A two-step high-order compact scheme for the Laplacian operator and its implementation in an explicit method for integrating the nonlinear Schrödinger equation

R.M. Caplan *, R. Carretero-González

Nonlinear Dynamical Systems Group,[1] San Diego State University, San Diego, CA 92182-7720, USA
Computational Science Research Center,[2] San Diego State University, San Diego, CA 92182-7720, USA
Department of Mathematics and Statistics, San Diego State University, San Diego, CA 92182-7720, USA

A B S T R A C T

We describe and test an easy-to-implement two-step high-order compact (2SHOC) scheme for the Laplacian operator and its implementation into an explicit finite-difference scheme for simulating the nonlinear Schrödinger equation (NLSE). Our method relies on a compact 'double-differencing' which is shown to be computationally equivalent to standard fourth-order non-compact schemes. Through numerical simulations of the NLSE using fourth-order Runge–Kutta, we confirm that our scheme shows the desired fourth-order accuracy. A computation and storage requirement comparison is made between the 2SHOC scheme and the non-compact equivalent scheme for both the Laplacian operator alone, as well as when implemented in the NLSE simulations. Stability bounds are also shown in order to get maximum efficiency out of the method. We conclude that the modest increase in storage and computation of the 2SHOC schemes is well worth the advantages of having the schemes compact, and their ease of implementation makes their use very useful for practical implementations.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The nonlinear Schrödinger equation (NLSE) is a universal model describing the evolution and propagation of complex field envelopes in nonlinear dispersive media. As such, it is used to describe many physical systems including the evolution of water waves, nonlinear optics, thermodynamic pulses, nonlinear waves in fluid dynamics, and waves in semiconductors [1–3]. The general form of the NLSE can be written as

$$i\frac{\partial \Psi}{\partial t} + a\nabla^2 \Psi - V(\mathbf{r})\Psi + s|\Psi|^2\Psi = 0, \tag{1}$$

where $\Psi(\mathbf{r}, t) \in \mathbb{C}$ is the value of the wavefunction, ∇^2 is the Laplacian operator, and $a > 0$ and s are parameters defined by the system being modeled. $V(\mathbf{r})$ is an external potential term, which when included, makes Eq. (1) known as the Gross–Pitaevskii equation [4].

---

* Correspondence to: Predictive Science Inc. 9990 Mesa Rim Rd, Suite 170, San Diego, CA 92121, USA. Tel.: +1 858 225 2314.
  E-mail addresses: sumseq@gmail.com, caplanr@predsci.com (R.M. Caplan).
  URL: http://www.predsci.com (R.M. Caplan).
[1] http://nlds.sdsu.edu.
[2] http://www.csrc.sdsu.edu.

The nonlinearity in the NLSE allows the prediction and description of important, and experimentally relevant, nonlinear effects and nonlinear waves, such as solitons and vortices [4]. In the context of nonlinear optics, this nonlinearity emerges from the nonlinear response of the optical medium. The cubic-type nonlinearity in this context is the so-called Kerr nonlinearity of the medium [5,6]. Another context where the NLSE serves as a prototypical model is in the realm of Bose–Einstein condensates (BECs) [4]. In this context, the nonlinearity takes into account the *mean-field* interaction of the atoms in the condensate. The sign of the nonlinearity in the NLSE (i.e., the sign of s) defines the type of interactions. For example, in the nonlinear optics application a self-focusing (self-defocusing) medium corresponds to $s > 0$ ($s < 0$), while in the BEC context, attractive (repulsive) interatomic interactions correspond to $s > 0$ ($s < 0$). In the case of self-focusing or attracting nonlinearity the prototypical (nonlinear) solutions of the NLSE are in the form of *bright* solitons in one dimension, while in two or three dimensions bright vortices and vortex rings do exist but are typically unstable. On the other hand, the case of self-defocusing or repulsive nonlinearity bears *dark* (nonlinear) solutions that in one dimension take the shape of dark or gray solitons which in two and three dimensions correspond to, typically stable, (dark) vortices and rings, respectively.

For almost every nonlinear partial differential equation (PDE) (including the NLSE), the interesting solutions require simulations using numerical methods. A standard methodology includes discretizing the solution in space by using finite-difference approximation schemes for the required derivatives. A *high-order* scheme is one which exhibits greater than second-order ($O(h^2)$, where h is the grid spacing) accuracy. While there are a large number of scenarios where higher-order schemes are a necessity due to the desired accuracy of the simulations, often the higher accuracy of high-order schemes is unnecessary, and second-order accuracy is sufficient for the problem. However, in such a case, using high-order schemes is still desirable because they allow one to simulate a solution with many fewer grid points, while maintaining the same accuracy as a second-order scheme. In those situations, the desired grid point size is based on the ability to resolve the structure of the solution, and not on the accuracy of computation.

Higher-order finite-difference schemes are typically achieved by computing derivatives with a wider scheme stencil. This adds a difficulty near the boundary, as one must be able to calculate the inner point near the boundary with the same accuracy as the internal scheme which can be complicated to implement. If this is not done, the entire simulation can eventually become lower-order. Another disadvantage of wide-stencils is that they cause many parallel implementations to be more difficult to realize, as different compute nodes must share or transfer more boundary values to their neighboring nodes. Besides the added complexity of the codes, the extra communication/global-memory-access (in the case of graphical processing units) reduces the overall parallel performance [7].

For the aforementioned reasons, there is great interest in high-order compact (HOC) schemes. These are finite-difference schemes which exhibit higher-order accuracy but still only rely on the closest neighboring points for computations. HOC schemes have been developed for various multidimensional steady-state PDEs including the convection–diffusion equation [8], Poisson's equation [9], the stream-function vorticity form of the Navier–Stokes equations [10], as well as generalized linear elliptical PDEs with variable coefficients [11] to name a few. Many HOC implementations of time-dependent PDEs have also been formulated including Burger's equation [12], the wave equation [13], the Euler equations [14] and the time-dependent convection–diffusion equation [15,16] as well as others. Time-dependent HOC schemes have also been developed for the one- and two-dimensional linear [17,18] and one-dimensional nonlinear Schrödinger equations [19,20]. One drawback of HOC schemes are that their formulations are typically specific for the model equation being used, and requires re-deriving the schemes for each different PDE to be simulated (see Ref. [11], where the authors developed a Maple program to symbolically generate a HOC scheme for three-dimensional linear elliptical models).

The time-dependent HOC schemes developed so far are all implicit (even if the time-stepping is done with an otherwise explicit scheme), requiring solving a linear system in each time-step, as well as iterative processes when simulating nonlinear PDEs such as the NLSE (see however Ref. [19] where the author's second implicit scheme implementation was able to be formulated to get around this requirement). The computational and storage requirements for implicit schemes can be prohibitive in large multidimensional settings. They are also, in general, difficult to optimally parallelize, especially in higher dimensions.

For these reasons, we wish to develop HOC formulations that are *fully explicit* time-dependent schemes (in our case, for the NLSE). To do this, we formulate a two-step procedure in computing the spatial derivatives, where each individual step is a compact computation. The first step of the 2SHOC scheme computes the standard second-order finite-difference approximation to the derivatives of the Laplacian, while the second step uses these computed derivative values to compactly approximate the Laplacian to fourth-order accuracy. Since the 2SHOC scheme computes the Laplacian independent of the governing PDE, it can be used in a multitude of time-dependent and time-independent PDEs which contain the Laplacian operator, increasing its generality. An explicit 2SHOC scheme for the NLSE can be formed by utilizing standard explicit ordinary differential equation (ODE) solvers. For the scheme described in this paper, we use the classic fourth-order Runge–Kutta (RK4) [21].

The paper is organized as follows. In Section 2 we show the formulation of the 2SHOC scheme for the Laplacian operator in one, two and three dimensions. Then, in Section 3, we use the scheme to form an explicit method for simulating the NLSE. In Section 4, we show numerical tests of each scheme from Section 3 and confirm their accuracy. In Section 5 we compare the storage and computation requirements for the 2SHOC schemes versus the standard fourth-order non-compact equivalent schemes. We conclude in Section 6.

2. Formulation of 2SHOC schemes for the Laplacian operator

It is well known that one can derive a fourth-order accurate finite-difference scheme for the second spatial derivative of a function by applying a 'double-differencing' approach. This works by noting that the first truncation term in the standard central-difference scheme contains the fourth spatial derivative. A second-order approximation to the fourth derivative can be obtained by applying a central-difference operator to the result of applying a central-difference operator to the function. When multiplied by the h^2 (where h is the spatial step-size of the computational grid) in the truncation term, the error in the truncation term becomes $O(h^4)$, and therefore, the resulting scheme becomes a fourth-order accurate approximation to the second derivative. When these two steps are algebraically combined and simplified, one yields the standard non-compact fourth-order stencil. Due to the increase in the number of computations and storage, the 'double-differencing' procedure is not actually implemented, rather, the resulting non-compact stencil is used directly.

