algorithms

MDPI

Article

Optimizing Finite-Difference Operator in Seismic Wave Numerical Modeling

Hui Li *, Yuan Fang, Zhiguo Huang, Mengyao Zhang and Qing Wei

Department of Computer Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; 2019200793@mail.buct.edu.cn (Y.F.); srnallblack@gmail.com (Z.H.); 2019200802@mail.buct.edu.cn (M.Z.); 2019200794@mail.buct.edu.cn (Q.W.)

* Correspondence: ray@mail.buct.edu.cn

Abstract: The finite-difference method is widely used in seismic wave numerical simulation, imaging, and waveform inversion. In the finite-difference method, the finite difference operator is used to replace the differential operator approximately, which can be obtained by truncating the spatial convolution series. The properties of the truncated window function, such as the main and side lobes of the window function's amplitude response, determine the accuracy of finite-difference, which subsequently affects the seismic imaging and inversion results significantly. Although numerical dispersion is inevitable in this process, it can be suppressed more effectively by using higher precision finite-difference operators. In this paper, we use the krill herd algorithm, in contrast with the standard PSO and CDPSO (a variant of PSO), to optimize the finite-difference operator. Numerical simulation results verify that the krill herd algorithm has good performance in improving the precision of the differential operator.

Keywords: seismic numerical modeling; finite-difference scheme; nature-inspired optimization algorithms; particle swarm optimization; center-decenter particle swarm optimization; krill herd algorithm

check for updates

Citation: Li, H.; Fang, Y.; Huang, Z.; Zhang, M.; Wei, Q. Optimizing Finite-Difference Operator in Seismic Wave Numerical Modeling. *Algorithms* **2022**, *15*, 132. https://doi.org/10.3390/a15040132

Academic Editors: Bogdan Dumitrescu and Frank Werner

Received: 27 February 2022
Accepted: 14 April 2022
Published: 18 April 2022

1. Introduction

Seismic wave numerical modeling is a widely used method to simulate the propagation of seismic waves in a known geological landscape. So far, it has been used in seismic exploration and seismology, e.g., earthquake disaster prediction [1], seismic performance assessment of roads, buildings, bridges, and other materials [2,3], seismic oceanography [4], ship detection, and identification [5].

The wave equation is of fundamental importance for describing the propagation of seismic waves [6]. The methods of seismic wave numerical modeling based on the wave equation include the finite-difference (FD) method [7], the finite element method [8,9], and the pseudo spectrum method [10]. The FD method, first presented in 1968 [7], simulates the propagation of seismic waves in various media of the earth and thus can help predicate what happens during the propagation process [11]. Compared to other techniques, the FD method is more efficient and easily implemented [6].

Generally, the wave equation of seismic wave propagation is a partial differential equation. The FD method transforms the continuous problem into the discrete problem, and one can obtain the approximate solution of the original wave equation by solving the difference equation. As the pseudospectral method is considered as the highest order finite-difference scheme [12], we shall calculate the finite-difference operator by truncating the convolution series of the pseudospectral method with the window function. However, the discretization of wave equation inevitably introduces numerical dispersion, which deteriorates the accuracy of seismic wave simulation. The window function method and optimization method are two essential ways to suppress numerical dispersion.

The size of the window function grid plays an influential role in the modeling precision. Although the refined difference grids offer better performance, sometimes, coarse difference grids are preferably used to reduce the computational cost. For this reason, choosing a suitable window function is demanding in practice. The traditional window functions include Hanning [13], Gaussian [14], Kaiser [15], and so on. In the past decades, some new window functions were also reported in the literature. Wang et al. [16] proposed a Chebyshev auto-convolution combined window. Zheng et al. [17] developed a cosine modulated Chebyshev window. Wang and Hong [18] constructed a new window function by combining the cosine function with the original window function.

Optimizing the difference operator is another strategy. In this category, the Newton method, the least squares method, and the swarm intelligence algorithm are frequently used. For example, Zhang and Yao [19] applied the simulated annealing algorithm to search for the operator that best satisfies the constraints. Z.Y. Wang et al. [20] used the improved particle swarm optimization (PSO) algorithm to optimize the finite-difference operator. Ren et al. [21] developed the least-squares method to optimize the staggered-grid finite-difference operator.

Both the window function and optimization method aim to find the optimal window function coefficients. The difference is that window functions are directly designed parameters, while optimization methods generate the optimal coefficients through the iteration. In this paper, we use the nature-inspired optimization (NIO) algorithms to find the fittest finite-difference coefficients. The NIO algorithms are widely used to solve the optimal solution problem in continuous and discrete space. In this paper, we implement the krill herd algorithm (KH), which is a recently proposed NIO algorithm, the PSO, and the center–decenter PSO, which is a variant of PSO. We derive the finite-difference operator through sinc function interpolation and Taylor series expansions. We define the objective function as the error function of the first and second derivatives. Compared with Wang's Chebyshev auto-convolution combined window, all the NIO algorithms show outstanding performances in seismic wave numerical simulation. Among them, KH exhibits greater stability at most orders.

The rest of the paper is structured as follows. Section 2 introduces the finite-difference method and deduces the objective function of the optimization method. The seismic numerical modeling and the three NIO algorithms are also introduced in this section. Section 3 presents the numerical simulations. Section 4 concludes the paper.

2. Methods

2.1. Finite-Difference Method

We start with a brief introduction of the FD method. Based on studies of Chu and Stoffa [22] and Bai et al. [23], we describe the derivation process of FD operators as follows. First, the FD operators can be derived by the sinc interpolator. According to Nyquist's theorem, a continuous signal $f(x)$ with limited bandwidth can be reconstructed by interpolating the normalized sinc function. Roughly speaking, the FD method represents waveforms with discrete samples.

$$sinc(x) = \begin{cases} 1, \text{ if } x = 0 \\ \frac{sin(\pi x)}{\pi x}, \text{ otherwise} \end{cases} \qquad f(x) = \sum_{n=-\infty}^{+\infty} sinc\left(\frac{x - n\Delta x}{\Delta x}\right) f(n\Delta x), \tag{1}$$

where Δx is the sampling interval. Thus, $1/\Delta x$ is the sampling frequency, and $f(n\Delta x)$ is the infinite uniformly spaced discrete sequence of $f(x)$.

