

Article

Iterated Crank–Nicolson Method for Peridynamic Models

Jinjie Liu ^{1,*}, Samuel Appiah-Adjei ¹ and Moysey Brio ²

¹ Division of Physics, Engineering, Mathematics, and Computer Science, Delaware State University, Dover, DE 19901, USA; sappiahadjei18@students.desu.edu

² Department of Mathematics, The University of Arizona, Tucson, AZ 85721, USA; brio@math.arizona.edu

* Correspondence: jliu@desu.edu

Abstract: In this paper, we explore the iterated Crank–Nicolson (ICN) algorithm for the one-dimensional peridynamic model. The peridynamic equation of motion is an integro-differential equation that governs structural deformations such as fractures. The ICN method was originally developed for hyperbolic advection equations. In peridynamics, we apply the ICN algorithm for temporal discretization and the midpoint quadrature method for spatial integration. Several numerical tests are carried out to evaluate the performance of the ICN method. In general, the ICN method demonstrates second-order accuracy, consistent with the Störmer–Verlet (SV) method. When the weight is 1/3, the ICN method behaves as a third-order Runge–Kutta method and maintains strong stability-preserving (SSP) properties for linear problems. Regarding energy conservation, the ICN algorithm maintains at least second-order accuracy, making it superior to the SV method, which converges linearly. Furthermore, selecting a weight of 0.25 results in fourth-order superconvergent energy variation for the ICN method. In this case, the ICN method exhibits energy variation similar to that of the fourth-order Runge–Kutta method but operates approximately 20% faster. Higher-order convergence for energy can also be achieved by increasing the number of iterations in the ICN method.

Keywords: peridynamics; nonlocal wave equation; iterated Crank–Nicolson method; Störmer–Verlet method; superconvergence

MSC: 65M06; 65M12; 74A70



Citation: Liu, J.; Appiah-Adjei, S.; Brio, M. Iterated Crank–Nicolson Method for Peridynamic Models. *Dynamics* **2024**, *4*, 192–207. <https://doi.org/10.3390/dynamics4010011>

Academic Editor: Christos Volos

Received: 16 February 2024

Revised: 8 March 2024

Accepted: 10 March 2024

Published: 15 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Peridynamics, introduced by Stewart Silling in 2000 [1], is a nonlocal theory designed to model structural deformations such as fractures [2–4]. Its applications extend beyond mechanics and material science to encompass fields like biology and multi-physics problems [5]. The peridynamic equation of motion is an integro-differential equation governing nonlocal wave interaction. Unlike classical continuum mechanics, which relies on partial differential equations and faces challenges in handling discontinuities, the peridynamic model employs summation of bond forces over a finite region known as the horizon. This unique feature enables the effective handling of discontinuous solutions. Numerical methods for solving nonlocal wave equations involve temporal discretization of derivatives and numerical integration in the spatial domain. Various numerical algorithms have been adopted into peridynamics, including the meshfree method [6], quadrature methods [7,8], the finite element method [9], the discontinuous Galerkin method [10], and spectral methods [11–14].

The Störmer–Verlet (SV) algorithm [15] is a popular temporal discretization method for peridynamics, implemented in software packages like PDMATLAB2D [16]. The SV method offers advantages such as time reversal and symplectic properties, leading to total energy conservation in a Hamiltonian system. However, as noted by [17], the magnitude of

the energy variation being greater than zero is influenced by the choice of the numerical quadrature method.

The Crank–Nicolson (CN) algorithm [18] is another widely used numerical method capable of preserving the total energy of a Hamiltonian system. The CN method is second-order accurate, implicit, and unconditionally stable. Because it is implicit, the discretized equations can be solved using iterations, leading to the explicit iterated Crank–Nicolson (ICN) algorithm [19]. Originally developed for hyperbolic advection equations with applications in numerical relativity [20–22], the ICN method has been extended to other fields, including beam propagation equations [23,24] and Maxwell’s equations for electromagnetic waves [25,26].

In this paper, we explore the ICN method for peridynamics. We employ the weighted ICN algorithm for temporal discretization and the midpoint quadrature method to evaluate the integral in the spatial domain. Teukolsky [20] discovered that the number of iterations does not affect the convergence rate of the ICN method for hyperbolic advection equations. As long as the number of iterations is greater than one, the method maintains second-order accuracy. Nevertheless, employing a large number of iterations can enhance its energy conservation property, as it gradually converges to the energy-conserving solution of the CN method. To enhance stability, the ICN method is generalized by introducing a weight θ , leading to the θ -ICN method [27]. When the weight equals 0.5, the method reverts to the original ICN method. However, the θ -ICN method degrades to first-order accuracy when θ is not equal to 0.5. In another study [28], the θ -ICN method is refined to achieve second-order accuracy by employing two different weights in consecutive iterations, with the geometric mean of the two weights set to 0.5. Our investigation of the ICN method for peridynamics aims to identify an efficient method with enhanced performance in terms of energy conservation.

The paper is organized as follows: In Section 2, we review the peridynamic equation and its corresponding numerical algorithms. In Section 3, we present the ICN method. Numerical examples are presented in Section 4, followed by the conclusion.

2. Peridynamic Model

In this section, we briefly review the peridynamic equation and the corresponding numerical methods.

Consider a one-dimensional homogeneous and infinitely long bar. The peridynamic equation of motion can be written as

$$\rho(x)u_{tt}(x, t) = \int_{-\infty}^{\infty} f(\tilde{x} - x, u(\tilde{x}, t) - u(x, t))d\tilde{x} + b(x, t), \quad (1)$$

where u , ρ , b , and f represent the displacement field, the density, the external force, and the pairwise force function, respectively. The limits of the integral in (1) are usually replaced with finite numbers by ignoring the interaction between particles with a distance greater than the horizon δ , or equivalently, $f = 0$ when $|\tilde{x} - x| > \delta$. Equation (1) can be written as

$$u_{tt} = A(x, u) + B(x, t), \quad (2)$$

and a system of first-order equations

$$u_t = v, \quad (3)$$

$$v_t = A(x, u) + B(x, t), \quad (4)$$

where $A(x, u) = \frac{1}{\rho} \int_{-\infty}^{\infty} f(\tilde{x} - x, u(\tilde{x}, t) - u(x, t))d\tilde{x}$ and $B = b/\rho$. If $A(x, u)$ is linear, then it can be written as $A(x, u) = \bar{A}u$, where \bar{A} is a matrix that does not depend on u . For the linear case, the total energy of the system is given by [17]

$$E = \frac{1}{2}v^T v - \frac{1}{2}u^T \bar{A}u - u^T B. \quad (5)$$

By viewing the system (3) and (4) as a Hamiltonian system, we see that E is the Hamiltonian (energy), so it is a conserved quantity.

The system of peridynamic Equations (3) and (4) can be written as a vector equation:

$$U_t = F(U), \quad (6)$$

where $U = (u, v)^T$ and $F(U) = (v, A(x, u) + B)^T$. In the linear case, $A(x, u) = \bar{A}u$, and we obtain

$$U_t = LU, \quad (7)$$

where

$$L = \begin{pmatrix} 0 & I \\ \bar{A} & 0 \end{pmatrix}. \quad (8)$$

A numerical algorithm for the system (3) and (4) involves the evaluation of the integral $A(x, u)$ in the spatial domain and the temporal discretization of the derivatives u_t and v_t . Using the Störmer–Verlet (SV) algorithm, we obtain the following discretized equations:

$$v^{n+1/2} = v^n + \frac{\Delta t}{2} [A(x, u^n) + B(x, t_n)], \quad (9)$$

$$u^{n+1} = u^n + \Delta t v^{n+1/2}, \quad (10)$$

$$v^{n+1} = v^{n+1/2} + \frac{\Delta t}{2} [A(x, u^{n+1}) + B(x, t_{n+1})]. \quad (11)$$

