

A Modulus-Squared Dirichlet Boundary Condition for Time-Dependent Complex Partial Differential Equations and its Application to the Nonlinear Schrödinger Equation

R. M. Caplan* and R. Carretero-González

Nonlinear Dynamical System Group,[†] Computational Science Research Center,[‡] and
Department of Mathematics and Statistics, San Diego State University,
San Diego, California 92182-7720, USA

October 3, 2011

Abstract

An easy to implement modulus-squared Dirichlet (MSD) boundary condition is formulated for numerical simulations of time-dependent complex partial differential equations in multidimensional settings. The MSD boundary condition approximates a constant modulus-square value of the solution at the boundaries. Application of the MSD boundary condition to the nonlinear Schrödinger equation is shown, and numerical simulations are performed to demonstrate its usefulness and advantages over other simple boundary conditions.

1 Introduction

When utilizing numerical methods to approximate the solutions to time-dependent partial differential equations (PDEs), proper handling of boundary conditions can be quite challenging. Sometimes, an otherwise stable numerical scheme will become unstable depending on how the boundary conditions are computed [1]. In addition, high-order schemes can degrade in accuracy to lower-order when using boundary conditions which are not compatible with the high-order accuracy [2]

Often, researchers will forgo a complicated boundary condition implementation and instead use tried-and-true boundary conditions techniques which are very simple yet provide acceptable results. One of the most common is the use of Dirichlet boundary conditions, where the function value at the boundary of the domain is set to a constant value (commonly this value is zero). A Dirichlet boundary condition of zero-value is often used when simulating solutions which decay towards zero at infinity, and where most of the dynamics (or ‘action’) is expected to remain in the central regions of the computational grid.

Problems involving PDEs whose function values are complex cannot, in general, make use of standard Dirichlet boundary conditions because of the constant oscillation of the real and imaginary parts of the function due to the intrinsic frequency of the system and the dynamics of the solutions (when the solution decays to zero at infinity, standard Dirichlet conditions of zero-value *can* be

*Corresponding author, email: sumseq@gmail.com, phone: 510-673-0720

[†]URL: <http://nlds.sdsu.edu>

[‡]URL: <http://www.csrc.sdsu.edu>

used). There are situations (for examples, see Sec. 4) where the modulus-squared of the solution converges to a constant value at the boundary. In such a case, a modulus-squared Dirichlet (MSD) boundary condition (which keeps the modulus-squared of the solution at the boundaries constant) would be desirable.

In this paper, we present a simple way to simulate a modulus-squared Dirichlet boundary condition in time-dependent complex-valued PDEs. The MSD boundary condition is very easy to implement and has an accuracy equal to the interior numerical scheme (as long as the assumption of a constant modulus-squared value at the boundary is valid). This new boundary condition eliminates the need for overly large grids or expensive and complicated boundary conditions for many problems.

A very common time-dependent complex-valued PDE used in a wide range of applications is the nonlinear Schrödinger equation (NLSE). The 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 [3] including nonlinear optics [4] and the mean-field dynamics of Bose-Einstein condensates (BECs) (in which case the NLSE is typically modified to include an external potential term, in which case the NLSE is referred to as the Gross-Pitaevskii equation) [5]. In systems such as optics and BECs, the modulus-squared of the solution (referred to as the ‘wavefunction’) represents the observable (intensity of light and atomic density respectively). In this situation, often the dynamics of ‘dark’ structures (dark solitons, vortices, vortex lines, vortex rings, etc.) which reside inside medium are studied. These are coherent structures of very low (or zero) central density which exist inside the bulk of the medium. The most basic form of the structures can be examined by assuming an infinite-extent domain, in which case the solutions exist within an infinite constant-density background. Such a situation is very well-suited for the use of MSD boundary conditions. As such, in Sec. 4, we use simulations of dark coherent structures in the NLSE to test the MSD boundary condition.

Many sophisticated boundary conditions have been developed for both the linear and nonlinear Schrödinger equations which simulate transparent or artificial boundaries (see the review of Ref. [6] and references therein). Most of these boundary conditions focus on eliminating reflections off the boundaries (the biggest problem when using Dirichlet boundary conditions) when studying dynamics which trail off to zero at infinity. Since our main concern are boundary conditions in a constant-density scenario, we do not make use of the boundary conditions described there. Additionally, one of the main features of the MSD boundary condition is that its simplicity of implementation allows small research projects to make common use of them. The boundary conditions shown in Ref. [6] can be very complicated to implement, and are therefore best suited for large projects.

The paper is organized as follows: In Sec. 2 we formulate the MSD boundary condition in general for any time-dependent complex-valued PDE. Then, in Sec. 3, we apply the MSD boundary condition to the NLSE. In Sec. 4 we numerically test the MSD boundary condition for simulating the NLSE using a Runge-Kutta finite-difference scheme. Stability effects when using the MSD boundary condition are discussed in Sec. 5. We conclude in Sec. 6.

2 Formulation of the MSD Boundary Condition

2.1 Notation

For this paper we use the notation Ψ_b to describe the grid-points on a boundary and Ψ_{b-1} to describe the grid-points one cell in from the boundary in the normal direction to the boundary

points (i.e. in two-dimensions, the top-left corner point would be a Ψ_b and the point one cell to the right and one cell down would be Ψ_{b-1}). We denote the real part of Ψ as Ψ^R and the imaginary part as Ψ^I . We also use the notation Ψ_α to denote the first derivative with respect to α ($\partial\Psi/\partial\alpha$), and so $\Psi_{\alpha,b}$ would refer to the first partial derivative with respect to α at the boundary points, where α can represent a spatial (x) or temporal (t) dependence.

2.2 Derivation of the MSD boundary Condition

We start out by stating the condition, that for all times, the modulus-squared of the function Ψ at the boundaries is equal to a constant, positive, real value B ,

$$|\Psi_b|^2 = B. \quad (1)$$

In such a case we have that

$$\frac{\partial}{\partial t} |\Psi_b|^2 = 0,$$

and thus

$$\frac{\partial}{\partial t} (\Psi_b^R)^2 + \frac{\partial}{\partial t} (\Psi_b^I)^2 = 0. \quad (2)$$

By using the chain rule and rearranging, Eq. (2) is equivalent to

$$\Psi_b^R \frac{\partial \Psi_b^R}{\partial t} = -\Psi_b^I \frac{\partial \Psi_b^I}{\partial t}. \quad (3)$$

A non-trivial solution that satisfies Eq. (3) is given by

$$\frac{\partial \Psi_b^R}{\partial t} = C \Psi_b^I \quad \text{and} \quad \frac{\partial \Psi_b^I}{\partial t} = -C \Psi_b^R, \quad (4)$$

where C is a constant. Since $\Psi_{t,b} = \Psi_{b,t}^R + i \Psi_{b,t}^I$, Eq. (4) can be expressed as

$$\Psi_b = \frac{i}{C} \Psi_{t,b}. \quad (5)$$

Eq. (5) gives the form of the MSD boundary condition on Ψ_t in terms of the boundary point Ψ_b and the constant C . In order to find the correct value of C , we differentiate Eq. (5) in the normal direction to the boundary (here represented as the x -direction) to get

$$\frac{\partial}{\partial x} \Psi_b = \frac{i}{C} \frac{\partial}{\partial x} \Psi_{t,b}. \quad (6)$$

