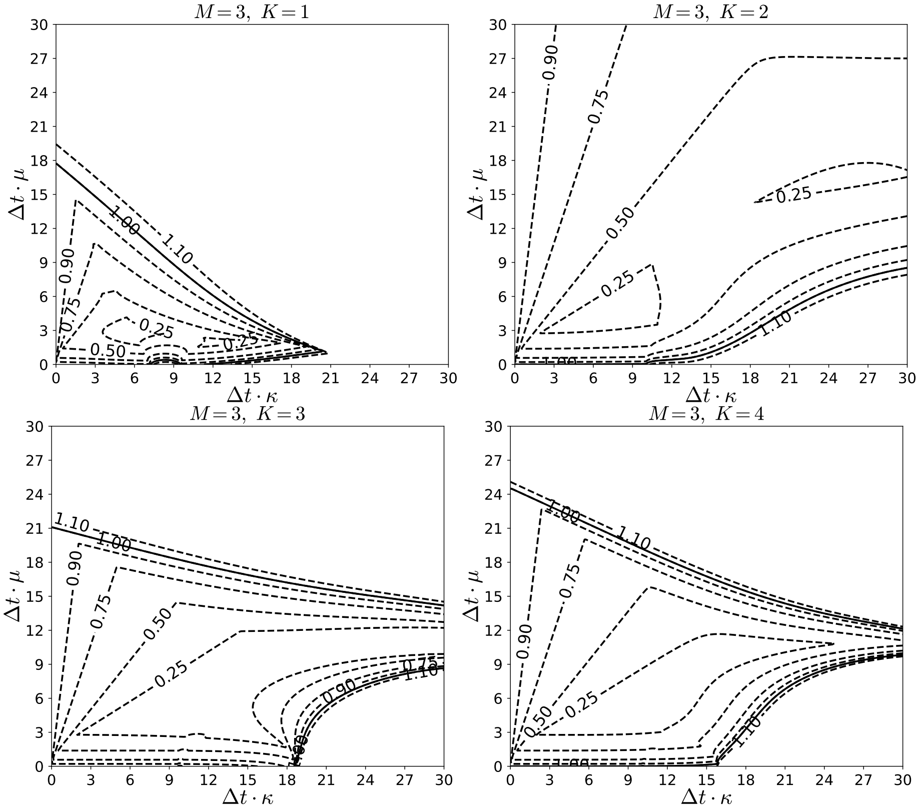

FIG. 2. *Stability domains of SDC with $M = 3$ Gauss–Legendre nodes and $K = 1, 2, 3, 4$ iterations.*

Figure 2 illustrates how the stability domain of SDC changes with the number of iterations K. For $K = 1$, the stability domain is noticeably smaller than the convergence domain. Surprisingly, for $K = 2$, the whole shown range of parameters becomes stable—at the moment, we have no theoretical explanation. A preliminary parameter search suggests that Lobatto nodes in particular often produce methods that remain stable for extremely strong damping (up to $\Delta t\mu \approx 100$) but we were not able to identify a robust heuristic for this behavior. For $K = 3$ iterations, parts of the shown parameter range are unstable again but the stability domain is still significantly larger than for $K = 1$. Increasing to $K = 4$ iterations increase the stability domain into the direction of stronger damping but slightly decreases it in the direction of a larger spring constant. However, in all cases the stability domain of SDC is much larger than that of Picard or RKN-4.

Stability for the purely oscillatory case. Table 1 shows the maximum stable values for SDC and Picard iteration for $\Delta t\kappa$ along the x-axis, that is, for the purely oscillatory case with no damping ($\mu = 0$). Choosing an even number of iterations K seems to be a poor choice for purely oscillatory systems as both methods are either unstable or have very restrictive stability limits. By contrast, if K is odd, both methods are stable for very large steps with SDC allowing even larger stable time steps than Picard iterations. At the moment, we cannot offer a hypothesis as to what causes this very different behavior for odd and even K.

Table 1

Stability limit for $\Delta t\kappa$ for $\mu = 0$ (purely oscillatory case with no damping) rounded to the first digit for SDC and Picard iteration (in brackets).

	SDC (Picard)				
K	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$
1	6.0 (4.7)	7.2 (4.7)	7.8 (4.7)	8.4 (4.7)	8.6 (4.7)
2	0.0 (12.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
3	0.0 (0.0)	9.6 (7.1)	26.5 (4.0)	35.3 (4.0)	55.1 (4.0)
4	11.6 (7.0)	0.2 (0.1)	0.4 (0.2)	0.4 (0.2)	0.6 (0.2)

4. Consistency and convergence order.

We state and prove our main theoretical result on the convergence rate of SDC for second-order problems in section 4.1 and then validate the theory against numerical examples for the special case of a single charged particle in a Penning trap in section 4.2.

4.1. Theory.

The strategy we use to prove the theorem below follows approaches used for the first-order case [8, 21].

Theorem 4.1. *Consider the initial value problem* (1.2) *with a Lipschitz continuous function f with Lipschitz constant L. Let p denote the order of the quadrature rule, and assume that $f \circ (x, v) \in \mathbf{C}^p([t_n, t_{n+1}])$ and that there exists a positive constant G such that $\|\frac{d^p}{dt^p}(f \circ (x, v))\| \le G$. Let $(x(t_{n+1}),\ v(t_{n+1}))$ be the exact solutions to* (1.2) *and $(x^k_{n+1},\ v^k_{n+1})$ be the approximate solutions to* (2.3) *provided by SDC after k iterations. If the step size Δt is sufficiently small, then*

$$|x(t_{n+1}) - x^k_{n+1}| = \mathcal{O}\left(G\Delta t^{p+1}\right) + \mathcal{O}\left(L^{k+1}\Delta t^{k+k_0+2}\right), \tag{4.1a}$$

$$|v(t_{n+1}) - v^k_{n+1}| = \mathcal{O}\left(G\Delta t^{p+1}\right) + \mathcal{O}\left(L^{k+1}\Delta t^{k+k_0+1}\right), \tag{4.1b}$$

where k_0 denotes the approximation order of the base method used to generate $\mathbf{U}^0$; see Remark 2.2. *Moreover, if f is independent of v, we have*

$$|x(t_{n+1}) - x^k_{n+1}| = \mathcal{O}\left(G\Delta t^{p+1}\right) + \mathcal{O}\left(L^{k+1}\Delta t^{2k+k_0+2}\right), \tag{4.2a}$$

$$|v(t_{n+1}) - v^k_{n+1}| = \mathcal{O}\left(G\Delta t^{p+1}\right) + \mathcal{O}\left(L^{k+1}\Delta t^{2k+k_0+1}\right). \tag{4.2b}$$

Proof. We substitute (2.5) into (2.12b) to find the updates (x_{n+1}, v_{n+1}) for the collocation method. Additionally, we determine the SDC method update formula (x^k_{n+1}, v^k_{n+1}) from (2.23) by plugging it into (2.5), subtract, and use the Cauchy-Schwarz inequality [31, pp. 171–177] and Lipschitz continuity to get

$$|v_{n+1} - v^k_{n+1}| = \Delta t|\mathbf{q}(F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^k, \mathbf{V}^k))| \tag{4.3}$$

$$\le \Delta t\|\mathbf{q}\|\|F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^k, \mathbf{V}^k)\| \tag{4.4}$$

$$\le \Delta t L\|\mathbf{q}\|(\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|). \tag{4.5}$$