Another popular method is the Runge–Kutta (RK) algorithm. A general s -stage RK method for Equation (6) can be written in the following standard form:

$$k_i = \Delta t F\left(U^n + \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad i = 1, 2, \dots, s \quad (12)$$

$$U^{n+1} = U^n + \sum_{i=1}^s b_i k_i, \quad (13)$$

where a_{ij} and b_i are the coefficients in the Butcher tableau [29]. In particular, when $s = 4$, $a_{21} = a_{32} = 1/2$, $a_{43} = 1$, $a_{31} = a_{41} = a_{42} = 0$, $b_1 = b_4 = 1/6$, and $b_2 = b_3 = 1/3$, we obtain the classical fourth-order RK (RK4) method [30]

$$k_1 = \Delta t F(U^n), \quad (14)$$

$$k_2 = \Delta t F\left(U^n + \frac{1}{2} k_1\right), \quad (15)$$

$$k_3 = \Delta t F\left(U^n + \frac{1}{2} k_2\right), \quad (16)$$

$$k_4 = \Delta t F(U^n + k_3), \quad (17)$$

$$U^{n+1} = U^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (18)$$

To evaluate the integral $A(x, u)$, a commonly used method is the quadrature algorithm. For example, using the midpoint rule, we have

$$A(x, u) = \frac{1}{\rho} \int_{-\infty}^{\infty} f(\tilde{x} - x, u(\tilde{x}, t) - u(x, t)) d\tilde{x} \approx \frac{1}{\rho} \sum_{j=1}^N f_j \Delta x, \quad (19)$$

where $f_j = f(x_j - x, u(x_j, t) - u(x, t))$. In the linear case, using the midpoint method, the matrix \bar{A} is symmetric and negative definite [17]. Other Newton–Cotes quadrature rules include the composite trapezoidal rule

$$A(x, u) \approx \frac{1}{2\rho} \left(f_1 + 2 \sum_{j=2}^{N-1} f_j + f_N \right) \Delta x, \quad (20)$$

and the composite Simpson's rule

$$A(x, u) \approx \frac{1}{3\rho} \left(f_1 + 4 \sum_{j=1}^{(N-1)/2} f_{2j} + 2 \sum_{j=1}^{(N-3)/2} f_{2j+1} + f_N \right) \Delta x, \quad (21)$$

where $N \geq 3$ must be an odd number for Simpson's rule.

In general, the SV method (9)–(11) is second-order in time, and the midpoint quadrature is second-order in space. Consequently, the resulting midpoint SV method attains second-order accuracy in both space and time. The SV method exhibits time reversal and symplectic properties. Therefore, it preserves the total energy of the Hamiltonian system.

In contrast, although commonly used RK methods have orders larger than two, they do not preserve the total energy. Let E_0 represent the energy at the initial time $t = 0$ and E_n the energy at time step t_n . For an energy-conserving method, we should have $|E_n - E_0| = 0$ (or of the order of the machine round-off error), where $|E_n - E_0|$ is the total energy variation. However, it has been reported in [17] that $|E_n - E_0| > 0$ for the SV method.

3. The Iterated Crank–Nicolson Method

In [19,20,27,28], the iterated Crank–Nicolson (ICN) and the weighted ICN algorithms are developed as predictor–corrector methods for hyperbolic partial differential equations. The peridynamic equation is a second-order integro-differential Equation (1), which can be expressed as a system of first-order differential Equations (3) and (4) and in its vector form (6).

In this section, we start with the Crank–Nicolson (CN) method for system of first-order differential equations and derive the ICN method as an iterative solver for the discretized equations from the CN algorithm.

There are two versions of the CN algorithms for the system (6):

$$U^{n+1} = U^n + \Delta t \frac{F(U^n) + F(U^{n+1})}{2}, \quad (22)$$

and

$$U^{n+1} = U^n + \Delta t F\left(\frac{U^n + U^{n+1}}{2}\right), \quad (23)$$

where U^n and U^{n+1} represent the solution at time $t_n = n\Delta t$ and $t_{n+1} = (n+1)\Delta t$, respectively. If the function F is linear in terms of the variable U , then Equations (22) and (23) are equivalent.

The implicit nature of the method requires the solution of a system of linear or nonlinear equations. When solved by iterations, it leads to the ICN method. In [28], the ICN method is developed using a predictor–corrector strategy, and it can also be derived using the CN Equation (23). Here, we use the CN Equation (22) to demonstrate the ICN method, and the other one can be derived following a similar procedure. The iterative solver is a fixed-point iteration. Replacing U^{n+1} on the right-hand side of Equation (22) with the k -th approximation U_k and letting the initial guess $U_0 = U^n$, we obtain the following ICN solver:

$$U_0 = U^n, \quad (24)$$

$$U_{k+1} = U^n + \Delta t \left(\frac{F(U^n) + F(U_k)}{2} \right), \quad k = 0, 1, 2, \dots, m \quad (25)$$

$$U^{n+1} = U_{m+1}, \quad (26)$$

where m is the number of iterations.

In the ICN Equations (24)–(26), if we use a weight θ when averaging $F(U_k)$ and $F(U^n)$ in Equation (25), then we obtain the θ -ICN method:

$$U_0 = U^n, \quad (27)$$

$$U_{k+1} = U^n + \Delta t[(1 - \theta)F(U^n) + \theta F(U_k)], \quad k = 0, 1, 2, \dots, m \quad (28)$$

$$U^{n+1} = U_{m+1}. \quad (29)$$

For hyperbolic equations, the stability region of the θ -ICN method depends on the weight θ . As a result, by changing the weight θ , we can obtain more stable solutions. However, the θ -ICN method is only first-order accurate [27]. In [28], the θ -ICN method is improved to second-order accuracy by employing different weights at consecutive steps where the number of iterations is fixed at two. The modified θ -ICN method can be written as

$$U_1 = U^n + \Delta t F(U^n), \quad (30)$$

$$U_2 = U^n + 2\theta \Delta t[(1 - \theta)F(U^n) + \theta F(U_1)], \quad (31)$$

$$U^{n+1} = U^n + \Delta t[(1 - \tilde{\theta})F(U^n) + \tilde{\theta}F(U_2)], \quad (32)$$

and it is second-order accurate when $\tilde{\theta} = 0.25/\theta$. This method is referred as the geometric-averaging (GA) θ -ICN method as the geometric mean of θ and $\tilde{\theta}$ is 0.5. When $\theta = 0.5$, it becomes the original ICN method. For simplicity, we call it the ICN method in the proceedings part of the paper.

The ICN algorithm (30)–(32) is a three-stage RK method and can be written in standard form as:

$$k_1 = \Delta t F(U^n), \quad (33)$$

$$k_2 = \Delta t F(U^n + k_1), \quad (34)$$

$$k_3 = \Delta t F(U^n + 2\theta(1 - \theta)k_1 + 2\theta^2 k_2), \quad (35)$$

$$U^{n+1} = U^n + (1 - \tilde{\theta})k_1 + \tilde{\theta}k_3. \quad (36)$$

A special case is when $\theta = 1/3$:

$$U_1 = U^n + \Delta t F(U^n), \quad (37)$$

$$U_2 = U^n + \frac{4}{9}\Delta t F(U^n) + \frac{2}{9}\Delta t F(U_1), \quad (38)$$

$$U^{n+1} = U^n + \frac{1}{4}\Delta t F(U^n) + \frac{3}{4}F(U_2), \quad (39)$$

and it is a third-order RK method as it satisfies the criteria given in Gottlieb and Shu's paper [31]. Furthermore, if F is linear, this method can further be written as

$$U^{(1)} = U^n + \Delta t F(U^n), \quad (40)$$

$$U^{(2)} = U^{(1)} + \Delta t F(U^{(1)}), \quad (41)$$

$$U^{n+1} = \frac{1}{3}U^n + \frac{1}{2}U^{(1)} + \frac{1}{6}(U^{(2)} + \Delta t F(U^{(2)})), \quad (42)$$

which makes it a strong stability-preserving (SSP) method [32].

4. Numerical Examples

In this section, we evaluate the performance of the ICN method for peridynamics through a series of numerical simulations. Our computer codes were implemented in MATLAB and executed on a 64-bit computer equipped with a 12th generation Intel(R) Core(TM) i7-12700 CPU @ 2.10GHz.

For simplicity, we set $b(x, t) = 0$ and $\rho(x) = 1$. The initial condition is $u_0(x) = \exp(-x^2)$ and $v_0 = 0$ unless otherwise specified. We consider the following pairwise force function:

$$f(\tilde{x} - x, u(\tilde{x}, t) - u(x, t)) = C(\tilde{x} - x)(u(\tilde{x}, t) - u(x, t))^r, \quad (43)$$

where the micromodulus function C is given by:

$$C(x) = \frac{4}{\sqrt{\pi}} e^{-x^2}. \quad (44)$$

The problem is linear when $r = 1$ and nonlinear otherwise. We use a finite horizon δ , so that $C(x) = 0$, if $|x| > \delta$.