Discretizing the spatial derivatives of Eq. (6) using forward-differencing yields

$$\frac{\Psi_{b-1} - \Psi_b}{h} - \frac{h}{2} \Psi_{xx,b} + O(h^2) = \frac{i}{C} \left[\frac{\Psi_{t,b-1} - \Psi_{t,b}}{h} - \frac{h}{2} \Psi_{txx,b} + O(h^2) \right], \quad (7)$$

where h is the grid spacing between the boundary and the interior point in the normal direction. Inserting Eq. (5) into Eq. (7) to eliminate temporal derivatives at the boundary points yields

$$\frac{\Psi_{b-1} - \Psi_b}{h} - \frac{h}{2} \Psi_{xx,b} + O(h^2) = \frac{i}{C} \left[\frac{\Psi_{t,b-1} + i C \Psi_b}{h} + \frac{i h C}{2} \Psi_{xx,b} + O(h^2) \right]. \quad (8)$$

It is observed that the truncation terms shown in Eq. (8) cancel out, and by the same formulation, all the additional truncation terms will also cancel out. Therefore, we have

$$\frac{\Psi_{b-1} - \Psi_b}{h} = \frac{i}{C} \left[\frac{\Psi_{t,b-1} + i C \Psi_b}{h} \right], \quad (9)$$

which yields the expression for C given by

$$C = i \frac{\Psi_{t,b-1}}{\Psi_{b-1}}. \quad (10)$$

Eq. (10) is computable since the value of $\Psi_{t,b-1}$ is computed using the interior scheme being implemented. Inserting the value of C of Eq. (10) into the MSD boundary condition formulation of Eq. (5) yields

$$\Psi_{t,b} \approx \frac{\Psi_{t,b-1}}{\Psi_{b-1}} \Psi_b. \quad (11)$$

The MSD boundary condition of Eq. (11) can be obtained in an alternative way by noting that a steady-state (in the modulus-squared) of a complex time-dependent function can be given as

$$\Psi = |\Psi| \exp(-i \Omega t),$$

where Ω is the frequency and $|\Psi|$ is the constant-valued amplitude of Ψ . If we assume that this relationship is true at the boundary points of the grid, we have

$$\frac{\partial \Psi_b}{\partial t} = -i \Omega_b |\Psi_b| \exp(i \Omega_b t) = -i \Omega_b \Psi_b. \quad (12)$$

From Eq. (12), we get

$$\Omega_b = i \frac{\Psi_{t,b}}{\Psi_b}. \quad (13)$$

Assuming that the closest interior grid point is nearly steady-state, we can infer that

$$\Omega_{b-1} \approx i \frac{\Psi_{t,b-1}}{\Psi_{b-1}}. \quad (14)$$

If it is assumed that the frequency of the interior point is approximately equal to the frequency at the boundary (i.e. $\Omega_b \approx \Omega_{b-1}$), then by inserting Eq. (14) as Ω_b in Eq. (12), Eq. (12) becomes the MSD boundary condition of Eq. (11).

This formulation of the MSD boundary condition shows that the $\Psi_{t,b}/\Psi_b$ term in Eq. (11) should be equal to i times the frequency. Since the frequency of the solution is a real value, computing $\Psi_{t,b-1}/\Psi_{b-1}$ may introduce a small imaginary part to the frequency due to numerical errors. This would lead the solution at the boundaries to undergo exponential growth. In order to ensure that the computed frequency is real-valued (i.e. that $\Psi_{t,b-1}/\Psi_{b-1}$ is purely imaginary), we modify the MSD boundary condition of Eq. (11) to be

$$\Psi_{t,b} \approx i \operatorname{Im} \left[\frac{\Psi_{t,b-1}}{\Psi_{b-1}} \right] \Psi_b. \quad (15)$$

When using the MSD boundary condition in programming environments that do not intrinsically handle complex variables, Eq. (15) must be explicitly split into its real and imaginary parts. We

begin by expanding the unmodified MSD boundary condition of Eq. (11) into its real and imaginary parts, given by

$$\Psi_{t,b}^R + i \Psi_{t,b}^I \approx \frac{\Psi_{t,b-1}^R + i \Psi_{t,b-1}^I}{\Psi_{b-1}^R + i \Psi_{b-1}^I} (\Psi_b^R + i \Psi_b^I).$$

which leads to

$$\Psi_{t,b}^R + i \Psi_{t,b}^I \approx \left[\frac{\Psi_{t,b-1}^R \Psi_{b-1}^R + \Psi_{t,b-1}^I \Psi_{b-1}^I}{(\Psi_{b-1}^R)^2 + (\Psi_{b-1}^I)^2} + i \left(\frac{\Psi_{t,b-1}^I \Psi_{b-1}^R - \Psi_{t,b-1}^R \Psi_{b-1}^I}{(\Psi_{b-1}^R)^2 + (\Psi_{b-1}^I)^2} \right) \right] (\Psi_b^R + i \Psi_b^I). \quad (16)$$

Going back to Eq. (3), we see that the first term in the brackets of Eq. (16) is equal to zero at the boundary. If it is assumed that the interior points are similar, the term can be dropped. Dropping the term is equivalent to the numerical fix used in Eq. (15), ensuring that the frequency of the solution Ψ at the boundary is real-valued. Simplifying Eq. (16) with this in mind yields the separated MSD boundary condition

$$\begin{aligned} \Psi_{t,b}^R &= -\tilde{\Omega} \Psi_b^I \\ \Psi_{t,b}^I &= \tilde{\Omega} \Psi_b^R, \end{aligned} \quad (17)$$

where

$$\tilde{\Omega} = \frac{\Psi_{t,b-1}^I \Psi_{b-1}^R - \Psi_{t,b-1}^R \Psi_{b-1}^I}{(\Psi_{b-1}^R)^2 + (\Psi_{b-1}^I)^2}. \quad (18)$$

The MSD boundary condition of Eqs. (15), (17), and (18) is given for the temporal derivative of the boundary point. This is ideally suited for Runge-Kutta type solvers, as the right hand side of the PDEs (Ψ_t) are evaluated and used for the time-stepping [7]. For other methodologies, or in situations where the boundary value of the spatial derivatives is needed, the MSD boundary condition can be inserted into the governing equation to formulate the required boundary conditions. An example of this is shown in Sec. 3 for the NLSE.

In situations where the sequential computation of the internal scheme followed by the boundary condition computations is not appropriate (for example, implicit finite-difference schemes) one can substitute the internal scheme of $\Psi_{t,b-1}$ into Eq. (11) and implement the boundary condition concurrently with the internal scheme (this would add extra computational steps as $\Psi_{t,b-1}$ would essentially be computed twice). Alternatively, if the frequency, Ω , of the overall solution is known, the $\Psi_{t,b-1}/\Psi_{b-1}$ term can be replaced with $i\Omega$ directly as shown above.

If it happens that $\Psi_{b-1} = 0$, the MSD boundary condition of Eq. (11) has a singularity. However, in most situations, Ψ_{b-1} only takes a zero value when the solution is tending toward zero at the boundary. In that case, the standard Dirichlet boundary condition of $\Psi_b = 0$ can be used instead of the MSD boundary condition.

2.3 Key Features of the MSD Boundary Condition

The following are a few key features of the MSD boundary condition:

- The formulation of the MSD boundary condition does not depend at all on the specific PDE being simulated, only that it is time-dependent, and is complex-valued.
- The MSD boundary condition can be used in multidimensional settings without modification since each boundary point only uses one interior point in the normal direction to the boundary.