However, when the advantages of using a compact scheme are considered, the numerical implementation of the 'double-differencing' procedure becomes an overlooked, viable alternative to other more complicated HOC schemes. In time-dependent systems, the sequential nature of the 'double-differencing' procedure makes it best suited for use with explicit time-stepping schemes. This may be a reason why this simple compact solution has been overlooked, as many high-order numerical schemes for time-dependent PDEs are typically implicit.

2.1. One-dimensional formulation

Although the methodology for the 2SHOC scheme is to use the 'double-differencing' approach, it is nevertheless worthwhile to show that the scheme can also be formed as a result of attempting to make other time-dependent HOC scheme formulations explicit. In Ref. [15], the authors formulate an implicit time-dependent HOC scheme for the transient convection–diffusion equation based on their method from Ref. [9]. We can obtain a compact high-order scheme for the time-dependent NLSE following the same methodology, and with a modification, can transform it into an explicit scheme which is equivalent to using the 2SHOC scheme for the Laplacian operator in the NLSE.

In one dimension, we discretize the wavefunction of the NLSE as $\Psi(x_i, t_n) \equiv \Psi_i^n$ with a grid spacing of size h ($x_i = x_{\min} + hi$) and a time-step of size k ($t_n = kn$). The Laplacian operator applied to Ψ in one dimension can be represented as

$$\nabla^2 \Psi_i = \left.\frac{\partial^2 \Psi}{\partial x^2}\right|_i = \delta_x^2 \Psi_i - \frac{h^2}{12}\left(\left.\frac{\partial^4 \Psi}{\partial x^4}\right|_i\right) + O(h^4), \tag{2}$$

where the central-difference operator δ_x^2 is defined as

$$\delta_x^2 \Psi_i = \frac{1}{h^2}(\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}). \tag{3}$$

As per the technique of Ref. [15], to formulate a high-order compact scheme, the NLSE of Eq. (1) is differentiated in space twice in order to form an expression for the fourth derivative in Eq. (2) which yields

$$\left.\frac{\partial^4 \Psi}{\partial x^4}\right|_i = -\frac{1}{a}\delta_x^2\left[i\left.\frac{\partial \Psi}{\partial t}\right|_i + \left(s|\Psi_i|^2 - V(x_i)\right)\Psi_i\right] + O(h^2). \tag{4}$$

Due to the h^2 in the truncation term of Eq. (2), when Eq. (4) is inserted into Eq. (2), the resulting approximation of the Laplacian becomes $O(h^4)$. Inserting Eqs. (2) and (4) into Eq. (1) yields the fourth-order in space semi-discrete HOC scheme

$$\left.\frac{\partial \Psi}{\partial t}\right|_i = i\left[a\left(\delta_x^2 \Psi_i + \frac{h^2}{a\,12}\delta_x^2\left[i\left.\frac{\partial \Psi}{\partial t}\right|_i + \left(s|\Psi_i|^2 - V(x_i)\right)\Psi_i\right]\right) + \left(s|\Psi_i|^2 - V(x_i)\right)\Psi_i\right]. \tag{5}$$

Due to the central-difference operator operating on the temporal derivative $\partial \Psi/\partial t$, the resulting scheme of Eq. (5) will be implicit even if the time-stepping is chosen to be otherwise explicit (for example, forward differencing).

In order to retain a fully explicit scheme, a two-step approach is used. First, an approximation of $\partial \Psi_i/\partial t$ is made and then is inserted into Eq. (5). The approximation of $\partial \Psi_i/\partial t$ must be $O(h^2)$ for the scheme to retain its fourth-order accuracy. The most straight-forward way to make such an approximation is to apply a standard second-order central differencing to the NLSE yielding

$$T_i = \left.\frac{\partial \Psi}{\partial t}\right|_i + O(h^2) = i\left[a\delta_x^2 \Psi_i + \left(s|\Psi_i|^2 - V(x_i)\right)\Psi_i\right]. \tag{6}$$

Once computed, T_i is inserted into Eq. (5), and the central difference operator can then be applied to T_i as

$$\delta_x^2 T_i = \frac{T_{i+1} - 2\,T_i + T_{i-1}}{h^2}.$$

It is important that the boundary conditions for Eq. (6) be computed to at least $O(h^2)$ accuracy since the boundary points will be used for interior calculations of the spatial derivative of Eq. (5) (see Section 3.2 for details on boundary conditions for the 2SHOC schemes). Inserting Eq. (6) into Eq. (5) yields the semi-discrete equation

$$\frac{\partial \Psi}{\partial t}\bigg|_i = i \left[a \left(\delta_x^2 \Psi_i + \frac{h^2}{a\,12} \delta_x^2 \left[i\,T_i + \left(s\,|\Psi_i|^2 - V(x_i) \right) \Psi_i \right]_i \right) + \left(s\,|\Psi_i|^2 - V(x_i) \right) \Psi_i \right] + O(h^4). \tag{7}$$

Any desired explicit ODE solver (that is stable for the problem with its parameters) can then be used to integrate Eq. (7).

Algebraically combining the two stages of Eqs. (6) and (7) together yields

$$\frac{\partial \Psi}{\partial t}\bigg|_i = i \left[a \left(\delta_x^2 \Psi_i - \frac{h^2}{12} \delta_x^2 (\delta_x^2 \Psi_i) \right) + \left(s\,|\Psi_i|^2 - V(x_i) \right) \Psi_i \right] + O(h^4), \tag{8}$$

and therefore the two-stage HOC scheme described in Eqs. (6) and (7) is computationally equivalent to simply taking the central-difference of the central-difference to approximate the fourth derivative truncation term in the Laplacian. Therefore, we have recovered the 2SHOC scheme approach for approximating the Laplacian which does not depend on the other terms of the governing equation. Therefore the 2SHOC is a stand-alone scheme for the Laplacian operator which can be used in multiple governing equations.

After collecting terms and simplifying the 2SHOC 'double-differencing', the resulting two-step scheme for the one-dimensional Laplacian is given by

$$(1)\ D_i = \frac{1}{h^2} \left(\Psi_{i+1} - 2\Psi_i + \Psi_{i-1} \right), \tag{9}$$

$$(2)\ \nabla^2 \Psi_i \approx \frac{7}{6} D_i - \frac{1}{12} \left(D_{i+1} + D_{i-1} \right). \tag{10}$$

When the two steps of Eqs. (9) and (10) are combined algebraically and simplified, the standard five-point non-compact fourth-order finite-difference approximation is recovered:

$$\nabla^2 \Psi_i = \frac{\partial^2 \Psi}{\partial x^2}\bigg|_i = \frac{-\Psi_{i+2} + 16\Psi_{i+1} - 30\Psi_i + 16\Psi_{i-1} - \Psi_{i-2}}{12h^2} + O(h^4).$$

A potential drawback in using the 2SHOC scheme is that it requires extra storage space (the D array) and more computations than using the standard fourth-order five-point stencil. However, as will be discussed in Section 5, the compact scheme's advantages can outweigh this deficiency.

2.2. Two-dimensional formulation

In two dimensions, the Laplacian operator applied to Ψ at grid location (i, j) can be represented as

$$\nabla^2 \Psi_{i,j} = \frac{1}{h^2} \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -4 & 1 \\ \hline & 1 & \\ \hline \end{array} \Psi_{i,j} - \frac{h^2}{12} \left(\frac{\partial^4 \Psi}{\partial x^4}\bigg|_{i,j} + \frac{\partial^4 \Psi}{\partial y^4}\bigg|_{i,j} \right) + O(h^4). \tag{11}$$

Unlike the one-dimensional case, there are additional compact grid points which are not being used in Eq. (11) (the four corner points). It is known that these points can be added to make a more accurate nine-point Laplacian operator given as [22]

$$\nabla^2 \Psi_{i,j} = \frac{1}{6\,h^2} \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array} \Psi_{i,j} + O(h^2). \tag{12}$$

However, the nine-point Laplacian of Eq. (12), while more accurate, is still second-order. In fact, it can easily be shown that there *cannot* exist a fourth-order nine-point Laplacian operator [23] (this fact should not be confused with the well-known fourth-order nine-point scheme for the Laplace and Poisson equations [24]). Even though the nine-point Laplacian of Eq. (12) is a more accurate second-order approximation, in order to minimize the amount of computation needed for the 2SHOC scheme, we will only utilize the standard five-point Laplacian of Eq. (11) for the required second-order derivatives. Once again, 'double-differencing' is used to obtain a second-order accurate approximation for the truncation term's fourth derivatives. However, in this case, the result yields two variations of the 2SHOC scheme.

