ФИЗИКА / PHYSICS

УДК 519.6
DOI: https://doi.org/10.52754/16948645_2026_1(8)_39

A HYBRID SPECTRAL DEFERRED CORRECTION METHOD WITH PHYSICS-INFORMED NEURAL NETWORK INITIALIZATION

Akramov Ikrom
Associate Professor
Bukhara state university
OrcidID: 0000-0002-8869-0784 *ikrom960623@gmail.com*

Rasulov Tulqin
Professor
Bukhara state university
OrcidID: 0000-0002-2868-4390 *t.h.rasulov@buxdu.uz*

Kurbonova Dilnora
Student
Bukhara state university
OrcidID: 0009-0006-3015-1928 *dilnoraqurbonova3@gmail.com*

Abstract: Spectral Deferred Correction (SDC) methods are iterative methods used to solve ordinary differential equations. The SDC method can achieve high-order convergence by adjusting the initial guess, quadrature nodes, and the number of iterations. Therefore, its convergence strongly depends on the initial approximation. In this work, we introduce a hybrid method in which a Physics-Informed Neural Network (PINN) provides the initial guess for SDC iterations. We present a rigorous formulation of both SDC and PINN, along with a convergence analysis of the proposed method. Finally, we validate our results using numerical examples. The proposed method significantly reduces the number of SDC sweeps required to achieve high-order accuracy.

Keywords: Spectral Deferred Correction (SDC), collocation problem, Physics-Informed Neural Network (PINN), initial guess.

ГИБРИДНЫЙ МЕТОД СПЕКТРАЛЬНОЙ ОТЛОЖЕННОЙ КОРРЕКЦИИ С ИНИЦИАЛИЗАЦИЕЙ НА ОСНОВЕ ФИЗИЧЕСКИ-ИНФОРМИРОВАННОЙ НЕЙРОННОЙ СЕТИ

Акрамов Икром
доцент
Бухарский государственный университет
OrcidID: 0000-0002-8869-0784, *ikrom960623@gmail.com*

Расулов Тўлқин
профессор
Бухарский государственный университет
OrcidID: 0000-0002-2868-4390, *t.h.rasulov@buxdu.uz*

Курбанова Дилнора
студентка

Бухарский государственный университет
OrcidID: 0009-0006-3015-1928, dilnoraqurbonova3@gmail.com

Аннотация: Методы спектральной отложенной коррекции (SDC) являются итерационными методами для решения обыкновенных дифференциальных уравнений. Метод SDC может достигать высокого порядка сходимости за счёт изменения начального приближения, квадратурных узлов и числа итераций. Поэтому его сходимость существенно зависит от начального приближения. В данной работе предлагается гибридный метод, в котором физически-информированная нейронная сеть (PINN) используется для задания начального приближения для итераций SDC. Представлена строгая формулировка методов SDC и PINN, а также анализ сходимости предложенного метода. В заключение результаты подтверждаются численными примерами. Предложенный метод существенно уменьшает количество итераций SDC, необходимых для достижения высокой точности.

Ключевые слова: спектральная отложенная коррекция (SDC), задача коллокации, физически-информированная нейронная сеть (PINN), начальное приближение.

Introduction

High-order numerical methods play a fundamental role in the accurate simulation of nonlinear and stiff differential equations arising in physics, engineering, and applied sciences [1-3]. In many applications, such as fluid dynamics, plasma physics, and kinetic theory, achieving high accuracy in time integration is essential for capturing complex dynamical behavior [4].

Spectral Deferred Correction (SDC) methods, originally introduced by Dutt, Greengard, and Rokhlin [5], provide a flexible and systematic framework for constructing high-order time integration schemes. The main idea behind SDC is to iteratively correct a low-order approximation using defect correction based on an integral formulation of the differential equation. Each iteration, often referred to as a sweep, increases the order of accuracy and drives the numerical solution toward the corresponding collocation solution. Over the years, SDC methods have been further developed and analyzed in various contexts, including stiff problems and multi-level extensions [5-9].

Despite their advantages, classical SDC methods rely heavily on the quality of the initial approximation. Typically, this initial guess is obtained using a low-order time-stepping scheme such as the forward or backward Euler method. While such approaches are inexpensive, they may produce relatively large initial errors, especially for highly nonlinear or stiff systems. As a result, a larger number of SDC iterations may be required to achieve the desired accuracy, which reduces the overall efficiency of the method [10, 11].