- The MSD boundary condition does not depend on the size of the grid spacing, and therefore does not need to be altered when using unequal grid spacing (for example, in adaptive mesh refinement applications).
- The MSD boundary condition can be considered compact, in that after computing the internal scheme, the boundary values only depend on their nearest neighboring grid point. This allows the MSD boundary condition to be easily implemented in parallel environments.
- In general, the MSD boundary condition is extremely easy to implement (see Appendix A for MATLAB code showing an implementation of the MSD boundary condition in one-dimension). This makes the MSD boundary condition an attractive alternative to more complicated boundary condition methodologies.

3 Application of the MSD boundary condition to the NLSE

Here we show an implementation of the MSD boundary condition for the NLSE. The NLSE with an external potential can be given in general form as

$$i \frac{\partial \Psi}{\partial t} + a \nabla^2 \Psi - V(\mathbf{r}) \Psi + s |\Psi|^2 \Psi = 0, \quad (19)$$

where Ψ is the wavefunction, $V(\mathbf{r})$ is an external potential, and a and s are parameters determined by the physical system being modeled.

As mentioned in Sec. 2.2, since the MSD boundary condition of Eq. (15) is given in terms of the temporal derivative, it is well suited for Runge-Kutta schemes, in which case it can be applied directly. However, other numerical schemes require boundary conditions on the Laplacian operator itself. This can be worked out by inserting Eq. (19) into Eq. (15) yielding

$$\nabla^2 \Psi_b \approx \left[\text{Im} \left(i \frac{\nabla^2 \Psi_{b-1}}{\Psi_{b-1}} \right) + \frac{1}{a} (N_{b-1} - N_b) \right] \Psi_b, \quad (20)$$

where

$$N_b = s |\Psi_b|^2 - V_b, \quad N_{b-1} = s |\Psi_{b-1}|^2 - V_{b-1}. \quad (21)$$

Splitting Eq. (20) into real and imaginary parts yields

$$\begin{aligned} \nabla^2 \Psi_b^R &\approx \left[A + \frac{1}{a} (N_{b-1} - N_b) \right] \Psi_b^R, \\ \nabla^2 \Psi_b^I &\approx \left[A + \frac{1}{a} (N_{b-1} - N_b) \right] \Psi_b^I, \end{aligned} \quad (22)$$

where

$$A = \frac{\nabla^2 \Psi_{b-1}^R \Psi_{b-1}^R + \nabla^2 \Psi_{b-1}^I \Psi_{b-1}^I}{(\Psi_{b-1}^R)^2 + (\Psi_{b-1}^I)^2}, \quad (23)$$

and N_{b-1} and N_b are as defined in Eq. (21).

As discussed in Sec. 2.2 the MSD boundary condition can be expanded out, expressing the $b - 1$ Laplacian in terms of the internal scheme in order to be able to evaluate the MSD boundary condition simultaneously with the interior points. (When using explicit time-stepping schemes, this

is usually unnecessary). As an example, using central-differencing in space for the one-dimensional NLSE, Eq. (20) becomes

$$\nabla^2 \Psi_b \approx \left[\text{Im} \left(i \frac{\Psi_b + \Psi_{b-2}}{\Psi_{b-1}} \right) - 2 + \frac{1}{a} (N_{b-1} - N_b) \right] \Psi_b, \quad (24)$$

where N_{b-1} and N_b are as defined in Eq. (21).

4 Numerical Results

In order to demonstrate the usefulness and advantages of the MSD boundary condition, we show a few example simulations of the NLSE. The MSD boundary condition is compared to a Laplacian-zero (L0) boundary condition defined as $\nabla^2 \Psi_b = 0$. The Laplacian-zero boundary condition is chosen for comparison because it is an easy-to-implement boundary condition that is possible to use for constant background-density simulations in specific cases.

Each boundary condition, including the MSD, are only valid given their own assumptions. Therefore if the modulus-squared of a solution is changing at the boundary, the MSD is not expected to work well, just as if the Laplacian of the wavefunction is far from zero at the boundary, the Laplacian-zero boundary condition is not expected to do well. Therefore, comparisons of which boundary condition is best is very often problem-specific. That being said, comparing the MSD to the L0 is still valuable in that for some problems, both boundary conditions are suitable allowing for a fair comparison and, in addition, since limiting the size of the required grid is very important (especially for higher-dimensional simulations), it is useful to see which boundary condition allows for the use of the smallest (tightest) grid within acceptable accuracy limits.

While there are numerous numerical methods that can be used to simulate solutions to the NLSE, we choose to perform the simulations using the code package NLSEmagic [8] which uses the fourth-order in time Runge-Kutta method with second-order central-differencing in space (RK4+CD).

4.1 One-Dimensional Dark Solitons in the NLSE

For the one-dimensional tests, a co-moving dark soliton solution with $V(\mathbf{r}) = 0$ is used, given by [5]

$$\Psi(x, t) = \sqrt{\frac{\Omega}{s}} \tanh \left[\sqrt{\frac{\Omega}{2a}} (x - ct) \right] \exp \left(i \left[\frac{c}{2a} x + \left(\Omega - \frac{c^2}{4a} \right) t \right] \right) \quad (25)$$

where c is the velocity of the soliton and Ω is a chosen parameter representing the frequency of the solution.

The first test is to compare the error for a steady-state dark soliton with $c = 0$. In such a case, for a large enough domain, the L0 boundary condition can be used since Ψ flattens out at infinity. In contrast, the MSD boundary condition should work on any sized domain, since its underlying assumption of constant density is valid anywhere along the steady-state solution. A one-sided second-order differencing (1SD) boundary condition defined in one-dimension as

$$\frac{\partial^2 \Psi}{\partial x^2} \approx \frac{1}{h^2} (-\Psi_{b-3} + 4\Psi_{b-2} - 5\Psi_{b-1} + 2\Psi_b),$$

is also used for comparison in this case.

We define a radius, r , which represents the distance from the center of the soliton to the edge of the computational domain. Simulations are performed for various lengths of r ranging from $r = 5$ (approximately equal to the width of the soliton) to $r = 25$ (the distance where the boundary value of the soliton is approximately equal to the infinite background density minus machine epsilon $\epsilon \approx 10^{-16}$). The average of the maximum errors in the real and imaginary part of Ψ over the length of the simulations are recorded. The results are shown in Fig. 1. Keeping in mind that $h^2 = 10^{-2}$