The first and second derivatives of f are derived from Equation (1). The derivation can be simplified by letting $x = 0$ (see Appendix A.1).

$$\left.\frac{\partial f}{\partial x}\right|_{x=0} = \frac{1}{\Delta x} \sum_{n=-\infty}^{+\infty} \left(-\frac{1}{n}\cos(n\pi)\right) f(n\Delta x), \tag{2}$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x=0} = \frac{1}{\Delta x^2}\sum_{n=-\infty}^{+\infty}\left(-\frac{2}{n^2}\cos(n\pi)\right)f(n\Delta x). \tag{3}$$

We use a window function $\omega(n)$ with length $N+1$ to truncate the derivative function Equations (2) and (3). N is an even number, which is related to the accuracy. Windowing is a weighting process that truncates the function of infinite length to $[-N/2, N/2]$. The FD operators can be calculated as:

$$\left.\frac{\partial f}{\partial x}\right|_{x=0} \approx \frac{1}{\Delta x}\sum_{n=-N/2}^{N/2}\omega(n)\left(-\frac{1}{n}\cos(n\pi)\right)f(n\Delta x), \tag{4}$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x=0} \approx \frac{1}{\Delta x^2}\sum_{n=-N/2}^{N/2}\omega(n)\left(-\frac{2}{n^2}\cos(n\pi)\right)f(n\Delta x). \tag{5}$$

Chu and Stoffa [22] pointed out that $w(n)$ in the conventional FD operator is a binomial constant coefficient, and they referred to the conventional $w(n)$ as a binomial window. One method to obtain a better FD operator is to design a more suitable $w(n)$ function, which requires more control over the parameters. For the optimization algorithms, we need to set an objective function. Consider the coefficients of the function Equations (4) and (5) as a whole:

$$\left.\frac{\partial f}{\partial x}\right|_{x=0} \approx \frac{1}{\Delta x}\sum_{n=-N/2}^{N/2} b_n f(n\Delta x), \qquad \left.\frac{\partial^2 f}{\partial x^2}\right|_{x=0} \approx \frac{1}{\Delta x^2}\sum_{n=-N/2}^{N/2} c_n f(n\Delta x), \tag{6}$$

where

$$b_n = -\frac{1}{n}\cos(n\pi)\omega(n), \qquad c_n = -\frac{2}{n^2}\cos(n\pi)\omega(n),$$

b_n and c_n are the finite-difference coefficients for the first and second derivative, respectively. They are related by $c_n = 2b_n/n$.

The FD weights can also be obtained through the Taylor series expansions (Thongyoy and Boonyasiriwat [24]). It reflects the relation between finite-difference operators and the pseudospectral method. The finite-difference operator is formulated as (see Appendix A.2):

$$\frac{\partial f}{\partial x} \approx \frac{1}{\Delta x}\sum_{n=1}^{N/2} b_n(f_n - f_{-n}), \tag{7}$$

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{1}{\Delta x^2}\left[c_0 f_0 + \sum_{n=1}^{N/2} c_n(f_n + f_{-n})\right], \tag{8}$$

where $f_n = f(x + n\Delta x)$, b_n and c_n are defined similarly as before. $\sum_{n=-N/2}^{N/2} b_n = \sum_{n=-N/2}^{N/2} c_n = 0$, $b_{-n} = -b_n$, $c_{-n} = c_n$. Hence, for $n = 0$,

$$b_0 = 0, \qquad c_0 = -2\sum_{n=1}^{N/2} c_n. \tag{9}$$

Performing the Fourier transform on Equations (7) and (8), the data can be transformed from the conventional spatial and time domains to the frequency-wavenumber domain, which is also known as the F-k domain (see Appendix A.3). We obtain the first derivative in the F-k domain:

$$k_x\Delta x \approx 2\sum_{n=1}^{N/2} b_n \sin(nk_x\Delta x), \tag{10}$$

and the second derivative:

$$-(k_x\Delta x)^2 \approx c_0 + 2\sum_{n=1}^{N/2} c_n \cos(nk_x\Delta x), \tag{11}$$

where k_x is the wavenumber corresponding to the spatial derivative $\partial/\partial x$ and $\partial^2/\partial x$, which is the variable in the F-k domain.

The truncation error of the first derivative and second derivative can be expressed as:

$$E_1 = 2\sum_{n=1}^{N/2} b_n \sin(nk_x\Delta x) - k_x\Delta x, \tag{12}$$

$$E_2 = -\left[c_0 + 2\sum_{n=1}^{N/2} c_n \cos(nk_x\Delta x)\right] - (k_x\Delta x)^2. \tag{13}$$

The FD method replaces the differential in the wave equation with difference, namely, the difference form of the derivative obtained above. To minimize the numerical dispersion introduced by the replacement, we shall find the optimal coefficients b_n and c_n that minimize the truncation errors E_1 and E_2. The optimization problem of seismic wave simulation has been transformed into a coefficient optimization problem. Since b_n or c_n can be derived from each other, we attempt to obtain the optimal b_n using the NIO algorithms. Zhang and Yao [19] expressed the objective function of the optimization method as follows:

$$\underset{0 \le k_x \le k_x^{max}}{max} |E_1| \le \epsilon, \tag{14}$$

where k_x^{max} is the max wavenumber the FD can reach, and ϵ is the error limitation. We list the frequently used symbols as in Table 1.

Table 1. List of symbols.

Symbol	Meaning
Δx	sampling interval
$n\Delta x$	sampling point
N	order
$f(x)$	signal
$f(n\Delta x)$	sample of $f(x)$
f_n	$f(x + n\Delta x)$
$w(n)$	window function
E_1, E_2	truncation errors of the first and second derivative
b_n, c_n	FD coefficients of the first and second derivative
ϵ	error limitation
k_x	wavenumber
k_x^{max}	max wavenumber
k_{cutoff}	cutoff wavenumber

2.2. Modeling

We use the elastic wave equation to simulate the seismic wave propagation. The earthquake hypocenter function is given as follows:

$$f(i) = \left(1 - 2(\pi f_{peak}(i - \frac{1}{f_{peak}}))^2\right) e^{-(\pi f_{peak}(i - \frac{1}{f_{peak}}))^2}, \tag{15}$$

where f_{peak} is the dominant frequency, and $i \in \{0, 1, \ldots, n\}$ is the propagation cycle. We set $f_{peak} = 50$ Hz.

The max iteration n is calculated by:

$$n = round(\frac{t_s}{\Delta t}), \tag{16}$$

where t_s is the simulation duration and Δt is the discrete time interval. Let $t_s = 0.3$ s and Δt be 0.75 ms; then, $n = 400$.

We use the homogeneous medium model and the more complex Marmousi velocity model [25] as the basic model of propagation geology. The Marmousi velocity model is an anisotropic media with very large velocity variations in the horizontal and vertical directions. The grid size of the homogeneous medium model is 200 × 200, and that of Marmousi is 767 × 751, as shown in Figure 1. Different media are distinguished by different colors. The position of the earthquake hypocenter is at the center and that of the receiver is at 50 grids right to the center. The grid spacing is set to 10 m.

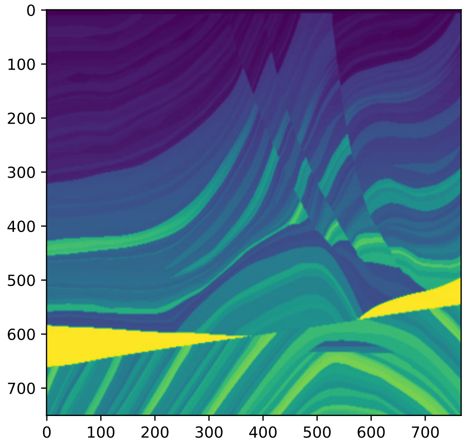


Figure 1. The landscape of the seismic wave propagation. Different colors in the model represent different media which impact the speed of wave propagation.

2.3. Three Optimization Algorithms

We optimize the finite-difference operator by minimizing the difference between the optimizing difference operator and the differential operator. A smaller truncation error means lower numerical dispersion. Next, we briefly introduce the three NIO algorithms: PSO, CDPSO, and KH.