The computational spatial domain $[L_x, U_x]$ is divided into N grid cells so $\Delta x = (U_x - L_x)/N$. We let the solution be defined at the center of each cell, so $U_j^n = U(x_j, n\Delta t)$ where $x_j = L_x + (j - \frac{1}{2})\Delta x$, $j = 1, 2, \dots, N$. Similar to [17], a periodic boundary condition is implemented to approximate the original problem of an infinitely long bar. The simulation stops before the solution reaches the boundaries.

To test the rate of convergence, we compute the L_2 error norm

$$\epsilon_{L_2} = \frac{\sqrt{\sum_{j=1}^N |u(x_j, t) - u_e(x_j, t)|^2}}{N}, \quad (45)$$

and the L_∞ error norm

$$\epsilon_{L_\infty} = \max_j |u(x_j, t) - u_e(x_j, t)|, \quad (46)$$

where u_e is the exact solution.

In our simulations, we primarily test the ICN methods with three weights: $\theta = 0.5$, $1/3$, and 0.25 . The original ICN method corresponds to the case when $\theta = 0.5$. The number of iterations for the ICN method is fixed at two unless specified.

4.1. Linear Peridynamic Equation

In the linear case where $r = 1$, the computational domain is chosen to be $[-20, 20]$, and the mesh size is N . We let $\delta = 5$ and $\Delta t = \Delta x$. The exact solution is given by [33]

$$u_e(x, t) = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-s^2} \cos(2sx) \cos\left(2t\sqrt{1-e^{-s^2}}\right) ds. \quad (47)$$

The solutions of the ICN method ($\theta = 0.5$) at four time instances are shown in Figure 1. The results are very similar for other values of θ and the SV method. The grid size is $N = 800$. We can see that the numerical solutions agree with the exact solutions very well. Table 1 shows the rates of convergence of the solutions measured in the L_2 and L_∞ norms, together with the energy variation $|E_n - E_0|$. In terms of L_2 and L_∞ norms, the SV method converges quadratically, the same as the ICN methods with $\theta = 0.5$ and 0.25 . When $\theta = 1/3$, the ICN method is third-order.

A symplectic method preserves the total energy, meaning the energy variation $|E_n - E_0| = 0$ is ideally zero or is on the order of the machine round-off error. However, as shown in Figure 2, the energy variation of the SV method is approximately 10^{-3} , and as illustrated in Table 1, besides being zero, it converges linearly. In the case of the ICN method with $\theta = 0.5$, the magnitude of energy variation grows and reaches approximately 10^{-2} when $t = 10$. On the other hand, when $\theta = 0.25$, the energy variation grows much more slowly, with its magnitude around 10^{-6} , several orders of magnitude lower than the other two methods. From Table 1, we have the following observations regarding the convergence of the energy variation. Firstly, the ICN method is at least second-order accurate, surpassing the linear convergence of the SV method. Secondly, when $\theta = 0.5$, both the energy and the solution exhibit second-order convergence. Thirdly, when $\theta = 1/3$, the energy variation

converges quadratically, despite its third-order solution. Lastly, when $\theta = 0.25$, the ICN method achieves fourth-order superconvergence.

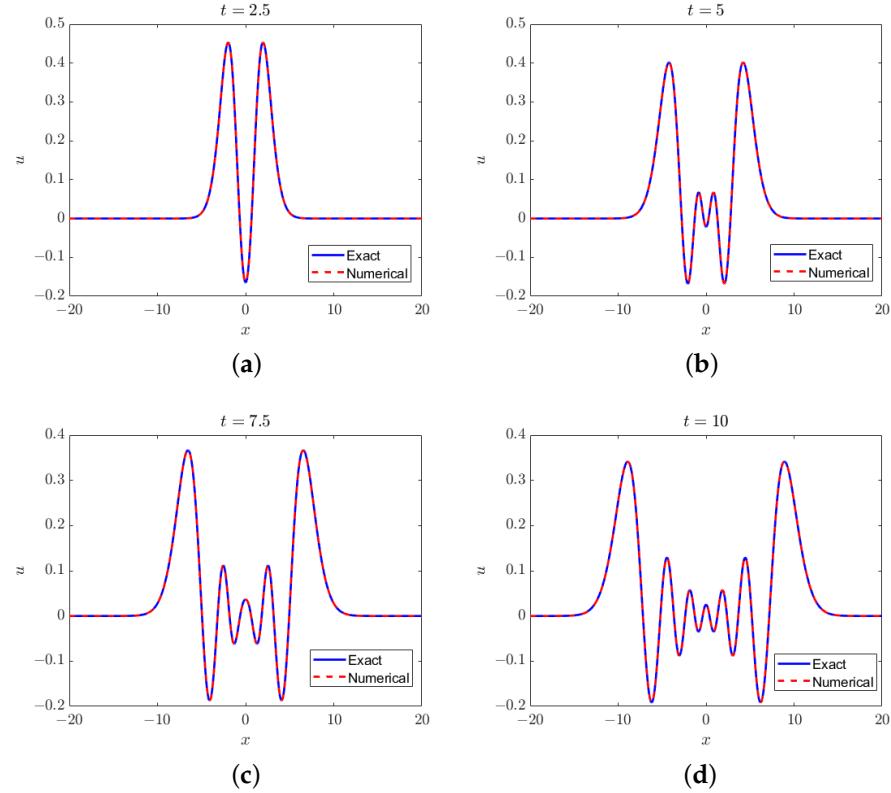


Figure 1. Solutions to the linear peridynamic equation at (a) $t = 2.5$, (b) $t = 5$, (c) $t = 7.5$, and (d) $t = 10$. Method: the ICN method with $\theta = 0.5$. Mesh size $N = 800$.

Table 1. Comparison of the SV method and ICN methods with various weights for the linear peridynamic equation. ϵ_{L_2} and ϵ_{L_∞} represent the numerical errors measured in the L_2 and L_∞ norms, respectively. $|E_n - E_0|$ denotes the energy variation from $t = 0$ to $t = 10$.

Method	N	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate	$ E_n - E_0 $	Rate
SV	100	1.36×10^{-3}	-	3.65×10^{-2}	-	3.67×10^{-2}	-
	200	2.34×10^{-4}	2.54	8.84×10^{-3}	2.05	1.83×10^{-2}	1.00
	400	4.10×10^{-5}	2.51	2.19×10^{-3}	2.01	9.18×10^{-3}	1.00
	800	7.24×10^{-6}	2.50	5.47×10^{-4}	2.00	4.59×10^{-3}	1.00
	1600	1.28×10^{-6}	2.50	1.37×10^{-4}	2.00	2.30×10^{-3}	1.00
ICN ($\theta = 0.5$)	100	2.34×10^{-3}	-	6.18×10^{-2}	-	3.87×10^{-1}	-
	200	4.67×10^{-4}	2.32	1.76×10^{-2}	1.81	1.45×10^{-1}	1.42
	400	8.27×10^{-5}	2.50	4.42×10^{-3}	2.00	3.89×10^{-2}	1.90
	800	1.45×10^{-5}	2.51	1.10×10^{-3}	2.01	9.84×10^{-3}	1.98
	1600	2.56×10^{-6}	2.50	2.74×10^{-4}	2.00	2.47×10^{-3}	2.00
ICN ($\theta = \frac{1}{3}$)	100	7.59×10^{-4}	-	2.11×10^{-2}	-	1.60×10^{-1}	-
	200	7.30×10^{-5}	3.38	2.79×10^{-3}	2.92	4.99×10^{-2}	1.68
	400	6.57×10^{-6}	3.47	3.56×10^{-4}	2.97	1.30×10^{-2}	1.94
	800	5.83×10^{-7}	3.49	4.45×10^{-5}	3.00	3.28×10^{-3}	1.99
	1600	5.16×10^{-8}	3.50	5.57×10^{-6}	3.00	8.22×10^{-4}	2.00
ICN ($\theta = 0.25$)	100	1.45×10^{-3}	-	3.91×10^{-2}	-	1.58×10^{-2}	-
	200	2.38×10^{-4}	2.61	8.99×10^{-3}	2.12	9.70×10^{-4}	4.02
	400	4.12×10^{-5}	2.53	2.20×10^{-3}	2.03	6.06×10^{-5}	4.00
	800	7.25×10^{-6}	2.51	5.48×10^{-4}	2.00	3.79×10^{-6}	4.00
	1600	1.28×10^{-6}	2.50	1.37×10^{-4}	2.00	2.37×10^{-7}	4.00