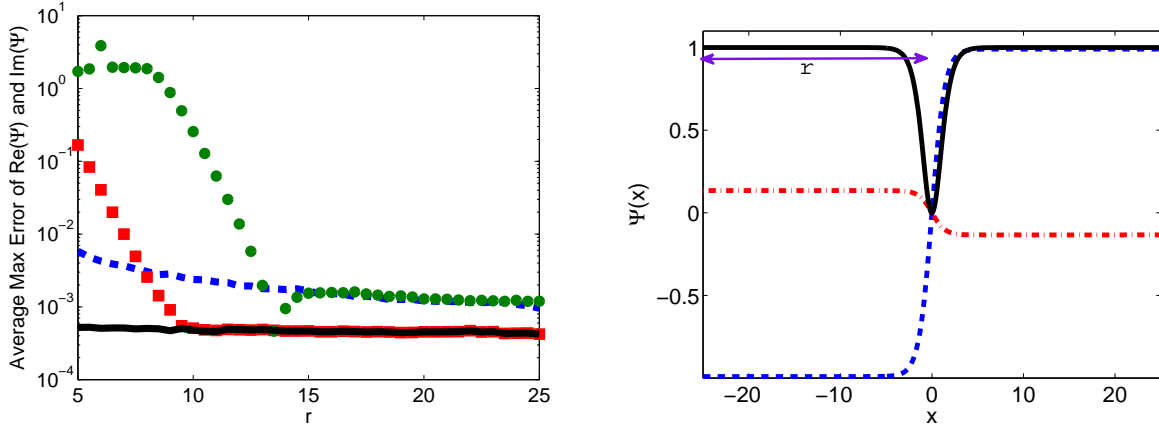


Figure 1: Left: Comparisons of errors for simulating the dark-soliton solution (with $c = 0$, $\Omega = -1$, $a = 1$, and $s = -1$) of Eq. (25) using MSD (blue dashed line), L0 (red squares), 1SD (green dots), and exact (black solid line) boundary conditions for various domain sizes. The simulations are run to an end time of $t = 50$ with a time-step of $k = 0.006$ and spatial step of $h = 0.1$. Right: Depiction of the soliton for the maximum domain ($r = 25$) simulation at time $t = 50$. The dashed (blue) and dotted (red) lines are the real and imaginary parts of Ψ respectively, while the solid line is the modulus-squared, $|\Psi|^2$. The definition of the radius r in the error plots is illustrated.

and the spatial scheme is $O(h^2)$, it is clear that the MSD boundary condition performs well even when the domain is small. Both the L0 and the MSD boundary conditions outperform the 1SD boundary condition. It is understandable that the L0 boundary condition outperforms the MSD when r get large, as the Laplacian tends towards zero rapidly as r increases, and the L0 boundary condition has no additional error associated with it. Even so, the MSD can simulate the solution at a smaller grid size than the L0 can to acceptable accuracy, demonstrating its usefulness in this case.

For the next test, the soliton is given a velocity of $c = 0.5$. Since the L0 assumptions are completely invalid at any domain size, it is not used for comparison (the 1SD is also not used as it fails quickly at any domain size as well). Therefore, the MSD boundary condition is only compared to the exact boundary condition. In this case the domain is set to be a distance r to the left of the initial position of the soliton, and a distance $r + cT$ to the right (where T is the simulation end-time) in order to account for its movement. In Fig. 2, we show the results of the simulations. We see that the MSD boundary condition performs quite well as long as the soliton is far enough from the boundaries (in this case ‘far enough’ is about equal to where the background density minus Ψ at the boundary is appropriately h^2). The observation that the MSD boundary condition has less error than the exact boundary condition can possibly be explained by the fact that using exact boundary conditions in fourth-order Runge-Kutta schemes can actually introduce errors in

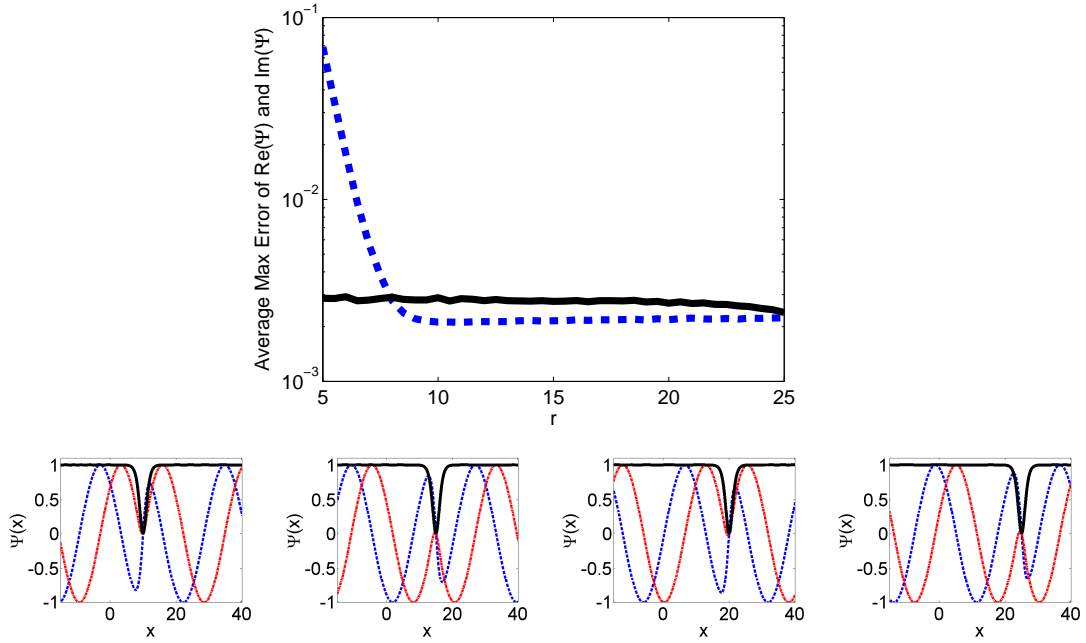


Figure 2: Top: Comparisons of errors for simulating the dark-soliton solution of Eq. (25) using MSD (blue dashed line), and exact (black solid line) boundary conditions for various domain sizes. The simulation parameters and figure descriptions are the same as in Fig. 1 except that here, the velocity is $c = 0.5$. Bottom: Depiction of the soliton during the simulation with $r = 15$ at times $t = 20, 30, 40, 50$ using the MSD boundary condition.

the scheme as described in Ref. [2]. As far as our review of the literature has gone, the MSD seems to be the *only* simple-to-implement boundary condition that can handle such a co-moving back-flow.

4.2 Two-Dimensional Dark Vortices in the NLSE

To test the MSD boundary condition in a more complicated setting, we use the known dynamics of dark vortices in the two-dimensional NLSE. Dark vortices are described as

$$\Psi(r, \theta, t) = f(r) \exp [i(m\theta + \Omega t)] \quad (26)$$

where m is the vortex charge (also known as the winding number) and Ω is the frequency which is directly related to the background density, ρ , as $\rho = \Omega/s$. The real-valued radial profile $f(r)$ can be obtained numerically by inserting Eq. (26) into Eq. (19) and solving the resulting ODE for $f(r)$ using a nonlinear equation solver (in our case, a Newton-Krylov GMRES(m) solver in a package called `nsoli` [9]). As an initial iterate for the solver, we use the asymptotic profile approximation given by [10]

$$f(r > 0) \approx \text{Re} \left[\sqrt{\frac{\Omega}{s} + \frac{am^2}{sr^2}} \right]. \quad (27)$$

It is important to note that the tails of the dark vortices converge to the background density much slower than the one-dimensional dark solitons of Eq. (25). For example, in the one-dimensional dark soliton, we could extend the domain to a size so the wave-function is at a value of $\rho - \epsilon$ ($\epsilon \approx 10^{-16}$), and this would give a radius of around 25 (for our parameter choices). To get the same boundary value in the two-dimensional vortex of charge $m = 1$, we would require a radius of over 1 million! Therefore, the ability of each boundary condition to not effect the dynamics of the system on a small grid is vital.

The first test is to simulate a single unitary-charged ($m = 1$) vortex which is known to be a stable, steady-state solution to the NLSE [5]. An important quantity when discussing vortices is the phase of the solution, ϕ , defined as $\phi = \arg(\Psi)$. The gradient of the phase is regarded as the fluid velocity of the solution, and determines the interaction dynamics of the vortices [11].