2.3.1. Particle Swarm Optimization

PSO is a well-reputed nature-inspired algorithm proposed by Kennedy and Eberhart [26], which mimics bird flocking or fish schooling. In the PSO algorithm, each agent has two attributes: the position and the velocity. The new position of the agent is updated by the velocity. Each agent has a current fittest position *pbest*. *gbest* represents the global fittest position of all agents. The maximum velocity method is used as the update rule [27]. Specifically, the position is updated as follows:

$$position^{i+1} = position^i + v^i, \tag{17}$$

where $position^i$ is the current position of the agents at the ith iteration, and v^i is the velocity at the ith iteration that is given as:

$$v^{i+1} = wv^i + c_1r_1(pbest^i - position^i) + c_2r_2(gbest^i - position^i), \tag{18}$$

where w is the inertia weight coefficient; c_1 and c_2 are the acceleration constants of individual and society; r_1 and r_2 are random numbers between 0 and 1.

2.3.2. Center–Decenter PSO

CDPSO, proposed by Wang [23], is a variant of PSO that has the advantage of avoiding premature convergence and local trap in PSO. It combines two learning strategies: centralized learning and decentralized learning. The agent switches its strategy to update the velocity.

In the centralized learning strategy, the velocity of the ith iteration is updated as follows, and the meaning of the same coefficients is the same as that of PSO.

$$\begin{aligned} v^{i+1} &= wv^{i} + c_1 r_1 (E^{i} - position^{i}), r_1 \in [0,1], \\ E^{i} &= \frac{1}{L}\sum_{l=1}^{L} pbest_l^{i}, \end{aligned} \tag{19}$$

where L is a random number, which represents the number of partial optimal solutions.

The velocity in decentralized learning is updated as follows:

$$\begin{aligned} v^{i+1} &= wv^{i} + c_2 r_2 (pbest_{\gamma^i}^{i} - position^{i}), r_2 \in [0,1], \\ \gamma^{i} &= rand()\%P, \gamma^{i} \in [1,P], \end{aligned} \tag{20}$$

where γ is the serial number of the particle, and P is the number of agents in the range of all possible solutions.

2.3.3. Krill Herd Algorithm

The Krill herd (KH) algorithm was proposed in 2012 by Gandomi and Alavi [28]. KH is inspired by the herding behavior of the krills. The position of the ith krill individuals X_i is influenced by three main factors: movement induced by the presence of other individuals N_i, foraging activity F_i, and random diffusion D_i, respectively:

$$\frac{dX_i}{dt} = N_i + F_i + D_i. \tag{21}$$

The individual motion is induced by the neighbors within a certain range and the position of the elite individual. The formula for induced speed N is given as follows:

$$\begin{aligned} N_i^{new} &= N^{max} * \alpha_i + w_n * N_i^{old}, w_n \in [0,1], \\ \alpha_i &= \alpha_i^{local} + \alpha_i^{target}, \end{aligned} \tag{22}$$

where w_n is the inertia weight, α_i is the direction of the induced, α_i^{local} is the influence of neighbors, and α_i^{target} is the influence of individuals with optimal fitness values.

The foraging motion is relevant to the location of food and previous experience. The foraging speed F is formulated as follows:

$$\begin{aligned} F_i^{new} &= V_f * \beta_i + w_f * F_i^{old}, w_f \in [0,1], \\ \beta_i &= \beta_i^{food} + \beta_i^{best}, \end{aligned} \tag{23}$$

where w is the inertia weight; β_i is the direction of the forage.

The physical diffusion is the random movement behavior of krill populations. The diffusion speed D is as follows:

$$D_i = D^{max}\left(1 - \frac{iter}{iter_{max}}\right)\delta, \delta \in [-1,1], \tag{24}$$

where δ is a random value.

We give the flow charts of the above-mentioned algorithms in Figure 2. PSO and CDPSO have similar processes. The difference between them is that CDPSO alters the updating strategy of the velocity.

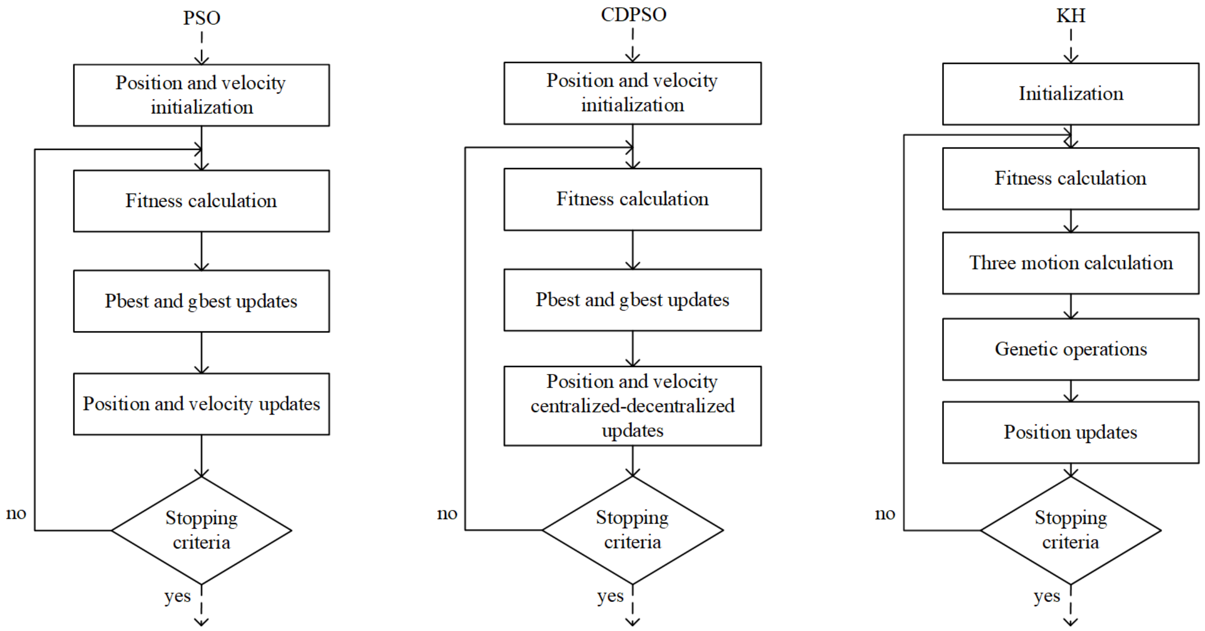


Figure 2. The flow charts of PSO, CDPSO, and KH.

3. Numerical Simulation

3.1. Simulation Setup

The parameters regarding the seismic wave landscape have been given previously. We let population = 30, iteration = 500 for PSO and KH, population = 100, and iteration = 1500 for CDPSO. The coefficients of PSO and CDPSO are assigned as: $w = 0.8$, $c_1 = c_2 = 1.5$, and the velocity is between -0.001 and 0.001.