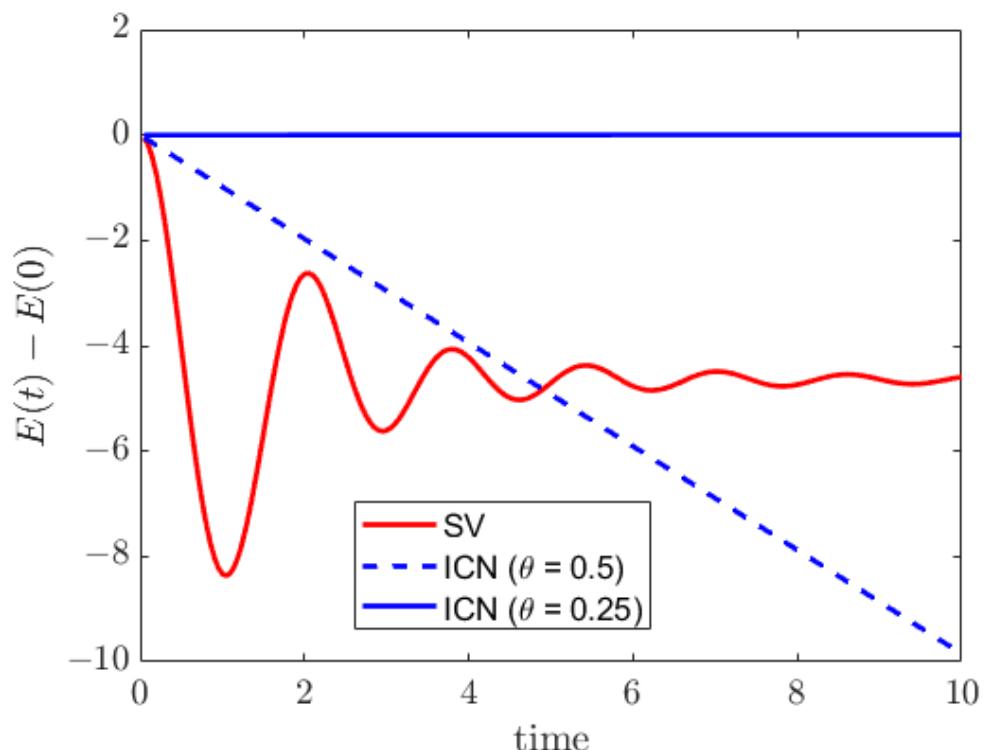


Figure 2. Time history of the total energy variation $E(t) - E(0)$. The grid size is $N = 800$.

Table 2 compares the SV, the ICN with $\theta = 0.25$, and the classical RK4 methods. For RK4, both the solution and the energy variation have fourth-order convergence, making it one of the optimal choices for temporal discretization. The energy variation of the SV method is four or five orders of magnitude larger than the other two methods. The ICN method operates approximately 30% slower than the SV method due to the additional evaluation of integrals in the updating equations. The ICN and RK4 methods yield similar energy variation for both resolutions $N = 1600$ and $N = 3200$, while the ICN solver runs approximately 20% faster than the RK4 method at both resolutions.

Table 2. Comparison of energy variation and run time for three methods (SV, ICN, and RK4).

Method	$ E_n - E_0 (N = 1600)$	Run Time (s)	$ E_n - E_0 (N = 3200)$	Run Time (s)
SV	2.30×10^{-3}	20	1.15×10^{-3}	139
RK4	2.10×10^{-7}	33	1.32×10^{-8}	236
ICN ($\theta = 0.25$)	2.37×10^{-7}	26	1.48×10^{-8}	184

4.2. Energy Conservation for the ICN Method with Different Weights

In this test, we investigate the relationship between the energy variation and the weight θ in the ICN method. The simulation setup remains consistent with the previous section, where the grid size is fixed at $N = 800$. The results presented in Table 3 demonstrate that as the weight θ decreases from 0.7 to 0.2, the energy variation $E_n - E_0$ increases from -1.7×10^{-2} to 1.9×10^{-3} . Notably, when $\theta = 0.25$, the energy variation reaches its minimum amplitude of 3.7×10^{-6} .

Table 3. Energy variation of the ICN method with different weights.

θ	$E_n - E_0$
0.70	-1.77×10^{-2}
0.60	-1.38×10^{-2}
0.50	-9.84×10^{-3}
0.40	-5.91×10^{-3}
1/3	-3.28×10^{-3}
0.30	-1.97×10^{-3}
0.25	3.79×10^{-6}
0.20	1.98×10^{-3}

4.3. The ICN Method with Multiple Iterations

In this example, we study the original ICN method ($\theta = 0.5$) with a varying number of iterations. Firstly, we examine the relationship between the energy variation and the number of iterations in the ICN method. The simulation setup remains consistent with Section 4.1. We consider two grid sizes: $N = 800$ and 1600 . As shown in Table 4, as m increases, the energy variation decreases. For $N = 800$ and $N = 1600$, the energy variations reach the order of the machine round-off error of 10^{-15} when m is greater than 10 and 8, respectively. This example illustrates that the ICN method can preserve energy up to machine precision by employing a large number of iterations. Furthermore, when the grid size increases, the energy variation reaches machine precision at a faster rate. For $N = 1600$, achieving the order of 10^{-7} requires 3 iterations, with a CPU run time of approximately 37 s. This runtime is not substantially greater than that required for the ICN method with $\theta = 0.25$ (26 s) and the RK4 method (33 s) in the previous test, as shown in Table 2.

Table 4. Energy variation of the ICN method when the number of iterations changes. E_n is the total energy at $t = 10$.

Number of Iterations m	$E_n - E_0 (N = 800)$	Run Time (s)	$E_n - E_0 (N = 1600)$	Run Time (s)
1	9.88×10^{-3}	2.65	2.47×10^{-3}	20.44
2	-9.84×10^{-3}	3.56	-2.47×10^{-3}	29.37
3	-1.51×10^{-5}	4.37	-9.47×10^{-7}	37.64
4	1.51×10^{-5}	5.23	9.47×10^{-7}	44.99
5	2.54×10^{-8}	6.07	3.98×10^{-10}	52.17
6	-2.54×10^{-8}	6.92	-3.98×10^{-10}	58.95
7	-4.55×10^{-11}	7.72	-1.78×10^{-13}	66.40
8	4.55×10^{-11}	8.58	1.78×10^{-13}	72.49
9	8.53×10^{-14}	9.45	$\sim 10^{-15}$	80.32
10	-8.70×10^{-14}	10.27	$\sim 10^{-15}$	89.71
11	$\sim 10^{-15}$	11.15	$\sim 10^{-15}$	95.04
12	$\sim 10^{-15}$	11.97	$\sim 10^{-15}$	101.81

From the previous multi-iteration test, we observe that when the number of iterations is three, the original ICN method exhibits an energy variation at the same order of magnitude as the RK4 method. This motivates us to conduct a convergence test for the three-iteration ICN method. The results are presented in Table 5. We find that the three-iteration ICN method demonstrates second-order convergence in terms of the L_2 and L_∞ error norms, and fourth-order convergence in terms of energy conservation. Since both the three-iteration ICN and the RK4 methods are four-stage methods, they have comparable CPU run times.

Furthermore, we conduct a series of simulations to examine the rate of convergence of the ICN method with varying numbers of iterations, as shown in Table 6. We observe that as the number of iterations increases, the rate of convergence also increases. When the number of iterations is an even number, the rate of convergence remains the same as the

number of iterations. However, for an odd number of iterations m , the rate of convergence is $m + 1$. These results indicate that a higher order of convergence (in terms of energy conservation) can be achieved by increasing the number of iterations in the ICN method.

Table 5. The three-iteration ICN method ($\theta = 0.5$) for linear peridynamic equation.

N	L_2	Rate	L_∞	Rate	$ E_n - E_0 $	Rate
100	3.11×10^{-3}	-	7.78×10^{-2}	-	-5.22×10^{-2}	-
200	4.90×10^{-4}	2.66	1.86×10^{-2}	2.06	-3.77×10^{-3}	3.79
400	8.31×10^{-5}	2.56	4.44×10^{-3}	2.07	-2.41×10^{-4}	3.97
800	1.45×10^{-5}	2.52	1.10×10^{-3}	2.02	-1.51×10^{-5}	3.99
1600	2.56×10^{-6}	2.50	2.74×10^{-4}	2.00	-9.47×10^{-7}	4.00

Table 6. Energy convergence rate as a function of the number of iterations in the ICN method.

