Applied Mathematics and Computation 218 (2011) 3596–3614

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Boosting the accuracy of finite difference schemes via optimal time step selection and non-iterative defect correction

Kevin T. Chu

Serendipity Research, Mountain View, CA 94041, United States

ARTICLE INFO

Keywords:
Optimal time step
Non-iterative defect correction
Finite difference schemes
High-order accurate numerical methods
Time dependent PDEs

ABSTRACT

In this article, we present a unified analysis of the simple technique for boosting the order of accuracy of finite difference schemes for time dependent partial differential equations (PDEs) by optimally selecting the time step used to advance the numerical solution and adding defect correction terms in a non-iterative manner. The power of the technique, which is applicable to time dependent, semilinear, scalar PDEs where the leading-order spatial derivative has a constant coefficient, is its ability to increase the accuracy of formally low-order finite difference schemes without major modification to the basic numerical algorithm. Through straightforward numerical analysis arguments, we explain the origin of the boost in accuracy and estimate the computational cost of the resulting numerical method. We demonstrate the utility of optimal time step (OTS) selection combined with non-iterative defect correction (NIDC) on several different types of finite difference schemes for a wide array of classical linear and semilinear PDEs in one and more space dimensions on both regular and irregular domains.

1. Introduction

High-order numerical methods for partial differential equations (PDEs) will always be valuable for increasing the computational efficiency of numerical simulations. Thus, it is not at all surprising that a great deal of effort in numerical PDEs continues to be focused on the development of high-order numerical schemes [1–7]. Typically, high-order accuracy is achieved by constructing schemes that are formally high-order accurate. However, high-order accuracy can also be obtained by using formally low-order schemes in clever ways (*e.g.* Richardson extrapolation and compact finite difference schemes [1]). When possible, the latter approach can be a powerful way to boost the accuracy of a numerical method without introducing too much additional algorithmic (and programming) complexity.

One increasingly important approach for improving the accuracy of a numerical algorithm is optimization of its numerical parameters (*e.g.* Δx and Δt). For some problems, this optimization can dramatically increase the accuracy of the numerical solution. Two well-known examples are: (1) the unit Courant–Friedrichs–Lewy (CFL) condition, $\Delta t = \Delta x/a$, for the first-order upwind forward Euler discretization of the linear advection equation, which completely eliminates the numerical error introduced by the scheme [8] and (2) the boost from second- to fourth-order accuracy that results if the time step is set to $\Delta t = \Delta x^2/6D$ when solving the 1D diffusion equation using forward Euler time integration with the standard second-order central difference approximation for the Laplacian.

In this article, we present a unified analysis of the simple technique for boosting the *order* of accuracy of finite difference schemes for time dependent PDEs by optimally choosing the time step and, when necessary, adding defect correction terms in a *non-iterative* manner. This technique, which is applicable when the leading-order spatial derivative in a semilinear, scalar PDE has a constant coefficient, is a systematic generalization of the ideas underlying the previously mentioned

E-mail address: ktchu@serendipityresearch.org

0096-3003/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2011.08.108

examples and specific examples that are scattered throughout the literature. Optimal time step (OTS) selection is based on the observation that a carefully chosen time step can *simultaneously* eliminate low-order terms in both the spatial and temporal discretization errors. At its essence, OTS selection combines the powerful idea of optimally choosing numerical parameters with the approach of using knowledge of the structure of the PDE to reduce numerical error. The end result is a strategy for boosting the order of accuracy of algorithmically simple and computationally low-cost numerical schemes.

While OTS selection alone is sufficient to allow a formally low-order method to deliver high-order accuracy for some problems, defect correction terms are generally required to eliminate residual terms in the leading-order truncation error. However, unlike traditional defect correction methods [9–13], using OTS selection eliminates the need to incorporate correction terms that involve derivatives of higher-order than those that appear in the original PDE. Another significant feature of OTS selection with defect correction is that the defect terms are incorporated in a *non-iterative* fashion. In contrast, traditional defect/deferred correction approaches typically require several refinement iterations at each time step to boost the order of accuracy of the numerical solution. OTS with non-iterative defect correction (OTS–NIDC) also differs from the method of modified equations [14] for eliminating higher-order corrections to the truncation error because it uses an optimal time step, not explicit inclusion of correction terms, to regularize the singularly perturbed modified equations – the correction terms that remain when using OTS–NIDC are all *regular* perturbations to the original equation.

As a technique for enhancing numerical methods, OTS–NIDC has several desirable features. Like any technique that leads to high-order methods, OTS–NIDC produces schemes that greatly reduce the computational cost required to obtain a numerical solution of a desired accuracy. However, its real power comes from the fact that it achieves high-order accuracy while being based solely on very simple, formally low-order finite difference schemes *without* requiring special algorithmic steps. In other words, OTS–NIDC allows us to extract as much accuracy as possible from simple numerical schemes with minimal additional work. For instance, we will see how OTS–NIDC makes it possible to achieve fourth-order accuracy for the diffusion equation in any number of space dimensions using only a second-order stencil for the Laplacian and explicit time integration. Moreover, high-order accuracy is not restricted to problems on simple, rectangular domains; irregular domains are straightforward to handle by appropriately setting the values in ghost cells [3,4,15,16]. OTS–NIDC is even beneficial when using finite difference schemes to solve some nonlinear PDEs.

This article is organized as follows. In Section 2, we compare OTS–NIDC to related numerical techniques. In Section 3, we present the theoretical foundations of OTS–NIDC and show how it boosts the order of accuracy for formally low-order numerical schemes. In Section 4, we demonstrate the utility of OTS–NIDC to problems in one space dimension by applying it to finite difference schemes for several classical linear and nonlinear PDEs. In Section 5, we show that OTS–NIDC is useful for problems in multiple space dimensions by applying it to the 2D linear advection equation and the 2D diffusion equation. To illustrate the ease with which high-order accuracy can be achieved for problems on arbitrary domains, the 2D diffusion equation is solved on both regular and irregular domains. For each PDE in Sections 4 and 5, we provide the numerical analysis required to calculate the optimal time step size, the defect correction terms, and the order of accuracy. Note that in order to emphasize the method, we have chosen to apply it to relatively simple problems rather than use it to tackle "real-life" applications. The method, however, is not restricted to these simple examples. Finally, Section 6 offers some concluding remarks.

2. Theoretical comparisons with related numerical techniques

2.1. Comparison with defect/deferred correction methods

Defect/deferred correction is a well-known strategy for boosting the order of accuracy of finite difference schemes for time dependent PDEs. It usually takes one of two forms: (1) iterative refinement of the numerical solution [9–13] or (2) derivation of high-order compact schemes for the spatial derivative operator treating time derivatives as source terms [1,4,6].

For iterated defect/deferred correction, the numerical solution at each time step is computed by iteratively adding correction terms that are calculated from lower accuracy iterates. When used to improve the temporal order of accuracy, the iterative approach helps to ensure stability of the time integration scheme [13]. Although there are instances where iterative defect correction has been used to eliminate both spatial and temporal errors [11], it is usually used to improve only one of these errors [9,12,13]. Unlike iterated defect correction, OTS–NIDC requires no iteration during each time step because it computes correction terms using only the solution at the previous time step(s) (as opposed to requiring an estimate of the solution at the current time).

High-order compact schemes focus on eliminating spatial discretization errors when deriving the semi-discrete equations. When applied to time dependent PDEs, a high-order compact scheme for the spatial derivative operator is derived treating time derivatives as source terms [1,4,6]. After deriving the high-order spatial discretization, any time differencing scheme may be applied to discretize the time derivatives. While this approach is very general, it really only provides a mechanism for reducing spatial discretization error. Temporal discretization errors are completely controlled by the choice of time differencing scheme. Note that because high-order compact schemes for the spatial derivative operators involve spatial derivatives of the source term, the resulting schemes are always effectively implicit (even if all of the spatial derivative terms are treated explicitly).

OTS–NIDC has several advantages over both of these defect correction techniques. First, OTS–NIDC is generally more computationally efficient than both iterated defect correction and the high-order compact scheme approach because it requires less memory and, for many problems, admits simple explicit time integration schemes. Second, OTS–NIDC inherently controls both spatial and temporal numerical errors at the same time. Finally, taking advantage of relationships between spatial derivatives, temporal derivatives and source terms, OTS–NIDC avoids the need to explicitly incorporate finite differences of high-order derivatives of intermediate solutions in order to eliminate discretization errors. Compared with iterated defect correction, OTS–NIDC makes it easier to avoid instabilities that arise from the presence of finite difference discretizations of high-order temporal derivatives by automatically eliminating the most problematic correction terms. Comparing to the high-order compact scheme approach, the use of an optimal time step in OTS–NIDC means that mixed spatio-temporal partial derivatives do not need to appear in the fully discrete equations.