Using that $\|q\| \le 1$ [25] and Theorem A.3 we find that

$$|v_{n+1} - v^k_{n+1}| \le \Delta t L\|\mathbf{q}\|(\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|) \tag{4.6}$$

$$\le \Delta t L^{k+1}\left(\tilde{C}_1\Delta t^{k+k_0+1} + \tilde{C}_2\Delta t^{k+k_0}\right) \tag{4.7}$$

$$\le L^{k+1}\left(\tilde{C}_1 + \tilde{C}_2\Delta t\right)\Delta t^{k+k_0+1}. \tag{4.8}$$

The entries of the qQ satisfy [16, pp. 208–210]

$$\sum_{i=0}^{M} q_i q_{i,j} = q_j(1-\tau_j),\ \ j=0,\ldots,M. \tag{4.9}$$

Because $\tau_j \leq 1$ for all $j=1,\ldots,M$ on the unit interval, it holds that

$$\|\mathbf{qQ}\| = \max_{j=1,\ldots,M} |q_j(1-\tau_j)| \leq 1. \tag{4.10}$$

Plugging (2.5) and (2.23) into (2.12a), subtracting, and using the Cauchy–Schwarz inequality and Lipschitz continuity gives

$$\begin{aligned} |x_{n+1} - x^k_{n+1}| &= \Delta t^2 |\mathbf{qQ}(F(\mathbf{X},\mathbf{V}) - F(\mathbf{X}^k,\mathbf{V}^k))| && (4.11)\\ &\leq L\Delta t^2 \|\mathbf{qQ}\|(\|\mathbf{X}-\mathbf{X}^k\| + \|\mathbf{V}-\mathbf{V}^k\|). && (4.12)\end{aligned}$$

Using Theorem A.3 yields

$$\begin{aligned} |x_{n+1} - x^k_{n+1}| &\leq \Delta t^2 L \|\mathbf{qQ}\|(\|\mathbf{X}-\mathbf{X}^k\| + \|\mathbf{V}-\mathbf{V}^k\|) && (4.13)\\ &\leq \Delta t^2 L^{k+1}\left(\tilde{C_1}\Delta t^{k+k_0+1} + \tilde{C_2}\Delta t^{k+k_0}\right) && (4.14)\\ &\leq L^{k+1}(\tilde{C_2} + \tilde{C_1}\Delta t)\Delta t^{k+k_0+2}. && (4.15)\end{aligned}$$

Assuming that $\Delta t \leq 1$, we have

$$\tilde{C_2} + \tilde{C_1}\Delta t \leq \tilde{C_2} + \tilde{C_1} =: C_L$$

and therefore

$$\begin{aligned} \left|x_{n+1} - x^k_{n+1}\right| &\leq C_L L^{k+1}\Delta t^{k+k_0+2}, && (4.16a)\\ \left|v_{n+1} - v^k_{n+1}\right| &\leq C_L L^{k+1}\Delta t^{k+k_0+1}. && (4.16b)\end{aligned}$$

Gauss quadrature nodes satisfy the orthogonality condition

$$\int_0^1 s^{j-1}\prod_{i=1}^{M}(s-\tau_i)ds = 0, \quad j=1,2,\ldots,\xi$$

[16, Theorem 7.9]. Thus, the following estimates hold,

$$\begin{aligned} |x(t_{n+1}) - x_{n+1}| &\leq C_G G\Delta t^{p+1}, && (4.17a)\\ |v(t_{n+1}) - v_{n+1}| &\leq C_G G\Delta t^{p+1}, && (4.17b)\end{aligned}$$

where $p = M+\xi$ and C_G is a constant. We have $\xi = M$ and $p = 2M$ for Legendre nodes, $\xi = M-1$ and $p = 2M-1$ for Radau nodes, and $\xi = M-2$ and $p = 2M-2$ for Lobatto nodes. Subtracting the analytical solution from the SDC solution at time t_{n+1} and using the triangle inequality along with (4.16) and (4.17) gives the bound

$$\begin{aligned} |x(t_{n+1}) - x^k_{n+1}| &\leq |x(t_{n+1}) - x_{n+1}| + |x_{n+1} - x^k_{n+1}|\\ &\leq C_G G\Delta t^{p+1} + C_L L^{k+1}\Delta t^{k+k_0+2} \end{aligned} \tag{4.18}$$

for the position error and the bound

$$\begin{aligned} |v(t_{n+1}) - v^k_{n+1}| &\leq |v(t_{n+1}) - v_{n+1}| + |v_{n+1} - v^k_{n+1}|\\ &\leq C_G G\Delta t^{p+1} + C_L L^{k+1}\Delta t^{k+k_0+1} \end{aligned} \tag{4.19}$$

for the velocity error. In summary, the local error of second-order SDC satisfies

$$|x(t_{n+1}) - x^k_{n+1}| = \mathcal{O}\left(G\Delta t^{p+1}\right) + \mathcal{O}\left(L^{k+1}\Delta t^{k+k_0+2}\right), \tag{4.20a}$$

$$|v(t_{n+1}) - v^k_{n+1}| = \mathcal{O}\left(G\Delta t^{p+1}\right) + \mathcal{O}\left(L^{k+1}\Delta t^{k+k_0+1}\right). \tag{4.20b}$$

When f is independent of v, (4.3) and (4.11) become

$$|x_{n+1} - x^k_{n+1}| = \Delta t^2 |\mathbf{qQ}(F(\mathbf{X}) - F(\mathbf{X}^k))| \le L\Delta t^2 \|\mathbf{qQ}\| \|\mathbf{X} - \mathbf{X}^k\| \le L\Delta t^2 \|\mathbf{X} - \mathbf{X}^k\|$$

and

$$|v_{n+1} - v^k_{n+1}| = \Delta t |\mathbf{q}(F(\mathbf{X}) - F(\mathbf{X}^k))| \le L\Delta t \|\mathbf{q}\| \|\mathbf{X} - \mathbf{X}^k\| \le L\Delta t \|\mathbf{X} - \mathbf{X}^k\|.$$

Using the triangle inequality, (4.17), and Theorem A.3 yields

$$|x(t_{n+1}) - x^k_{n+1}| \le |x(t_{n+1}) - x_{n+1}| + |x_{n+1} - x^k_{n+1}| \tag{4.21}$$

$$\le C_G G\Delta t^{p+1} + C_L L^{k+1}\Delta t^{2k+k_0+2} \tag{4.22}$$

and

$$|v(t_{n+1}) - v^k_{n+1}| \le |v(t_{n+1}) - v_{n+1}| + |v_{n+1} - v^k_{n+1}| \tag{4.23}$$

$$\le C_G G\Delta t^{p+1} + C_L L^{k+1}\Delta t^{2k+k_0+1}. \tag{4.24}$$

Thus we obtain

$$|x(t_{n+1}) - x^k_{n+1}| = \mathcal{O}(G\Delta t^{p+1}) + \mathcal{O}(L^{k+1}\Delta t^{2k+k_0+2}), \tag{4.25a}$$

$$|v(t_{n+1}) - v^k_{n+1}| = \mathcal{O}(G\Delta t^{p+1}) + \mathcal{O}(L^{k+1}\Delta t^{2k+k_0+1}). \tag{4.25b}$$

□

A direct consequence of Theorem 4.1 and [28, Definition 2.1] is the following.