Choosing a moderate domain size (a 120×120 grid with spacing of $h = 0.25$), we integrate the NLSE for considerable time (up to $T = 50,000$) using both the MSD and L0 boundary conditions. The results are shown in Fig. 3. It is easy to see that the L0 boundary condition is only useful for shorter simulations, as it quickly suffers from phase discrepancies which get worse as time progresses, until the point where the solution is distorted badly enough to be unusable. From successive tests it is found that this effect occurs for the L0 boundary condition for even very large domains, but the time of the onset of the distortions is delayed longer as the domain size is increased. In contrast to this, the MSD boundary condition creates *no distortions* in the phase or modulus-squared of the solution for very long simulations (in this case up to $t = 50,000$).

Given a set end-time, it is useful to determine the size of the grid needed to properly simulate the vortex for a given boundary condition. Using the initial vortex solution as representing the ‘exact’ steady-state solution, we can track the error in the modulus-squared of the solution over the course of the simulations as the grid-size is increased. A radius r is defined as the distance from the center of the vortex to the edge of the domain in the x or y direction. Fig. 4 shows the results of varying r from 5 to 35 for a simulation with an end-time of $t = 300$. It is clearly seen that a much larger grid is required for the L0 boundary condition to be close to the effectiveness of the MSD boundary condition in this case. For example, it took nearly a 200×200 grid for the L0 boundary condition to produce the same accuracy as the MSD boundary condition did on a 20×20 grid!

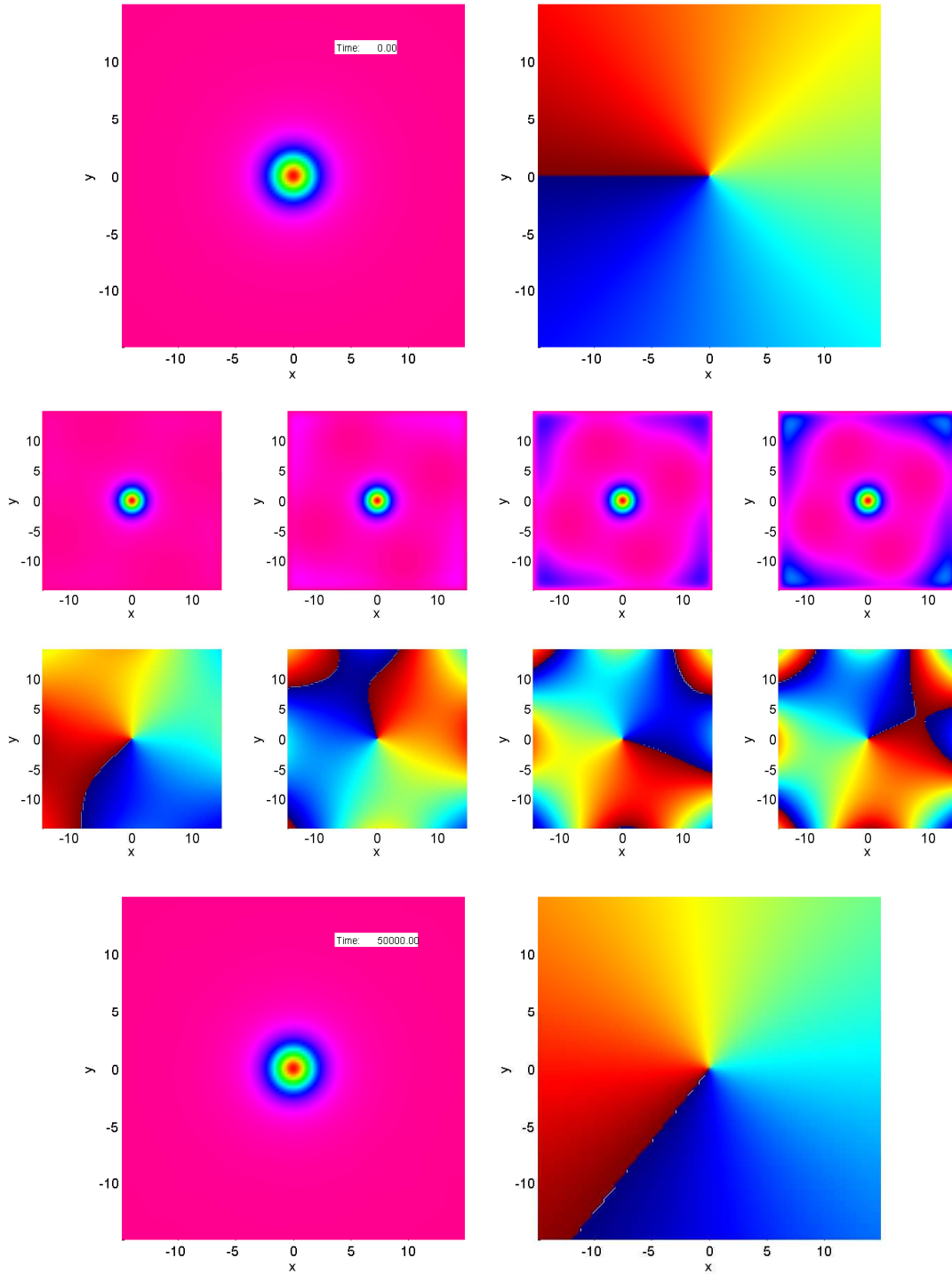


Figure 3: Top row: Modulus-squared and phase of the initial condition of a single dark vortex solution. Second (third) row: Snapshots of the density (phase) using the Laplacian-zero boundary condition at times 600, 1500, 2400, and 3000. Bottom row: Modulus-squared and phase of simulation using the MSD boundary condition at time $t = 50,000$. All simulations use a spatial-step of $h = 0.25$ and a time-step of $k = 0.01$ on a grid size of 120×120 .

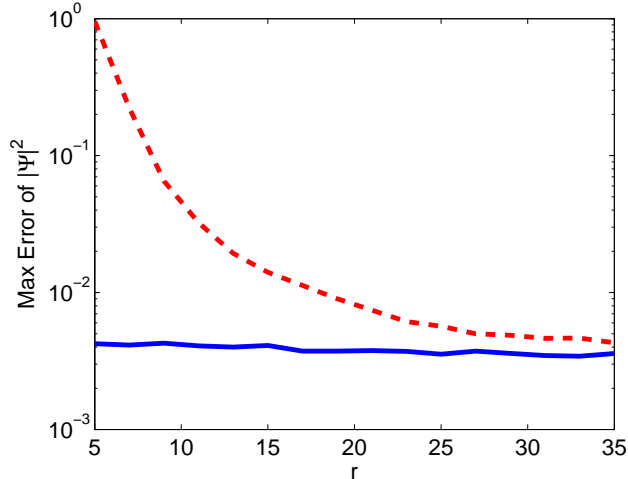


Figure 4: Maximum error of the modulus-squared of a steady-state dark vortex using the L0 (red dashed line) and the MSD (solid blue line) boundary conditions for various domain sizes. The initial numerically optimized solution of the vortex is used as the ‘true solution’ for error comparisons. The simulations were run to an end-time of $t = 300$ with a spatial-step of $h = 0.25$ and time-step $k = 0.01$.