We use the first-order derivative truncation error E_1 in Equation (12) as the objective. The value of E is corresponding to the value of k_x. We denote this one–one relation as $E(k_x)$. To make the image of coefficient convergence more demonstrative, the above error value $E(k_x)$ is used to calculate the number of agents that meet the requirement of the accuracy. Given the accuracy $\epsilon = 5e^{-3}$, the cost function value (CFV) is determined as follows:

$$C(k_x) = \begin{cases} 1 & |E(k_x)| \geq \epsilon, \\ 0 & else; \end{cases} \tag{25}$$

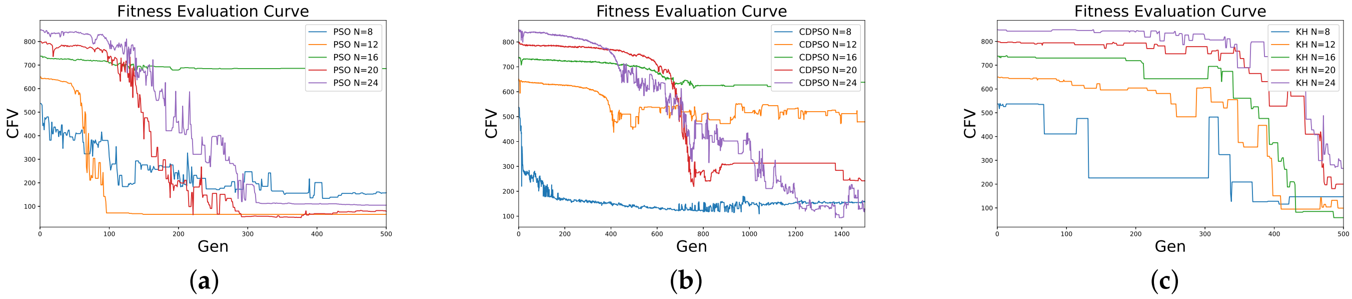
$$CFV = \sum_{k_x=0}^{k_{cutoff}} C(k_x). \tag{26}$$

where k_{cutoff} is the cutoff wavenumber, corresponding to the upper limit of k_x. As $k_x \in [0, \pi/\Delta x]$ and the sampling interval Δx is set to π, $k_x \in [0, 1]$. $k_{cutoff} \in [0, 1000]$, then the k_x is an increasing sequence $0, 0.001, \cdots, k_{cutoff}/1000$.

We choose the order N as $[8, 12, 16, 20, 24]$ with $k_{cutoff} = [0.54, 0.65, 0.74, 0.8, 0.85] \times 1000$ for the following tests of the three algorithms.

3.2. Coefficients Convergence

To test the performance of the algorithms, we use CFV in Equation (26) to evaluate the fitness of the solution. The fitness convergence curves of the three NIO algorithms are illustrated in Figure 3. The closer CFV value is to 0, the better the precision of the algorithm is.



(**a**) (**b**) (**c**)

Figure 3. The convergence curves of PSO (**a**), CDPSO (**b**), and KH (**c**).

As illustrated in Figure 3, the KH algorithm with 500 generations shows better accuracy than the CDPSO algorithm with 1500 generations. When the order $N = 8$, the KH algorithm is slightly better than the PSO algorithm. However, when $N = 16$, the KH algorithm converges to a good optimum, while PSO and CDPSO do not converge.

We give the first finite-difference coefficients b_n optimized by the three algorithms in Tables 2–4.

Table 2. The finite-difference coefficients for the first derivative optimized by PSO.

	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
b_1	0.849367818	0.920977162	0.877587089	0.965287713	0.973893871
b_2	−0.25408939	−0.3577484	−0.288519626	−0.433682998	−0.449521466
b_3	0.065186786	0.153433266	0.079759689	0.24111591	0.261752507
b_4	−0.009222605	−0.058927291	0	−0.139204287	−0.161705926
b_5		0.017640291	0	0.078452231	0.1
b_6		−0.003169356	−0.006035286	−0.041571492	−0.06002697
b_7			0.013566605	0.020009949	0.034157136
b_8			−0.013048505	−0.008365952	−0.017995371
b_9				0.002801649	0.008513142
b_{10}				−0.00060467	−0.003450766
b_{11}					0.001103313
b_{12}					−0.000224999

Table 3. The finite-difference coefficients for the first derivative optimized by CDPSO.

	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
b_1	0.860321205	0.86109756	0.897879165	0.945915193	0.973635331
b_2	−0.26835603	−0.266916073	−0.319377704	−0.399220361	−0.449084566
b_3	0.074799005	0.070567648	0.112048086	0.198575497	0.261348878
b_4	−0.012312666	−0.008783105	−0.026268214	−0.096211902	−0.161442267
b_5		7.08×10^{-7}	0.000414949	0.041381255	0.099999422
b_6		−0.000970876	$-0.1.99 \times 10^{-7}$	−0.014264206	−0.06040469
b_7			0.002950187	0.003122273	0.034861698
b_8			−0.002579216	−0.000113252	−0.018844889
b_9				$0.5.24 \times 10^{-6}$	0.009398521
b_{10}				−0.000148269	−0.004199285
b_{11}					0.001607051
b_{12}					−0.000460175

Table 4. The finite-difference coefficients for the first derivative optimized by KH.

	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
b_1	0.85637065	0.91565981	0.953061594	0.956976127	0.973409618
b_2	−0.263266133	−0.349228563	−0.411757109	−0.418632603	−0.448750956
b_3	0.071389175	0.144791061	0.213788114	0.222144041	0.260615495
b_4	−0.011184471	−0.052565668	−0.111261167	−0.119424376	−0.160355857
b_5		0.014337537	0.053990931	0.060681453	0.098818855
b_6		−0.002178404	−0.023007227	−0.027737712	−0.059118172
b_7			0.007967749	0.010823998	0.033637319
b_8			−0.00182424	−0.003382806	−0.01775192
b_9				0.000823068	0.008525232
b_{10}				−0.000185621	−0.003794
b_{11}					0.00150656
b_{12}					−0.000518958

3.3. Stability

We calculate second coefficients c_n by using the formula mentioned in Section 2.1: $c_n = 2b_n/n$ $(n = 1, 2, \ldots, N/2)$, $c_0 = -2\sum_{n=1}^{N/2} c_n$. Then, we use the absolute error E_1 (Equation (12)) and E_2 (Equation (13)) to evaluate the stability of the optimized difference operators. The absolute error is illustrated in Figure 4.