In parallel, recent advances in scientific machine learning have introduced Physics-Informed Neural Networks (PINNs) as a powerful tool for approximating solutions of differential equations [12]. PINNs incorporate the governing equations directly into the training process of a neural network, allowing the model to learn solutions that satisfy the underlying physical laws. Unlike classical numerical methods, PINNs provide a global, smooth approximation over the entire domain and do not require a predefined

The main objective of this work is to develop and analyze such a hybrid PINN-SDC method. In this framework, the neural network provides a physics-consistent initial guess at the quadrature nodes, and the SDC method acts as an efficient iterative solver for the associated collocation problem. We show that this combination leads to faster convergence and improved efficiency, particularly for nonlinear problems.

The remainder of the paper is organized as follows. In Section 2, we introduce the integral formulation and the associated collocation problem. Section 3 presents the derivation of the SDC method. Section 4 reviews the formulation of Physics-Informed Neural Networks. In Section 5, we describe the proposed hybrid method. Section 6 provides an error decomposition and convergence analysis. Finally, numerical results and conclusions are presented in the subsequent sections.

Collocation problem

In this section, we explain carefully how the Spectral Deferred Correction (SDC) method is obtained starting from the original differential equation

Consider the following ordinary differential equation

$$\frac{du}{dt} = f(u,t), u(t_0) = u_0 \tag{1}$$

Instead of working directly with the derivative, we rewrite the problem in an integral form. This is done by integrating both sides from t_0 to some time t

$$u(t) = u_0 + \int_{t_0}^{t} f(u(\tau),\tau)d\tau \tag{2}$$

This equation tells us something very important. The value of the solution at time t is equal to the initial value plus the total accumulated effect of the function f from time t_0 to t.

Suppose we divide the interval into M sub-intervals

$$t_0 < t_1 < t_2 < \cdots < t_M$$

These points are called quadrature nodes. At each of these points, we want to approximate the solution.

For each point t_m, the integral equation becomes:

$$u(t_m) = u_0 + \int_{t_0}^{t_m} f(u(\tau),\tau)d\tau \tag{3}$$

The integral is usually difficult to compute exactly because we do not know the exact solution $u(\tau)$. Therefore, we approximate the integral using a numerical rule called quadrature.

We replace the integral by a weighted sum:

$$\int_{t_0}^{t_m} f(u(\tau),\tau)d\tau \approx \sum_{j=0}^{M} q_{mj} f(u_j, t_j) \tag{4}$$

where u_j is an approximation of $u(t_j)$, and q_{mj} are known weights that depend on the chosen nodes.

Now the equation becomes:

$$u_m = u_0 + \sum_{j=0}^{M} q_{mj} f(u_j, t_j) \tag{5}$$

This is called the *collocation problem*. It is a system of equations where all unknown values $u_0, u_1, \ldots, u_M$ appear together.

The important thing to notice is that the values u_j appear inside the function $f(u_j, t_j)$. This means the equations are nonlinear. Because of this, solving the system directly can be very expensive, especially when M is large.

Instead of solving it all at once, we try to improve an approximate solution step by step. This idea leads to the SDC method.

We begin by choosing an initial approximation. For example, we can use a simple method like Euler's method to compute values:

$$u_0^0, u_1^0, \dots, u_M^0$$

Here, the superscript 0 means this is the first guess.

This guess is not very accurate, but it gives us a starting point.

Now we check how good our approximation is. We do this by plugging it into the collocation equation. We define the defect:

$$r_m^k = u_0 + \sum_{j=0}^{M} q_{mj} f\left(u_j^k, t_j\right) - u_m^k \tag{6}$$

If our approximation were perfect, this value would be zero. In reality, it is not zero, and this tells us how much error we have.

We now try to correct our approximation. We write:

$$u_m^{k+1} = u_m^k + \delta_m^k$$

where δ_m^k is a correction that we want to compute.

The idea is simple: we improve the solution little by little.

Spectral deferred correction method

Instead of solving a complicated global problem, we go back to the integral form and focus on small intervals.

Between two points t_{m-1} and t_m, we write:

$$u(t_m) = u(t_{m-1}) + \int_{t_{m-1}}^{t_m} f(u(\tau), \tau) d\tau \tag{7}$$

Now we approximate this step by step.

We split the integral into two parts. One part uses the old approximation, and the other part corrects the error:

$$u_m^{k+1} = u_{m-1}^{k+1} + \int_{t_{m-1}}^{t_m} f(u^k(\tau), \tau) d\tau + \int_{t_{m-1}}^{t_m} [f(u^{k+1}(\tau)) - f(u^k(\tau))] d\tau \tag{8}$$

The first integral is computed using the known values from the previous iteration. The second integral is small and can be approximated using a simple method.