The first is a direct parallel to the one-dimensional 2SHOC scheme in which the second derivative in the *x* and *y* directions are approximated with second-order central-differencing. These are then used to form second-order approximations to the

fourth derivatives in the truncation terms in Eq. (11). After simplification, this results in the 2SHOC scheme

$$(1)\ D_{i,j}^x = \delta_x^2 \Psi_{i,j} = \frac{\Psi_{i+1,j} - 2\Psi_{i,j} + \Psi_{i-1,j}}{h^2},$$

$$D_{i,j}^y = \delta_y^2 \Psi_{i,j} = \frac{\Psi_{i,j+1} - 2\Psi_{i,j} + \Psi_{i,j-1}}{h^2},$$ (13)

$$(2)\ \nabla^2 \Psi_{i,j} \approx \frac{7}{6}\left(D_{i,j}^x + D_{i,j}^y\right) - \frac{1}{12}\left(D_{i+1,j}^x + D_{i-1,j}^x + D_{i,j+1}^y + D_{i,j-1}^y\right).$$ (14)

The 2SHOC scheme as described in Eqs. (13) and (14) require two storage arrays (D_x and D_y) in addition to that for Ψ. In large-scale computations, memory use can be a major bottleneck and so it is useful to limit the amount of extra storage required as much as possible. We find that the two-dimensional 2SHOC scheme can be re-formulated to only require *one* extra storage matrix, at the cost of requiring more computations (a trade-off that is evaluated in Section 5).

To formulate the lower-storage version of the two-dimensional 2SHOC, the standard five-point second-order finite-difference approximation to the two-dimensional Laplacian is used for the first stage (stored in D), and then the result is combined with a second-order cross-derivative stencil to yield the second-order approximations to the fourth derivative truncation terms in Eq. (11). First, we note that

$$\nabla^2\left(\nabla^2\Psi\right) = \frac{\partial^2}{\partial x^2}\nabla^2\Psi + \frac{\partial^2}{\partial y^2}\nabla^2\Psi = \frac{\partial^4\Psi}{\partial x^4} + \frac{\partial^4\Psi}{\partial y^4} + 2\frac{\partial^4\Psi}{\partial x^2\,\partial y^2},$$

in which case the fourth derivative truncation terms in Eq. (11) can be written as

$$\frac{\partial^4\Psi}{\partial x^4} + \frac{\partial^4\Psi}{\partial y^4} = \frac{\partial^2}{\partial x^2}\nabla^2\Psi + \frac{\partial^2}{\partial y^2}\nabla^2\Psi - 2\frac{\partial^4\Psi}{\partial x^2\,\partial y^2}.$$ (15)

The cross-derivative in Eq. (15) is known to have the nine-point second-order compact stencil [25]

$$\left.\frac{\partial^4\Psi}{\partial x^2\,\partial y^2}\right|_{i,j} = \frac{1}{h^4}\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\\hline -2 & 4 & -2 \\\hline 1 & -2 & 1 \\\hline\end{array}\ \Psi_{i,j} + O(h^2).$$ (16)

Part of the stencil of Eq. (16) can be written in terms of the five-point second-order approximation to the Laplacian stored in D, in which case the single-storage 2SHOC scheme (after simplification) is given as

$$1)\quad D_{i,j} = \delta_x^2\Psi_{i,j} + \delta_y^2\Psi_{i,j} = \frac{1}{h^2}\begin{array}{|c|c|c|}\hline & 1 & \\\hline 1 & -4 & 1 \\\hline & 1 & \\\hline\end{array}\ \Psi_{i,j}$$ (17)

$$2)\quad \nabla^2\Psi_{i,j} \approx -\frac{1}{12}\begin{array}{|c|c|c|}\hline & 1 & \\\hline 1 & -12 & 1 \\\hline & 1 & \\\hline\end{array}\ D_{i,j} + \frac{1}{6h^2}\begin{array}{|c|c|c|}\hline 1 & & 1 \\\hline & -4 & \\\hline 1 & & 1 \\\hline\end{array}\ \Psi_{i,j}.$$ (18)

As in the one-dimensional case, both formulations of the 2SHOC schemes in two dimensions are computationally equivalent to the non-compact fourth-order stencil

$$\nabla^2\Psi_{i,j} = -\frac{1}{12h^2}\begin{array}{|c|c|c|c|c|}\hline & & 1 & & \\\hline & & -16 & & \\\hline 1 & -16 & 60 & -16 & 1 \\\hline & & -16 & & \\\hline & & 1 & & \\\hline\end{array}\ \Psi_{i,j} + O(h^4).$$ (19)

2.3. Three-dimensional formulation

In three dimensions, the Laplacian operator applied to Ψ can be represented as

$$\nabla^2\Psi_{i,j,k} = \left.\frac{\partial^2\Psi}{\partial x^2}\right|_{i,j,k} + \left.\frac{\partial^2\Psi}{\partial y^2}\right|_{i,j,k} + \left.\frac{\partial^2\Psi}{\partial z^2}\right|_{i,j,k}$$

$$= \delta_x^2\Psi_{i,j,k} + \delta_y^2\Psi_{i,j,k} + \delta_z^2\Psi_{i,j,k} - \frac{h^2}{12}\left(\left.\frac{\partial^4\Psi}{\partial x^4}\right|_{i,j,k} + \left.\frac{\partial^4\Psi}{\partial y^4}\right|_{i,j,k} + \left.\frac{\partial^4\Psi}{\partial y^4}\right|_{i,j,k}\right) + O(h^4).$$ (20)

There are again two formulations of the 2SHOC scheme which trade off storage requirement versus number of computations. The first formulation takes *three* storage matrices (D^x, D^y, D^z), while, as in the two-dimensional case, the other formulation takes only *one* (D). The three-storage 2SHOC directly follows from the two-dimensional version and is defined as

$$(1)\ D^x_{i,j,k} = \delta^2_x \Psi_{i,j,k} = \frac{\Psi_{i+1,j,k} - 2\Psi_{i,j,k} + \Psi_{i-1,j,k}}{h^2} \tag{21}$$

$$D^y_{i,j,k} = \delta^2_y \Psi_{i,j,k} = \frac{\Psi_{i,j+1,k} - 2\Psi_{i,j,k} + \Psi_{i,j-1,k}}{h^2}$$

$$D^z_{i,j,k} = \delta^2_z \Psi_{i,j,k} = \frac{\Psi_{i,j,k+1} - 2\Psi_{i,j,k} + \Psi_{i,j,k-1}}{h^2}$$

$$(2)\ \nabla^2 \Psi_{i,j,k} \approx \frac{7}{6}\left(D^x_{i,j,k} + D^y_{i,j,k} + D^z_{i,j,k}\right) - \frac{1}{12}\left(D^x_{i+1,j,k} + D^x_{i-1,j,k} + D^y_{i,j+1,k} + D^y_{i,j-1,k} + D^z_{i,j,k+1} + D^z_{i,j,k-1}\right). \tag{22}$$

The single storage formulation requires the use of the cross-derivatives of $\nabla^2(\nabla^2\Psi)$ as before. We write the fourth derivative truncation terms of Eq. (20) as

$$\frac{\partial^4\Psi}{\partial x^4} + \frac{\partial^4\Psi}{\partial y^4} + \frac{\partial^4\Psi}{\partial z^4} = \nabla^2\left(\nabla^2\Psi\right) - 2\left(\frac{\partial^4\Psi}{\partial x^2 \partial y^2} + \frac{\partial^4\Psi}{\partial x^2 \partial z^2} + \frac{\partial^4\Psi}{\partial y^2 \partial z^2}\right), \tag{23}$$

therefore the cross-derivatives in Eq. (23) can be approximated to second-order accuracy using nine-point compact stencils in each direction which contain the same coefficients as the two-dimensional stencil of Eq. (16).