As an additional example to test the MSD boundary condition, we simulate two equal-charge vortices whose interaction is known to produce a rotating circular motion of the two vortices orbiting each other [12]. Using a fixed grid size of 171×171 , the simulations are run for long times using the L0 and MSD boundary conditions. The results are shown in Fig. 5. We see that once again, using the L0 boundary condition causes a break-down in the dynamics, eventually causing the two vortices to decouple from each other and fling into the boundaries. The MSD boundary condition, by contrast, allows for near-perfect rotational dynamics for indefinite simulation times, even for small grid sizes. It is also noticeable that the period of rotation of the vortices is shorter when using the MSD boundary condition when compared to using the L0 boundary condition. Through further simulations with larger grid sizes, we have observed that the period of rotation converges to the same value (between the two values of the period displayed) for both boundary conditions at approximately the same grid-size (around 250×250). Therefore, the MSD boundary condition performs as well as the L0 in terms of rotational period, but much better in terms of long-term dynamics and minimum grid-size requirements.

If the end-time of the simulation is fixed to be such that the vortices will rotate at least one complete rotation (here we use $t = 480$), we can record the deviation from perfect circular motion as the grid size (given as the distance, d , from the center of the grid to the edge along the x or y direction) is varied. The vortices are tracked during the simulations and the maximum deviation from a constant radius to the center of the rotating vortices is recorded. The results are shown in Fig. 6. We see that the L0 boundary condition requires a very large grid size to capture the correct dynamics (and completely fails for smaller grids), while the MSD is able to capture the dynamics to an acceptable degree on a much smaller grid-size. The discrepancies when using the MSD boundary condition at low grid sizes (up to 20% radius variation) is understandable since at those distances the boundaries become far from steady-state due to the indentations in the background density due caused by the motion of the vortices.

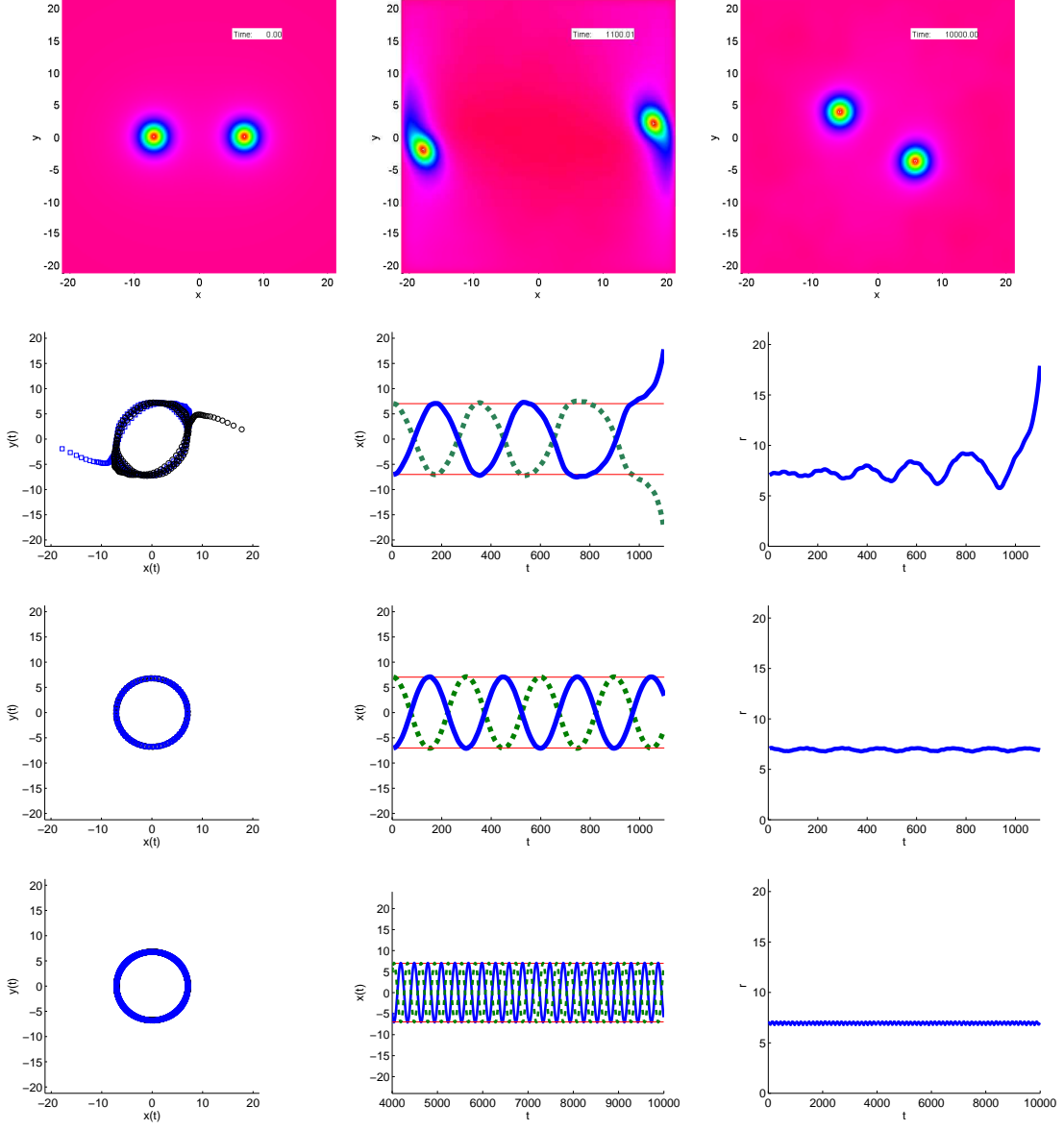


Figure 5: Top row, left to right: Modulus-squared of initial condition of two single-charge vortices separated by a distance of 7 from the center of the grid, snapshot of simulation using L0 boundary condition at time $t = 1100$, and snapshot using MSD boundary condition at time $t = 10,000$. Second row, left to right: Traced positions of the two vortices (square and circles respectively) over the course of the simulation using L0 boundary conditions, x positions of the two vortices (solid and dashed line respectively) versus time, and the computed radius of one vortex from the center of the grid for an end time of $t = 1100$. Third row: Same description as the second row, but using the MSD boundary condition. Bottom row: Same as in the third row, but with an end-time of $t = 10,000$ (only times 5000 through 10,000 shown in the x versus t plot). In all simulations the spatial-step is $h = 0.25$ and the time-step is $k = 0.01$ with a grid size of 171×171 .

4.3 Three-Dimensional Dark Vortex Rings in the NLSE

To further show the usefulness of the MSD boundary condition in multi-dimensional settings, we provide one additional example simulation — that of a steady-state vortex ring amidst a co-moving back-flow.

Unlike two-dimensional vortices, vortex rings have an intrinsic transverse velocity associated with them [13]. Therefore, in order to run long simulations of the rings (for instance, to study stability under small perturbations, or interactions between multiple co-moving rings), a very large grid size is required. Often, due to the size of the simulations, a large enough grid is not within the memory limitations of the computers being used for the simulations. To avoid this problem, a vortex ring can be made to be a ‘steady-state’ by applying a background velocity equal and opposite to the vortex ring’s intrinsic velocity. By doing this, long-term simulations of the rings can be performed, but with many fewer grid points.