2.2. Comparison with the modified equation method

Both defect and deferred correction can be viewed as special cases of the method of modified equations [14]. When using any modified equation method, the basic approach is to derive a modified version of the original PDE through the introduction of the leading-order terms of the discretization error into the PDE. Unfortunately, the modified PDE is a singular perturbation to the original PDE due to the presence of the discretization parameters in the correction terms. Because the singular perturbation can lead to numerical instabilities, finding a way regularize the modified equation is an important step for developing any numerical method based on the modified equations method.

Whereas deferred correction methods deal with the singular perturbation by using an iterative procedure to avoid numerical instabilities, the traditional method of modified equations [14] handles the singularity by "pushing" the singularity to a higher-order term in the discretization error. The basic procedure involves substituting derivatives that appear in the singular term with expressions obtained by taking derivatives of the original PDE. This procedure yields a new modified equation where high-order derivatives of the solution are replaced by derivatives of source terms and the new singular term is of sufficiently high-order in the discretization parameter that it can be neglected in the construction of the numerical scheme.

While OTS–NIDC, like defect/deferred correction, falls within the general framework of the method of modified equations, it is an important special case because regularization of the singular perturbation in the modified equations is *automatic*. Regularization results from selecting the time step so that the spatial and temporal singular perturbation terms exactly cancel each other rather than through an iterative refinement procedure. By constraining the relationship between the time step and grid spacing, the modified equation becomes a regular perturbation to the original PDE. In the special case when OTS–NIDC does not require defect correction, the modified equation is actually exactly the same as the original equation – no regular perturbation terms remain after the singular perturbation terms have been eliminated.

There are several advantages of OTS–NIDC over traditional modified equation methods. First, because no finite difference approximations of higher-order derivatives are required, numerical schemes based on OTS–NIDC do not require additional boundary conditions. Second, derivation of the optimal time step and correction terms for OTS–NIDC requires a comparable amount of analysis as the traditional modified equation method. In both cases, we need explicit knowledge of the leading-order terms in the discretization error. Finally, because the leading-order terms in the discretization error are eliminated implicitly rather than explicitly as additional terms in the numerical scheme, the computational performance of OTS–NIDC is often superior to the performance of traditional modified equation methods.

2.3. Comparison with adaptive time stepping methods

OTS–NIDC is distinct and separate from traditional adaptive time stepping techniques [17,18] that are used in the context of the method of lines [17,19]. While both methods reduce numerical error by controlling the time step, the numerical errors they control are fundamentally different. Adaptive time stepping can only be used to reduce temporal discretization errors because it controls the errors that arise when solving the coupled system of ODEs for the semi-discrete approximation to the PDE; the spatial discretization errors are completely controlled by the finite difference scheme selected to approximate the spatial derivatives. In contrast, optimal time step selection simultaneously reduces *both* the spatial and temporal discretization errors because it uses information from the PDE to choose the time step. Another important distinction between the two techniques is that optimal time step selection uses a *fixed* time step which makes it significantly less computationally complex than adaptive time stepping methods.

3. Theoretical foundations of OTS–NIDC

Technically, OTS–NIDC is not by itself a method for constructing finite difference schemes. Rather, it is a way to enhance to the performance of existing finite difference schemes by carefully choosing the time step to eliminate leading-order terms in the discretization error. The two key ideas underlying OTS–NIDC are (1) a judicious choice of time step can be used to eliminate the leading-order (and possibly higher-order) terms in the error and (2) the PDE provides valuable insight into the relationships between terms in the discretization error. In this section, we present and analyze the technique of OTS–NIDC for finite difference schemes constructed to solve scalar evolution equations of the form

$$\frac{\partial u}{\partial t} = F(x, u(x), Du(x), D^2u(x), \ldots), \tag{1}$$

where the right hand side is an arbitrary function of the solution u, its spatial derivatives, and the independent variable x. Because this class of time dependent PDEs is so prevalent and important, restricting our attention to problems of this form is not a serious limitation. The same principles apply to finite difference schemes for general PDEs.

3.1. Optimal time step selection for a model PDE

Let us begin by considering a finite difference approximation for the following time dependent PDE in one spatial dimension:

$$\frac{\partial u}{\partial t} = A\frac{\partial^n u}{\partial x^n}, \tag{2}$$

where A is a constant of the appropriate sign so that (2) is well-posed. Because we will be boosting the accuracy of the finite difference approximation by optimizing the time step, there is no reason to explicitly construct the scheme to be high-order. In fact, using a high-order finite difference scheme only complicates the analysis required to calculate the optimal time step.

To compute the optimal time step, we first analyze the discretization errors for the finite difference approximation, explicitly keeping track of the leading-order terms. Because each term in the error is proportional to a partial derivative of the solution, we can use the PDE to relate the leading-order terms to each other. For many finite difference schemes, this careful analysis yields a direct relationship between the leading-order spatial and temporal errors that allows them to be combined as a single term of the form:

$$(Lu)P(\Delta x, \Delta t), \tag{3}$$

where L is a differential operator and P is a simple function of the numerical parameters Δx and Δt. By selecting the time step so that $P = 0$, we can completely eliminate the leading-order term in the discretization error. The time step selected in this way is the *optimal time step* for the finite difference scheme. Using the optimal time step for time stepping yields a numerical solution with an *effective* order of accuracy that is higher than the formal order of the original finite difference scheme.

An important class of finite difference schemes for time dependent PDEs are those based on the method of lines. When the method of lines [17,19–21] approach is used for (2), P often takes a particularly simple form:

$$P(\Delta x, \Delta t) = (\alpha\Delta x^r - \beta\Delta t^s)\Delta t, \tag{4}$$

where r and $(s + 1)$ are the orders of the leading spatial and temporal discretization errors, and α and β are constants that depend on the details of the PDE and finite difference scheme. Setting $P = 0$, we find that the optimal time step is given by[1]

$$\Delta t_{opt} = (\alpha/\beta)^{1/s}\Delta x^{r/s}. \tag{5}$$

An important assumption in the above analysis is that α and β have the same sign. If this condition is not satisfied, (*e.g.* finite difference scheme for the diffusion equation based on the standard central difference stencil for the Laplacian and backward Euler time integration), then OTS selection is not applicable because none of the zeros of (4) is positive. Fortunately, in this situation, it is usually possible to find a closely related finite difference scheme that has a positive optimal time step.

3.1.1. Error analysis

The form of the local error in (4) follows from the fact that the method of lines first approximates the PDE as a system of ODEs in time by only discretizing the spatial derivatives:

$$\mathbf{u}_t = F(\mathbf{u}), \tag{6}$$

where $\mathbf{u}$ is the vector of values of u at the grid points and $F(\mathbf{u})$ is the finite difference approximation of the spatial derivative operator acting on $\mathbf{u}$. The second term in the local error (4) arises directly from temporal discretization errors associated with the numerical scheme used to integrate the system of ODEs in time. The first term arises because F is only an approximation of the spatial derivative operators in the PDE. As a result, the system of ODEs (6) has an error due to the spatial discretization at *every* point in time. For a spatial discretization error of order r, the error in (6) is $O(\Delta x^r)$, so the error accumulated during each time step due to spatial discretization error is $O(\Delta t\Delta x^r)$. Because the method of lines is so frequently used to construct finite difference approximations for time dependent PDEs, we will focus primarily on finite difference schemes of this type for the remainder of the article. It is straightforward to extend the results we obtain to more general finite difference schemes (*e.g.* the Lax–Wendroff scheme for the linear advection equation).

To analyze the accuracy of the finite difference scheme for (2) when the optimal time step is used, we examine the higher-order terms in the discretization error. Elimination of the leading terms in the discretization error leaves a local error of the form $O(\Delta t\Delta x^p) + O(\Delta t^{q+1})$, where $p > r$ and $(q + 1) > (s + 1)$ are the orders of the spatial and temporal discretization errors, respectively, after the leading-order terms in the error have been eliminated. From the local error, we can compute the global

[1] The $\Delta t = 0$ root of P is discarded because it is not numerically meaningful.