After extracting D terms out of the cross derivatives and simplifying, the single-storage 2SHOC scheme in three dimensions becomes

$$(1)\quad D_{i,j,k} = \frac{1}{h^2}\left(\boxed{\begin{smallmatrix} & & \\ & 1 & \\ & & \end{smallmatrix}}\ \Psi_{i,j+1,k} + \boxed{\begin{smallmatrix} & 1 & \\ 1 & -6 & 1 \\ & 1 & \end{smallmatrix}}\ \Psi_{i,j,k} + \boxed{\begin{smallmatrix} & & \\ & 1 & \\ & & \end{smallmatrix}}\ \Psi_{i,j-1,k}\right), \tag{24}$$

$$(2)\quad \nabla^2 \Psi_{i,j,k} \approx -\frac{1}{12}\left(\boxed{\begin{smallmatrix} & & \\ & 1 & \\ & & \end{smallmatrix}}\ D_{i,j+1,k} + \boxed{\begin{smallmatrix} & 1 & \\ 1 & -10 & 1 \\ & 1 & \end{smallmatrix}}\ D_{i,j,k} + \boxed{\begin{smallmatrix} & & \\ & 1 & \\ & & \end{smallmatrix}}\ D_{i,j-1,k}\right) \tag{25}$$

$$+\frac{1}{6h^2}\left(\boxed{\begin{smallmatrix} & 1 & \\ 1 & & 1 \\ & 1 & \end{smallmatrix}}\ \Psi_{i,j+1,k} + \boxed{\begin{smallmatrix} 1 & & 1 \\ & -12 & \\ 1 & & 1 \end{smallmatrix}}\ \Psi_{i,j,k} + \boxed{\begin{smallmatrix} & 1 & \\ 1 & & 1 \\ & 1 & \end{smallmatrix}}\ \Psi_{i,j-1,k}\right).$$

Again, both formulations of the 2SHOC scheme in three dimensions are computationally equivalent to the standard non-compact fourth-order stencil.

3. Implementation of the 2SHOC schemes for solving the NLSE

In this section, we show the implementation of the 2SHOC schemes into explicit methods for simulating the NLSE. The NLSE is a good model for use with the 2SHOC schemes, as it is very relevant in current research, and as mentioned in Section 1, its nonlinear terms make the use of explicit time-stepping desirable. We emphasize however, that the NLSE is being used as one example of implementing the 2SHOC scheme for the Laplacian, and that the 2SHOC scheme is independent of any additional terms in the governing PDE, and can therefore be used with a multitude of models.

There exists many numerical methods for integrating the NLSE (see reviews of Refs. [26,27]) including the widely-used split-step Fourier (SSF) method [28]. Although the SSF method can be very accurate and efficient, it is not used in all cases since it is typically formulated to be only second-order accurate in time, can be computationally expensive in three-dimensional settings, is somewhat difficult to parallelize (see Ref. [29]), and its required use of periodic boundary conditions can make in unsuitable for certain problems (such as homogeneous background density solutions such as the dark soliton example shown in Section 4). Other split-step methods which use finite-difference for the linear part of the solve are also used (see for example, the HOC split-step method of Ref. [20]) but those scheme typically require the use of tri-diagonal solvers for each dimension to be stable which can complicate their parallelization. Although for simplicity and time-accuracy we use a standard finite-difference approach for time-stepping the NLSE with the 2SHOC scheme, we note that since the 2SHOC scheme for the Laplacian is independent of the time-stepping methodology, it could also be implemented into a split-step method for the NLSE.

3.1. Explicit time integration using Runge–Kutta

To formulate a fully explicit scheme for simulating the NLSE, a method of lines approach is utilized, where the 2SHOC is used for the spatial Laplacian and the resulting semi-discrete system of ODEs is integrated using an explicit time-stepping scheme. As shown in Refs. [30,31], the first-order ($O(k)$ where k is the time-step) forward difference and the

second-order ($O(k^2)$) Runge–Kutta scheme (Heun's method) for simulating the NLSE are unconditional *unstable* (unless, as shown in Ref. [32], the real and imaginary parts of the NLSE are computed in staggered time steps). Therefore, for our implementation, a higher-order Runge–Kutta method is required and is chosen to be the standard fourth-order Runge–Kutta (RK4) scheme [21]. To facilitate our computational cost comparisons in Section 5, we write the RK4 scheme applied to the NLSE algorithmically as

$$\begin{aligned}
&(1)\, K_{\text{tot}} = F(\Psi^n) && (6)\, K_{\text{tmp}} = F(\Psi_{\text{tmp}}) \\
&(2)\, \Psi_{\text{tmp}} = \Psi^n + \frac{k}{2} K_{\text{tot}} && (7)\, K_{\text{tot}} = K_{\text{tot}} + 2\, K_{\text{tmp}} \\
&(3)\, K_{\text{tmp}} = F(\Psi_{\text{tmp}}) && (8)\, \Psi_{\text{tmp}} = \Psi^n + k\, K_{\text{tmp}} \\
&(4)\, K_{\text{tot}} = K_{\text{tot}} + 2\, K_{\text{tmp}} && (9)\, K_{\text{tmp}} = F(\Psi_{\text{tmp}}) \\
&(5)\, \Psi_{\text{tmp}} = \Psi^n + \frac{k}{2} K_{\text{tmp}} && (10)\, \Psi^{n+1} = \Psi^n + \frac{k}{6}\, (K_{\text{tot}} + K_{\text{tmp}}),
\end{aligned} \tag{26}$$

where

$$F(\Psi) = \frac{\partial \Psi}{\partial t} = i \left[a\nabla^2 \Psi + \left(s\, |\Psi|^2 - V(\mathbf{r}) \right) \Psi \right],$$

and $\nabla^2 \Psi$ inside $F(\Psi)$ is computed with either a standard central differencing (CD) or the 2SHOC scheme. We denote the combined time–space schemes as RK4+CD and RK4+2SHOC respectively.

The RK4 time-stepping requires three storage matrices (K_{tmp}, K_{tot}, Ψ_{tmp}) in addition to the storage for the solution Ψ and external potential $V(\mathbf{r})$ (if $V(\mathbf{r})$ is chosen to be stored rather than computed at each evaluation). Lower-storage fourth-order Runge–Kutta schemes with comparable accuracy have been developed (a 3*N* in Ref. [33], and a five-stage 2*N* in Ref. [34] which requires an additional function evaluation, and whose coefficients are numerically derived), but for simplicity we use the classic RK4 of Eq. (26). Using the 2SHOC scheme inside $F(\Psi)$ requires additional storage, which we discuss in Section 5.

The RK4+2SHOC scheme is conditionally stable, in that the size of the time-step is limited by a bound based on the spatial step h (see Section 3.3 for the stability bounds of the RK4+2SHOC scheme applied to the NLSE). Conditional stability is one of the only drawbacks of using explicit schemes. However, even though implicit schemes are usually unconditionally stable, error requirements and algorithm complexity often make the explicit schemes more efficient even taking the time-step limitations into account.

3.2. Boundary conditions

The use of proper boundary conditions is very important when performing numerical simulations. The 2SHOC scheme requires two boundary conditions, one for each step. The first step is a boundary condition on the Laplacian of the wavefunction (or on the separate spatial derivatives if the high-storage 2SHOC is being used). The second step either requires a boundary condition for the fourth spatial derivatives, or in many cases, the scheme for the overall model equations dictate the boundary condition (for example, in our RK4 scheme for the NLSE, the boundary condition of the second step of the 2SHOC is not required, as the boundary condition of the time derivative of the wavefunction (Ψ_t) overwrites any condition that would be in the 2SHOC scheme).

As mentioned in Ref. [15], it appears that in general, compact boundary conditions are not possible to realize for one-stage HOC schemes. However, even when the use of one-sided differencing (or any other non-compact technique) is necessary, it does not negate the boundary advantage of using HOC schemes (since the advantage is not having to alter the scheme *near* the boundaries). There are some boundary condition techniques that can be seen as compact, and hence fit very well into the framework of HOC schemes. These conditions are a Dirichlet ($\Psi = $ const) and modulus-squared Dirichlet (MSD) ($|\Psi|^2 = $ const) [35] boundary conditions.

In many scenarios, one would ideally want a transparent boundary condition. However, such conditions can be complicated to implement (see Ref. [36] for a review of different approaches). For many problems, an easy alternative is to make the computational grid large enough so that one can reasonably use Dirichlet conditions at the boundaries. When $s > 0$ (attractive or focusing nonlinearity) in the NLSE, most dynamics are of areas of density which decay into a surrounding background area of zero-density, in which case a Dirichlet boundary condition of zero is useful. In order to use the Dirichlet boundary condition with the 2SHOC scheme, the boundary condition on the Laplacian must be computed in the first step. This is easily found by setting $\Psi_t = 0$ in Eq. (1) yielding

$$\nabla^2 \Psi_b = -\frac{(s|\Psi_b|^2 - V_b)\Psi_b}{a}, \tag{27}$$

where b represents a boundary grid point. In the second stage of the 2SHOC, the boundary condition of Eq. (27) would also be used for the overall Laplacian boundary. In the RK4+2SHOC method for the NLSE, the second stage boundary condition would simply apply the Dirichlet condition directly as

$$\left. \frac{\partial \Psi}{\partial t} \right|_b = 0. \tag{28}$$

In the case where $s < 0$ (repulsive or defocusing nonlinearity) and $V(\mathbf{r}) = 0$ in the NLSE, most interesting dynamics are within areas of low densities which increase towards an area with a constant background density. In such a case, one cannot use standard Dirichlet boundary conditions since the constant boundary is in the modulus-squared of the wavefunction, not at a single real and imaginary value. To solve this problem, we recently have developed a new modulus-squared Dirichlet boundary condition that accurately simulates a constant density at the boundaries [35]. This boundary condition is described as

$$\Psi_{t,b} \approx i \operatorname{Im}\left[\frac{\Psi_{t,b-1}}{\Psi_{b-1}}\right] \Psi_b, \tag{29}$$

where Ψ_t is the temporal derivative of Ψ and $b - 1$ represents the closest internal point relative to a boundary grid point in the normal direction. The standard form of the MSD boundary condition requires to first compute all interior points and then compute the boundaries, making the general form of the MSD not compatible with implicit schemes (see Ref. [35] for details on possible implicit scheme implementation strategies of the MSD).