THEOREM 4.2. *Let the right-hand side function f in* (1.2) *satisfy the assumptions of Theorem* 4.1. *Then, the global convergence rate of SDC is $p^* := \min\{p, k + k_0\}$. If f does not depend on v, we have $p^* = \min\{p, 2k + k_0\}$.*

4.2. Numerical examples.

We validate our convergence analysis for the Penning trap benchmark [24]. The equations of motion are the Lorentz equations

$$\dot{x}(t) = v(t), \tag{4.26a}$$

$$\dot{v}(t) = f(x(t), v(t)) := \alpha[\mathbf{E}(x(t)) + v(t) \times \mathbf{B}(x(t))] \tag{4.26b}$$

with a constant magnetic field $\mathbf{B} = \frac{\omega_B}{\alpha} \cdot \hat{\mathbf{e}}_z$ along the z-axis with frequency ω_B. Let $\alpha = \frac{q}{m}$ denote that particle's charge-to-mass ratio so that

$$v \times \mathbf{B} = \frac{\omega_B}{\alpha} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} v. \tag{4.27}$$

The electric field with frequency ω_E is given by

$$\mathbf{E}(x) = -\epsilon \frac{\omega_E^2}{\alpha} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix} x. \tag{4.28}$$

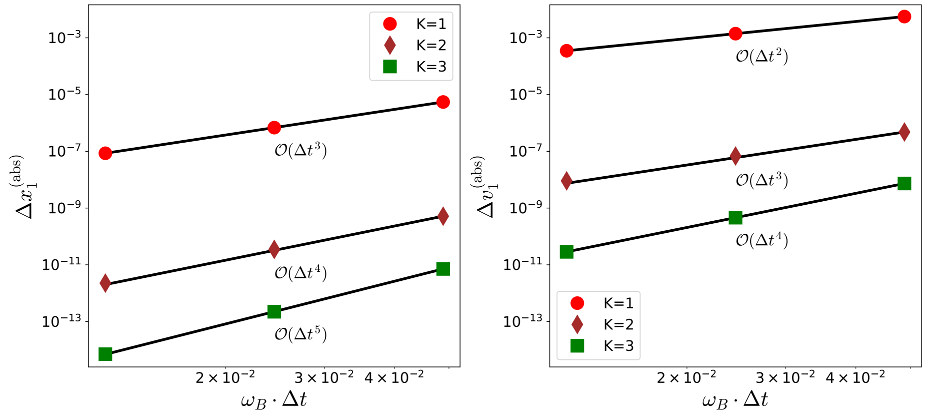

FIG. 3. *Absolute local error $\Delta x_1^{(\mathrm{abs})}$ in the first component of the particle's position (left) and velocity (right) using $K = 1, 2, 3$ SDC iterations and $M = 5$. Note: color appears only in the online article.*

We use the same parameter as Winkel, Speck, and Ruprecht [37, Table 1]. For (4.26) with magnetic field (4.27) and electric field (4.28) and a single particle inside the Penning trap, an analytic solution can be found [6]. Note that because of the zero row in the matrix in (4.27), the force along the third component is independent of the velocity, while the forces along the first or second component are not. By looking at the error in the first and third component separately, we will confirm below the different convergence orders that our theory predicts for these cases.

4.2.1. Local error. Figure 3 shows the local position (left) and velocity (right) of SDC along the first axis error against the time step Δt scaled with the frequency of the magnetic field. The local error is computed by taking the difference $\Delta x_i^{(\mathrm{abs})} := |x_i^{(\mathrm{approx})} - x_i^{(\mathrm{analyt})}|$ between numerical and analytic solution after a single time step. The index $i = 1, 2, 3$ indicates the two horizontal and one vertical axes. In line with our theoretical predictions, the order of the local error increases by one for every iteration and the order of the local error in the position is always one higher than the order of the local error in the velocity.

Figure 4 shows the local error for position and velocity in the third component where the force is independent of v for SDC using $M = 5$ Gauss–Legendre quadrature nodes. As predicted by Theorem 4.1, the one order difference between position and velocity error remains, but the order of the local error increases by two orders per iteration.

4.3. Global error. Figure 5 shows the relative global error in the v_1-component (left) and v_3-component (right) of the velocity for $M = 3$ Gauss–Legendre nodes with fixed final time $t_{\mathrm{end}} = 2$. Since the velocity depends on the position because of the inhomogeneous magnetic and electric field, the global error will have the order of the lower local order of the velocity. In line with Theorem 4.2 we see that in the v_1-direction every iteration increases the global convergence order by one. By contrast, in the v_3-direction, where the force is independent of the velocity, every iteration increases the global order by two.

Table 2 shows measured convergence rates rounded to two digits for $M = 2, 3, 4$ nodes and $K = 1, 2, 3$ and $K = 10$ iterations. The theoretically predicted convergence rates according to Theorem 4.2 are shown in brackets. The left table shows the error in the x_1-component and the right table the error in the x_3-component. In line

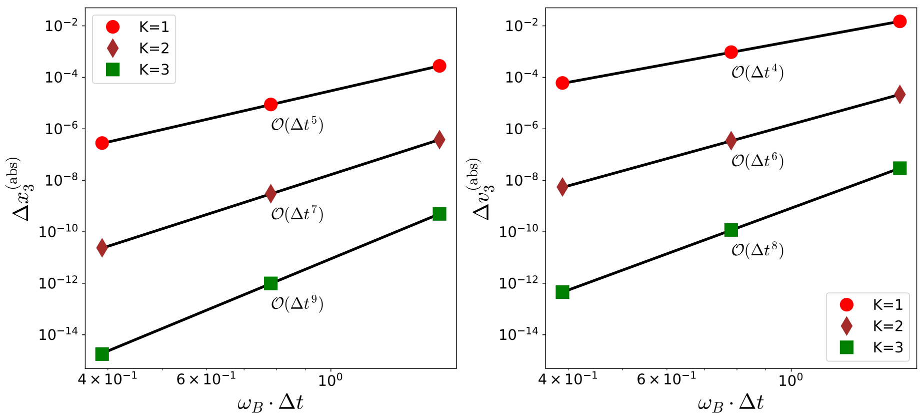

FIG. 4. *Absolute local error $\Delta x_3^{(\mathrm{abs})}$ in the third component of the particle's position (left) and velocity (right) using one, two, and three SDC iterations and* 5 *quadrature nodes. In line with Theorem* 4.1*, the order increases by two per iteration. Note: color appears only in the online article.*

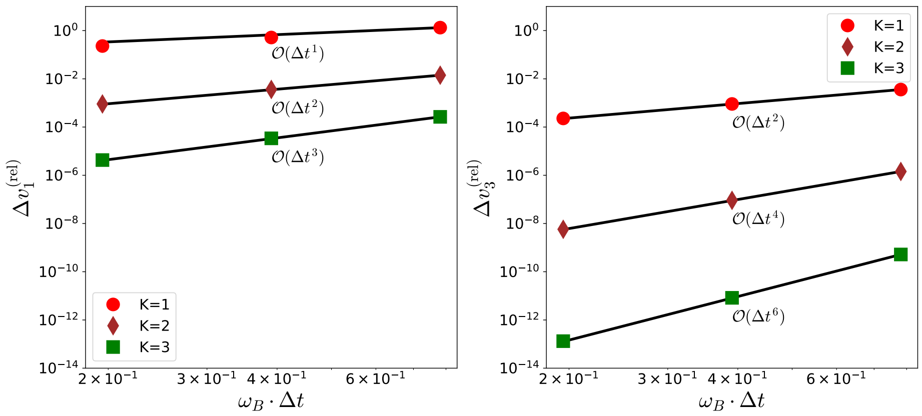

FIG. 5. *Relative global error $\Delta v_i^{(\mathrm{rel})}$, $i=1,3$ in the first component of the particle's horizontal (left) and vertical (right) velocity in the Penning trap versus time step size for* 3 *Gauss–Legendre collocation nodes using one, two, and three SDC iterations. Note: color appears only in the online article.*

with theory, the order increases by one per iteration in the former and by two per iteration in the latter case. For $K=10$ iterations, the order is governed by the order of the underlying quadrature rule and is therefore the same in both first and third components.