Table 1
Computational cost of various finite difference schemes for the 1D diffusion equation as a function of the global numerical error e. For all of the schemes, the standard second-order central difference stencil is used to discretize the Laplacian. Note that the time step for the backward Euler scheme is chosen so that the temporal error is subdominant to the spatial error and that the computation time for the Crank–Nicholson method assumes an $O(N) = O(1/\Delta x)$ matrix inversion algorithm.

Numerical scheme	Δx	Δt	Memory	Compute time
Forward Euler	$O(e^{1/2})$	$O(e)$	$O(e^{-1/2})$	$O(e^{-3/2})$
Backward Euler	$O(e^{1/2})$	$O(e)$	$O(e^{-1/2})$	$O(e^{-3/2})$
Crank–Nicholson	$O(e^{1/2})$	$O(e^{1/2})$	$O(e^{-1/2})$	$O(e^{-1})$
Forward Euler with OTS–NIDC	$O(e^{1/4})$	$O(e^{1/2})$	$O(e^{-1/4})$	$O(e^{-3/4})$

Table 2
Computational cost of the forward Euler scheme with OTS–NIDC and the Crank–Nicholson method for solving the diffusion equation as a function of the global numerical error e. Note that the computation time for the Crank–Nicholson method assumes an $O(N) = O(1/\Delta x^d)$ matrix inversion algorithm. A comparison of computational costs in higher spatial dimensions continues along the same trends but requires replacement of the conventional forward Euler time integration scheme with an alternative for which the optimal time step is stable (*e.g.* the DuFort–Frankel scheme).

Dimensions	Forward Euler with OTS–NIDC		Crank–Nicholson	
	Memory	Compute time	Memory	Compute time
1	$O(e^{-1/4})$	$O(e^{-3/4})$	$O(e^{-1/2})$	$O(e^{-1})$
2	$O(e^{-1/2})$	$O(e^{-1})$	$O(e^{-1})$	$O(e^{-3/2})$
3	$O(e^{-3/4})$	$O(e^{-5/4})$	$O(e^{-3/2})$	$O(e^{-2})$

error of the numerical solution over extended intervals of time by using the heuristic argument that the global error is equal to the local error divided by Δt [19]:

$$e = O(\Delta x^p) + O(\Delta t^q). \tag{7}$$

Because $p > r$ and $q > s$, we see that using an optimal time step boosts the overall accuracy of the finite difference scheme.

It is important to emphasize that Δx and Δt are *not* independent in (7) because the time step must be set to Δt_{opt} in order to derive the error estimate. In other words, there is really only one parameter, Δx, that controls the numerical accuracy of the finite difference scheme. Since $\Delta t_{opt} = O(\Delta x^{r/s})$, the global error (7) reduces to

$$e = O(\Delta x^{\min(p,rq/s)}). \tag{8}$$

3.1.2. Stability considerations

We have not yet considered the issue of numerical stability of finite difference schemes when the time step is set to Δt_{opt}. For explicit time integration schemes, OTS–NIDC is ineffective when the optimal time step is greater than the smallest stable time step. In this situation, it is necessary to replace the finite difference approximation of the PDE. For example, if the diffusion equation is solved using a finite difference scheme based on forward Euler time integration and a discretization of the Laplacian with an isotropic leading-order truncation error (see Section 5.3), the optimal time step becomes unstable when the dimension of space is greater than 3. To make OTS–NIDC useful for diffusion problems in four and more space dimensions, we can switch to the DuFort–Frankel scheme, which is explicit and unconditionally stable [19].

3.1.3. Computational performance

For many problems, the optimal time step is on the order of the maximum time step allowed by the stability constraint for an explicit method. As a result, using the optimal time step to advance the finite difference scheme in time may appear to be overly restrictive. However, this apparent drawback is more than compensated by the boost in the order of accuracy. For example, when solving the diffusion equation using a forward Euler time integration scheme with a second-order central difference stencil for the Laplacian, we obtain a fourth-order scheme when Δt is set equal to the optimal time step (discussed further in Sections 4.3 and 5.3). As Table 1 shows, this boost in accuracy leads to a scheme that is asymptotically computationally cheaper than traditional implicit schemes that use the same low-order finite difference stencils. The performance gain for problems in higher spatial dimensions is even more impressive. As we can see in Table 2, the gap between the performance of a simple forward Euler scheme with the optimal time step and the Crank–Nicholson method grows with the spatial dimension of the problem.

3.2. OTS–NIDC for general PDEs in one space dimension

In the previous section, we analyzed optimal time step selection in the context of a simple model PDE in one space dimension. For more general PDEs, OTS selection continues to be useful, but it needs to be augmented with defect correction (incorporated in a non-iterative manner) to achieve the same improvement in the order of accuracy. In this section, we discuss OTS–NIDC in the context of general constant coefficient, linear PDEs and semilinear PDEs.

3.2.1. General constant coefficient, linear PDEs

Let us consider a general constant coefficient, linear PDE of the form

$$\frac{\partial u}{\partial t} = \sum_{k=1}^{n} A_k \frac{\partial^k u}{\partial x^k} + f(x,t). \tag{9}$$

Even if we use the PDE to relate terms in the discretization error, it is unlikely that all of the dominant terms in the error can be combined into a single term. As a result, the leading-order discretization error is generally a sum of several terms:

$$\Delta t\left[(Lu)(\alpha\Delta x^r - \beta\Delta t^s) + (L'u)\Delta x^r + (L''u)\Delta t^s + (Gf)\Delta t^s\right], \tag{10}$$

where L, u, α, β, r, and s are defined as in (3) and (4), L' and L'' are the linear spatial differential operators associated with the leading-order terms in the discretization error that are not involved in the choice of the optimal time step, and G is a spatio-temporal differential operator that acts on the source term f. Note that neither L' nor L'' involve any temporal differential operators because any that arise can be eliminated using the PDE (9).

From (10), it is clear that setting $\Delta t = \Delta t_{opt}$ is not enough to completely eliminate the leading-order error. To eliminate the vestigial errors, we need to add defect correction terms to cancel out the second, third, and fourth terms in (10). To advance the solution to the next time step, the second and third terms in (10) can be approximated using finite differences applied to the solution u at the current time. The last term can be computed using either a finite difference approximation or a direct analytic calculation. With the addition of these correction terms, the global error can be determined using the analysis in Section 3.1.1.

Some care must be exercised when choosing the finite difference stencils for the correction terms. While the order of accuracy will always be boosted, the finite difference stencils used for the correction terms can introduce errors into the numerical solution that will cause the order of accuracy to fall below the estimate given by (8).

3.2.2. Semilinear PDEs

OTS–NIDC can also be applied to semilinear PDEs of the form

$$\frac{\partial u}{\partial t} = A\frac{\partial^n u}{\partial x^n} + F\left(\frac{\partial^{n-1} u}{\partial x^{n-1}}, \frac{\partial^{n-2} u}{\partial x^{n-2}}, \ldots, \frac{\partial u}{\partial x}, u\right) + f(x,t), \tag{11}$$

where F is an arbitrary function of the lower-order spatial derivatives of u and $f(x,t)$ is a source term. While the coefficient on the leading-order spatial derivative for a semilinear PDE is generally allowed to be a function of space and time, optimal time step selection can only be applied when it is a constant. Because the PDE is linear in the leading-order spatial derivative, it is natural to construct a finite difference scheme that treats F in an explicit manner and lets stability considerations determine whether the $A\frac{\partial^n u}{\partial x^n}$ term is handled explicitly or implicitly. Assuming that sufficiently high-order stencils are used to compute F, the leading-order discretization error for such a finite difference scheme is given by

$$\Delta t\left[(Lu)(\alpha\Delta x^r - \beta\Delta t^s) + (Hu)\Delta x^r + (H'u)\Delta t^s + (Gf)\Delta t^s\right], \tag{12}$$

where L, u, α, β, r, and s are defined as in (3) and (4), H and H' are spatial differential operators (potentially nonlinear) associated with the leading-order terms in the discretization error that arise from the nonlinear term in (11), and G is a spatio-temporal differential operator that acts on f. With the discretization error written in the form (12), the optimal time step and defect correction terms can be derived by proceeding in the same manner as in the previous section.

3.3. Boundary conditions

In general, high-order discretizations of boundary conditions are required to ensure that errors introduced at the domain boundary do not pollute the numerical solution in the domain interior. For Dirichlet boundaries, it is sufficient to impose boundary conditions by using boundary values that have the same order of accuracy as the numerical scheme. More effort is required to ensure that Neumann boundary conditions do not reduce the order of accuracy of the solution. In general, Neumann boundary conditions require that the accuracy of ghost cells slightly exceeds the desired order of accuracy for the solution. A detailed discussion of methods for imposing boundary conditions is beyond the scope of this article. We refer interested readers to the thorough analysis presented in [19]. To avoid complicating the discussion in Sections 4 and 5, Neumann boundary conditions for all examples are imposed by taking advantage of knowledge of the analytical solution.