The MSD boundary condition can be used for any time-dependent complex PDE and in multiple dimensions. For our application to the NLSE, we require boundary conditions for each step of the 2SHOC scheme. For the first step, one can substitute the NLSE in the MSD boundary condition and get an MSD condition for the Laplacian given by

$$\nabla^2 \Psi_b \approx \left[\operatorname{Im}\left(i\,\frac{\nabla^2 \Psi_{b-1}}{\Psi_{b-1}}\right) + \frac{1}{a}\,(N_{b-1} - N_b)\right] \Psi_b, \tag{30}$$

where

$$N_b = s\,|\Psi_b|^2 - V_b, \qquad N_{b-1} = s\,|\Psi_{b-1}|^2 - V_{b-1}, \tag{31}$$

while for the second step, the boundary condition of Eq. (29) can (in the RK4+2SHOC scheme) be used directly.

It should be noted that the MSD boundary condition cannot be directly applied to the multi-storage version of the 2SHOC schemes because the MSD computes the boundary value of $\nabla^2 \Psi$, not the individual second spatial derivatives needed for Eq. (13) or Eq. (21). This can be overcome by rearranging the boundary condition taking advantage of the cross derivative stencils. However, since we will be only using the single-storage 2SHOC schemes in our examples, we do not show this alteration to the MSD boundary condition here.

There are numerous other boundary conditions one might use for the NLSE including other compact-friendly ones such as Laplacian-zero ($\nabla^2 \Psi = 0$) and periodic. However, in this paper we have limited ourselves to Dirichlet and MSD, as they will be used in the numerical tests of Section 4.

3.3. Stability

Since the most important drawback in using explicit schemes is that they are conditionally stable, it is very important to limit this deficiency as much as possible by computing the stability bound to determine the largest time-step that is usable.

Since the 2SHOC schemes are algebraically equivalent to the standard non-compact fourth-order schemes, the stability bounds on the time-step should not adversely be affected by the use of the compact schemes. However, the form of the boundary condition implementation in the 2SHOC schemes will change the way the near-boundary points are computed. In Ref. [30] we used an extension to the methodology employed in Ref. [31] to conduct an analysis of the stability bounds for simulating the multidimensional NLSE with RK4 for various boundary conditions including those in Section 3.2. The analysis was done for both the standard second-order central differencing, as well as fourth-order differencing using the 2SHOC formulation. The results utilizing the two boundary conditions discussed in Section 3.2 are summarized as follows:

In the linear case where $s = 0$ and with no external potential ($V(\mathbf{r}) = 0$), utilizing Dirichlet boundary conditions, the stability bound on the time-step k using a fourth-order central-difference scheme (with interior points computed in the 2SHOC methodology) in a d-dimensional setting is

$$k_{\text{linear}} < \left(\frac{3}{4}\right) \frac{h^2}{d\,\sqrt{2}\,a}, \tag{32}$$

which is just $3/4$ of the stability bound in the second-order central differencing case.

In the general NLSE case, one can get a linearized stability bound by treating the nonlinearity $|\Psi|^2$ (and the MSD boundary condition terms) as a constant, yielding

$$k < \frac{\sqrt{8}}{\max\{\|\vec{B}\|_\infty,\,\|\forall L_i,\,L_i - \vec{G}\|_\infty\}} \frac{h^2}{a}, \tag{33}$$

where $\vec{B}$ are the boundary points as defined by Table 1, $\vec{L}$ is defined as

$$L_i = \frac{h^2}{a}\left(s|\Psi_i|^2 - V(\mathbf{r}_i)\right),$$

and $\vec{G}$ is a set of values defined in Table 2, determined by the dimension being used.

Table 1
Values for B_b in Eq. (33).

	Dirichlet ($\Psi_b = $ const)	MSD ($|\Psi_b|^2 = $ const)
B_b	0	$\dfrac{h^2}{i\,a}\dfrac{1}{\Psi_{b-1}}\dfrac{\partial \Psi}{\partial t}\bigg|_{b-1}$

Table 2
Values for $\vec{G}$ in Eq. (33) for the one-, two-, and three-dimensional NLSEs.

	$d = 1$	$d = 2$	$d = 3$
$\vec{G}$	$\dfrac{1}{12} \times \{64, 63, 46, 12, -3, -4\}$	$\dfrac{1}{12} \times \{128, 127, 126, 110, 109, 92, 24, 9, 8, -6, -7, -8\}$	$\dfrac{1}{12} \times \{192, 191, 190, 189, 174, 173, 172, 156, 155,$ $138, 36, 21, 20, 6, 5, 4, -9, -10, -11, -12\}$

As mentioned in Ref. [30], since these stability results are based on a linearized approximation to the full nonlinear problem, in practice, one must use a time-step that is lower than the bounds given. In our experience, setting the time-step to be 80% of the given bounds ensures stability in most cases (in one-dimensional simulations, the step size can be up to 90% of the given bounds for most cases).

4. Numerical results

Here we show our numerical results for using the RK4+2SHOC scheme to integrate the NLSE. Since the 2SHOC schemes are algebraically equivalent to the standard fourth-order non-compact schemes, they should exhibit the same order of accuracy as well. However, because the order of computation is altered (which could introduce numeric cancellation or round-off errors), and the implementation of boundary conditions is different than when using wide stencils, numerical tests of the accuracy of the 2SHOC scheme is justified.

To test the accuracy, we integrate the NLSE using non-trivial initial conditions and record the error versus the exact solutions for various spatial step-sizes. To compute the error in the simulations, the real and imaginary parts of the wavefunction Ψ are compared to the true solution at 100 equal-spaced intervals throughout the simulation. The averaged L2-norm of the wavefunction error ($E^n = \|\Psi^n - \Psi_{\text{exact}}^n\|_2/\mathbf{N}$, where n is the current time step and $\mathbf{N}$ is the total number of grid points) is computed at each time interval. Using the averaged L2-norm error is necessary since we are using a fixed domain and therefore the total L2-norm would be greater for smaller h due to the increase in the number of grid points. When the simulation is completed, we define the error of the real and imaginary parts of the whole simulation as the mean of the errors at each of the 100 intervals ($E = \Sigma E^n/K$, where $K = 100$ is the number of time intervals). To compute the order of error, we define the error order between two simulations with spatial steps h_1 and h_2 as $O = \ln(E_{h_1}) - \ln(E_{h_2})$. The overall order of the scheme is determined by taking the mean of the average of the real and imaginary error orders for each run. For comparison, we run each simulation using the classic second-order central-differencing (CD) in space as well.

We note that since the RK4 scheme is $O(k^4)$, and the stability bounds require that $k \propto h^2$, the errors in the simulations attributed to the time-stepping are negligible compared to those due to the spatial differencing, and therefore should not effect the accuracy tests.