4.4. Conservation properties. Many second-order problems are Hamiltonian systems for which conservation properties of the time integrator are important. We consider the undamped harmonic oscillator (3.1) with $\mu = 0.0$ and $\kappa = 1.0$ and a resulting oscillation frequency of $\omega = 1.0$. The continuous Hamiltonian $H(t) = \frac{1}{2}\left(x(t)^2 + v(t)^2\right)$ is constant so that $H(t) = H(0)$. Figure 6 shows the relative error $|H_n - H_0|/H_0$ in the discrete Hamiltonian $H_n = \frac{1}{2}\left(x_n^2 + v_n^2\right)$ for a time step of $\Delta t = 2\pi/10$ until $t_{\mathrm{end}} = 1\times 10^6$ for a total of 1,591,551 steps for RKN and SDC with $M=3$ and $M=5$ Gauss–Legendre quadrature nodes and $K=2,3,4$ iterations. Since the collocation method is symplectic, we expect bounded long term error for large K. However, already for $K=2$ second-order SDC shows no discernable drift.

Table 2

Measured convergence rate rounded to two digits followed by convergence rate predicted by Theorem 4.1 in brackets for different numbers of Gauss–Legendre quadrature nodes.

Horizontal axis			
K	$M=2$	$M=3$	$M=4$
1	1.28(1)	1.30(1)	1.61(1)
2	1.99(2)	1.99(2)	2.14(2)
3	2.99(3)	2.99(3)	2.98(3)
10	3.99(4)	5.99(6)	7.77(8)

Vertical axis			
K	$M=2$	$M=3$	$M=4$
1	1.99(2)	2.00(2)	1.99(2)
2	4.00(4)	3.99(4)	3.98(4)
3	3.99(4)	5.96(6)	5.97(6)
10	3.99(4)	5.99(6)	7.91(8)

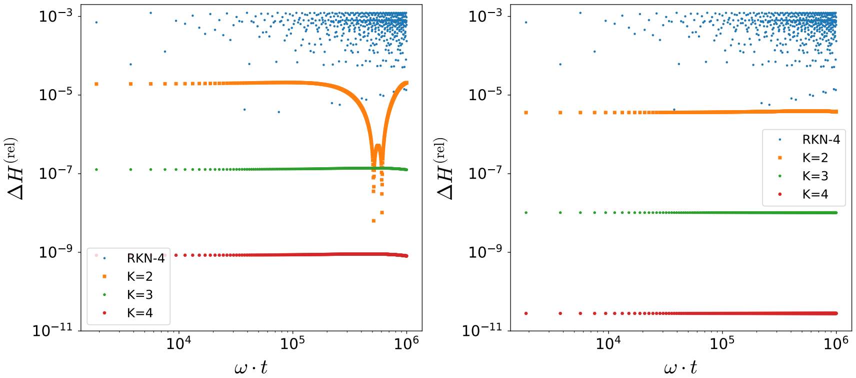

Fig. 6. *Relative error in the discrete Hamiltonian for the undamped harmonic oscillator over* 1.5 *million time steps for $M = 3$ (left) and $M = 5$ (right). Note: color appears only in the online article.*

Furthermore, the relative error in the Hamiltonian from SDC is smaller than from RKN-4 and decreases by about two orders of magnitude per iteration. This is in line with previous findings for the Lorentz equations that showed low energy errors and little to no drift for SDC, even for very long simulation times [33, 37].

5. Computational efficiency. SDC requires more function evaluation per time step than the Picard iteration or an RKN method. However, for the same Δt, it will produce a smaller error. This allows SDC to achieve accuracy comparable to Picard or RKN-4 with a larger time step.

For a fair comparison in terms of efficiency, Figure 7 shows the relative error in the first component (left) and third component (right) of the position for the Penning trap benchmark for SDC, Picard iteration, and RKN-4 against the total number of f evaluations required. Note that the different iteration numbers for SDC in the left plot ($K = 2, 4, 6$) and the right plot ($K = 1, 2, 3$) are chosen to achieve the same global convergence rates in both cases. In all cases, SDC is more efficient than Picard using the same number of iterations. The advantage of SDC is more pronounced for the case where the force does depend on velocity. Furthermore, with sufficiently many iterations, the increasing order of SDC allows it to eventually outperform RKN-4. For the error in the third component, $K = 3$ iterations are enough for SDC to become more efficient than RKN-4 while in the first component it requires $K = 4$ iterations. While not shown, the provided code can also generate work-precision results for the velocity-Verlet integrator. For the error in the first component, we found it to be marginally less efficient than SDC with $k = 2$ iterations while for the error in the third component was slightly more efficient than SDC with $k = 1$ iteration. In both

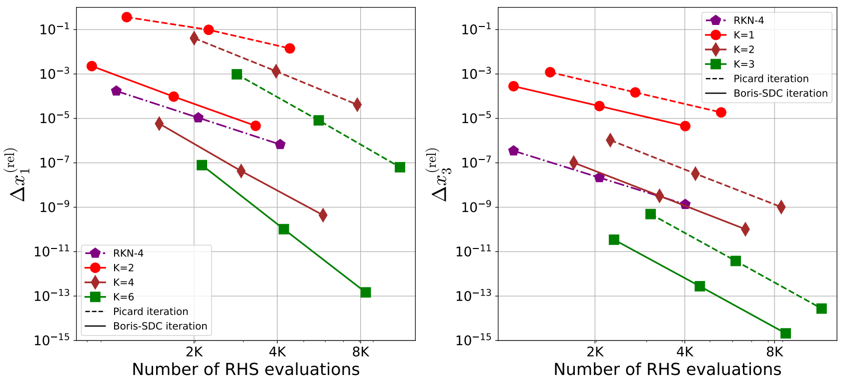

FIG. 7. *Relative position error in the x_1-direction (left) and x_3-direction (right) for SDC (solid lines), Picard (dashed lines) with different iteration numbers K with $M = 5$ quadrature nodes and RKN-4 against the total number of f evaluations. Note: color appears only in the online article.*

cases, the higher-order variants of SDC are significantly more efficient than velocity-Verlet (not shown).

6. Conclusions. We provide a theoretical analysis of spectral deferred corrections applied to second-order problems. Using the damped harmonic oscillator as a test problem, similarly to how the Dahlquist equation is used for first-order problems, we investigate convergence and stability of SDC compared against a collocation method using Picard iteration and an RKN-4 method. The main theoretical result of the paper is a proof that every SDC iteration increases the global convergence order by one for problems where the force depends on the velocity and by two if the force is independent of the velocity. We also show that the order of the local position error is one higher than the order of the local velocity error. Our theoretical predictions are validated against numerical examples for the Penning trap benchmark. We compare SDC against Picard and RKN-4 with respect to work precision and find it to be competitive for medium to high accuracies.