3.4. OTS–NIDC selection in multiple space dimensions

OTS–NIDC is applicable to PDEs in any number of space dimensions. While problems in higher dimensions generally require a more careful choice of the finite difference stencil used to approximate spatial derivatives, there are no fundamental barriers that prevent the use of OTS–NIDC. Moreover, by using a ‘ghost cell’ or ‘ghost value’ method for imposing boundary conditions [3,4,15,16], optimal time step selection can even be directly applied to problems on irregular domains.

One important issue that arises in multiple space dimensions is the choice of grid spacing. In general, the grid spacing need not be equal in all spatial directions. For some problems, the ratio of the grid spacing in different coordinate directions may need to be carefully chosen for an optimal time step to exist. In other words, it may be necessary to optimally choose both the time step *and* the mesh in order to boost the order of accuracy.

4. Application to PDEs in one space dimension

Because there are no geometry or anisotropy considerations, OTS–NIDC is very useful for boosting the accuracy of many finite difference schemes in one spatial dimension. In this section, we demonstrate its utility in designing finite difference schemes that solve several classical linear and semilinear PDEs. To emphasize the value of including numerical parameter optimization as part of the design of any finite difference scheme, we include applications of OTS–NIDC to several different types of finite difference schemes. The examples in this section were selected because they have analytical solutions that are useful to compare numerical solutions against. However, the utility of OTS–NIDC is by no means limited to these simple PDEs.

4.1. Unit CFL conditions for wave equations

In this section, we show how OTS selection naturally leads to unit CFL conditions for the linear advection equation and second-order wave equation in one space dimension.

4.1.1. Linear advection equation

When the linear advection equation, $u_t + Au_x = 0$, is solved using a first-order upwind scheme with forward Euler time integration, it is well-known that choosing $\Delta t = \Delta x/|A|$ completely eliminates all numerical error in the solution (except for errors in the initial conditions). This result is known as the unit CFL condition [8] and is usually proved by examining how characteristic lines of the advection equation pass through grid points. Interestingly, the same optimal choice of time step arises by applying OTS selection.

Using OTS selection, we begin by analyzing the discretization error for the first-order upwind, forward Euler scheme:

$$u_j^{n+1} = u_j^n - A\Delta t\left(\frac{u_j^n - u_{j-1}^n}{\Delta x}\right), \tag{13}$$

where we have taken A to be positive for convenience. When applying OTS selection, we generally only need to keep track of the leading-order terms in the discretization error. However, for the advection equation, it is valuable to retain all of them. Using Taylor series expansions of the true solution of the PDE, it is straightforward to show that the evolution equation for the error $e \equiv u - \tilde{u}$ is

$$e_j^{n+1} = e_j^n - A\Delta t\left(\frac{e_j^n - e_{j-1}^n}{\Delta x}\right) + A\Delta t\sum_{k=2}^{\infty}\frac{(-1)^k}{k!}\frac{\partial^k \tilde{u}^n}{\partial x^k}\left(\Delta x^{k-1} - (A\Delta t)^{k-1}\right), \tag{14}$$

where $\tilde{u}$ denotes the true solution of the PDE, all derivatives are evaluated at (t_n, x_j), and we have used the PDE to replace all of time derivatives with spatial derivatives.

From (14), we see that choosing the time step to eliminate the leading-order term of the discretization error leads to the unit CFL condition and *complete elimination* of the local truncation error at each time step. It is interesting to note that OTS selection applied to the Lax–Wendroff scheme [8,22] yields the same optimal time step and the same total elimination of the truncation error.

4.2. Second-order wave equation

Numerical methods for the wave equation $u_{tt} - c^2u_{xx} = f$ are typically derived by transforming the second-order equation into an equivalent first-order system of equations. However, as Kreiss, Petersson, and Yström showed in [23], direct discretization of the second-order wave equation is a viable option and leads to the following finite difference scheme that is second-order accurate in both space and time:

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} = c^2\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}\right) + f. \tag{15}$$

The stability constraint for this scheme of the form $\Delta t = K\Delta x$, where K is a constant.

To compute the optimal time step and correction terms for this scheme, we begin by deriving the truncation error for the scheme. Employing Taylor series expansions and the observation that the true solution satisfies $u_{tttt} = c^4u_{xxxx} + c^2f_{xx} + f_{tt}$, we find that the error equation for (15) is given by

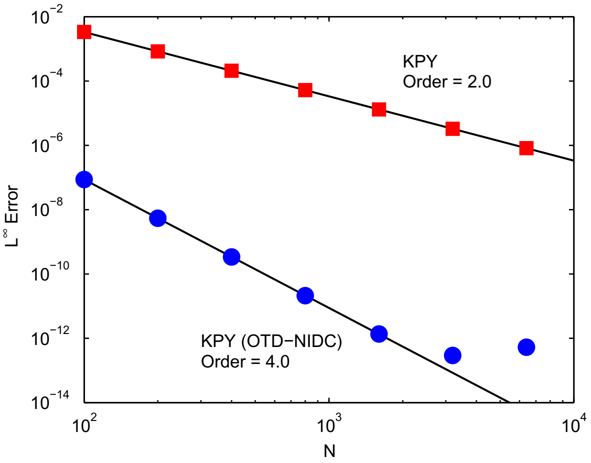

Fig. 1. L^∞ error as a function of number of grid points for the KPY discretization of the second-order wave equation with OTS–NIDC (blue circles) and without OTS–NIDC (red squares). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$e_i^{n+1} = 2e_i^n - e_i^{n-1} + \Delta t^2 c^2 \left(\frac{e_{i+1}^n - 2e_i^n + e_{i-1}^n}{\Delta x^2} \right) + \frac{\Delta t^2}{12} c^2 \frac{\partial^4 \tilde{u}}{\partial x^4} (\Delta x^2 - \Delta t^2 c^2) - \frac{\Delta t^4}{12} \left(c^2 \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial t^2} \right) + O(\Delta t^2 \Delta x^4) + O(\Delta t^6). \tag{16}$$

Choosing $\Delta t = \Delta x/c$ and adding the correction term $\Delta t^2(c^2 f_{xx} + f_{tt})/12$ to the right-hand side of (15) eliminates the leading-order term in the local truncation error. Using the heuristic for two-step methods that the global error should be approximately $1/\Delta t^2$ times the local error leads to a global error of $O(\Delta x^4) = O(\Delta t^4)$–a boost in the order of accuracy from second- to fourth-order. Notice that as for first-order upwind, forward Euler and Lax–Wendroff schemes for the linear advection equation, applying OTS selection to the KPY scheme leads to an optimal time step that satisfies the unit CFL condition. However, unlike the schemes for the linear advection equation, the unit CFL condition for the KPY scheme eliminates only the leading-order term in the truncation error. Fig. 1 demonstrates that the expected accuracy is indeed achieved by applying OTS–NIDC to the KPY scheme.

An important property of the KPY scheme is that the error introduced during the first time step affects the global error at all times. More specifically, a $(p+1)$-order accurate first time introduces an $O(\Delta x^p)$ error into the solution at all future times. Thus, we must ensure that the error introduced by the first time step is at least one order of accuracy higher than desired accuracy for the numerical solution[2].

Constructing a fifth-order approximation for the first time step is straightforward using a fifth-order Taylor series expansion in time:

$$\begin{aligned} u^1 &= u^0 + \Delta t \frac{\partial u^0}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u^0}{\partial t^2} + \frac{\Delta t^3}{6} \frac{\partial^3 u^0}{\partial t^3} + \frac{\Delta t^4}{24} \frac{\partial^4 u^0}{\partial t^4} \\ &= u^0 + \Delta t \frac{\partial u^0}{\partial t} + \frac{\Delta t^2}{2} \left(c^2 \frac{\partial^2 u^0}{\partial x^2} + f \right) + \frac{\Delta t^3}{6} \left(c^2 \frac{\partial^2}{\partial x^2} \left(\frac{\partial u^0}{\partial t} \right) + \frac{\partial f}{\partial t} \right) + \frac{\Delta t^4}{24} \left(c^4 \frac{\partial^4 u^0}{\partial x^4} + c^2 \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial t^2} \right). \end{aligned} \tag{17}$$

Note that special care must be exercised if finite differences are used to compute the derivatives in (17) because higher-order terms may be implicitly included by lower-order terms in the expansion if the initial time step is taken using Δt_{opt}.