Each numerical test utilizes the single-storage formulation of the RK4-2SHOC schemes where applicable and is performed using our freely distributed[3] MATLAB-interfaced GPU-accelerated code package NLSEmagic [37]. Specifically, we make use of the 'full research scripts' and the double-precision GPU-accelerated 2SHOC integrators compiled from the version 'v013' source packages for one, two, and three dimensions. The precise configuration of the scripts used in computing the results given in this paper can be obtained from the 'Reproducible Figure Package' available on the NLSEmagic website.[3]
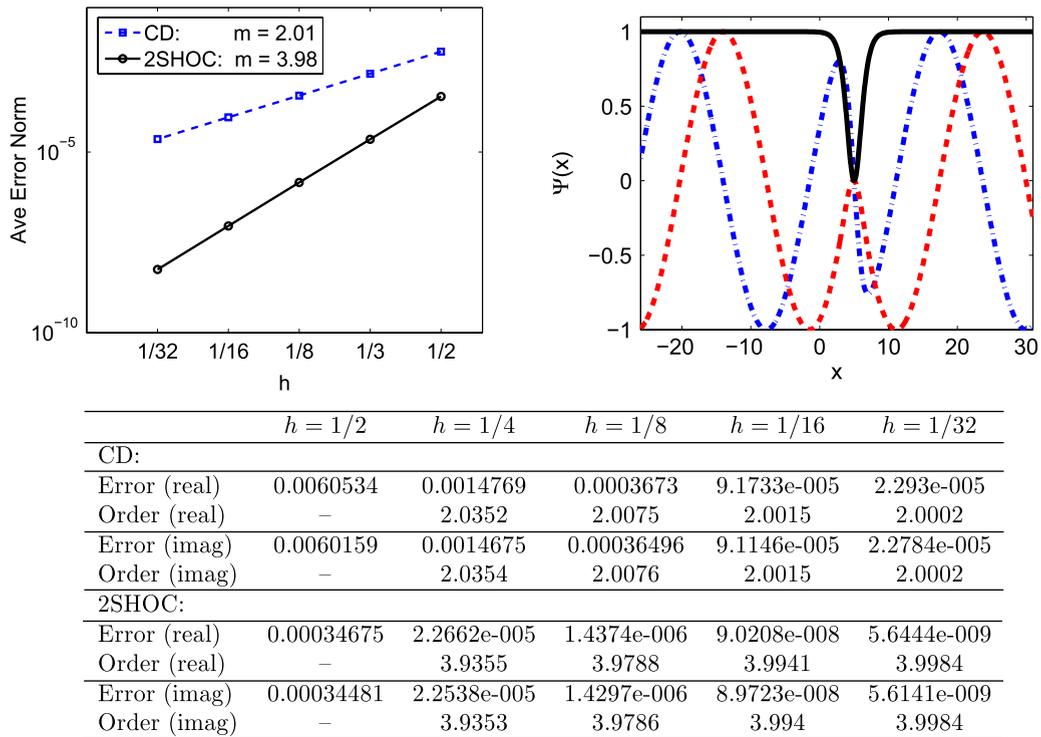
4.1. One-dimensional test

For the one-dimensional test of the RK4+2SHOC scheme, we use a known dynamical solution to the one-dimensional NLSE (with $V(\mathbf{r}) = \mathbf{0}$ and $s < 0$) of a co-moving dark soliton given by [4]

$$\Psi(x, t) = \sqrt{\frac{\Omega}{s}} \tanh\left[\sqrt{\frac{|\Omega|}{2\,a}}\,(x - c\,t)\right] \exp\left(i\left[\frac{c}{2\,a}x + \left(\Omega - \frac{c^2}{4\,a}\right)t\right]\right), \tag{34}$$

where Ω is the frequency and c is the velocity of the soliton. We use the modulus-squared Dirichlet boundary condition ($|\Psi|^2 = $ constant) of Eqs. (29) and (30) and set the size of the computational domain large enough so that the wavefunction's

---

	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$	$h = 1/32$
CD:					
Error (real)	0.0060534	0.0014769	0.0003673	9.1733e-005	2.293e-005
Order (real)	–	2.0352	2.0075	2.0015	2.0002
Error (imag)	0.0060159	0.0014675	0.00036496	9.1146e-005	2.2784e-005
Order (imag)	–	2.0354	2.0076	2.0015	2.0002
2SHOC:					
Error (real)	0.00034675	2.2662e-005	1.4374e-006	9.0208e-008	5.6444e-009
Order (real)	–	3.9355	3.9788	3.9941	3.9984
Error (imag)	0.00034481	2.2538e-005	1.4297e-006	8.9723e-008	5.6141e-009
Order (imag)	–	3.9353	3.9786	3.994	3.9984

Fig. 1. (Color online) Top left: overall order of accuracy (m) computed from simulating the one-dimensional NLSE with the initial condition of Eq. (34) for the RK4+CD and RK4+2SHOC schemes. Top right: snap-shot of the simulation with $h = 1/32$. The solid (black) line is the modulus-squared $|\Psi|^2$, while the dot-dashed (blue) and dashed (red) lines are the real and imaginary parts of Ψ respectively. Bottom: table of error values and order of accuracy values for each spatial step h. The parameters used for the solution are $s = -1$, $a = 1$, $c = 0.5$, and $\Omega = -1$. The solution is integrated to an end-time of $t = 10$ with a time-step size of $k = 0.0005$.

density is machine epsilon ($\epsilon \approx 10^{-16}$) lower than the background density at the boundaries throughout the entire time of the simulation. We set $s = -1$, $a = 1$, $c = 0.5$, and $\Omega = -1$ in Eq. (34). The grid spatial-step h is varied from 1/2 to 1/32. We run the simulation to an end-time of $t = 10$ with a time-step size of $k = 0.0005$ (which is slightly less than the maximum stability bound for the smallest value of h used) resulting in 20,000 time steps.

The results of the simulations are shown in Fig. 1, where the fourth-order accuracy of the RK4+2SHOC scheme is easily observed.

### 4.2. Two-dimensional test

There is no readily available, non-trivial, exact two-dimensional solution to the NLSE to use for order comparisons. However, since the RK4+2SHOC scheme is explicit (and therefore does not require any special handling of the nonlinearity such as iterative methods), the accuracy of the scheme can be tested reliably in a linear setting (where $s = 0$). The chosen test problem is the Gaussian wavepacket solution

$$\Psi(x, y, t) = \exp\left(-\frac{x^2 + y^2}{2a}\right)\exp(-i\,2\,t), \tag{35}$$

where $V(x, y)$ is the external potential

$$V(x, y) = \frac{x^2 + y^2}{a}. \tag{36}$$

We use Dirichlet boundary conditions ($\Psi = 0$) and set $a = 1$. We set the size of the computational domain large enough so that the exact solution has a value of $\Psi_b = \sqrt{\epsilon}$ (where $\epsilon \approx 10^{-16}$) at the boundaries. The simulation time and number of intervals for error computations are the same as in the one-dimensional test in Section 4.1. We vary h from $h = 1$ to $h = 1/16$ and use a time-step of $k = 0.001$ resulting in 10,000 time steps. The results are shown in Fig. 2. Like in the one-dimensional case, the fourth-order accuracy of the RK4+2SHOC scheme is observed.
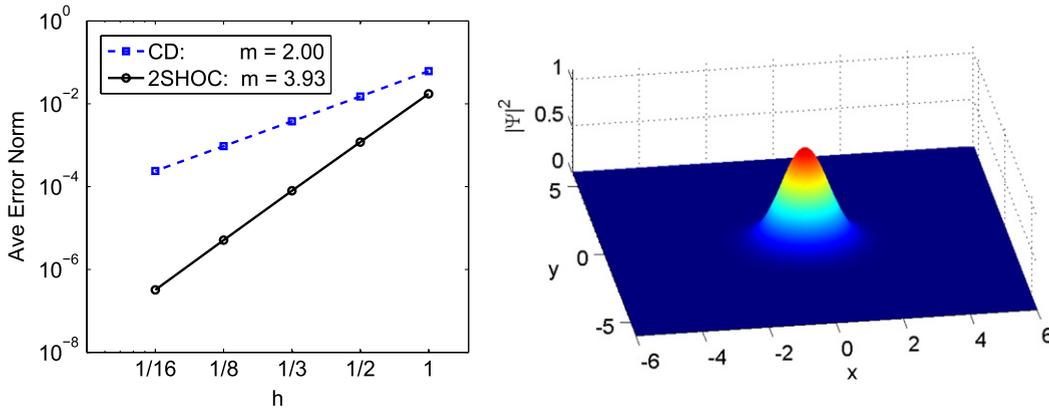
	$h = 1$	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
CD:					
Error (real)	0.059782	0.01451	0.0036626	0.00092345	0.00023196
Order (real)	–	2.0426	1.9861	1.9878	1.9932
Error (imag)	0.062855	0.015442	0.0038944	0.00097946	0.00024585
Order (imag)	–	2.0252	1.9874	1.9913	1.9942
2SHOC:					
Error (real)	0.017015	0.0011492	7.7072e-005	4.9388e-006	3.1147e-007
Order (real)	–	3.8882	3.8982	3.964	3.987
Error (imag)	0.017684	0.001228	8.2419e-005	5.2815e-006	3.3308e-007
Order (imag)	–	3.848	3.8972	3.964	3.987

Fig. 2. (Color online) Top left: overall order of accuracy (m) computed from simulating the two-dimensional linear Schrödinger equation with the initial condition of Eq. (35) for the RK4+CD and RK4+2SHOC schemes. Top right: depiction of the modulus-squared $|\Psi|^2$ of the wavefunction with $h = 1/16$ at $t = 0$. Bottom: table of error values and order of accuracy values for each spatial step h. The parameters used for the solution are $s = 0$, and $a = 1$. The solution is integrated to an end time of $t = 10$ with a time-step size of $k = 0.001$.