Number of Iterations m	Rate of Convergence of Energy
1	2.0
2	2.0
3	4.0
4	4.0
5	6.0
6	6.0
7	8.0
8	8.0

4.4. Comparison of Three Quadrature Formulas for Linear Peridynamics

As shown in [17], the choice of quadrature formula for integration impacts the accuracy of energy conservation. In this example, we examine three different Newton–Cotes quadrature formulas: the midpoint rule (Equation (19)), the composite trapezoidal rule (Equation (20)), and the composite Simpson’s rule (Equation (21)). For Simpson’s rule, we adjust the grid size to $N + 1$, as N is an even number in our simulations. Results are presented in Tables 7 and 8. We do not observe significant differences among the three quadrature formulas. All three methods exhibit the same order of convergence. The L_2 , L_∞ , and energy variation are of comparable magnitudes across all three methods, with the differences being negligible. This example suggests that the degree of the Newton–Cotes quadrature formulas does not significantly affect the convergence rate.

Table 7. Comparison of three quadrature formulas for the peridynamic simulation using the SV method. ϵ_{L_2} and ϵ_{L_∞} represent the numerical errors measured in the L_2 and L_∞ norms, respectively. $|E_n - E_0|$ denotes the energy variation from $t = 0$ to $t = 10$.

Method	N	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate	$ E_n - E_0 $	Rate
SV + midpoint	100	$1.3626781938 \times 10^{-3}$	-	$3.6546374629 \times 10^{-2}$	-	$-3.6686416132 \times 10^{-2}$	-
	200	$2.3386240754 \times 10^{-4}$	2.54	$8.8420752239 \times 10^{-3}$	2.05	$-1.8341493309 \times 10^{-2}$	1.00
	400	$4.1038150512 \times 10^{-5}$	2.51	$2.1896453314 \times 10^{-3}$	2.01	$-9.1848205222 \times 10^{-3}$	1.00
	800	$7.2412556153 \times 10^{-6}$	2.50	$5.4733686142 \times 10^{-4}$	2.00	$-4.5945520876 \times 10^{-3}$	1.00
	1600	$1.2794967586 \times 10^{-6}$	2.50	$1.3681318781 \times 10^{-4}$	2.00	$-2.2975549644 \times 10^{-3}$	1.00
SV + composite trapezoidal	100	$1.3626781938 \times 10^{-3}$	-	$3.6546374631 \times 10^{-2}$	-	$-3.6686416132 \times 10^{-2}$	-
	200	$2.3386240741 \times 10^{-4}$	2.54	$8.8420752309 \times 10^{-3}$	2.05	$-1.8341493309 \times 10^{-2}$	1.00
	400	$4.1038150469 \times 10^{-5}$	2.51	$2.1896453350 \times 10^{-3}$	2.01	$-9.1848205222 \times 10^{-3}$	1.00
	800	$7.2412556001 \times 10^{-6}$	2.50	$5.4733686322 \times 10^{-4}$	2.00	$-4.5945520876 \times 10^{-3}$	1.00
	1600	$1.2794967533 \times 10^{-6}$	2.50	$1.3681318870 \times 10^{-4}$	2.00	$-2.2975549644 \times 10^{-3}$	1.00

Table 7. Cont.

Method	<i>N</i>	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate	$ E_n - E_0 $	Rate
SV + composite Simpson	100	$1.3532560898 \times 10^{-3}$	-	$3.6461913740 \times 10^{-2}$	-	$-3.6637130961 \times 10^{-2}$	-
	200	$2.3386240745 \times 10^{-4}$	2.53	$8.8420752291 \times 10^{-3}$	2.04	$-1.8341493309 \times 10^{-2}$	1.00
	400	$4.1038150476 \times 10^{-5}$	2.51	$2.1896453345 \times 10^{-3}$	2.01	$-9.1848205222 \times 10^{-3}$	1.00
	800	$7.2412556013 \times 10^{-6}$	2.50	$5.4733686308 \times 10^{-4}$	2.00	$-4.5945520876 \times 10^{-3}$	1.00
	1600	$1.2794967535 \times 10^{-6}$	2.50	$1.3681318867 \times 10^{-4}$	2.00	$-2.2975549644 \times 10^{-3}$	1.00

Table 8. Comparison of three quadrature formulas for the peridynamic simulation using the ICN method ($\theta = 0.25$). ϵ_{L_2} and ϵ_{L_∞} represent the numerical errors measured in the L_2 and L_∞ norms, respectively. $|E_n - E_0|$ denotes the energy variation from $t = 0$ to $t = 10$.

Method	<i>N</i>	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate	$ E_n - E_0 $	Rate
ICN + midpoint	100	$1.4531942146 \times 10^{-3}$	-	$3.9071358030 \times 10^{-2}$	-	$1.5796799025 \times 10^{-2}$	-
	200	$2.3797519266 \times 10^{-4}$	2.61	$8.9850019534 \times 10^{-3}$	2.12	$9.7041252646 \times 10^{-4}$	4.02
	400	$4.1222863495 \times 10^{-5}$	2.53	$2.1984031675 \times 10^{-3}$	2.03	$6.0617629614 \times 10^{-5}$	4.00
	800	$7.2494508340 \times 10^{-6}$	2.51	$5.4789636190 \times 10^{-4}$	2.00	$3.7885364605 \times 10^{-6}$	4.00
	1600	$1.2798591070 \times 10^{-6}$	2.50	$1.3684841395 \times 10^{-4}$	2.00	$2.3678339289 \times 10^{-7}$	4.00
ICN + composite trapezoidal	100	$1.4531942145 \times 10^{-3}$	-	$3.9071358029 \times 10^{-2}$	-	$1.5796799025 \times 10^{-2}$	-
	200	$2.3797519254 \times 10^{-4}$	2.61	$8.9850019605 \times 10^{-3}$	2.12	$9.7041252645 \times 10^{-4}$	4.02
	400	$4.1222863452 \times 10^{-5}$	2.53	$2.1984031712 \times 10^{-3}$	2.03	$6.0617629611 \times 10^{-5}$	4.00
	800	$7.2494508188 \times 10^{-6}$	2.51	$5.4789636369 \times 10^{-4}$	2.00	$3.7885364605 \times 10^{-6}$	4.00
	1600	$1.2798591017 \times 10^{-6}$	2.50	$1.3684841484 \times 10^{-4}$	2.00	$2.3678340710 \times 10^{-7}$	4.00
ICN + composite Simpson	100	$1.4433953589 \times 10^{-3}$	-	$3.8867087321 \times 10^{-2}$	-	$1.5770288980 \times 10^{-2}$	-
	200	$2.3797519257 \times 10^{-4}$	2.60	$8.9850019587 \times 10^{-3}$	2.11	$9.7041252645 \times 10^{-4}$	4.02
	400	$4.1222863459 \times 10^{-5}$	2.53	$2.1984031706 \times 10^{-3}$	2.03	$6.0617629612 \times 10^{-5}$	4.00
	800	$7.2494508201 \times 10^{-6}$	2.51	$5.4789636355 \times 10^{-4}$	2.00	$3.7885364570 \times 10^{-6}$	4.00
	1600	$1.2798591019 \times 10^{-6}$	2.50	$1.3684841481 \times 10^{-4}$	2.00	$2.3678339645 \times 10^{-7}$	4.00

4.5. Linear Peridynamic Model with Discontinuous Initial Condition

In this example, we explore our method's capability for handling discontinuities, focusing on discontinuous initial conditions [34,35]. The simulation setup mirrors the previous linear example in Section 4.1, except for the initial condition.

First, we utilize the following piecewise constant function as the initial displacement field u_0 :

$$u_0(x) = \begin{cases} 1, & |x| < 2 \\ 0.5, & 2 \leq |x| < 5 \\ 0, & \text{otherwise.} \end{cases} \quad (48)$$

The initial velocity field v_0 is set to zero. The solutions of the ICN method ($\theta = 0.25$) at four time instances are shown in Figure 3, and the convergence rates are listed in Table 9.