We approximate the correction term using a simple Euler-type step:

$$\int_{t_{m-1}}^{t_m} [f(u^{k+1}) - f(u^k)] d\tau \approx \Delta t_m \left[f\left(u_{m-1}^{k+1}\right) - f\left(u_{m-1}^k\right)\right] \tag{9}$$

The first integral is approximated using quadrature weights. Combining everything, we obtain:

$$u_m^{k+1} = u_{m-1}^{k+1} + \Delta t_m \left[f\left(u_{m-1}^{k+1}\right) - f\left(u_{m-1}^k\right)\right] + \sum_{j=0}^{M} q_{mj} f\left(u_j^k\right) \tag{10}$$

This is exactly the SDC iteration formula.

The method works as follows. We start with a rough approximation. Then we repeatedly improve it. Each correction reduces the error and increases the accuracy.

After several iterations, the solution becomes very close to the true collocation solution.

The collocation method gives a very accurate solution but is hard to solve directly. The SDC method solves it step by step in a simple and efficient way.

In your work, instead of starting with a rough initial guess, a Physics-Informed Neural Network provides a much better starting point. This means fewer correction steps are needed to reach high accuracy.

Physics-Informed Neural Networks

In this section, we explain what a Physics-Informed Neural Network (PINN) is and how it is used to approximate the solution of a differential equation [12].

A neural network is a function with parameters. We denote it by $u_\theta(t)$, where θ represents all adjustable parameters (weights and biases). The goal of the neural network is to learn a function that behaves like the true solution $u(t)$.

The key idea of a PINN is the following. Instead of only learning from data, the neural network is trained using the differential equation itself. This means that the network is forced to satisfy the physics of the problem.

We recall the differential equation

$$\frac{du}{dt} = f(u,t) \tag{11}$$

If $u_\theta(t)$ is a good approximation, then it should satisfy

$$\frac{du_\theta}{dt}(t) \approx f(u_\theta(t),t) \tag{12}$$

To enforce this, we define a loss function. The loss function measures how well the neural network satisfies the equation and the initial condition.

We write:

$$\mathcal{L} = \frac{1}{N}\sum_i |u_\theta(t_i) - u_i|^2 + \frac{1}{N}\sum_j \left|\frac{du_\theta}{dt}(t_j) - f\big(u_\theta(t_j),t_j\big)\right|^2 \tag{13}$$

The first term measures how well the neural network matches known values, such as the initial condition. The second term measures how well the differential equation is satisfied.

During training, the parameters θ are adjusted so that this loss becomes as small as possible. When the loss is small, the neural network behaves like a function that satisfies the differential equation.

An important property of PINNs is that they produce a smooth function defined on the entire time interval. This is different from classical numerical methods, which only give values at discrete points.

Proposed Hybrid Method

When we solve the problem using SDC, we need an initial approximation:

$$u_m^0 \approx u(t_m)$$

If this initial guess is poor, then many SDC iterations are required to correct the error. This increases computational cost.

A simple method like Euler produces a rough approximation [13, 14]. The error can be quite large, especially for nonlinear problems.

In contrast, a PINN has already learned the structure of the solution. It approximately satisfies the differential equation everywhere in the interval. This means that the initial error is already small before starting SDC.

Another important point is that the PINN gives values at all points in a continuous way. Therefore, it naturally provides values at all quadrature nodes without additional computation.

Because of these reasons, the PINN initialization is much closer to the true collocation solution than a classical low-order method.

As a result, the SDC method needs fewer correction steps to reach high accuracy.

We now describe in detail how the Physics-Informed Neural Network (PINN) is combined with the Spectral Deferred Correction (SDC) method.

The central idea is to use the neural network as a high-quality approximation of the solution before applying any classical numerical method. The neural network produces a function $u_\theta(t)$ that approximately satisfies the differential equation on the entire interval.

Once the neural network is trained, we evaluate it at the quadrature nodes:

$$u_m^0 := u_\theta(t_m), m = 0,1, \dots, M \tag{14}$$

These values serve as the initial approximation for the SDC method. Unlike standard approaches where the initial guess is obtained using a low-order method, here the approximation already incorporates global information about the solution.

After initialization, the SDC iteration is applied. At each sweep, the method corrects the current approximation using the defect of the collocation problem. Because the initial error is already small, the correction process converges faster.

Thus, the hybrid method consists of two stages. In the first stage, a global approximation is constructed using the neural network. In the second stage, this approximation is refined locally through SDC iterations until high-order accuracy is achieved.