4.3. Three-dimensional test

As in the two-dimensional case, there is no readily available non-trivial exact solution to test the three-dimensional NLSE. Therefore, we again use a linear example choosing the three-dimensional analog of Eq. (35) defined as

$$\Psi(x, y, t) = \exp\left(-\frac{x^2 + y^2 + z^2}{2\,a}\right)\exp(-i\,3\,t), \tag{37}$$

with external potential

$$V(x, y) = \frac{x^2 + y^2 + z^2}{a}. \tag{38}$$

As before, Dirichlet boundary conditions ($\Psi = 0$) are used and a is set to 1. The size of the computational domain is once again set to be large enough so that the exact solution has a value of $\Psi_b = \sqrt{\epsilon}$ at the boundaries. The simulation time and number of intervals for error computations are the same as in the one-dimensional test in Section 4.1. The step-size is varied from $h = 1$ to $h = 1/16$ and a time-step of $k = 0.0005$ is used resulting in 20,000 time steps. The results are shown in Fig. 3, where once again the fourth-order accuracy of the scheme is clearly seen. The average order of accuracy given in Fig. 3 is 3.91 which is slightly smaller than expected for the fourth-order scheme. However as seen in the corresponding table, the order of accuracy starts at around 3.8 for the first spatial step-size reduction, while in the smaller reductions, the order increases to around 3.98. This means that the slightly lower-order values are most likely due to the inability of the coarse grid to adequately resolve the solution, and not on the scheme itself. Lower values of h were not used due to computational memory constraints.

From the table in Fig. 3, it is easy to illustrate the advantage of using high-order schemes for large three-dimensional problems mentioned in the beginning of this chapter—that of being able to use a much smaller grid while maintaining the desired accuracy. For example, in Fig. 3, using the second-order scheme for $h = 1/8$ (making the grid resolution $97 \times 97 \times 97 = 912{,}673$ grid points) yielded an error of around 0.0005. Roughly the same error (0.0006) was found using the fourth-order scheme with a spatial-step size of $h = 1/2$, requiring the grid size of only $25 \times 25 \times 25 = 15{,}625$ grid points. This is a 98.3% reduction in the number of grid points required! When combined with the ease of parallelization that the 2SHOC compact scheme provides, its usefulness for large three-dimensional problems is apparent.
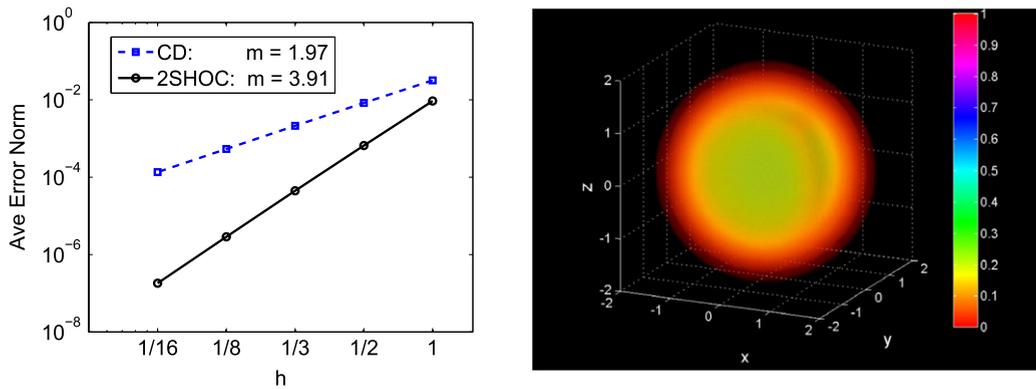
	$h = 1$	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
CD:					
Error (real)	0.03125	0.0083706	0.0021518	0.00054571	0.00013743
Order (real)	–	1.9004	1.9598	1.9793	1.9894
Error (imag)	0.032541	0.0083009	0.0021026	0.00053144	0.00013374
Order (imag)	–	1.9709	1.9811	1.9842	1.9905
2SHOC:					
Error (real)	0.0093495	0.000666	4.5165e-005	2.9087e-006	1.839e-007
Order (real)	–	3.8113	3.8822	3.9567	3.9834
Error (imag)	0.0093489	0.00064941	4.4012e-005	2.8344e-006	1.792e-007
Order (imag)	–	3.8476	3.8832	3.9568	3.9834

Fig. 3. (Color online) Top left: overall order of accuracy (m) computed from simulating the three-dimensional linear Schrödinger equation with the initial condition of Eq. (37) for the RK4+CD and RK4+2SHOC schemes. Top right: volumetric rendering of the modulus-squared $|\Psi|^2$ of the wavefunction with $h = 1/16$ at $t = 0$ (displayed on a smaller grid than used in the simulations). Bottom: table of error values and order of accuracy values for each spatial step h. The parameters used for the solution are $s = 0$, and $a = 1$. The solution is integrated to an end time of $t = 10$ with a time-step size of $k = 0.0005$.

Table 3
One-dimensional storage and computational cost analysis for the 2SHOC scheme compared to the equivalent non-compact scheme. The storage is given in terms of N, the total number of grid points.

Method	Operations	Storage	Op ratio	Storage ratio
Laplacian:				
Non-compact	7	N	–	–
2SHOC	8	$2N$	1.14	2
NLSE RK4 step:				
Non-compact	$4(7+7)+13 = 69$	$5N$	–	–
2SHOC	$4(8+7)+13 = 73$	$6N$	1.06	1.2

5. Computation and storage comparisons

In this section we compare the storage and computational requirements of the 2SHOC schemes compared to the standard fourth-order non-compact equivalent schemes for one, two and three dimensions. We count the number of operations (in terms of the number of elements in the domain, N) needed for each method ignoring the boundaries. For simplicity, we also ignore the added operations needed due to Ψ being complex, and treat all operations as acting on real variables. Also, since floating-point division operations are far more computationally expensive than additions and multiplications, we record the operations required in an optimized form of the schemes, in which all divisions are only computed once (and hence ignored) by pre-computing the constant terms. We record the number of computations and storage space required for the Laplacian operator alone using the 2SHOC, as well as when implemented into the NLSE simulations using the RK4+2SHOC schemes.

The one-dimensional analysis is shown in Table 3. We see that for the RK4+2SHOC NLSE implementation, the scheme only requires about 6% more computations and 20% more storage than the equivalent non-compact scheme. This small increase in storage and computations is minor compared to the advantages of having a compact scheme.

The two-dimensional analysis is given in Table 4. Both the double-storage version of the 2SHOC of Eqs. (13) and (14) and the single-storage version of Eqs. (17) and (18) are analyzed. Here, using the 2SHOC scheme is a greater increase in additional computations, taking 50% more (for the 1X storage version) than the non-compact scheme in the RK4 step of the NLSE. The

Table 4
Two-dimensional storage and computational cost analysis for 2SHOC schemes compared to equivalent non-compact schemes. The storage is given in terms of N, the total number of grid points.

Method	Operations	Storage	Op ratio	Storage ratio
Laplacian:				
Non-compact	9	N	–	–
2SHOC ($2X$ storage)	15	$3N$	1.66	3
2SHOC ($1X$ storage)	19	$2N$	2.11	2
NLSE RK4 step:				
Non-compact	$4(9+7)+13=77$	$5N$	–	–
2SHOC ($2X$ storage)	$4(15+7)+13=101$	$7N$	1.31	1.4
2SHOC ($1X$ storage)	$4(19+7)+13=117$	$6N$	1.52	1.2

Table 5
Three-dimensional storage and computational cost analysis for 2SHOC schemes compared to equivalent non-compact schemes. The storage is given in terms of N, the total number of grid points.

Method	Operations	Storage	Op ratio	Storage ratio
Laplacian:				
Non-compact	14	N	–	–
2SHOC ($3X$ storage)	22	$4N$	1.57	4
2SHOC ($1X$ storage)	31	$2N$	2.21	2
NLSE RK4 step:				
Non-compact	$4(14+7)+13=97$	$5N$	–	–
2SHOC ($3X$ storage)	$4(22+7)+13=129$	$8N$	1.33	1.6
2SHOC ($1X$ storage)	$4(31+7)+13=165$	$6N$	1.70	1.2

two-storage version of 2SHOC reduces this to about 30%, but with a 20% increase in the storage required when compared to the single-storage 2SHOC (which like the one-dimensional case is only 20% more than the non-compact scheme).