Appendix A. Auxiliary results. This appendix collects a number of technical results we need for the proof of the main convergence results in section 4.

PROPOSITION A.1. *For the matrices introduced in* (2.5)*,* (2.15)*, and* (2.17)*, we have the following bounds*

$$\|Q_{\mathrm{T}}\| \leq 1, \qquad \|Q_{\mathrm{x}}\| \leq \frac{3}{2}, \tag{A.1a}$$

$$\|Q\| \leq 1, \qquad \|QQ\| \leq 1. \tag{A.1b}$$

Proof. From Ruprecht and Speck [25, Lemma 3.1] we know that

$$\|Q_{\mathrm{I}}\| \leq 1, \ \|Q_{\mathrm{E}}\| \leq 1. \tag{A.2}$$

Furthermore, it holds that

$$\|Q_{\mathrm{E}} \circ Q_{\mathrm{E}}\| \leq \Delta\tau_1^2 + \cdots + \Delta\tau_M^2 \leq \Delta\tau_1 + \cdots + \Delta\tau_M \leq 1.$$

Then,

$$\|Q_{\mathrm{T}}\| \leq \frac{1}{2}(\|Q_{\mathrm{E}}\| + \|Q_{\mathrm{I}}\|) \leq 1$$

and

$$\|Q_{\mathrm{x}}\| \leq \|Q_{\mathrm{E}}\| \, \|Q_{\mathrm{T}}\| + \frac{1}{2} \|Q_{\mathrm{E}}\| \leq \frac{3}{2}.$$

The bounds for the norm of the Q matrix were proven by Caklovic [7]. Furthermore, we have $\|QQ\| \leq \|Q\| \|Q\| \leq 1$ which completes the proof. □

PROPOSITION A.2. *Let f be a Lipschitz continuous function with Lipschitz constant L and $(\mathbf{X}, \mathbf{V})$ be the solution to the collocation problem* (2.5)*. Let $(\mathbf{X}^k, \mathbf{V}^k)$ be approximations provided by the SDC iteration* (2.23)*. Suppose Δt satisfies*

$$\Delta t \leq (1-\delta)/L \text{ and } \Delta t^2 < \frac{1}{3} \tag{A.3}$$

for some positive number $0 < \delta < 1$. Then, the following hold,

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_1 L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^k\|), \tag{A.4a}$$

$$\|\mathbf{V} - \mathbf{V}^k\| \leq C_2 L \Delta t (\|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^k\|) \tag{A.4b}$$

with constants C_1, C_2 independent of Δt. If f does not depend on v we have

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_1 L \Delta t^2 \|\mathbf{X} - \mathbf{X}^{k-1}\|, \tag{A.5a}$$

$$\|\mathbf{V} - \mathbf{V}^k\| \leq 2 L \Delta t (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{X} - \mathbf{X}^k\|). \tag{A.5b}$$

Proof. To prove (A.4a), subtract (2.23a) from (2.5a) to get

$$\begin{aligned}\mathbf{X} - \mathbf{X}^k = {} & \Delta t^2 \mathbf{QQ}(F(\mathbf{X}, \mathbf{V}) - F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1})) + \Delta t^2 \mathbf{Q}_{\mathrm{x}} (F(\mathbf{X}^{k-1}, \mathbf{V}^{k-1}) \\ & - F(\mathbf{X}^k, \mathbf{V}^k)).\end{aligned}$$

Using the triangle inequality and Lipschitz continuity we have

$$\begin{aligned}\|\mathbf{X} - \mathbf{X}^k\| \leq {} & \|\mathbf{QQ}\| L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) \\ & + \|\mathbf{Q}_{\mathrm{x}}\| L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) \\ & + \|\mathbf{Q}_{\mathrm{x}}\| L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|).\end{aligned}$$

Since $\|\mathbf{Q}_{\mathrm{x}}\| \leq \frac{3}{2}$ and $\|\mathbf{QQ}\| \leq 1$ by Proposition A.1, we can simplify to

$$\begin{aligned}\|\mathbf{X} - \mathbf{X}^k\| \leq {} & \left(1 + \frac{3}{2}\right) L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) \\ & + \frac{3}{2} L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^k\| + \|\mathbf{V} - \mathbf{V}^k\|).\end{aligned}$$

Because of $\Delta t^2 < \frac{1}{3}$ we have $1 - \frac{3}{2} L \Delta t^2 \geq 1 - L \Delta t \geq \delta > 0$ and thus

$$\|\mathbf{X} - \mathbf{X}^k\| \leq \frac{5 L \Delta t^2}{2 - 3 L \Delta t^2} (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\|) + \frac{3 L \Delta t^2}{2 - 3 L \Delta t^2} \|\mathbf{V} - \mathbf{V}^k\|.$$

Since $1 - \frac{3}{2} L \Delta t^2 > \delta$ for $\Delta t^2 < \frac{1}{3}$, this yields

$$\frac{5}{2 - 3 \Delta t^2 L} \leq \frac{5}{2\delta} =: C_1. \tag{A.6}$$

Hence,

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_1 L \Delta t^2 (\|\mathbf{X} - \mathbf{X}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^{k-1}\| + \|\mathbf{V} - \mathbf{V}^k\|). \tag{A.7}$$

We can prove (A.4b) in a similar way. Subtract (2.5b) from (2.23b) to get

$$\mathbf{V}-\mathbf{V}^k=\Delta t\mathbf{Q}(F(\mathbf{X},\mathbf{V})-F(\mathbf{X}^{k-1},\mathbf{V}^{k-1}))+\Delta t\mathbf{Q}_{\mathrm{T}}(F(\mathbf{X}^{k-1},\mathbf{V}^{k-1})-F(\mathbf{X}^k,\mathbf{V}^k)).$$

Then,

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\|\le\Delta t\|\mathbf{Q}\|\|F(\mathbf{X},\mathbf{V})-F(\mathbf{X}^{k-1},\mathbf{V}^{k-1})\|\\+\Delta t\|\mathbf{Q}_{\mathrm{T}}\|\|F(\mathbf{X}^{k-1},\mathbf{V}^{k-1})-F(\mathbf{X}^k,\mathbf{V}^k)\|.\end{aligned}$$

By using that F is Lipschitz continuous we obtain

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\|&\le\|\mathbf{Q}\|L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|)\\&\quad+\|\mathbf{Q}_{\mathrm{T}}\|L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^k\|+\|\mathbf{V}-\mathbf{V}^k\|).\end{aligned}$$

Since $\|\mathbf{Q}_{\mathrm{T}}\|\le 1$ and $\|\mathbf{Q}\|\le 1$ by Proposition A.1, it follows that

$$\|\mathbf{V}-\mathbf{V}^k\|\le 2L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|)+L\Delta t(\|\mathbf{X}-\mathbf{X}^k\|+\|\mathbf{V}-\mathbf{V}^k\|).$$

Since, $1-L\Delta t\ge\delta>0$, we get

$$\|\mathbf{V}-\mathbf{V}^k\|\le\frac{2L\Delta t}{1-L\Delta t}(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|)+\frac{L\Delta t}{1-L\Delta t}(\|\mathbf{X}-\mathbf{X}^k\|).$$