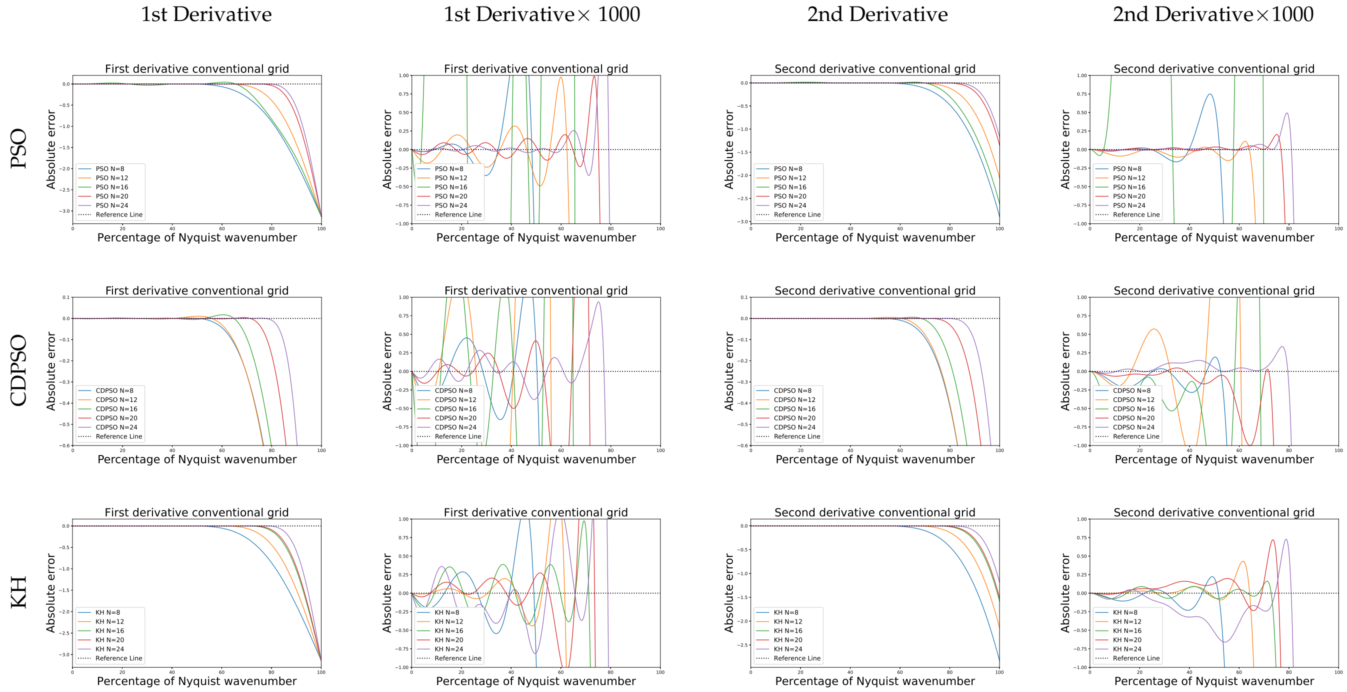


Figure 4. The first and second derivative accuracy of PSO, CDPSO, and KH. The horizontal and vertical coordinates are the percentage of Nyquist wavenumber k_x and absolute error E_1, respectively.

The curve is flat from the beginning. It decreases after the wavenumber reaches about 60. To make it easier to observe, the figure is magnified a thousand times (columns 2 and 4). The closer the curve is to 0, the higher the error stability is. It is observed that the curve fluctuates up and down at 0 and extends abruptly to negative infinity at a certain point. PSO performs well at $N = 12$ and 24 but not very well at $N = 8$ and 16. CDPSO is good at $N = 8$, but the overall fluctuation is large. The second derivative error of KH is between plus and minus 0.00075. The KH algorithm shows better stability than PSO and CDPSO. When the percentage of Nyquist wavenumber is less than 40 in the first derivative and less than 50 in the second derivative, the KH algorithm shows excellent stability.

3.4. Model Test

We choose the $N = 12$ finite-difference coefficients of the second derivative c_n for wave field simulation. Take Wang's [16] data as a baseline, which are denoted as 'source'. They proposed a Chebyshev auto-convolution combined window. The four finite-difference coefficients of the second derivative are illustrated in the Table 5.

Table 5. Finite-difference coefficients of the second derivative optimized by PSO, CDPSO, and KH ($N = 12$).

	c_0	c_1	c_2	c_3	c_4	c_5	c_6
Source	−3.138521049	1.854196962	−0.368305121	0.110295386	−0.034584857	0.009637126	−0.001978972
PSO	−3.126061573	1.841954323	−0.3577484	0.102288844	−0.029463646	0.007056116	−0.001056452
CDPSO	−2.995218501	1.722195119	−0.266916073	0.047045099	−0.004391552	2.83×10^{-7}	−0.000323625
KH	−3.114688956	1.831319621	−0.349228563	0.096527374	−0.026282834	0.005735015	−0.000726135

Take the wave field simulation of the homogeneous medium first. With the earthquake hypocenter obtained from Equation (15), we choose the snapshot at 150 ms, as shown in Figure 5. Since the homogeneous medium is isotropic, the entire wave field can be known by looking at the graph in only one direction. Therefore, we stitched the four figures together to facilitate the comparison. The operator with less waviness suppressed the numerical dispersion better, and the parameters calculated are better, too. It is observed that PSO is quite similar to the source, while KH is the best to have less wave. KH has significantly suppressed numerical dispersion.

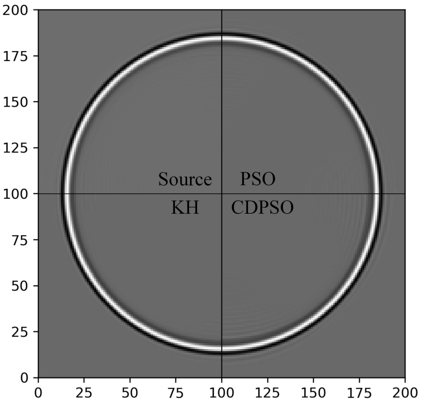


Figure 5. Impulse response of CDPSO, KH, PSO, and source in homogeneous medium at 150 ms.

We take the simulation on the Marmousi model shown in Figure 1. Figure 6 gives four wave field snapshots taken at 75 ms intervals.

Due to the complexity of the model and the small difference between the optimized parameters, the obtained wave field information is mixed and difficult to distinguish. It is not obvious which one is better. Therefore, we make the single traces figure from the receiver to show the differences. A portion of the trace after the arrival of the main waver is captured and shown in Figure 7. To see the differences more clearly, we zoomed in on the waveform after the arrival of the main frequency. CDPSO has a better suppression of numerical dispersion in the later stages, although the suppression is not as good as the source data in the early stage. The overall trend of PSO is the same as source data, while KH is slightly different from the source in the late stage. It is observed from the wave of about 625 that both KH and PSO have a better suppression effect than the source. KH is much better than CDPSO and slightly better than PSO. The differences caused by parameter optimization become less noticeable as the complexity of the model increases.