Algorithm

1. Train the neural network $u_\theta(t)$ on the interval $[0, T]$ so that it approximately satisfies the differential equation and initial condition.
2. Evaluate the trained network at the quadrature nodes to obtain the initial values u_m^0.
3. Apply SDC sweeps to iteratively improve the solution.

Error Decomposition

To understand why this hybrid method is effective, we analyze the error in a structured way.

Let $u(t)$ denote the exact solution, and let u_m^k be the approximation after k SDC iterations. We define the total error:

$$e_m^k = u(t_m) - u_m^k \tag{15}$$

We now split this error into two parts:

$$e_m^k = \big(u(t_m) - u_\theta(t_m)\big) + (u_\theta(t_m) - u_m^k). \tag{16}$$

The first term represents the error introduced by the neural network:

$$e_m^{\mathrm{PINN}} = u(t_m) - u_\theta(t_m) \tag{17}$$

The second term represents the error introduced by the SDC iterations:

$$e_m^{\mathrm{SDC},k} = u_\theta(t_m) - u_m^k \tag{18}$$

Thus, the total error can be written as:

$$e_m^k = e_m^{\mathrm{PINN}} + e_m^{\mathrm{SDC},k} \tag{19}$$

This decomposition clearly separates the contribution of the machine learning approximation and the numerical correction.

The PINN error depends on how well the neural network is trained. The SDC error depends on the number of iterations and the time step size.

Improved Hybrid Method Description

We now refine our understanding of the hybrid method using the error decomposition. At the beginning of the SDC iteration, we have:

$$u_m^0 = u_\theta(t_m) \tag{20}$$

which implies:

$$e_m^0 = u(t_m) - u_\theta(t_m) = e_m^{\text{PINN}} \tag{21}$$

This shows that the initial error is exactly the PINN approximation error.

During SDC iterations, the method reduces the numerical error component. More precisely, each SDC sweep improves the approximation by correcting the defect of the collocation equation.

As the number of iterations increases, the SDC error decreases, while the PINN error remains fixed. Therefore, after sufficiently many iterations, the dominant source of error is the neural network approximation.

This observation explains why a better PINN leads to fewer required SDC iterations.

Convergence Analysis

We now formalize the above observations.

Theorem 1. Assume that the function f is Lipschitz continuous with constant L. Let the neural network approximation satisfy:

$$\|u - u_\theta\| \le \varepsilon_{PINN}$$

Then the total error after K SDC iterations satisfies:

$$\|e^K\| \le C(\Delta t)^K + \varepsilon_{PINN} \tag{22}$$

where C is a constant independent of Δt and K.

Proof. From the theory of SDC methods, it is known that each iteration reduces the collocation error by a factor proportional to Δt. Therefore, after K iterations, the SDC error satisfies:

$$\left\|e^{\text{SDC},K}\right\| \le C(\Delta t)^K$$

From the error decomposition, we have:

$$\|e^K\| \le \|e^{\text{PINN}}\| + \left\|e^{\text{SDC},K}\right\|$$

Using the assumption on the PINN approximation, we obtain:

$$\|e^K\| \le \varepsilon_{\text{PINN}} + C(\Delta t)^K$$

This completes the proof.

Numerical Results

In this section, we demonstrate the performance of the proposed PINN-SDC method on two test problems. The first example is a simple linear equation with a known exact solution, and the second example is the nonlinear Penning trap system, which is commonly used in plasma physics.

Linear Equation: We first consider the scalar problem:

$$u' = -u, u(0) = 1 \tag{23}$$

The exact solution is given by:

$$u(t) = e^{-t} \tag{24}$$

We compare the performance of the SDC method initialized with a standard low-order method and with a PINN-based initialization.

The results show that the PINN provides a much better initial approximation. As a result, the number of SDC iterations required to achieve a given accuracy is significantly reduced. The final numerical solution obtained by the hybrid method matches the exact solution with high accuracy.

The table 1 shows the evolution of the training loss over time. At epoch 0, the initial loss is relatively large (around 10^{-3}), reflecting the fact that the network parameters are randomly initialized and the network does not yet satisfy the differential equation.

After 500 epochs, the loss drops dramatically to approximately 10^{-8}, indicating that the network has learned a solution that closely satisfies the governing differential equation and initial condition. Subsequent training (epochs 1000 and 1500) shows that the loss remains around 10^{-8}, suggesting convergence and stability of the network training.