4.3. Diffusion equation

Solution of the diffusion equation, $u_t = D\, u_{xx}$, using forward Euler time integration with a second-order central difference approximation for the Laplacian provides a familiar example where careful choice of the time step leads to cancellation of the leading order spatial and temporal truncation errors. It is well-known that choosing the time step to be $\Delta t_{opt} = \Delta x^2/6D$ boosts the order of accuracy for this simple scheme from $O(\Delta x^2)$ to $O(\Delta x^4)$.

When a source term is present, boosting the order of accuracy requires the addition of defect correction terms. These correction terms are straightforward to calculate by examining the discretization error for the scheme

$$u_j^{n+1} = u_j^n + \Delta t \left(D \left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \right] + f_j^n \right), \tag{18}$$

[2] This requirement follows from a straightforward analysis of the second-order difference equation for the normal modes of the error.

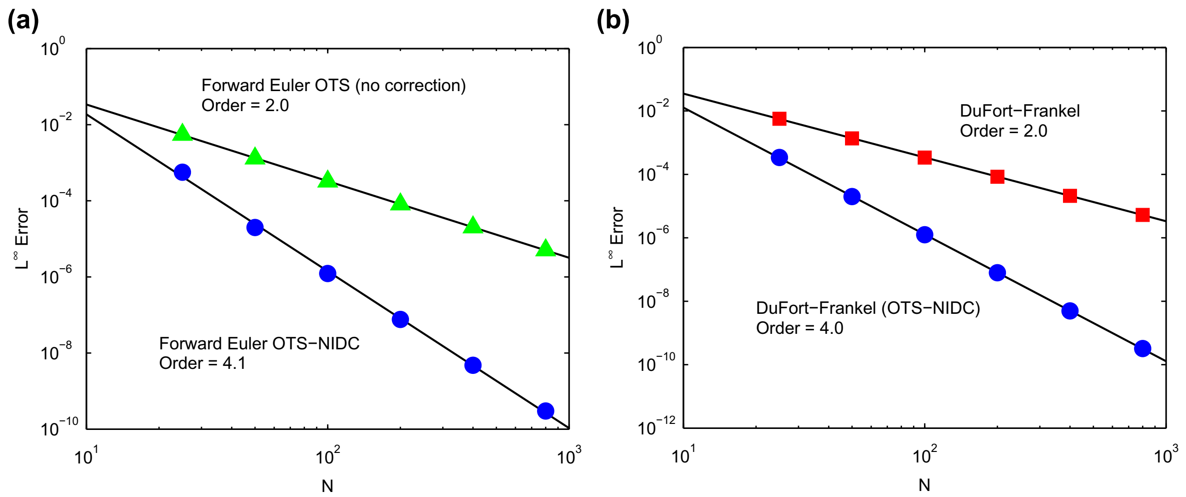

Fig. 2. (a) L^∞ error as a function of number of grid points for two finite difference schemes that solve the diffusion equation: forward Euler with optimal time step and defect correction term (circles) and forward Euler with optimal time step *without* defect correction term (triangles). (b) L^∞ error in the numerical solution of the 1D diffusion equation as a function of number of grid points for the DuFort–Frankel scheme with (circles) and without (squares) OTS–NIDC.

where f is the source term. Note that even though this scheme is formally first-order in time and second-order in space, the stability constraint $\Delta t \leqslant \Delta x^2/2D$ implies that the scheme is $O(\Delta x^2)$ accurate overall (*i.e.* inclusion of the temporal order of accuracy is irrelevant). Using Taylor series expansions and the PDE, $u_t = D\,u_{xx} + f$, we find that the truncation error is given by

$$\frac{\partial^4 \tilde{u}^n}{\partial x^4}\left[\frac{\Delta x^2}{12} - \frac{D\Delta t}{2}\right](D\Delta t) - \frac{\Delta t^2}{2}\left(D\frac{\partial^2 f}{\partial x^2} + \frac{\partial f}{\partial t}\right) + O(\Delta t \Delta x^4) + O(\Delta t^3). \tag{19}$$

From this expression, we easily recover the well-known optimal time step for the diffusion equation, $\Delta t_{opt} = \Delta x^2/6D$, and identify the correction term $\Delta t^2(Df_{xx} + f_t)/2$.

From (8), we see that using the optimal time step and adding this simple correction term to (18) boosts the accuracy of original scheme from second to fourth order. Note that to achieve fourth order accuracy, the spatial derivatives in the correction term must be calculated using a finite difference approximation that is at least second-order accurate; otherwise, the truncation error will be $O(\Delta t^2 \Delta x)$ which implies an $O(\Delta t \Delta x) = O(\Delta x^3)$ global error. The importance of the correction term is illustrated in Fig. 2(a), which compares the accuracy of the OTS forward Euler scheme with and without the correction term. As we can see, OTS forward Euler with correction term is fourth-order accurate while OTS forward Euler without correction term is only second-order accurate.

4.3.1. DuFort–Frankel scheme

As mentioned earlier, the forward Euler scheme for the diffusion equation is unstable for problems in more than 3 space dimensions. Fortunately, the OTS–NIDC makes it possible to construct a fourth-order accurate DuFort–Frankel scheme.

We begin by writing the DuFort–Frankel scheme in a form that is easily generalizable to higher space dimensions:

$$u_j^{n+1} = u_j^{n-1} + 2\Delta t\left(D\left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}\right] + f_j^n\right) - 2D\frac{\Delta t^3}{\Delta x^2}\left(\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\Delta t^2}\right). \tag{20}$$

Note that the DuFort–Frankel scheme is essentially a leap-frog time integration scheme that is stabilized by the addition of a term proportional to u_{tt} that vanishes in the limit $\Delta t \to 0$. This scheme is unconditionally stable and has a global error that is formally $O(\Delta t^2/\Delta x^2) + O(\Delta t^2) + O(\Delta x^2) + O(\Delta t^4/\Delta x^2)$ [19]. This expression for the formal error is "minimized" when $\Delta t = O(\Delta x^2)$, which yields an $O(\Delta x^2)$ global error.

Using Taylor series expansions centered at (t_n, x_j) and the PDE, we derive the local truncation error for the DuFort–Frankel scheme:

$$2D\Delta t\frac{\partial^4 \tilde{u}}{\partial x^4}\left(\frac{\Delta x^2}{12} - \frac{D^2\Delta t^2}{\Delta x^2}\right) - 2D\frac{\Delta t^3}{\Delta x^2}\left(D\frac{\partial^2 f}{\partial x^2} + \frac{\partial f}{\partial t}\right) + O(\Delta t^3) + O(\Delta t \Delta x^4) + O\left(\frac{\Delta t^5}{\Delta x^2}\right). \tag{21}$$

Therefore the optimal time step is $\Delta t_{opt} = \Delta x^2/D\sqrt{12}$ and the defect correction term is $2D\Delta t^3\,(Df_{xx} + f_t)/\Delta x^2$. As with the forward Euler scheme for the diffusion equation, OTS–NIDC boosts the order of accuracy for the DuFort–Frankel scheme from two to four. Fig. 2(b) shows that OTS–NIDC applied to the DuFort–Frankel scheme achieves the expected boost in accuracy.