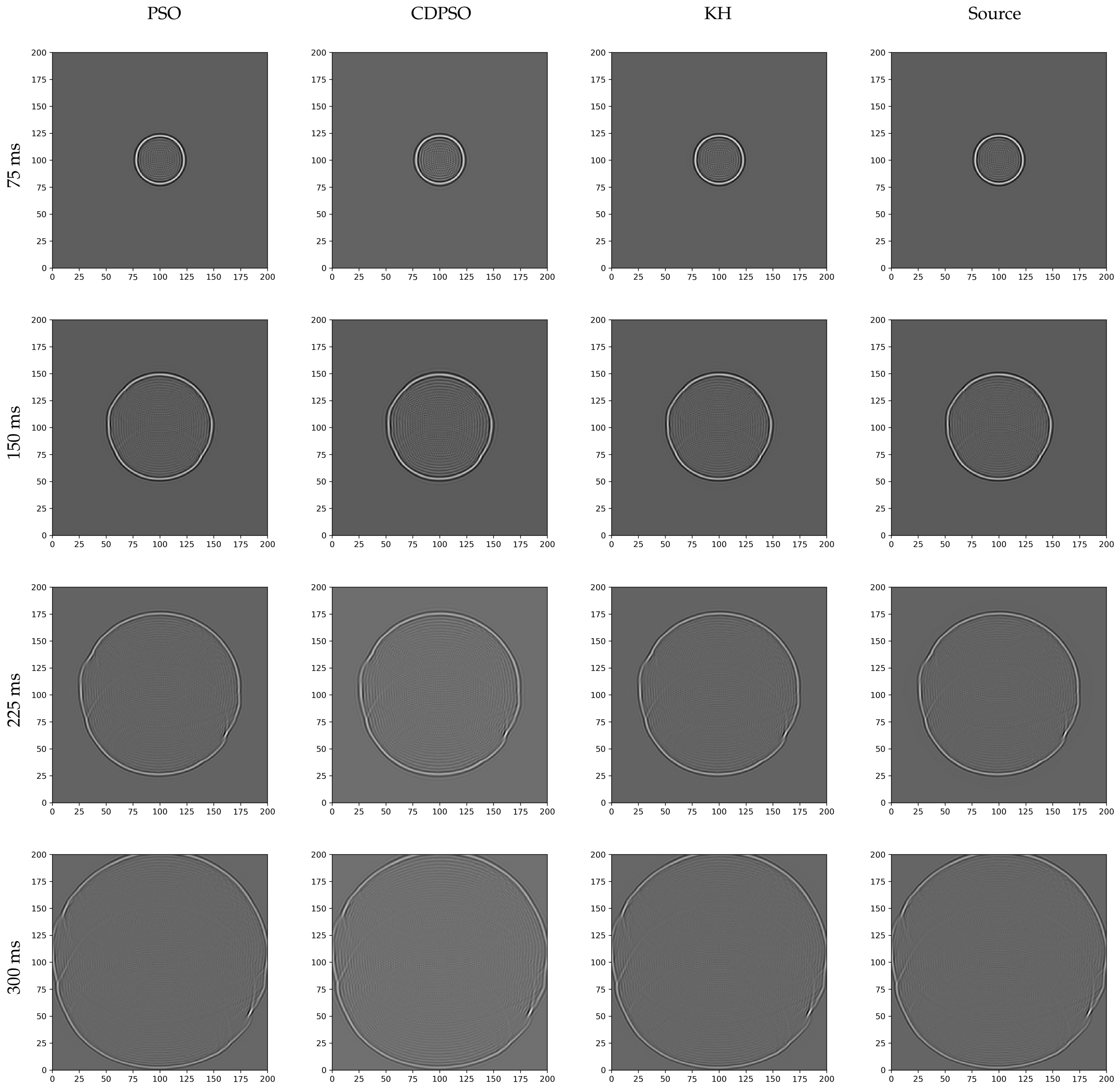
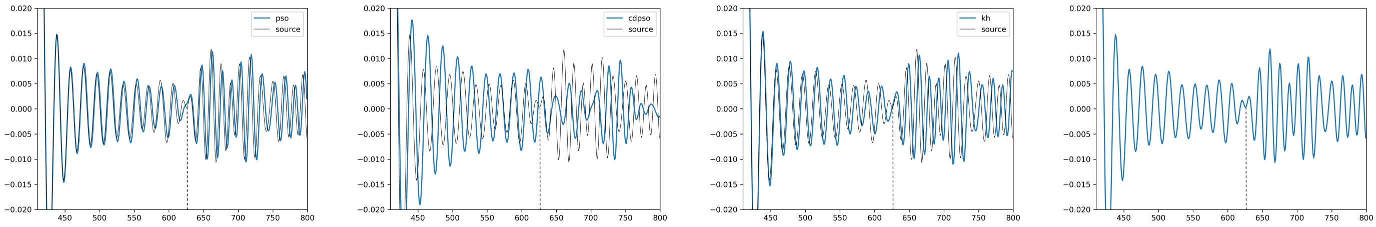


Figure 6. The Chebyshev auto-convolution windows of CDPSO, KH, and PSO.



(**a**) PSO (**b**) CDPSO (**c**) KH (**d**) Source

Figure 7. The trace of PSO, CDPSO, KH, and source within 300 ms.

In a nutshell, the KH shows the best performance among the three algorithms.

4. Conclusions

The finite-difference method uses the difference formulas to replace the derivatives approximately. We introduced the process of obtaining the FD operator and set the error as the optimization objective. We used the NIO algorithms to optimize the first and second derivative coefficients. We compared the performance of the three optimization algorithms (PSO, CDPSO, and KH) in terms of convergence and stability, and we compared them with Chebyshev auto-convolution combined window in a model test. Compared with the Chebyshev window, KH and PSO have better performance. Despite the fact that we did not perform further tuning operation, the NIO algorithms still have a good performance in spectral suppression. KH is the most stable algorithm among the three, which performs well at all N orders. It shows promising prospect in optimizing a finite-difference operator in seismic wave numerical modeling.

Author Contributions: Conceptualization, H.L. and Y.F.; methodology, H.L. and Y.F.; software, Z.H.; validation, H.L., Y.F. and Z.H.; formal analysis, Y.F., Z.H. and Q.W.; investigation, H.L., Y.F., Z.H. and M.Z.; resources, H.L., Z.H and Q.W.; data curation, Z.H.; writing—original draft preparation, Y.F., Z.H. and M.Z.; writing—review and editing, H.L., Y.F. and M.Z.; visualization, Z.H.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (No. 2021YFB3802200).

Data Availability Statement: Data is contained within the article.

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

Appendix A. FD Operator

Appendix A.1. The First and Second Derivatives of f

The first and second derivatives of f can be derived from Equation (1).

$$\begin{aligned}\frac{\partial f}{\partial x} &= \sum_{n=-\infty}^{+\infty}\left\{\frac{\frac{\pi}{\Delta x}cos[\frac{\pi}{\Delta x}(x-n\Delta x)]\times\frac{\pi}{\Delta x}(x-n\Delta x)-sin[\frac{\pi}{\Delta x}(x-n\Delta x)]\frac{\pi}{\Delta x}}{[\frac{\pi}{\Delta x}(x-n\Delta x)]^2}\right\}f(n\Delta x),\\ &= \sum_{n=-\infty}^{+\infty}\left\{\frac{cos[\frac{\pi}{\Delta x}(x-n\Delta x)]}{x-n\Delta x}-\frac{sin[\frac{\pi}{\Delta x}(x-n\Delta x)]}{\frac{\pi}{\Delta x}(x-n\Delta x)^2}\right\}f(n\Delta x),\end{aligned} \tag{A1}$$