Because $1-L\Delta t\ge\delta>0$, we have

$$\frac{2}{1-L\Delta t}\le\frac{2}{\delta}=:C_2 \tag{A.8}$$

and

$$\|\mathbf{V}-\mathbf{V}^k\|\le C_2L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^k\|). \tag{A.9}$$

Let f be independent of v, i.e., $F(\mathbf{X},\mathbf{V})=F(\mathbf{X})$. Subtracting (2.5) from (2.23) yields

$$\mathbf{X}-\mathbf{X}^k=\Delta t^2\mathbf{Q}\mathbf{Q}(F(\mathbf{X})-F(\mathbf{X}^{k-1}))+\Delta t^2\mathbf{Q}_{\mathrm{x}}(F(\mathbf{X}^{k-1})-F(\mathbf{X}^k)), \tag{A.10a}$$

$$\mathbf{V}-\mathbf{V}^k=\Delta t\mathbf{Q}(F(\mathbf{X})-F(\mathbf{X}^{k-1}))+\Delta t\mathbf{Q}_{\mathrm{T}}(F(\mathbf{X}^{k-1})-F(\mathbf{X}^k)). \tag{A.10b}$$

Using similar arguments as above, we obtain

$$\|\mathbf{X}-\mathbf{X}^k\|\le C_1L\Delta t^2\|\mathbf{X}-\mathbf{X}^{k-1}\|, \tag{A.11a}$$

$$\|\mathbf{V}-\mathbf{V}^k\|\le 2L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^k\|). \tag{A.11b}$$

□

The following theorem provides the error bound for SDC at the quadrature nodes.

THEOREM A.3. *Consider the initial value problem* (1.2) *and let f be Lipschitz continuous with Lipschitz constant L. If the step size Δt is sufficiently small, we have*

$$\|\mathbf{X}-\mathbf{X}^k\| \le \tilde{C}_1 L^k \Delta t^{k+k_0+1}, \tag{A.12a}$$
$$\|\mathbf{V}-\mathbf{V}^k\| \le \tilde{C}_2 L^k \Delta t^{k+k_0} \tag{A.12b}$$

with constants $\tilde{C}_1$, $\tilde{C}_2$ independent of Δt, and k_0 the order of the procedure used to generate the starting value $\mathbf{U}^0$ for the SDC iteration; see Remark 2.2. If f is independent of v, we have

$$\|\mathbf{X}-\mathbf{X}^k\| \le \hat{C}_1 L^k \Delta t^{2k+k_0}, \tag{A.13a}$$
$$\|\mathbf{V}-\mathbf{V}^k\| \le \hat{C}_2 L^k \Delta t^{2k+k_0-1} \tag{A.13b}$$

with constants $\hat{C}_1$, $\hat{C}_2$ independent of Δt.

Proof. First, consider a case where the right-hand side function f does not depend on v. Insert (A.5a) into (A.5b) to get

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\| &\le 2L\Delta t(\|\mathbf{X}-\mathbf{X}^{k-1}\| + C_1 L\Delta t^2\|\mathbf{X}-\mathbf{X}^{k-1}\|)\\ &= 2L\Delta t\|\mathbf{X}-\mathbf{X}^{k-1}\| + 2C_1L^2\Delta t^3\|\mathbf{X}-\mathbf{X}^{k-1}\|.\end{aligned}$$

As before,

$$\|\mathbf{X}-\mathbf{X}^k\| \le C_1 L\Delta t^2\|\mathbf{X}-\mathbf{X}^{k-1}\|,$$

such that

$$\|\mathbf{V}-\mathbf{V}^k\| \le 2L\Delta t\|\mathbf{X}-\mathbf{X}^{k-1}\| + 2C_1L^2\Delta t^3\|\mathbf{X}-\mathbf{X}^{k-1}\|.$$

By recursive insertion, we get

$$\|\mathbf{X}-\mathbf{X}^k\| \le C_1^k L^k \Delta t^{2k}\|\mathbf{X}-\mathbf{X}^0\|, \tag{A.14}$$
$$\|\mathbf{V}-\mathbf{V}^k\| \le 2C_1^{k-1}L^k\Delta t^{2k-1}\|\mathbf{X}-\mathbf{X}^0\| + 2C_1^kL^{k+1}\Delta t^{2k+1}\|\mathbf{X}-\mathbf{X}^0\|. \tag{A.15}$$

For a starting value $\mathbf{X}^0$ for the SDC iteration of order k_0 we have

$$\|\mathbf{X}-\mathbf{X}^0\| \le C_0\Delta t^{k_0}, \tag{A.16a}$$
$$\|\mathbf{V}-\mathbf{V}^0\| \le C_0\Delta t^{k_0}, \tag{A.16b}$$

where the constant C_0 is independent of Δt. Taken together, we find that

$$\|\mathbf{X}-\mathbf{X}^k\| \le C_1^kL^k\Delta t^{2k}\|\mathbf{X}-\mathbf{X}^0\| \le C_1^kC_0L^k\Delta t^{2k+k_0}.$$

Similarly, using (A.16) in (A.15) we obtain

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\| &\le 2C_1^{k-1}L^k\Delta t^{2k-1}\|\mathbf{X}-\mathbf{X}^0\| + 2C_1^kL^{k+1}\Delta t^{2k+1}\|\mathbf{X}-\mathbf{X}^0\|\\ &\le 2C_1^{k-1}C_0L^k\Delta t^{2k+k_0-1} + 2C_1^kC_0L^{k+1}\Delta t^{2k+k_0+1}\\ &= 2C_0C_1^{k-1}(1+C_1L\Delta t^2)L^k\Delta t^{2k+k_0-1}.\end{aligned}$$

Since $\Delta t^2 < \frac{1}{3}$, the following estimate is valid

$$2C_0C_1^{k-1}(1+C_1L\Delta t^2) \le 2C_0C_1^{k-1}\left(1+\frac{C_1L}{3}\right) =: \hat{C}_2.$$

Thus,

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\le \hat{C}_1L^k\Delta t^{2k+k_0},\\ \|\mathbf{V}-\mathbf{V}^k\| &\le \hat{C}_2L^k\Delta t^{2k+k_0-1},\end{aligned}$$

where $\hat{C}_1 := C_1^kC_0$.

For the general case where f depends on v, we use estimate (A.4) in Proposition A.2. First, we insert $\|\mathbf{X}-\mathbf{X}^k\|$ from (A.4a) on the right–hand side of the inequality (A.4b) to get

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\le C_1L\Delta t^2(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|\\ &\quad + C_2L\Delta t(\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^k\|))\\ &= (C_1L\Delta t^2+C_1C_2L^2\Delta t^3)(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|)\\ &\quad + C_1C_2L^2\Delta t^3\|\mathbf{X}-\mathbf{X}^k\|.\end{aligned}$$

If Δt is small enough such that $1-C_1C_2L^2\Delta t^3 \ge \tilde{\delta} > 0$, then

$$\|\mathbf{X}-\mathbf{X}^k\| \le \frac{C_1L\Delta t^2+C_1C_2L^2\Delta t^3}{1-C_1C_2L^2\Delta t^3}(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|). \tag{A.17}$$

Analogously, substitute the expression for $\|\mathbf{V}-\mathbf{V}^k\|$ from (A.4b) into (A.4a) to get

$$\begin{aligned}\|\mathbf{V}-\mathbf{V}^k\| &\le C_2L\Delta t(\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{X}-\mathbf{X}^{k-1}\|\\ &\quad + C_1L\Delta t^2(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^k\|))\\ &= (C_2L\Delta t+C_1C_2L^2\Delta t^3)(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|)\\ &\quad + C_1C_2L^2\Delta t^3\|\mathbf{V}-\mathbf{V}^k\|.\end{aligned}$$