For the second experiment in this section, we apply a piecewise constant function as the initial velocity field v_0 :

$$v_0(x) = \begin{cases} 0.5, & |x| < 2 \\ 0.25, & 2 \leq |x| < 5 \\ 0, & \text{otherwise.} \end{cases} \quad (49)$$

The initial displacement field is a continuous function $u_0(x) = \exp(-x^2)$. The results are presented in Figure 4 and in Table 10. When computing the convergence rates, we utilize the solution obtained on a fine mesh ($N = 3200$) as the reference solution. In both scenarios, the convergence rates are akin to those observed with continuous initial conditions. Specifically, with regard to the L_2 and L_∞ error norms, both the SV method and the ICN method demonstrate quadratic convergence. In terms of energy conservation,

the SV method displays linear convergence, whereas the ICN method with $\theta = 0.25$ achieves fourth-order accuracy.

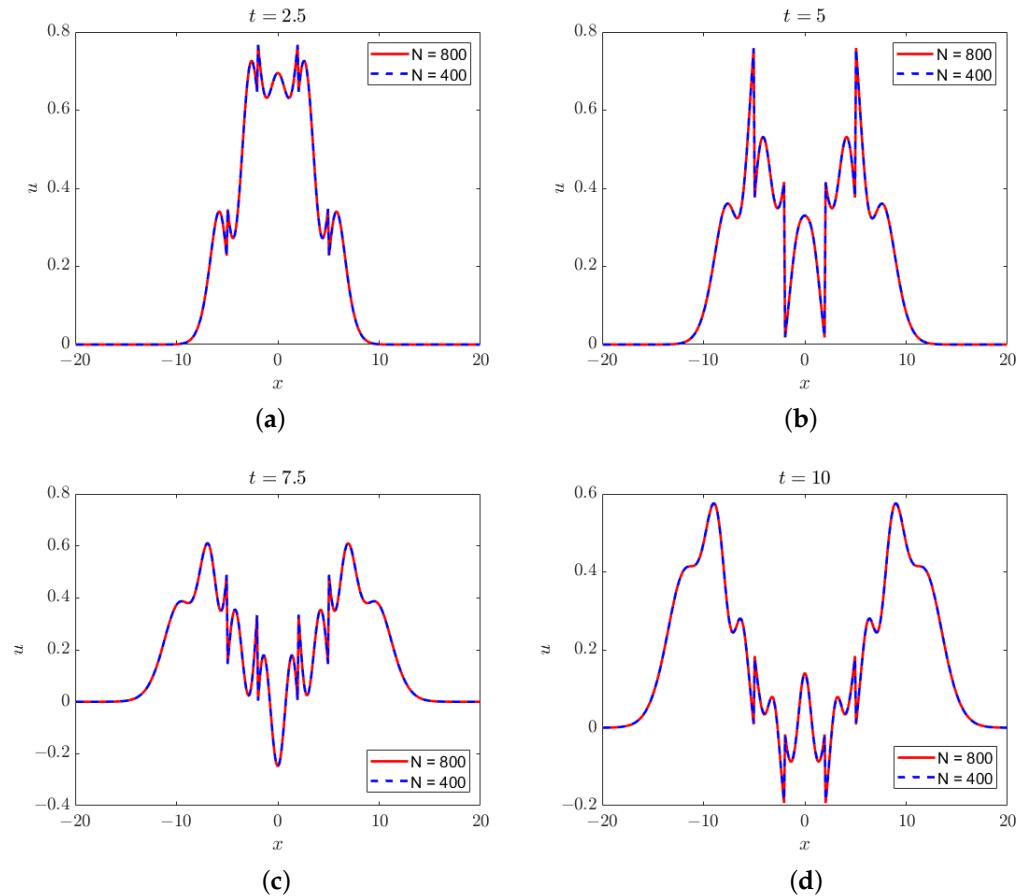


Figure 3. Solutions of the linear peridynamic equation with piecewise constant initial displacement field (48) at (a) $t = 2.5$, (b) $t = 5$, (c) $t = 7.5$, and (d) $t = 10$.

Table 9. Numerical errors and convergence rates of the SV and the ICN methods for the linear peridynamic equation with piecewise constant initial displacement field (48). ϵ_{L_2} and ϵ_{L_∞} represent the numerical errors measured in the L_2 and L_∞ norms, respectively. $|E_n - E_0|$ denotes the energy variation from $t = 0$ to $t = 10$.

Method	N	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate	$ E_n - E_0 $	Rate
SV	100	3.31×10^{-3}	-	1.03×10^{-1}	-	6.40×10^{-2}	-
	200	4.78×10^{-4}	2.79	2.74×10^{-2}	1.91	3.26×10^{-2}	0.97
	400	8.17×10^{-5}	2.55	7.53×10^{-3}	1.86	1.63×10^{-2}	1.00
	800	1.36×10^{-5}	2.58	1.90×10^{-3}	1.99	8.14×10^{-3}	1.00
	1600	1.89×10^{-6}	2.85	3.91×10^{-4}	2.28	4.06×10^{-3}	1.00
ICN ($\theta = 0.25$)	100	3.53×10^{-3}	-	1.09×10^{-1}	-	3.13×10^{-2}	-
	200	4.89×10^{-4}	2.85	2.81×10^{-2}	1.95	2.00×10^{-3}	3.96
	400	8.22×10^{-5}	2.57	7.59×10^{-3}	1.89	1.26×10^{-4}	3.99
	800	1.37×10^{-5}	2.59	1.90×10^{-3}	2.00	7.88×10^{-6}	4.00
	1600	1.89×10^{-6}	2.85	3.91×10^{-4}	2.28	4.92×10^{-7}	4.00

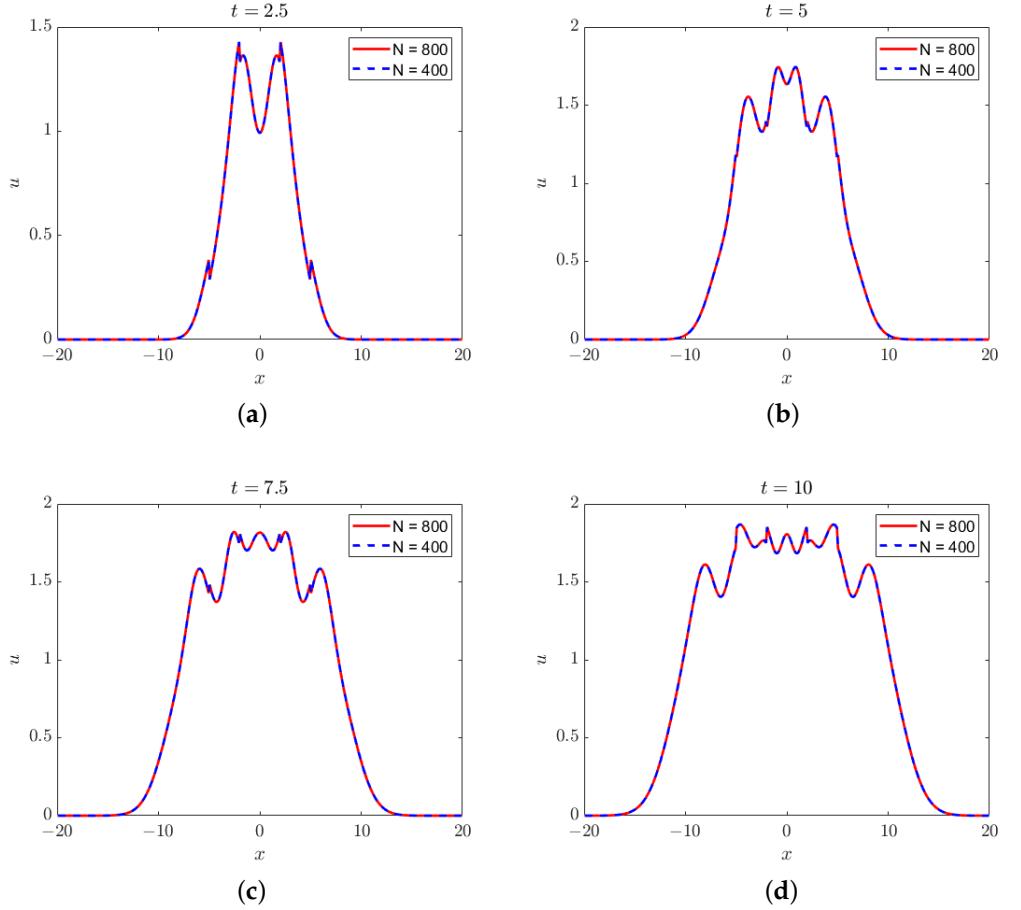


Figure 4. Solutions of the linear peridynamic equation with piecewise constant initial velocity field (49) at (a) $t = 2.5$, (b) $t = 5$, (c) $t = 7.5$, and (d) $t = 10$.