The three-dimensional analysis is given in Table 5. Here we see that the single-storage 2SHOC scheme in the RK4 step for the NLSE takes 70% more computations than the non-compact scheme, and for the Laplacian operator alone, takes a little more than twice the computations required for the non-compact scheme. The $1X$-storage 2SHOC scheme for the Laplacian only takes 57% more computations but also uses four times the storage. Since the main advantage of using a compact scheme is that it is easy to implement in a parallel environment and to realize near the boundaries, the increase in computation is easily offset by the parallelism (as most parallel implementations have speed-ups of well over two). In addition, often times, the number of computations an algorithm takes is not nearly as important as the memory bandwidth used. Therefore, using less memory but more computations may actually perform better than using less computations and more memory. Also, since memory capacity can be limited in certain computational environments such as GPUs, using the single-storage 2SHOC schemes allow for larger problem sizes (it is for this reason that the single-storage 2SHOC schemes were chosen for implementation in the NLSEmagic GPU code package [37] mentioned in the beginning of the previous section).

6. Conclusion

We have formulated two-step high-order compact (2SHOC) finite-difference schemes for the Laplacian operator in one, two, and three dimensions. Two forms of the multidimensional 2SHOC schemes were formulated, one requiring less storage and more computation than the other. When implementing the schemes in partial differential equation models, the schemes do not rely on other parts of the governing equation, and are therefore more general than other HOC schemes.

The schemes are well-suited for use in explicit finite-difference schemes for time-dependent problems. As an example, we have shown the implementation of the 2SHOC schemes into explicit schemes for solving the nonlinear Schrödinger equation and, through numerical simulations, have shown that the schemes display the desired accuracy.

A computation and storage analysis revealed that the 2SHOC schemes take more storage and computations (never more than roughly double) than their non-compact equivalent schemes, but that these increases are well offset due to the advantages of using a compact scheme, most notably, the easing of parallel implementations.

Acknowledgments

References

[1] L. Debnath, Nonlinear Partial Differential Equations for Scientists and Engineers, second ed., Birkhauser Boston, 2005.
[2] C. Sulem, P.L. Sulem, The Nonlinear Schrödinger Equation, Self-Focusing and Wave Collapse, in: Appl. Math. Sci., vol. 139, Springer, 1999.

[3] B.A. Malomed, Variational methods in nonlinear fiber-optics and related fields, Prog. Opt. 43 (2002) 71–193.
[4] P.G. Kevrekidis, D.J. Frantzeskakis, R. Carretero-González, Emergent Nonlinear Phenomena in Bose–Einstein Condensates: Theory and Experiment, in: Springer Series on Atomic, Optical, and Plasma Physics, vol. 45, 2008.
[5] J. Wyller, W. Królikowski, O. Bang, J.J. Rasmussen, Generic features of modulational instability in nonlocal kerr media, Phys. Rev. E 66 (2002) 066615.
[6] D.E. Pelinovsky, Y.A. Stepanyants, Yu.S. Kivshar, Self-focusing of plane dark solitons in nonlinear defocusing media, Phys. Rev. E 51 (1995) 5016.
[7] W.F. Spotz, G.F. Carey, Iterative and parallel performance of high-order compact systems, SIAM J. Sci. Comput. 19 (1) (1998) 1–14.
[8] J. Zhang, An explicit fourth-order compact finite difference scheme for three-dimensional convection–diffusion equation, Commun. Numer. Methods Eng. 14 (1998) 209–218.
[9] W.F. Spotz, G.F. Carey, A high-order compact formulation for the 3D Poisson equation, Numer. Methods Partial Differential Equations 12 (1996) 235–243.
[10] W.F. Spotz, G.F. Carey, High-order compact scheme for the steady stream-function vorticity equations, Internat. J. Numer. Methods Engrg. 38 (20) (1995) 3497–3512.
[11] L. Ge, J. Zhang, Symbolic computation of high-order compact difference schemes for three-dimensional linear elliptic partial differential equations with variable coefficients, J. Comput. Appl. Math. 143 (2002) 9–27.
[12] W. Liao, An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation, Appl. Math. Comput. 206 (2008) 755–764.
[13] M. Ciment, S.H. Leventhal, Higher order compact implicit schemes for the wave equation, Math. Comp. 29 (132) (1975) 985–994.
[14] S. Abarbanel, A. Kumar, Compact high-order schemes for the Euler equations, J. Sci. Comput. 3 (1988) 275–288.
[15] W.F. Spotz, G.F. Carey, Extension of high-order compact schemes to time-dependent problems, Numer. Methods Partial Differential Equations 17 (6) (2001) 657–672.
[16] Jiten C. Kalita, D.C. Dalal, Anoop K. Dass, A class of higher order compact schemes for the unsteady two-dimensional convection–diffusion equation with variable convection coefficients, Internat. J. Numer. Methods Fluids 38 (12) (2002) 1111–1131.
[17] A. Mohebbi, M. Dehghan, The use of compact boundary value method for the solution of two-dimensional Schrödinger equation, J. Comput. Appl. Math. 225 (2009) 124–134.
[18] J.C. Kalita, P. Chhabra, S. Kumar, A semi-discrete higher order compact scheme for the unsteady two-dimensional Shrödinger equation, J. Comput. Appl. Math. 197 (2006) 141–149.
[19] S. Xie, G. Li, S. Yi, Compact finite-difference schemes with high accuracy for one-dimensional nonlinear Schrödinger equation, Comput. Methods Appl. Mech. Engrg. 198 (2009) 1052–1060.
[20] M. Dehghan, A. Taleei, A compact split-step finite difference method for solving the nonlinear Schrödinger equations with constant and variable coefficients, Comput. Phys. Comm. 181 (2010) 43–51.
[21] J.C. Butcher, A history of Runge–Kutta methods, Appl. Numer. Math. 20 (3) (1996) 247–260.
[22] W. Milne, Numerical Solution of Differential Equations, second ed., Dover Publications Inc., 1970.
[23] Ronald M. Caplan, Study of vortex Ring dynamics in the nonlinear Schrödinger equation utilizing GPU-accelerated high-order compact numerical integrators, Ph.D. Thesis, Claremont Graduate University and San Diego State University, 2012.
[24] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, second ed., SIAM, 2004.
[25] W.F. Spotz, High-order compact finite difference schemes for computational mechanics, Ph.D. Thesis, University of Texas at Austin, 1995.
[26] V.M. Pérez-García, X. Liu, Numerical methods for the simulation of trapped nonlinear Schrödinger systems, Appl. Math. Comput. 144 (2–3) (2003) 215–234.
[27] Thiab R. Taha, Mark I. Ablowitz, Analytical and numerical aspects of certain nonlinear evolution equations, II, numerical, nonlinear Schrödinger equation, J. Comput. Phys. 55 (2) (1984) 203–230.
[28] J.A.C. Weideman, B.M. Herbst, Split-step methods for the solution of the nonlinear Schrödinger equation, SIAM J. Numer. Anal. 23 (3) (1986) 485–507.
[29] S. Zoldi, V. Ruban, A. Zenchuk, S. Burtsev, Parallel implementation of the split-step Fourier method for solving nonlinear Schrödinger systems, SIAM News 32 (1) (1999) 8–9.
[30] R.M. Caplan, R. Carretero-González, Numerical stability of explicit Runge–Kutta finite-difference schemes for the nonlinear Schrödinger equation, Appl. Numer. Math. (2013) (submitted for publication). arXiv:1107.4810.
[31] M.S. Ismail, A fourth-order explicit schemes for the coupled nonlinear Schrödinger equation, Appl. Math. Comput. 196 (2008) 273–284.
[32] M.M. Cerimele, M.L. Chiofalo, F. Pistella, S. Succi, M.P. Tosi, Numerical solution of the Gross–Pitaevskii equation using an explicit finite-difference scheme: an application to trapped Bose–Einstein condensates, Phys. Rev. E 62 (1) (2000) 1382–1389.
[33] D.J. Fyfe, Economical evaluation of Runge–Kutta formula, Math. Comp. 20 (1966) 392–398.
[34] M.H. Carpenter, C.A. Kennedy, Fourth-order 2N-storage Runge–Kutta schemes, NASA Tech. Mem., 1994.
[35] R.M. Caplan, R. Carretero-González, A modulus-squared Dirichlet boundary condition for time-dependant complex partial differential equations and its application to the nonlinear Schrödinger equation, 2013. arXiv:1110.0569.
[36] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt, A. Schädle, A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations, Commun. Comput. Phys. 4 (2008) 729–796.
[37] R.M. Caplan, NLSEmagic: nonlinear Schrödinger equation multidimensional Matlab-based GPU-accelerated integrators using compact high-order schemes, Comput. Phys. Comm. 184 (4) (2013) 1250–1271.