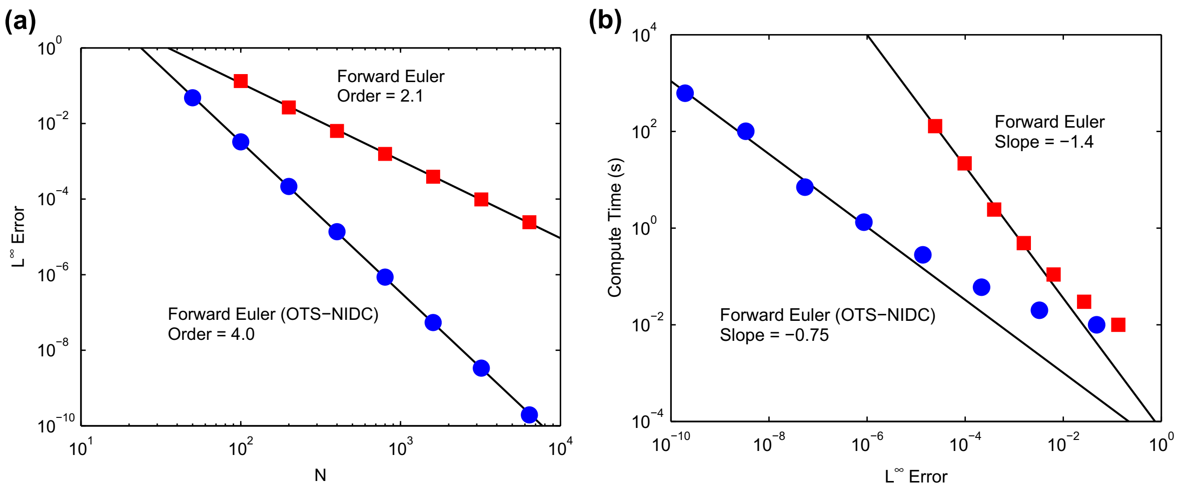

Fig. 3. (a) L^∞ error as a function of number of grid points for two finite difference schemes that solve the viscous Burgers equation: forward Euler with OTS–NIDC (circles) and forward Euler with suboptimal time step $\Delta t = \Delta x^2/4\nu$ and no correction term (squares). (b) Computation time as a function of L^∞ error. These results were obtained using MATLAB implementations of the finite difference schemes run on a dual-core 2.4 GHz MacBook Pro.

4.4. Viscous Burgers equation

We now demonstrate the use of OTS–NIDC in the context of a well-studied semilinear PDE – the viscous Burgers equation [24]:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \tag{22}$$

where ν is the viscosity. For this equation, let us consider the finite difference scheme constructed using a forward Euler time integration scheme and second-order, central difference approximations for the diffusion and nonlinear advection terms:

$$u_j^{n+1} = u_j^n + \Delta t\left(\nu\left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}\right] - u_j^n\left[\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}\right]\right). \tag{23}$$

This scheme is formally first-order in time and second-order in space. Using the stability condition for the advection–diffusion as a heuristic guide[3], we expect a stability constraint of the form $\Delta t \leqslant \Delta x^2/2\nu$ in the limit $\Delta x \to 0$ (assuming that the solution u remains bounded). Therefore, the scheme (23) is $O(\Delta x^2)$ accurate overall.

Using the usual OTS–NIDC analysis procedure, we obtain the local truncation error for (23):

$$\left(\nu\frac{\partial^4\tilde{u}^n}{\partial x^4} - 2\tilde{u}_j^n\frac{\partial^3\tilde{u}^n}{\partial x^3}\right)\left[\frac{\Delta x^2}{12} - \frac{\nu\Delta t}{2}\right]\Delta t + \frac{\Delta t^2}{2}\left(4\nu\frac{\partial\tilde{u}^n}{\partial x}\frac{\partial^2\tilde{u}^n}{\partial x^2} - 2\tilde{u}_j^n\left(\frac{\partial^2\tilde{u}^n}{\partial x^2}\right)^2 - \left(\tilde{u}_j^n\right)^2\frac{\partial^2\tilde{u}^n}{\partial x^2}\right) + O(\Delta t^3) + O(\Delta t\Delta x^4). \tag{24}$$

Therefore, the optimal time step is $\Delta t_{opt} = \Delta x^2/6\nu$ and the defect correction term is

$$-\frac{\Delta t^2}{2}\left(4\nu\frac{\partial\tilde{u}^n}{\partial x}\frac{\partial^2\tilde{u}^n}{\partial x^2} - 2\tilde{u}_j^n\left(\frac{\partial^2\tilde{u}^n}{\partial x^2}\right)^2 - \left(\tilde{u}_j^n\right)^2\frac{\partial^2\tilde{u}^n}{\partial x^2}\right) \tag{25}$$

Note that the optimal time step eliminates errors introduced by both the diffusion term *and* the nonlinear advection term so that the highest order spatial derivative in the defect correction terms only is no higher than the highest order spatial derivative in the original PDE.

By using the optimal time step and the correction term, the overall error for the finite difference scheme (23) is reduced from $O(\Delta x^2)$ to $O(\Delta x^4)$. As for the diffusion equation, we need to use a second-order accurate discretization for the spatial derivatives in the correction term. Fig. 3(a) demonstrates the boost in accuracy obtained by applying OTS–NIDC to the finite difference scheme (23).

The asymptotic computational performance of numerical schemes for the 1D viscous Burgers equation is the same as for the 1D diffusion equation. Fig. 3(b) shows that the theoretical estimates from Table 1 for the conventional forward Euler and OTS–NIDC forward Euler algorithms are in good agreement with timing data collected from computational experiments.

[3] The stability constraint for the advection–diffusion equation is $\Delta t < \min\{2D/A^2, \Delta x^2/2D\}$ [25].

Note that the data for the OTS–NIDC scheme bends upwards below an L^∞ error of about 10^{-5}, which indicates that the $O(1/\Delta x)$ work per time step does not dominate the computational work per time step until Δx is sufficiently small.

The Burgers equation example illustrates an important aspect of OTS–NIDC: the ease with which OTS–NIDC can be used to boost the order of accuracy depends on the choice of the finite difference scheme. Had we chosen an upwind approximation for the gradient term in the Burgers equation, the correction term would have been more complicated and included terms involving higher-order spatial derivatives than those that appear in the original equation. While it should be possible to handle this situation with OTS–NIDC selection, it is clearly less desirable.

4.5. Fourth-order parabolic equation

As a final example in one space dimension, we consider a PDE with high-order spatial derivatives – a fourth-order parabolic equation:

$$\frac{\partial u}{\partial t} = -\kappa \frac{\partial^4 u}{\partial x^4} + f(x,t), \tag{26}$$

where $\kappa > 0$. This example illustrates the importance of the choice of discretization for the spatial derivative in order for OTS–NIDC to be applicable.

From our previous examples, we know that optimal time step selection requires the coefficient on leading-order spatial discretization error to be directly related to the second derivative of the solution u with respect to time:

$$\frac{\partial^2 u}{\partial t^2} = \kappa^2 \frac{\partial^8 u}{\partial x^8} - \kappa \frac{\partial^4 f}{\partial x^4} + \frac{\partial f}{\partial t}. \tag{27}$$

Therefore, the leading-order term in the spatial discretization error should involve the eighth-derivative of u, which implies that the finite difference approximation to the bilaplacian needs to be at least fourth-order accurate. With this insight, let us consider a finite difference scheme for (26) that uses forward Euler for time integration and a fourth-order central difference approximation for the bilaplacian:

$$u_j^{n+1} = u_j^n + \Delta t\left(-\kappa B u^n + f_j^n\right), \tag{28}$$

where the fourth-order discrete bilaplacian is given by

$$Bu^n = \frac{-u_{j+3}^n + 12u_{j+2}^n - 39u_{j+1}^n + 56u_j^n - 39u_{j-1}^n + 12u_{j-2}^n - u_{j-3}^n}{6\Delta x^4}. \tag{29}$$

This scheme is formally first-order in time and fourth-order in space. Due to the stability constraint $\Delta t \leqslant 3\Delta x^4/40\kappa$, the scheme is $O(\Delta x^4)$ accurate overall.

Proceeding in the usual way and noting that the central difference approximation to the bilaplacian satisfies

$$B\tilde{u}^n = \frac{\partial^4 \tilde{u}}{\partial x^4} - \frac{7\Delta x^4}{240}\frac{\partial^8 \tilde{u}}{\partial x^8} + O(\Delta x^6), \tag{30}$$

we find that the local truncation error for (28) is given by

$$\frac{\partial^8 \tilde{u}^n}{\partial x^8}\left[\frac{7\Delta x^4}{240} - \frac{\kappa \Delta t}{2}\right](\kappa \Delta t) + \frac{\Delta t^2}{2}\left(\kappa \frac{\partial^4 f}{\partial x^4} - \frac{\partial f}{\partial t}\right) + O(\Delta t^3) + O(\Delta t \Delta x^6). \tag{31}$$

Therefore, the optimal time step is $\Delta t_{opt} = 7\Delta x^4/120\kappa$ and the correction term is

$$-\frac{\Delta t^2}{2}\left(\kappa \frac{\partial^4 f}{\partial x^4} - \frac{\partial f}{\partial t}\right). \tag{32}$$

Together, these modifications boost the accuracy of the original finite difference scheme from fourth- to sixth-order. As with the previous examples, the spatial derivatives in the correction term should be computed using a finite difference approximation that is at least second-order accurate. The improved accuracy of (28) when the optimal time step and defect correction terms are used is illustrated in Fig. 4(a).