Table 10. Numerical errors and convergence rates of the SV and the ICN methods for the linear peridynamic equation with piecewise constant initial velocity field (49). ϵ_{L_2} and ϵ_{L_∞} represent the numerical errors measured in the L_2 and L_∞ norms, respectively. $|E_n - E_0|$ denotes the energy variation from $t = 0$ to $t = 10$.

Method	N	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate	$ E_n - E_0 $	Rate
SV	100	3.47×10^{-3}	-	8.95×10^{-2}	-	2.99×10^{-2}	-
	200	2.69×10^{-4}	3.69	9.93×10^{-3}	3.17	1.51×10^{-2}	0.99
	400	4.69×10^{-5}	2.52	2.61×10^{-3}	1.93	7.54×10^{-3}	1.00
	800	7.93×10^{-6}	2.56	6.62×10^{-4}	1.98	3.76×10^{-3}	1.00
	1600	1.15×10^{-6}	2.78	1.41×10^{-4}	2.23	1.88×10^{-3}	1.00
ICN ($\theta = 0.25$)	100	3.50×10^{-3}	-	9.14×10^{-2}	-	1.04×10^{-2}	-
	200	2.64×10^{-4}	3.73	9.78×10^{-3}	3.22	6.35×10^{-4}	4.03
	400	4.55×10^{-5}	2.54	2.38×10^{-3}	2.04	3.94×10^{-5}	4.01
	800	7.67×10^{-6}	2.57	5.81×10^{-4}	2.04	2.45×10^{-6}	4.01
	1600	1.11×10^{-6}	2.78	1.25×10^{-4}	2.22	1.53×10^{-7}	4.00

4.6. Nonlinear Peridynamic Equation

For the nonlinear case, we set $r = 3$ in Equation (43). The computational domain spans $[-10, 10]$, with $\Delta t = \Delta x$. The solution obtained through the ICN method is presented in Figure 5, and Table 11 illustrates the convergence rates of these solutions. We use the solution on a fine mesh ($N = 3200$) as the exact solution when computing the convergence rates. From the tabulated data, it is evident that when $\theta = 1/3$, the ICN method demon-

strates third-order convergence. For the other two values of θ , the ICN method converges quadratically, similar to the SV method.

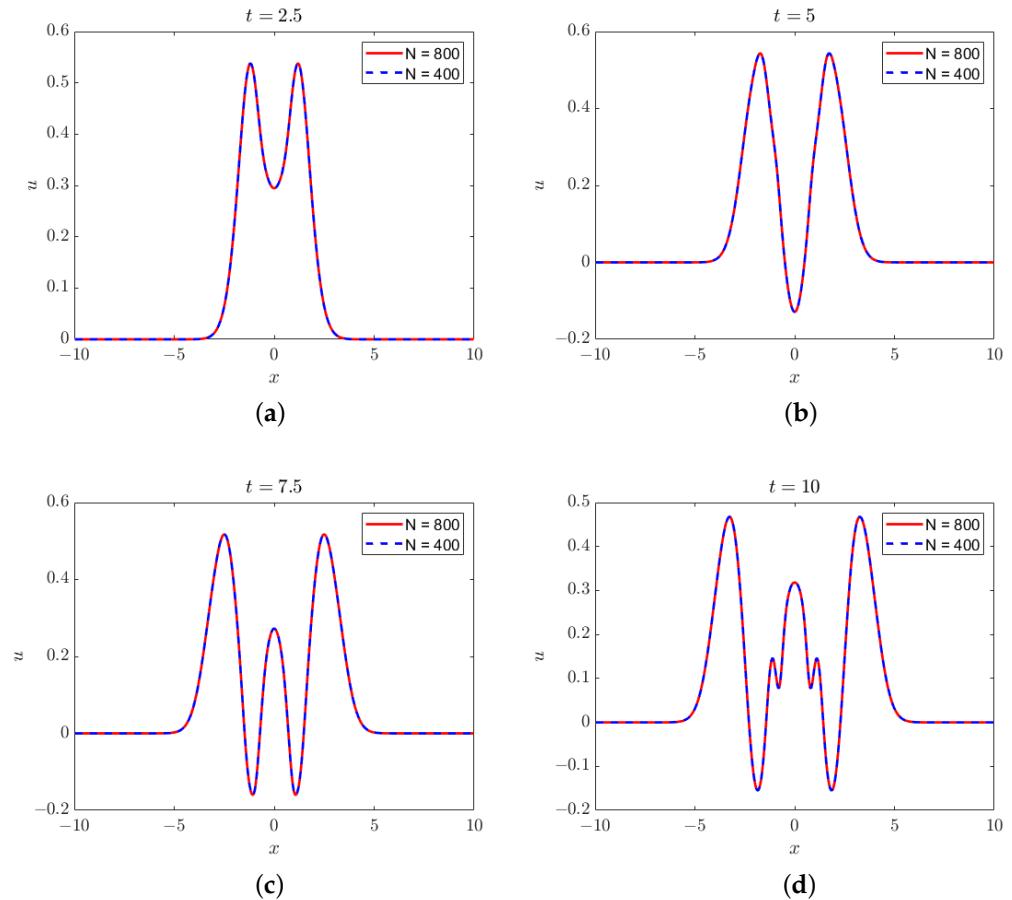


Figure 5. Solutions to the nonlinear peridynamic equation at (a) $t = 2.5$, (b) $t = 5$, (c) $t = 7.5$, and (d) $t = 10$.

Table 11. Comparison of the SV method and the ICN methods with various weights for the nonlinear peridynamic equation. ϵ_{L_2} and ϵ_{L_∞} represent the numerical errors measured in the L_2 and L_∞ norms, respectively.

Method	N	ϵ_{L_2}	Rate	ϵ_{L_∞}	Rate
SV	100	2.12×10^{-4}	-	4.45×10^{-3}	-
	200	2.33×10^{-5}	3.19	7.04×10^{-4}	2.66
	400	4.11×10^{-6}	2.50	1.91×10^{-4}	1.88
	800	7.65×10^{-7}	2.42	6.45×10^{-5}	1.57
ICN ($\theta = 0.5$)	100	2.15×10^{-4}	-	7.65×10^{-2}	-
	200	2.46×10^{-5}	3.13	1.54×10^{-2}	2.31
	400	4.20×10^{-6}	2.55	3.01×10^{-3}	2.36
	800	7.68×10^{-7}	2.45	6.26×10^{-4}	2.27
ICN ($\theta = \frac{1}{3}$)	100	8.39×10^{-4}	-	2.14×10^{-2}	-
	200	8.03×10^{-5}	3.39	2.84×10^{-3}	2.92
	400	7.36×10^{-6}	3.45	3.63×10^{-4}	2.97
	800	6.59×10^{-7}	3.48	5.17×10^{-5}	2.81
ICN ($\theta = 0.25$)	100	3.11×10^{-4}	-	8.13×10^{-3}	-
	200	3.34×10^{-5}	3.22	1.16×10^{-3}	2.81
	400	4.77×10^{-6}	2.81	2.40×10^{-4}	2.27
	800	7.92×10^{-7}	2.59	7.16×10^{-5}	1.75

5. Conclusions

In summary, we utilized the iterated Crank–Nicolson (ICN) algorithms to solve the integro-differential equation for peridynamic models. The weighted ICN method, a generalization of the original ICN method that introduces a weight θ while maintaining at least second-order accuracy, was employed for temporal discretization. Additionally, we employed a midpoint quadrature algorithm for numerical integration in the spatial domain.

A series of numerical simulations were conducted to assess the performance of the ICN method for linear and nonlinear peridynamic equations in one dimension. The results confirmed the convergence rates of the ICN methods with various weights, including $\theta = 0.5, 1/3$, and 0.25 . Additionally, we compared three quadrature formulas and observed similar results for both the Störmer–Verlet (SV) and ICN methods, indicating that the choice of quadrature formula does not significantly affect the rate of convergence.