In Fig. 7 we show a simulation of a moving vortex ring, as well as a steady-state vortex ring amidst a back-flow at various time intervals using the MSD boundary condition. The vortex ring solution is found by seeding the numerically-exact two-dimensional vortex described in Sec. 4.2 (with the correct back-flow velocity added to it) into a nonlinear optimization routine utilizing the two-dimensional axisymmetric version of the three-dimensional NLSE. The vortex ring remains stationary for very long simulation times (up to $t = 1500$ in this case), further demonstrating the MSD boundary condition’s usefulness in studying co-moving solutions. As in the case of the co-moving dark soliton of Sec. 4.1, the MSD boundary condition seems to be the *only* viable and easy-to-implement boundary condition for simulating a solution containing a co-moving back-flow.

5 Stability

It is well known that boundary conditions can adversely effect the stability of numerical simulations to the point where an otherwise stable scheme can become unstable [1]. Therefore, it is necessary to investigate the stability effects of the MSD boundary condition. However, since the MSD boundary condition is general in nature, its effect on the stability of the simulations will depend on the governing PDE being used, as well as the form of the overall numerical scheme being implemented. Therefore, no general statements about the MSD boundary condition’s stability effects can be made. That being said, in the specific case of the NLSE using the RK4+CD scheme that have we have used for the examples in Sec. 4, the stability effects of the MSD boundary condition can be addressed, the results of which may be useful for a variety of PDEs and numerical methods.

An analysis of the stability effects of using the MSD (and the L0) boundary conditions have been worked out as part of our prior study on the stability of RK4 schemes applied to the NLSE [14]. The relevant results are as follows.

First, in the linear case where $s = 0$, and with no external potential ($V(\mathbf{r}) = 0$), the stability bound on the time-step k of simulating the NLSE utilizing the RK4+CD method, and using standard Dirichlet boundary conditions in a d -dimensional setting is given by

$$k_{\text{linear}} < \frac{h^2}{d\sqrt{2}a}. \quad (28)$$

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

$$k < \frac{\sqrt{8}}{\max\{\|\vec{B}\|_{\infty}, \|\nabla L_i, L_i - \vec{G}\|_{\infty}\}} \frac{h^2}{a}, \quad (29)$$

where \vec{B} are the boundary points defined as

	L0 ($\nabla^2 \Psi_b = 0$)	MSD ($ \Psi_b ^2 = \text{const}$)
B_b	$\frac{h^2}{a} (s \Psi_b ^2 - V_b)$	$\frac{h^2}{a} \text{Im} \left[\frac{\Psi_{t,b-1}}{\Psi_{b-1}} \right]$

the elements of \vec{L} are defined as

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

and \vec{G} is a set of values determined by the dimension being used, defined as

	$d = 1$	$d = 2$	$d = 3$
\vec{G}	{4, 3, 1, 0}	{8, 7, 6, 2, 1, 0}	{12, 11, 10, 9, 3, 2, 1, 0}

As mentioned in Ref. [14], 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 slightly lower than the bounds given above. In our experience, setting the time-step to be 80% of the given bounds ensures stability in most cases.

The results show that the MSD boundary condition should not effect the stability greatly, as the B values are $O(h^2)$, and the value of $\Psi_{t,b-1}/\Psi_{b-1}$ (as discussed in Sec. 2.2) is roughly equal to i times the frequency of the solution at the boundaries, which is typically a reasonable value. Therefore the $L_i - \vec{G}$ terms in Eq. (29) will almost always be larger than those in \vec{B} , in which case the MSD boundary condition has no effect of the stability requirements. Also, it should be noted that the full stability bound of Eq. (29) is, in practice, often very similar in value to the simple linear bounds of Eq. (28), making the former bound a good practical bound to use.

6 Conclusion

We have shown the formulation of a modulus-squared Dirichlet (MSD) boundary condition for numerical simulations of time-dependent complex partial differential equations. The standard form of the MSD boundary condition is given as a boundary value of the time-derivative of the solution as a function of the solution at that point, as well as the solution and its time-derivative at the closest interior point. It is easily expressed as

$$\left. \frac{\partial \Psi}{\partial t} \right|_b \approx i \text{Im} \left[\frac{1}{\Psi_{b-1}} \left. \frac{\partial \Psi}{\partial t} \right|_{b-1} \right] \Psi_b,$$

where the subscripts b and $b - 1$ refer to a boundary point and the closest interior point to the boundary respectively.

Through multidimensional numerical examples of the MSD boundary condition applied to the nonlinear Schrödinger equation, we have shown that it is extremely effective in terms of non-interference with internal dynamics and accuracy, as well as in requiring smaller grid sizes when compared to other boundary conditions currently in use (such as setting the Laplacian to zero at the boundaries). This is especially true in simulations of coherent structures which exhibit vorticity (such as vortices and vortex rings) and in simulations of solutions which have a co-moving background velocity associated with them. In the latter case, the MSD boundary condition seems to be the *only* simple boundary condition which can handle such co-moving back-flows. We conclude that the MSD boundary condition is as effective as the standard Dirichlet boundary condition for problems with a constant value at the boundary, and nearly as easy to implement.

Appendix A Sample Implementation of the MSD Boundary Condition

In order to demonstrate the simplicity of implementing the MSD boundary condition, we include here the MATLAB implementation of the MSD boundary condition in one-dimension. We assume a time-dependent complex-valued partial differential equation in the form

$$U_t = F(U),$$

where $F(U)$ can contain spatial derivatives, nonlinearities, etc. Given a MATLAB function to compute $F(U)$, an initial condition U , storage vector Ut , and using a simple first-order time stepping scheme with time-step k , we arrive at the following code:

```
for(t=0:k:endtime)
    %Store F(U) into a vector. Here, F(U) sets boundary values to dummy values:
    Ut = F(U);
    %Apply MSD boundary condition to Ut:
    Ut(1) = 1i*imag(Ut(2)/U(2))*U(1);
    Ut(end) = 1i*imag(Ut(end-1)/U(end-1))*U(end);
    %Take step in scheme:
    U = k*Ut + U;
end
```

Acknowledgments

This research was supported by NSF-DMS-0806762 and the Computational Science Research Center at San Diego State University.

References

- [1] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and applied Mathematics, Philadelphia, PA, 2 edition, 2004.
- [2] B. Fornberg. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, 1996.
- [3] L. Debnath. *Nonlinear Partial Differential Equations for Scientists and Engineers*. Birkhauser Boston, New York, New York, 2 edition, 2005.
- [4] B. Malomed. Variational methods in nonlinear fiber-optics and related fields. *Progress in Optics*, **43** (2002) 71–193.
- [5] P. G. Kevrekidis, D. J. Frantzeskakis and R. Carretero-González. *Emergent Nonlinear Phenomena in Bose-Einstein Condensates: Theory and Experiment*, volume 45. Springer Series on Atomic, Optical, and Plasma Physics, 2008.
- [6] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt and A. Schädle. A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations. *Comm. in Comp. Phys.*, **4**, 4 (2008) 729–796.