Table 1. *The evolution of the PINN training loss over time for linear problem.*

Epoch	**Loss**
0	0.002496610628440976
500	1.3459487746558807e-08
1000	1.522908021911462e-08

Penning Trap System

We now consider the Penning trap model, which describes the motion of a charged particle in combined electric and magnetic fields. The equations of motion are given by:

$$\ddot{x} = \omega_c \dot{y} + \frac{\omega_z^2}{2} x, \ddot{y} = -\omega_c \dot{x} + \frac{\omega_z^2}{2} y, \ddot{z} = -\omega_z^2 z \tag{25}$$

We rewrite this system as a first-order system by introducing velocity variables:

$$v_x = \dot{x}, v_y = \dot{y}, v_z = \dot{z}$$

This leads to:

$$\dot{x} = v_x \tag{26}$$
$$\dot{y} = v_y \tag{27}$$
$$\dot{z} = v_z \tag{28}$$
$$\dot{v}_x = \omega_c v_y + \frac{\omega_z^2}{2} x \tag{29}$$
$$\dot{v}_y = -\omega_c v_x + \frac{\omega_z^2}{2} y \tag{30}$$
$$\dot{v}_z = -\omega_z^2 z \tag{31}$$

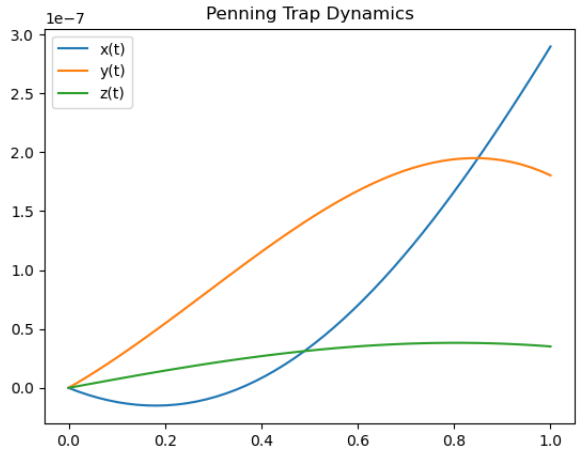

Figure 1. *The trajectory of Penning trap problem for 3-axes.*

Figure 1 shows the trajectory of Penning trap problem for 3 axes they are $x(t), y(t)$ and $z(t)$. The system exhibits oscillatory behavior and coupling between components, making it a suitable test for the proposed method.

Table 2. *PINN Training Loss for Penning Trap Problem.*

Epoch Loss	Epoch Loss
0	0.01792776770889759
500	3.1727699933981057e-07
1000	1.6572564618400065e-06
1500	2.3762851242281613e-07

This table 2 shows the evolution of the training loss for the Physics-Informed Neural Network (PINN) applied to the Penning trap system, which is a six-dimensional nonlinear dynamical system. At epoch 0, the loss is approximately 1.79e-2, indicating that the network starts with a poor approximation due to random initialization. After 500 epochs, the loss drops sharply to about 3.17e-7, demonstrating that the network has learned a solution that closely satisfies the system of differential equations. By epoch 1000, the loss slightly increases to 1.66e-6, which can happen due to stochastic variations during training, but it remains very small. At epoch 1500, the loss stabilizes at 2.38e-7, indicating that the network has effectively converged.

Conclusion

In this work, we proposed a hybrid numerical framework that combines Spectral Deferred Correction (SDC) methods with Physics-Informed Neural Networks (PINNs) for solving initial value problems. The main idea is to use the neural network as a physics-consistent initial approximation and then apply SDC iterations to recover high-order accuracy.

We began by revisiting the integral formulation of the differential equation and its associated collocation problem. Based on this formulation, we derived the SDC method as an iterative procedure for solving the collocation system. We then introduced the PINN approach, which provides a global approximation of the solution by incorporating the governing equation into the training process.

The central contribution of this work is the integration of these two approaches. By evaluating the trained neural network at the quadrature nodes, we obtain an initial approximation that is more accurate than traditional low-order initial guesses. This leads to a substantial reduction in the initial error of the SDC method.

The convergence analysis shows that the total error consists of two terms: one associated with the SDC iterations and another determined by the quality of the neural network approximation. As a result, high-order accuracy can be achieved with fewer SDC sweeps when the PINN approximation is sufficiently accurate.

Overall, the proposed PINN-SDC framework combines the global approximation capability of machine learning with the rigorous error control of classical numerical methods. This makes it particularly attractive for nonlinear problems where traditional initial guesses may be inadequate.

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. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Front. Appl. Math. 16, SIAM, Philadelphia, 1995, https://doi.org/10.1137/1.9781611970944.

11. 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.
12. M. Raissi, P. Perdikaris, G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics, 378, 2019, https://doi.org/10.1016/j.jcp.2018.10.045.
13. 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.
14. 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.