Applying optimal time step selection to high-order PDEs requires the use of high-order discretizations of the spatial derivatives. While the need to derive high-order finite difference stencils may seem like an unnecessary burden, it is important to remember that high-order finite difference schemes for spatial derivatives are almost always necessary to maximize the efficiency of explicit time integration schemes for high-order PDEs[4]. Optimal time step selection merely provides a valuable way

[4] Because restrictive time step constraints are always a major limitation of explicit schemes for time dependent PDEs with high-order spatial derivatives [19,26], the overall error of finite difference schemes for high-order PDEs is usually dominated by the error introduced by the spatial discretization. As a result, it is crucial that numerical schemes for high-order PDEs use high-order discretizations for the spatial derivatives.

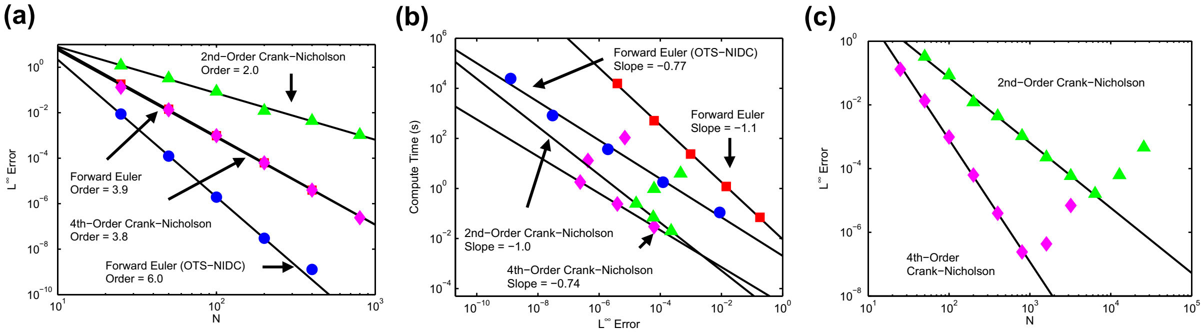

Fig. 4. (a) L^∞ error as a function of number of grid points for four finite difference schemes that solve the fourth-order parabolic Eq. (26): forward Euler with OTS–NIDC (circles), forward Euler with suboptimal time step and no correction term (squares), Crank–Nicholson with a fourth-order stencil for the bilaplacian and an $O(\Delta x^2)$ time step (diamonds), and Crank–Nicholson with a second-order stencil for the bilaplacian (triangles). Both forward Euler schemes use a fourth-order, central finite difference approximation for the bilaplacian. Note that the errors for the forward Euler with suboptimal time step and fourth-order Crank–Nicholson schemes lie almost directly on top of each other at the resolution of the figures. (b) Computation time as a function of L^∞ error. These results were obtained using MATLAB implementations of the finite difference schemes run on a dual-core 2.4 GHz MacBook Pro. Note the impact of round-off error associated with matrix inversion on the accuracy of the solution for the two Crank–Nicholson schemes. (c) Illustration of impact of round-off error associated with matrix inversion on the accuracy of the solution for the two Crank–Nicholson schemes when N is large.

Table 3
Computational cost as a function of the global numerical error e for various finite difference schemes that numerically solve the 1D fourth-order parabolic equation. For all finite difference schemes, the time step is chosen so that the temporal error is subdominant to the spatial error.

Numerical scheme	Δx	Δt	Memory	Compute time
Forward Euler with second-order Bilaplacian	$O(e^{1/2})$	$O(e^2)$	$O(e^{-1/2})$	$O(e^{-5/2})$
Forward Euler with fourth-order Bilaplacian	$O(e^{1/4})$	$O(e)$	$O(e^{-1/4})$	$O(e^{-5/4})$
Forward Euler with fourth-order Bilaplacian and OTS–NIDC	$O(e^{1/6})$	$O(e^{2/3})$	$O(e^{-1/6})$	$O(e^{-5/6})$
Crank–Nicholson with second-order Bilaplacian	$O(e^{1/2})$	$O(e^{1/2})$	$O(e^{-1/2})$	$O(e^{-1})$
Crank–Nicholson with fourth-order Bilaplacian	$O(e^{1/4})$	$O(e^{1/2})$	$O(e^{-1/4})$	$O(e^{-3/4})$

to leverage the effort spent deriving high-order finite difference schemes in order to further boost the overall accuracy of the numerical method.

Table 3 shows the asymptotic computational performance of various finite difference methods for solving Eq. (27). Observe the dramatic decrease in computational cost as a function of the error by simply replacing a second-order discretization for the bilaplacian operator with a fourth-order discretization. For the forward Euler scheme, using an OTS–NIDC drives the computational cost down even more. The performance plot in Fig. 4(b) shows that the theoretical estimates are in good agreement with timings from computational experiments. Note that the empirical performance for the forward Euler schemes is better than theoretically expected because the computational work per time step is not dominated by the $O(1/\Delta x)$ update calculations until Δx gets very small.

In Fig. 4(b), observe that the second-order Crank–Nicholson scheme actually outperforms the OTS–NIDC scheme over the high (but still practically useful) range of errors even though Table 3 predicts that the OTS–NIDC scheme should do better asymptotically. The origin of this discrepancy is the constants hidden by the big-O notation. In particular, the factor of 7/120 in the optimal time step for the forward Euler scheme with a fourth-order approximation of the bilaplacian leads to a large hidden constant for the compute time for the OTS–NIDC scheme, and the low cost of inverting the banded-diagonal bilaplacian matrix in 1D leads to a relatively small hidden constant in the Crank–Nicholson schemes.

As Fig. 4(b) and (c) show, there is an important advantage to explicit time integration schemes for the fourth-order parabolic equation – round-off error associated with matrix inversion does *not* limit the achievable accuracy of the numerical solution. In contrast, errors introduced in matrix inversion affects both the second- and fourth-order Crank–Nicholson schemes.

Comparing the computational cost of the best forward Euler and Crank–Nicholson schemes, we see that the OTS–NIDC forward Euler scheme more efficiently utilizes memory at the cost of requiring more computation time. This trade off between memory and computation only exists in one space dimension. In two-dimensions, both schemes require the same amount of compute time; in three or more space dimensions, the forward Euler scheme requires less compute time than the Crank–Nicholson scheme.

5. Application to PDEs in multiple space dimensions

OTS–NIDC can be used to boost the order of accuracy for finite difference schemes in any number of space dimensions. However, several important issues arise for problems in more than one space dimension. These issues are primarily related to the presence of cross-terms in the discretization error, multiple grid spacing parameters, and the domain geometry. In this section, we show how OTS–NIDC selection easily overcomes these issues by using it to construct high-order finite difference schemes for the 2D advection equation and the 2D diffusion equation. It is straightforward to generalize the approach presented in this section to more space dimensions and to construct methods for semilinear PDEs by incorporating ideas discussed in Sections 3 and 4.

5.1. Discretization of spatial derivatives

A significant distinction between the error for finite difference schemes in one space dimension and multiple space dimensions is the presence of cross-terms in the discretization error. Recall that OTS–NIDC requires the leading-order temporal errors to be directly related to the leading-order spatial errors via the PDE. This requirement can be satisfied in one of two ways:

- By carefully designing the spatial discretization so that the leading-order spatial errors are directly related to the spatial derivative operators in the PDE; or
- By explicitly adding defect correction terms to cancel out any cross-terms in the discretization error that cannot be eliminated by using the PDE.

These two approaches are actually equivalent because the correction terms can be combined with the finite difference stencils in the original scheme and be viewed as part of an alternate spatial discretization scheme.

5.2. Linear advection equation

For our first application of OTS–NIDC in multiple space dimensions, we consider the two-dimensional advection equation

$$\frac{\partial u}{\partial t} + \vec{A} \cdot \nabla u = 0, \tag{33}$$

where $\vec{A} = (A_x, A_y)$ is the advection velocity. The most interesting and important aspect of this problem is that elimination of the truncation error requires optimal selection of the grid spacing ratio as well as the time step.

For convenience, let us assume that $A_x, A_y > 0$. The simple forward Euler scheme for (33) with first-order upwind discretizations of the advection terms is

$$u_{i,j}^{n+1} = u_{i,j}^n - A_x \Delta t \left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} \right) - A_y \Delta t \left(\frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y} \right). \tag{34}$$

This method is formally first-order in both space and time with a stability constraint of $\Delta t < (A_x/\Delta x + A_y/\Delta y)^{-1}$.