Hence,

$$\|\mathbf{V}-\mathbf{V}^k\| \le \frac{C_2L\Delta t+C_1C_2L^2\Delta t^3}{1-C_1C_2L^2\Delta t^3}(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|). \tag{A.18}$$

Let

$$m_1 := \frac{L\Delta t^2(C_1+C_1C_2L\Delta t)}{1-C_1C_2L^2\Delta t^3},\ m_2 := \frac{L\Delta t(C_2+C_1C_2L\Delta t^2)}{1-C_1C_2L^2\Delta t^3}, \tag{A.19}$$

and we obtain the following system of inequalities

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\le m_1(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|),\\ \|\mathbf{V}-\mathbf{V}^k\| &\le m_2(\|\mathbf{X}-\mathbf{X}^{k-1}\|+\|\mathbf{V}-\mathbf{V}^{k-1}\|).\end{aligned}$$

These can be written in matrix form

$$\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^k\|\\ \|\mathbf{V}-\mathbf{V}^k\|\end{pmatrix} \le \begin{pmatrix}m_1 & m_1\\ m_2 & m_2\end{pmatrix}\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^{k-1}\|\\ \|\mathbf{V}-\mathbf{V}^{k-1}\|\end{pmatrix}. \tag{A.20}$$

Recursive insertion yields

$$\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^k\|\\ \|\mathbf{V}-\mathbf{V}^k\|\end{pmatrix} \le \begin{pmatrix}m_1 & m_1\\ m_2 & m_2\end{pmatrix}^k\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^0\|\\ \|\mathbf{V}-\mathbf{V}^0\|\end{pmatrix} =: M^k\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^0\|\\ \|\mathbf{V}-\mathbf{V}^0\|\end{pmatrix}. \tag{A.21}$$

It is easy to show by induction that $M^k=(m_1+m_2)^{k-1}M$ so that (A.21) becomes

$$\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^k\|\\ \|\mathbf{V}-\mathbf{V}^k\|\end{pmatrix} \le \begin{pmatrix}(m_1+m_2)^{k-1}m_1 & (m_1+m_2)^{k-1}m_1\\ (m_1+m_2)^{k-1}m_2 & (m_1+m_2)^{k-1}m_2\end{pmatrix}\begin{pmatrix}\|\mathbf{X}-\mathbf{X}^0\|\\ \|\mathbf{V}-\mathbf{V}^0\|\end{pmatrix} \tag{A.22}$$

or

$$\begin{aligned}\|\mathbf{X}-\mathbf{X}^k\| &\le m_1(m_1+m_2)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\|+\|\mathbf{V}-\mathbf{V}^0\|),\\ \|\mathbf{V}-\mathbf{V}^k\| &\le m_2(m_1+m_2)^{k-1}(\|\mathbf{X}-\mathbf{X}^0\|+\|\mathbf{V}-\mathbf{V}^0\|).\end{aligned}$$

For $\Delta t < \frac{1}{\sqrt{3}}$ and $1-C_1C_2L^2\Delta t^3 \ge \tilde{\delta} > 0$, we can write

$$\frac{C_2 + C_1C_2L\Delta t^2}{1 - C_1C_2L^2\Delta t^3} \leq \frac{3C_2 + C_1C_2L}{3\tilde{\delta}} =: C_3,$$
$$\frac{C_1 + C_1C_2L\Delta t}{1 - C_1C_2L^2\Delta t^3} \leq \frac{\sqrt{3}C_1 + C_1C_2L}{\sqrt{3}\tilde{\delta}} =: C_4,$$

and get the following inequalities

$$\|\mathbf{X} - \mathbf{X}^k\| \leq C_4L\Delta t^2(C_3L\Delta t + C_4L\Delta t^2)^{k-1}(\|\mathbf{X} - \mathbf{X}^0\| + \|\mathbf{V} - \mathbf{V}^0\|),$$
$$\|\mathbf{V} - \mathbf{V}^k\| \leq C_3L\Delta t(C_3L\Delta t + C_4L\Delta t^2)^{k-1}(\|\mathbf{X} - \mathbf{X}^0\| + \|\mathbf{V} - \mathbf{V}^0\|).$$

Using the results from (A.16) yields

$$\begin{aligned}\|\mathbf{X} - \mathbf{X}^k\| &\leq L^k\Delta t^{k+1}C_4(C_3 + C_4\Delta t)^{k-1}(\|\mathbf{X} - \mathbf{X}^0\| + \|\mathbf{V} - \mathbf{V}^0\|)\\ &\leq L^k\Delta t^{k+1}C_4(C_3 + C_4\Delta t)^{k-1}(C_0\Delta t^{k_0} + C_0\Delta t^{k_0})\\ &\leq 2C_4C_0(C_3 + C_4\Delta t)^{k-1}L^k\Delta t^{k+k_0+1}.\end{aligned}$$

Similar computations can be done for the variable $\mathbf{V}$ which yields

$$\begin{aligned}\|\mathbf{V} - \mathbf{V}^k\| &\leq L^k\Delta t^kC_3(C_3 + C_4\Delta t)^{k-1}(\|\mathbf{X} - \mathbf{X}^0\| + \|\mathbf{V} - \mathbf{V}^0\|)\\ &\leq 2C_3C_0(C_3 + C_4\Delta t)^{k-1}L^k\Delta t^{k+k_0}.\end{aligned}$$

With $\Delta t < \frac{1}{\sqrt{3}}$, we find that

$$2C_4C_0(C_3 + C_4\Delta t)^{k-1} \leq 2C_4C_0\left(C_3 + \frac{1}{\sqrt{3}}C_4\right)^{k-1} := \tilde{C}_1,$$
$$2C_3C_0(C_3 + C_3\Delta t)^{k-1} \leq 2C_3C_0\left(C_3 + \frac{1}{\sqrt{3}}C_4\right)^{k-1} := \tilde{C}_2.$$

Therefore,

$$\|\mathbf{X} - \mathbf{X}^k\| \leq \tilde{C}_1L^k\Delta t^{k+k_0+1},$$
$$\|\mathbf{V} - \mathbf{V}^k\| \leq \tilde{C}_2L^k\Delta t^{k+k_0},$$

which completes the proof. □

Reproducibility of computational results. This paper has been awarded the "SIAM Reproducibility Badge: code and data available," as a recognition that the authors have followed reproducibility principles valued by SISC and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://github.com/Parallel-in-Time/pySDC/tree/master/pySDC/projects/Second_orderSDC.

Acknowledgment. We gratefully acknowledge many helpful discussions with Thibaut Lunet as well as his assistance with the pySDC software.

REFERENCES

[1] P. Agarwal, J. Mohamed, and B. Samet, *Fixed Point Theory in Metric Spaces: Recent Advances and Applications*, Springer, Singapore, 2018, https://doi.org/10.1007/978-981-13-2913-5.

[2] X. Bai and J. L. Junkins, *Modified Chebyshev-Picard iteration methods for orbit propagation*, J. Astronaut. Sci., 58 (2011), pp. 583–613, https://doi.org/10.1007/BF03321533.