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \sum_{n=-\infty}^{+\infty}\left\{\frac{-\frac{\pi}{\Delta x}sin[\frac{\pi}{\Delta x}(x-n\Delta x)]\times(x-n\Delta x)-cos[\frac{\pi}{\Delta x}(x-n\Delta x)]}{(x-n\Delta x)^2}\right.\\ &\qquad\left.-\frac{\frac{\pi}{\Delta x}cos[\frac{\pi}{\Delta x}(x-n\Delta x)]\times\frac{\pi}{\Delta x}(x-n\Delta x)^2-2\frac{\pi}{\Delta x}(x-n\Delta x)sin[\frac{\pi}{\Delta x}(x-n\Delta x)]}{(\frac{\pi}{\Delta x})^2(x-n\Delta x)^4}\right\}f(n\Delta x),\\ &= \sum_{n=-\infty}^{+\infty}\left\{sin[\frac{\pi}{\Delta x}(x-n\Delta x)]\left(\frac{-\frac{\pi}{\Delta x}}{x-n\Delta x}+\frac{2}{\frac{\pi}{\Delta x}(x-n\Delta x)^3}\right)\right.\\ &\qquad\left.-cos[\frac{\pi}{\Delta x}(x-n\Delta x)]\frac{2}{(x-n\Delta x)^2}\right\}f(n\Delta x).\end{aligned} \tag{A2}$$

While $x=0$, $sin[\frac{\pi}{\Delta x}(x-n\Delta x)]=sin(-n\pi)=0$. The derivatives can be simplified as:

$$\left.\frac{\partial f}{\partial x}\right|_{x=0} = \frac{1}{\Delta x}\sum_{n=-\infty}^{+\infty}\left(-\frac{1}{n}\cos(n\pi)\right)f(n\Delta x), \tag{A3}$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x=0} = \frac{1}{\Delta x^2}\sum_{n=-\infty}^{+\infty}\left(-\frac{2}{n^2}\cos(n\pi)\right)f(n\Delta x). \tag{A4}$$

Appendix A.2. The FD Operator through Taylor's Expansion

According to Wisart and Chaiwoot [24], the finite-difference formulas for the derivatives can be obtained from Taylor's expansion.

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + O((x - x_0)^3), \\ f(x_0 + n\Delta x) &= f(x_0) + f'(x_0)n\Delta x + \frac{f''(x_0)}{2}(n\Delta x)^2 + O((n\Delta x)^3), \end{aligned} \tag{A5}$$

x_0 is a differentiable point on f. Δx is the sampling interval, $O((n\Delta x)^n)$ is the approximate error. Next, we calculate the normal derivative formula.

$$\begin{aligned} \sum_{n=-N/2}^{N/2} a_n f(x + n\Delta x) &\approx f(x) \sum_{n=-N/2}^{N/2} a_n + \Delta x f'(x) \sum_{n=-N/2}^{N/2} n a_n + \frac{f''(x)}{2}(\Delta x)^2 \sum_{n=-N/2}^{N/2} n^2 a_n \\ &\approx Af(x) + B\Delta x f'(x) + C(\Delta x)^2 f''(x) \\ A = \sum_{n=-N/2}^{N/2} a_n, \quad B &= \sum_{n=-N/2}^{N/2} n a_n, \quad C = \frac{1}{2}\sum_{n=-N/2}^{N/2} n^2 a_n, \end{aligned} \tag{A6}$$

where a_n are constant coefficients. For the calculation derivative, the constant A should be 0. For first derivative:

$$\begin{aligned} &\frac{\partial f}{\partial x} \approx \frac{1}{\Delta x}\sum_{n=-N/2}^{N/2} \frac{a_n}{B} f(x + n\Delta x) \approx \frac{1}{\Delta x}\sum_{n=-N/2}^{N/2} b_n f(x + n\Delta x), \\ &\sum_{n=-N/2}^{N/2} b_n = 0, \qquad \sum_{n=-N/2}^{N/2} n^2 b_n = \sum_{n=1}^{N/2} n^2(b_n + b_{-n}) = 0, \end{aligned} \tag{A7}$$

where b_n are the finite-difference coefficients for the first derivative. Let $b_{-n} = -b_n$; then, we can get $b_0 = 0$.

For the second derivative

$$\begin{aligned} &\frac{\partial^2 f}{\partial x^2} \approx \frac{1}{(\Delta x)^2}\sum_{n=-N/2}^{N/2} \frac{a_n}{C} f(x + n\Delta x) \approx \frac{1}{(\Delta x)^2}\sum_{n=-N/2}^{N/2} c_n f(x + n\Delta x), \\ &\sum_{n=-N/2}^{N/2} c_n = 0, \qquad \sum_{n=-N/2}^{N/2} n c_n = \sum_{n=1}^{N/2} n(c_n - c_{-n}) = 0, \end{aligned} \tag{A8}$$

where c_n are the finite-difference coefficients for the second derivative. Let $c_{-n} = c_n$; then, we can get $c_0 = -2\sum_{n=1}^{N/2} c_n$.

Then, the normal derivative formula is summarized as

$$\frac{\partial f}{\partial x} \approx \frac{1}{\Delta x}\sum_{n=1}^{N/2} b_n(f_n - f_{-n}), \tag{A9}$$

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{1}{(\Delta x)^2}\left[c_0 f_0 + \sum_{n=1}^{N/2} c_n(f_n + f_{-n})\right], \tag{A10}$$

where $f_n = f(x + n\Delta x)$, N is the order, c_n and b_n are the FD coefficients. The higher the N, the more approximate the result.

Appendix A.3. Fourier Transform

The Fourier transform in the wavenumber domain is

$$F(k_x) = \int_{-\infty}^{+\infty} f(x)e^{-ik_x x}dx, \tag{A11}$$

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}dk_x, \tag{A12}$$

where k_x is the wavenumber, and i is the imaginary number.

Substitute Equation (A12) into Equations (7) and (8) as follows. The first derivative is

$$\begin{aligned}
&\frac{\partial f}{\partial x} \approx \frac{1}{\Delta x}\sum_{n=1}^{N/2} b_n(f_n - f_{-n}),\\
&\frac{1}{2\pi}\int_{-\infty}^{+\infty} ik_x F(k_x)e^{ik_x x}dk_x \approx \frac{1}{\Delta x}\frac{1}{2\pi}\sum_{n=1}^{N/2} b_n\int_{-\infty}^{+\infty} F(k_x)(e^{ik_x(x+n\Delta x)} - e^{ik_x(x-n\Delta x)})dk_x,\\
&\Delta x\int_{-\infty}^{+\infty} ik_x F(k_x)e^{ik_x x}dk_x \approx \sum_{n=1}^{N/2} b_n\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}\times 2isin(k_x n\Delta x)dk_x,\\
&\int_{-\infty}^{+\infty} e^{ik_x x}F(k_x)\left[ik_x\Delta x - 2i\sum_{n=1}^{N/2} b_n isin(k_x n\Delta x)\right]dk_x \approx 0.
\end{aligned} \tag{A13}$$