Our focus on energy conservation in the linear peridynamic model led to several key observations. Firstly, for the original ICN method with $\theta = 0.5$, increasing the number of iterations reduced the energy variation until it reached machine precision, with the convergence rate increasing as iterations increased. Secondly, decreasing θ resulted in a corresponding decrease in the magnitude of the energy variation, with $\theta = 0.25$ yielding the minimum value. Thirdly, we noted that the rate of energy convergence was lower than that of the solution for the SV method and the ICN method with $\theta = 1/3$. While the SV method exhibited second-order convergence with linear energy convergence, the ICN method with $\theta = 1/3$ achieved third-order accuracy with quadratic energy convergence. When $\theta = 0.5$, both the solution and the energy variation were second-order convergent. Finally, when $\theta = 0.25$, the ICN method demonstrated fourth-order superconvergence regarding energy conservation, with energy variation at the same order of magnitude as the fourth-order Runge–Kutta method while saving approximately 20% of the CPU time. Similar results were obtained with piecewise constant initial conditions, and a nonlinear case was also tested to verify the rate of convergence of the ICN method.

In conclusion, our numerical tests demonstrated that the ICN method is well suited for peridynamic simulation and generally outperforms the SV method in terms of energy conservation.

Author Contributions: Methodology, J.L. and M.B.; Software, J.L. and S.A.-A.; Validation, J.L. and S.A.-A.; Writing—original draft, J.L.; Writing—review & editing, J.L., S.A.-A. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Silling, S.A. Reformulation of elasticity theory for discontinuities and long-range forces. *J. Mech. Phys. Solids* **2000**, *48*, 175–209. [[CrossRef](#)]
2. Silling, S.A.; Lehoucq, R.B. Peridynamic theory of solid mechanics. *Adv. Appl. Mech.* **2010**, *44*, 73–168.
3. Madenci, E.; Oterkus, E. *Peridynamic Theory and Its Applications*; Springer: New York, NY, USA, 2014.
4. Bobaru, F.; Foster, J.T.; Geubelle, P.H.; Silling, S.A. *Handbook of Peridynamic Modeling*; CRC Press: Boca Raton, FL, USA, 2016.
5. Javili, A.; Morasata, R.; Oterkus, E.; Oterkus, S. Peridynamics review. *Math. Mech. Solids* **2019**, *24*, 3714–3739. [[CrossRef](#)]
6. Silling, S.A.; Askari, E. A meshfree method based on the peridynamic model of solid mechanics. *Comput. Struct.* **2005**, *83*, 1526–1535. [[CrossRef](#)]
7. Weckner, O.; Emmrich, E. Numerical simulation of the dynamics of a nonlocal, inhomogeneous, infinite bar. *J. Comput. Appl. Mech.* **2005**, *6*, 311–319.
8. Emmrich, E.; Weckner, O. The peridynamic equation and its spatial discretisation. *Math. Model. Anal.* **2007**, *12*, 17–27. [[CrossRef](#)]
9. Macek, R.W.; Silling, S.A. Peridynamics via finite element analysis. *Finite Elem. Anal. Des.* **2007**, *43*, 1169–1178. [[CrossRef](#)]
10. Tian, X.; Du, Q. Nonconforming discontinuous Galerkin methods for nonlocal variational problems. *SIAM J. Numer. Anal.* **2015**, *53*, 762–781. [[CrossRef](#)]

11. Lopez, L.; Pellegrino, S.F. A spectral method with volume penalization for a nonlinear peridynamic model. *Int. J. Numer. Methods Eng.* **2021**, *122*, 707–725. [[CrossRef](#)]
12. Lopez, L.; Pellegrino, S.F. A nonperiodic Chebyshev spectral method avoiding penalization techniques for a class of nonlinear peridynamic models. *Int. J. Numer. Methods Eng.* **2022**, *123*, 4859–4876. [[CrossRef](#)]
13. Lopez, L.; Pellegrino, S.F. A space-time discretization of a nonlinear peridynamic model on a 2D lamina. *Comput. Math. Appl.* **2022**, *116*, 161–175. [[CrossRef](#)]
14. Lopez, L.; Pellegrino, S.F. A fast-convolution based space–time Chebyshev spectral method for peridynamic models. *Adv. Contin. Discret. Model.* **2022**, *2022*, 70. [[CrossRef](#)]
15. Hairer, E.; Lubich, C.; Wanner, G. Geometric numerical integration illustrated by the Störmer–Verlet method. *Acta Numer.* **2003**, *12*, 399–450. [[CrossRef](#)]
16. Seleson, P.; Pasetto, M.; John, Y.; Trageser, J.; Reeve, S.T. PDMATLAB2D: A Peridynamics MATLAB Two-dimensional Code. *J. Peridyn. Nonlocal Model.* **2024**, *1*, 1–57. [[CrossRef](#)]
17. Coclite, G.M.; Fanizzi, A.; Lopez, L.; Maddalena, F.; Pellegrino, S.F. Numerical methods for the nonlocal wave equation of the peridynamics. *Appl. Numer. Math.* **2020**, *155*, 119–139. [[CrossRef](#)]
18. Crank, J.; Nicolson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*; Cambridge University Press: Cambridge, UK, 1947; Volume 43, pp. 50–67.
19. Choptuik, M.W. Critical behaviour in scalar field collapse. In *Deterministic Chaos in General Relativity*; Springer: Boston, MA, USA, 1994; pp. 155–175.
20. Teukolsky, S.A. Stability of the iterated Crank–Nicholson method in numerical relativity. *Phys. Rev. D* **2000**, *61*, 087501. [[CrossRef](#)]
21. Duez, M.D.; Marronetti, P.; Shapiro, S.L.; Baumgarte, T.W. Hydrodynamic simulations in 3 + 1 general relativity. *Phys. Rev. D* **2003**, *67*, 024004. [[CrossRef](#)]
22. Duez, M.D.; Liu, Y.T.; Shapiro, S.L.; Stephens, B.C. General relativistic hydrodynamics with viscosity: Contraction, catastrophic collapse, and disk formation in hypermassive neutron stars. *Phys. Rev. D* **2004**, *69*, 104030. [[CrossRef](#)]
23. Yioultsis, T.V.; Ziogos, G.D.; Kriezis, E.E. Explicit finite-difference vector beam propagation method based on the iterated Crank–Nicolson scheme. *JOSA A* **2009**, *26*, 2183–2191. [[CrossRef](#)] [[PubMed](#)]
24. Ketzaki, D.A.; Rekanos, I.T.; Kosmanis, T.I.; Yioultsis, T.V. Beam Propagation Method Based on the Iterated Crank–Nicolson Scheme for Solving Large-Scale Wave Propagation Problems. *IEEE Trans. Magn.* **2015**, *51*, 7204404. [[CrossRef](#)]
25. Shibayama, J.; Nishio, T.; Yamauchi, J.; Nakano, H. Explicit FDTD method based on iterated Crank–Nicolson scheme. *Electron. Lett.* **2022**, *58*, 16–18. [[CrossRef](#)]
26. Wu, P.; Wang, X.; Xie, Y.; Jiang, H.; Natsuki, T. Iterated Crank–Nicolson Procedure with Enhanced Absorption for Nonuniform Domains. *IEEE J. Multiscale Multiphysics Comput. Tech.* **2022**, *7*, 61–68. [[CrossRef](#)]
27. Leiler, G.; Rezzolla, L. Iterated Crank–Nicolson method for hyperbolic and parabolic equations in numerical relativity. *Phys. Rev. D* **2006**, *73*, 044001. [[CrossRef](#)]
28. Tran, Q.; Liu, J. Modified iterated Crank–Nicolson method with improved accuracy for advection equations. *Numer. Algorithms* **2023**, *1*–22. [[CrossRef](#)]
29. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2016.
30. Burden, R.L. *Numerical Analysis*; Brooks/Cole Cengage Learning: Boston, MA, USA, 2011.
31. Gottlieb, S.; Shu, C.W. Total variation diminishing Runge–Kutta schemes. *Math. Comput.* **1998**, *67*, 73–85. [[CrossRef](#)]
32. Gottlieb, S.; Shu, C.W.; Tadmor, E. Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **2001**, *43*, 89–112. [[CrossRef](#)]
33. Weckner, O.; Abeyaratne, R. The effect of long-range forces on the dynamics of a bar. *J. Mech. Phys. Solids* **2005**, *53*, 705–728. [[CrossRef](#)]
34. Emmrich, E.; Weckner, O. Analysis and numerical approximation of an integro-differential equation modeling non-local effects in linear elasticity. *Math. Mech. Solids* **2007**, *12*, 363–384. [[CrossRef](#)]
35. Difonzo, F.V.; Di Lena, F. Numerical Modeling of Peridynamic Richards' Equation with Piecewise Smooth Initial Conditions Using Spectral Methods. *Symmetry* **2023**, *15*, 960. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.