[3] S. Blanes and P. Moan, *Practical symplectic partitioned Runge–Kutta and Runge–Kutta-Nyström methods*, J. Comput. Appl. Math., 142 (2002), pp. 313–330, https://doi.org/10.1016/S0377-0427(01)00492-7.

[4] J. P. Boris, *Relativistic plasma simulation-optimization of a hybrid code*, in Proceeding of Fourth Conference on Numerical Simulations of Plasmas, Naval Research Laboratory, Washington, DC, 1972.

[5] E. L. Bouzarth and M. L. Minion, *A multirate time integrator for regularized Stokeslets*, J. Comput. Phys., 229 (2010), pp. 4208–4224, https://doi.org/10.1016/j.jcp.2010.02.006.

[6] L. Brown and G. Gabrielse, *Geonium theory: Physics of a single electron or ion in a Penning trap*, Rev. Mod. Phys., 58 (1986), pp. 233–311.

[7] G. Caklovic, *The Infinity Norm Bounds and Characteristic Polynomial for High Order RK Matrices*, https://doi.org/10.48550/arXiv.2203.04086, 2022.

[8] M. F. Causley and D. C. Seal, *On the convergence of spectral deferred correction methods*, Commun. Appl. Math. Comput. Sci., 14 (2019), pp. 33–64, https://doi.org/10.2140/camcos.2019.14.33.

[9] C. Chicone, *Ordinary Differential Equations with Applications*, Texts Appl. Math. 34, Springer, NY, 2006, https://doi.org/10.1007/0-387-35794-7.

[10] A. Christlieb, B. W. Ong, and J.-M. Qiu, *Integral deferred correction methods constructed with high order Runge-Kutta integrators*, Math. Comp., 79 (2010), pp. 761–783, https://doi.org/10.1090/S0025-5718-09-02276-5.

[11] A. Dutt, L. Greengard, and V. Rokhlin, *Spectral deferred correction methods for ordinary differential equations*, BIT, 40 (2000), pp. 241–266, https://doi.org/10.1023/A:1022338906936.

[12] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Appl. Math. Sci. 95, Springer, New York, 1994.

[13] T. Hagstrom and R. Zhou, *On the spectral deferred correction of splitting methods for initial value problems*, Commun. Appl. Math. Comput. Sci., 1 (2006), pp. 169–205, https://doi.org/10.2140/camcos.2006.1.169.

[14] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-preserving Algorithms for Ordinary Differential Equations*, Springer, Berlin, 2002, https://doi.org/10.1007/3-540-30666-8.

[15] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration illustrated by the Störmer-Verlet method*, Acta Numer., 12 (2003), pp. 399–450, https://doi.org/10.1017/S0962492902000144.

[16] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I*: Nonstiff problems*, 2nd ed., Springer, Berlin, 1993, https://doi.org/10.1007/978-3-540-78862-1.

[17] A. C. Hansen and J. Strain, *On the order of deferred correction*, Appl. Numer. Math., 61 (2011), pp. 961–973, https://doi.org/10.1016/j.apnum.2011.04.001.

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

[19] S. O. Imoni, F. O. Otunta, and T. R. Ramamohan, *Embedded implicit Runge-Kutta Nyström method for solving second-order differential equations*, Int. J. Comput. Math., 83 (2006), pp. 777–784, https://doi.org/10.1080/00207160601084505.

[20] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Front. Appl. Math. 16, SIAM, Philadelphia, 1995, https://doi.org/10.1137/1.9781611970944.

[21] G. Kremling and R. Speck, *Convergence of multilevel spectral deferred corrections*, Commun. Appl. Math. Comput. Sci., 16 (2021), pp. 227–265, https://doi.org/10.2140/camcos.2021.16.227.

[22] M. L. Minion, *Semi-implicit spectral deferred correction methods for ordinary differential equations*, Commun. Math. Sci., 1 (2003), pp. 471–500, https://doi.org/10.4310/CMS.2003.v1.n3.a6.

[23] M. L. Minion, *Semi-implicit projection methods for incompressible flow based on spectral deferred corrections*, Appl. Numer. Math., 48 (2004), pp. 369–387.

[24] F. Penning, *Die Glimmentladung bei niedrigem Druck zwischen koaxialen Zylindern in einem axialen Magnetfeld*, Physica, 3 (1936), pp. 873–894, https://doi.org/10.1016/S0031-8914(36)80313-9.

[25] D. Ruprecht and R. Speck, *Spectral deferred corrections with fast-wave slow-wave splitting*, SIAM J. Sci. Comput., 38 (2016), pp. A2535–A2557, https://doi.org/10.1137/16M1060078.

[26] Q. Shen and B. Kochunas, *High-order accurate solutions of the point kinetics equations with the spectral deferred correction method*, Nucl. Sci. Eng., 197 (2023), pp. 1364–1385, https://doi.org/10.1080/00295639.2022.2159276.

[27] K. Smedt, D. Ruprecht, J. Niesen, S. Tobias, and J. Nättilä, *New applications for the Boris spectral deferred correction algorithm for plasma simulations*, Appl. Math. Comput., 442 (2023), 127706, https://doi.org/10.1016/j.amc.2022.127706.

[28] B. Sommeijer, *Explicit, high-order Runge-Kutta-Nyström methods for parallel computers*, Appl. Numer. Math., 13 (1993), pp. 221–240.

[29] R. Speck, *Algorithm* 997*: PySDC - prototyping spectral deferred corrections*, ACM Trans. Math. Software, 45 (2019), 35, https://doi.org/10.1145/3310410.

[30] R. Speck, T. Lunet, T. Baumann, L. Wimmer, and I. Akramov, *Parallel-in-Time/pysdc*, https://doi.org/10.5281/zenodo.10637162.

[31] G. Strang, *Linear Algebra and its Applications*, 4th ed., Brooks Cole, Pacific Grove, CA, 1993.

[32] K. Tretiak, J. Buchanan, R. Akers, and D. Ruprecht, *Performance of the BGSDC integrator for computing fast ion trajectories in nuclear fusion reactors*, Comput. Phys. Commun., 264 (2021), 107876, https://doi.org/10.1016/j.cpc.2021.107876.

[33] K. Tretiak and D. Ruprecht, *An arbitrary order time-stepping algorithm for tracking particles in inhomogeneous magnetic fields*, J. Comput. Phys. X, 4 (2019), 100036, https://doi.org/10.1016/j.jcpx.2019.100036.

[34] P. J. van der Houwen and B. P. Sommeijer, *Explicit Runge–Kutta (–Nyström) methods with reduced phase errors for computing oscillating solutions*, SIAM J. Numer. Anal., 24 (1987), pp. 595–617, https://doi.org/10.1137/0724041.

[35] P. J. Van der Houwen and B. P. Sommeijer, *Diagonally implicit Runge–Kutta–Nyström methods for oscillatory problems*, SIAM J. Numer. Anal., 26 (1989), pp. 414–429, https://doi.org/10.1137/0726023.

[36] L. Verlet, *Computer "Experiments" on classical fluids.* I. *Thermodynamical properties of Lennard-Jones molecules*, Phys. Rev. (2), 159 (1967), 98–103, https://doi.org/10.1103/PhysRev.159.98.

[37] M. Winkel, R. Speck, and D. Ruprecht, *A high-order Boris integrator*, J. Comput. Phys., 295 (2015), pp. 456–474, https://doi.org/10.1016/j.jcp.2015.04.022.