Hence,

$$k_x\Delta x \approx 2\sum_{n=1}^{N/2} b_n isin(k_x n\Delta x). \tag{A14}$$

The second derivative is

$$\begin{aligned}
&\frac{\partial^2 f}{\partial x^2} \approx \frac{1}{\Delta x^2}\left[c_0 f_0 + \sum_{n=1}^{N/2} c_n(f_n + f_{-n})\right],\\
&-\frac{1}{2\pi}\int_{-\infty}^{+\infty} k_x^2 F(k_x)e^{ik_x x}dk_x\\
&\approx \frac{1}{\Delta x^2}\left\{\frac{c_0}{2\pi}\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}dk_x + \frac{1}{2\pi}\sum_{n=1}^{N/2} c_n\int_{-\infty}^{+\infty} F(k_x)(e^{ik_x(x+n\Delta x)} + e^{ik_x(x-n\Delta x)})dk_x\right\},\\
&\approx \frac{1}{\Delta x^2}\frac{1}{2\pi}\left\{c_0\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}dk_x + \sum_{n=1}^{N/2} c_n\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}\times 2cos(k_x n\Delta x)dk_x\right\},\\
&\approx \frac{1}{\Delta x^2}\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}(c_0 + 2\sum_{n=1}^{N/2} c_n cos(k_x n\Delta x))dk_x,\\
&\int_{-\infty}^{+\infty} F(k_x)e^{ik_x x}\left[(k_x\Delta x)^2 + (c_0 + 2\sum_{n=1}^{N/2} c_n cos(k_x n\Delta x))\right]dk_x \approx 0.
\end{aligned} \tag{A15}$$

Hence,

$$-(k_x\Delta x)^2 \approx c_0 + 2\sum_{n=1}^{N/2} c_n cos(k_x n\Delta x). \tag{A16}$$

References

1. Favorskaya, A.; Petrov, I.; Golubev, V.; Khokhlov, N. Numerical simulation of earthquakes impact on facilities by grid-characteristic method. *Matem. Mod.* **2017**, *112*, 1206–1215. [CrossRef]
2. Sarchi, L.; Varum, H.; Monteiro, R.; Silveira, D. Seismic behavior of two Portuguese adobe buildings: Part II—Numerical modeling and fragility assessment. *Int. J. Archit. Herit.* **2018**, *12*, 936–950. [CrossRef]
3. Chen, X.; Fu, J.; Xue, F.; Wang, X. Comparative Numerical Research on the Seismic Behavior of RC Frames Using Normal and High-Strength Reinforcement. *Int. J. Civ. Eng.* **2017**, *15*, 531–547. [CrossRef]

4. Chen, J.; Tong, S.; Han, T.; Song, H.; Pinheiro, L.; Xu, H.; Azevedo, L.; Duan, M.; Liu, B. Modelling and detection of submarine bubble plumes using seismic oceanography. *J. Mar. Syst.* **2020**, *209*, 103375. [CrossRef]
5. Lu, Z.; Zhang, Z.; Gu, J. Modeling of low-frequency seismic waves in a shallow sea using the staggered grid difference method. *Chin. J. Oceanol. Limnol.* **2017**, *35*, 1010–1017. [CrossRef]
6. Yao, G.; Wu, D.; Debens, H.A. Adaptive finite difference for seismic wavefield modelling in acoustic media. *Sci. Rep.* **2016**, *6*, 30302. [CrossRef]
7. Alterman, Z.; Karal, F. Propagation of elastic waves in layered media by Finite Difference methods. *Bull. Seismol. Soc. Am.* **1968**, *58*, 367–398.
8. Lysmer, J.; Drake, L.A. A Finite Element Method for Seismology. *Methods Comput. Phys. Adv. Res. Appl.* **1972**, *11*, 181–216.
9. SU, B.; Li, H.; Liu, S.; Yang, D. Modified symplectic scheme with finite element method for seismic wavefield modeling. *Chin. J. Geophys.* **2019**, *62*, 1440–1452.
10. Jiang, X.; Adeli, H. Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings. *Int. J. Numer. Methods Eng.* **2007**, *71*, 606–629. [CrossRef]
11. Kalyani, V.K.; Sinha, A.; Pallavika; Chakraborty, S.K.; Mahanti, N.C. Finite difference modeling of seismic wave propagation in monoclinic media. *Acta Geophys.* **2008**, *56*, 1074–1089. [CrossRef]
12. Fornberg, B. The pseudospectral method; comparisons with finite differences for the elastic wave equation. *Geophysics* **1987**, *52*, 483–501. [CrossRef]
13. Zhou, B.; Greenhalgh, S. Seismic scalar wave equation modeling by a convolutional differentiator. *Bull. Seismol. Soc. Am.* **1992**, *82*, 289–303.
14. Igel, H.; Mora, P.; Riollet, B. Anisotropic wave propagation through finite-difference grids. *Geophysics* **1995**, *60*, 1203–1216. [CrossRef]
15. Hicks, G.J. Arbitrary source and receiver positioning in finite-difference schemes using Kaiser windowed sinc functions. *Geophysics* **2002**, *67*, 156–165. [CrossRef]
16. Wang, Z.; Liu, H.; Tang, X.D.; Wang, Y. Optimized Finite-Difference Operators Based on Chebyshev Auto-Convolution Combined Window Function. *Chin. J. Geophys.* **2015**, *58*, 628–642.
17. Zheng, W.Q.; Meng, X.H.; Liu, J.H.; Wang, J.H. High precision elastic wave equation forward modeling based on cosine modulated Chebyshev window function. *Chin. J. Geophys.* **2016**, *59*, 2650–2662.
18. Wang, J.; Hong, L. Stable optimization of finite-difference operators for seismic wave modeling. *Stud. Geophys. Geod.* **2020**, *64*, 452. [CrossRef]
19. Zhang, J.H.; Yao, Z.X. Optimized finite-difference operator for broadband seismic wave modeling. *Geophysics* **2012**, *78*, A13–A18. [CrossRef]
20. Wang, Z.Y.; Bai, W.L.; Liu, H. An optimized finite-difference scheme based on the improved PSO algorithm for wave propagation. In *SEG Technical Program Expanded Abstracts 2019*; Society of Exploration Geophysicists: Houston, TX, USA, 2019.
21. Ren, Y.J.; Huang, J.P.; Peng, Y.; Liu, M.L.; Chao, C.; Yang, M.W. Optimized staggered-grid finite-difference operators using window functions. *Appl. Geophys.* **2018**, *15*, 253–260. [CrossRef]
22. Chu, C.; Stoffa, P. Determination of finite-difference weights using scaled binomial windows. *Geophysics* **2012**, *77*, W17–W26. [CrossRef]
23. Bai, W.; Wang, Z.; Liu, H.; Yu, D.; Chen, C.; Zhu, M. Optimisation of the finite-difference scheme based on an improved PSO algorithm for elastic modelling. *Explor. Geophys.* **2020**, *52*, 419–430. [CrossRef]
24. Thongyoy, W.; Boonyasiriwat, C. Least-Squares Finite Difference Operator. 2015. Available online: https://mcsc.sc.mahidol.ac.th/publications/reports/2015/2015-01.pdf (accessed on 8 April 2022).
25. Lailly, P.; Rocca, F.; Versteeg, R. The Marmousi experience: Synthesis. In Proceedings of the EAEG Workshop-Practical Aspects of Seismic Data Inversion, Copenhagen, Denmark, 28 May–1 June 1990. [CrossRef]
26. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
27. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS'95, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
28. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [CrossRef]