- [7] G. H. Golub and J. M. Ortega. *Scientific Computing and Differential Equations An Introduction to Numerical Methods*. Academic Press, San Diego, California, 2 edition, 1992.
- [8] R. M. Caplan. NLSEmagic: Nonlinear Schrödinger equation multidimensional Matlab-based GPU-accelerated integrators using compact high-order schemes. <http://www.nlsemagic.com>, (2011).
- [9] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. SIAM (Fundamentals of Algorithms), 2003.
- [10] D.J. Frantzeskakis R. Carretero-González and P.G. Kevrekidis. Nonlinear waves in Bose-Einstein condensates: physical relevance and mathematical techniques. *Nonlinearity*, **21** (2008) R139–R202.
- [11] Yuri S. Kivshar, Jason Christou, Vladimir Tikhonenko, Barry Luther-Davies and Len M. Pismen. Dynamics of optical vortex solitons. *Optics Communications*, **152**, 1-3 (1998) 198 – 206.
- [12] D.J. Frantzeskakis P.G. Kevrekidis, R. Carretero-González and I.G. Kevrekidis. Vortices in Bose-Einstein condensates: Some recent developments. *Mod. Phys. Lett. B*, **18**, 30 (2004) 1481–1505.
- [13] P. H. Roberts and J. Grant. Motions in a Bose condensate I. The structure of the large circular vortex. *J. Phys. A: Gen. Phys.*, **4** (1971) 55–72.
- [14] R. M. Caplan and R. Carretero-González. Numerical stability of explicit Runge-Kutta finite-difference schemes for the nonlinear Schrödinger equation. *arXiv:1107.4810*, (2011).

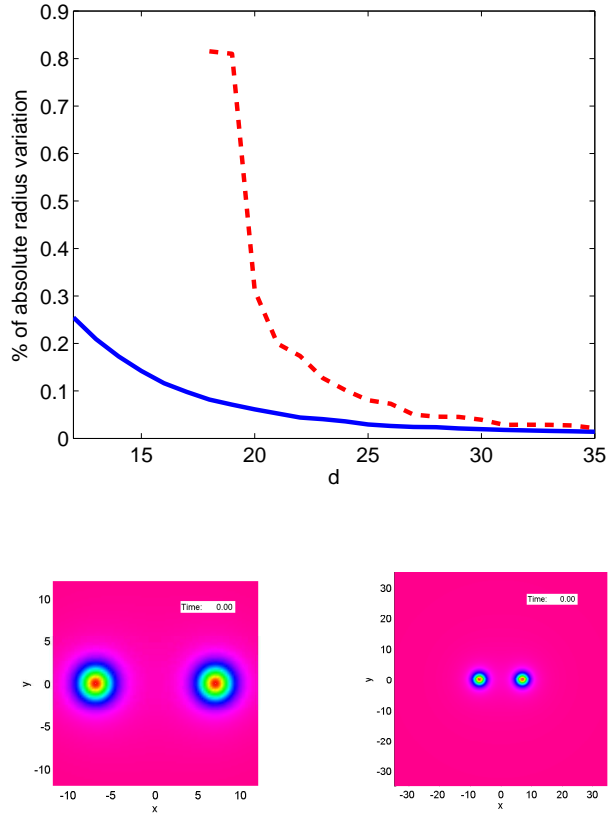


Figure 6: Top: Percentage of absolute radius variation when compared to the initial radius of the vortices to the center of the grid as a function of grid size d (defined as the distance from the center of the grid to its edge along the x or y direction) for a simulation with an end-time of $t = 480$ using both the L0 (dashed (red) line) and the MSD (solid (blue) line) boundary conditions. The grid size, d , is varied from 12 (the minimum size required to resolve both vortices adequately) to 35. Results for the L0 boundary condition below $d = 18$ are not shown as the vortices hit the grid wall before the simulation ends. The other simulation parameters are the same as those in Fig. 5. Bottom: Modulus-squared of the initial condition for grid-sizes $d = 12$ and $d = 35$.

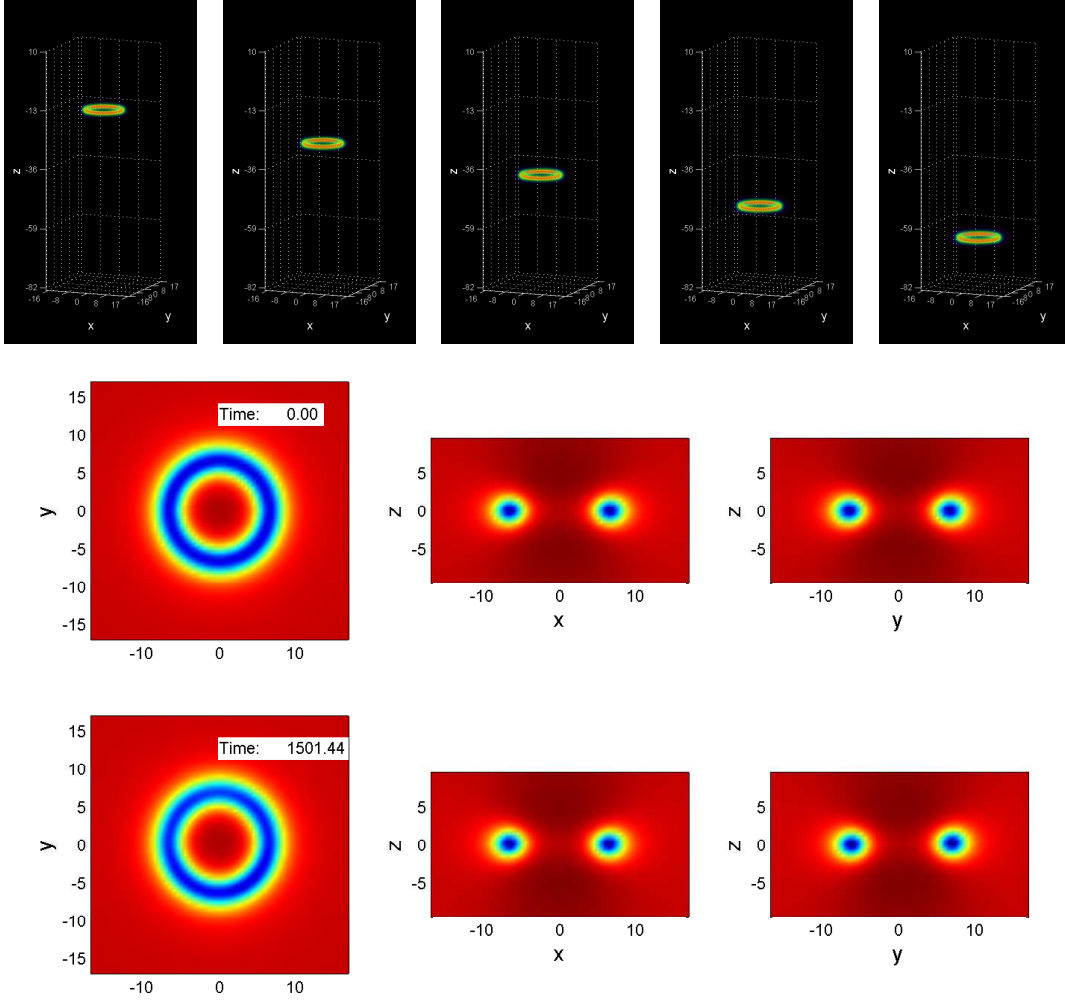


Figure 7: Top row: Snapshots of volumetric inverse renders of the modulus-squared of a vortex ring at equal time intervals from $t = 25$ to $t = 150$. Middle row: Two-dimensional cuts of the modulus-squared of the initial condition of a steady-state vortex ring amidst a back-flow. Bottom row: Two-dimensional cuts of the same vortex ring at simulation time $t = 1500$. All simulations use the MSD boundary condition, spatial-step size $h = 0.5$, and time-step $k = 0.035$.