To compute the optimal time step and grid spacing ratio, we begin by analyzing the discretization error for (34). As in our analysis of the finite difference scheme for the one-dimensional advection equation, we retain all of the terms in the discretization error. Using Taylor series expansions and the PDE (33), we find that the truncation error satisfies

$$\frac{A_x \Delta t}{2} \frac{\partial^2 \tilde{u}}{\partial x^2} (\Delta x - A_x \Delta t) + \frac{A_y \Delta t}{2} \frac{\partial^2 \tilde{u}}{\partial y^2} (\Delta y - A_y \Delta t) - A_x A_y \Delta t^2 \frac{\partial^2 \tilde{u}}{\partial x \partial y} + \frac{A_x \Delta t}{\Delta x} \sum_{k=3}^{\infty} \frac{(-\Delta x)^k}{k!} \frac{\partial^k \tilde{u}^n}{\partial x^k} + \frac{A_y \Delta t}{\Delta y} \sum_{k=3}^{\infty} \frac{(-\Delta y)^k}{k!} \frac{\partial^k \tilde{u}^n}{\partial y^k}$$
$$- \sum_{k=3}^{\infty} \frac{(-\Delta t)^k}{k!} (A_x \partial_x + A_y \partial_y)^k \tilde{u}^n. \tag{35}$$

From this expression, we see that choosing the ratio of the grid spacings to be $\Delta x/\Delta y = A_x/A_y$ and setting the time step to be $\Delta t_{opt} = \Delta x/A_x = \Delta y/A_y$ eliminates two of the three leading-order terms in the discretization error[5]. The last leading-order term can be eliminated by adding the defect correction terms

$$A_x A_y \Delta t^2 \frac{\partial^2 \tilde{u}}{\partial x \partial y}. \tag{36}$$

These choices enable us to boost the order of accuracy for the finite difference scheme (34) from first- to second-order.

[5] Note that the use of different values for Δx and Δy is consistent with physical intuition. When computing the upwind derivative in a given coordinate direction, the upstream value we use should be at a distance that is proportional to the flow speed in that direction. Otherwise, we will be using upstream data that is too far or too near in one of the coordinate directions.

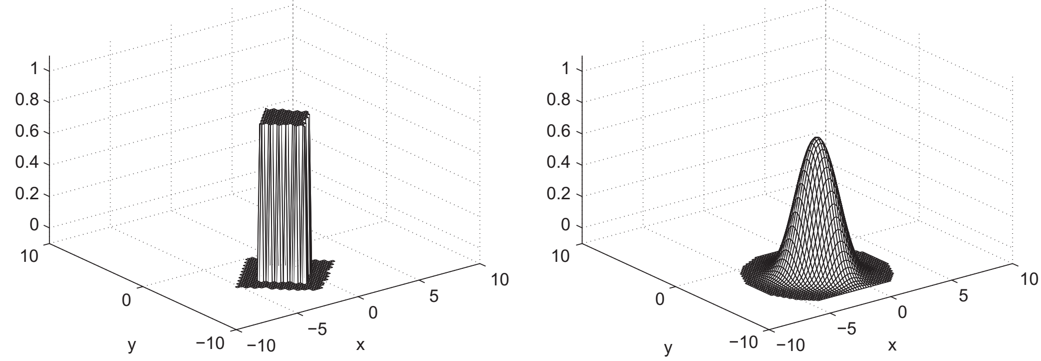

Fig. 5. Comparison of numerical solutions for the 2D advection Eq. (33) using forward Euler time stepping with OTS–NIDC (left) and forward Euler time stepping with suboptimal time step (right). The latter scheme uses a time step $\Delta t = \Delta x/2\ (|A_x| + |A_y|)$. Both figures show the numerical solution at the same time when the initial conditions are taken to be a 2D square function centered at the origin: $u(x,y,0) = 1$ if $|x| + |y| \leqslant 2$ and $u(x,y,0) = 0$ if $|x| + |y| > 2$.

With a careful choice of discretization for the correction term, we can boost the order of accuracy even further. Using the first-order upwind discretization for $\tilde{u}_{xy}$

$$\frac{(u_{i,j} - u_{i-1,j}) - (u_{i,j-1} - u_{i-1,j-1})}{\Delta x \Delta y}, \tag{37}$$

the truncation error (35) becomes

$$\begin{aligned}&\frac{A_x\Delta t}{2}\frac{\partial^2\tilde{u}}{\partial x^2}(\Delta x - A_x\Delta t) + \frac{A_y\Delta t}{2}\frac{\partial^2\tilde{u}}{\partial y^2}(\Delta y - A_y\Delta t) + \frac{A_x\Delta t}{\Delta x\Delta y}(\Delta y - A_y\Delta t)\sum_{k=3}^{\infty}\frac{(-\Delta x)^k}{k!}\frac{\partial^k\tilde{u}^n}{\partial x^k}\\ &+\frac{A_y\Delta t}{\Delta x\Delta y}(\Delta x - A_x\Delta t)\sum_{k=3}^{\infty}\frac{(-\Delta y)^k}{k!}\frac{\partial^k\tilde{u}^n}{\partial y^k} + \frac{A_xA_y\Delta t^2}{\Delta x\Delta y}\sum_{k=3}^{\infty}\sum_{l=0}^{k}(-1)^k\binom{k}{l}\frac{\Delta x^l\Delta y^{k-l}}{k!}\frac{\partial^k\tilde{u}^n}{\partial x^l\partial y^{k-l}} - \sum_{k=3}^{\infty}\frac{(-\Delta t)^k}{k!}(A_x\partial_x + A_y\partial_y)^k\tilde{u}^n,\end{aligned} \tag{38}$$

which vanishes when the optimal time step and grid spacing ratio are used. In other words, when the defect correction (37) is used in conjunction with the optimal time step and grid spacing ratio, we obtain a finite difference scheme that introduces no truncation error at each time step. Fig. 5 demonstrates the amazing accuracy of the defect corrected scheme over the first-order forward Euler scheme (34) without defect correction. Notice how the sharp corners of the initial conditions are perfectly preserved when the optimal time step, optimal grid spacing ratio, and correction term are used. In contrast, the sharp corners are completely smoothed out in the solution computed using the standard first-order upwind scheme.

It is interesting to push the analysis of the optimal finite difference scheme a little further and examine the defect-corrected finite difference scheme:

$$u_{i,j}^{n+1} = u_{i,j}^n - A_x\Delta t\left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x}\right) - A_y\Delta t\left(\frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y}\right) + A_xA_y\Delta t^2\left(\frac{u_{i,j} - u_{i-1,j} - u_{i,j-1} + u_{i-1,j-1}}{\Delta x\Delta y}\right). \tag{39}$$

Using the relationship between the optimal time step and the grid spacing ratio, this scheme can be rewritten as

$$u_{i,j}^{n+1} = u_{i,j}^n - \frac{A_x\Delta t}{2}\left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} + \frac{u_{i,j-1}^n - u_{i-1,j-1}^n}{\Delta x}\right) - \frac{A_y\Delta t}{2}\left(\frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y} + \frac{u_{i-1,j}^n - u_{i-1,j-1}^n}{\Delta y}\right), \tag{40}$$

which is essentially a forward Euler time integration scheme with first-order upwind discretizations of the spatial derivatives that are computed using all four upwind neighbors. This formulation of (39) suggests that the highest accuracy finite difference scheme for the 2D linear advection equation arises by using a fully upwind discretization of the advection terms with the optimal time step and optimal ratio of grid spacings.

In higher space dimensions, it becomes tedious to carry out the analysis to derive the infinite order accurate finite difference scheme for the advection equation because many high-order correction terms must be retained. However, it is straightforward to generalize (40) to d space dimensions by using first-order discretizations for the spatial derivatives that are averages over the 2^{d-1} upwind discretizations in each coordinate direction. As with the two-dimensional problem, the optimal grid spacings satisfy $\Delta x_k / A_{x_k} = \Delta t_{opt}$ for $1 \leqslant k \leqslant d$.

Like the unit CFL condition for the one-dimensional advection equation, the analogue of (40) in any number of space dimensions could also have been derived by examining how characteristic lines of the d-dimensional advection equation pass through grid points. For the two-dimensional problem, this perspective becomes apparent when we fully simplify (40) using the optimal time step and grid spacing ratio to obtain $u_{i,j}^{n+1} = u_{i-1,j-1}^n$.

5.3. Diffusion equation

As for PDEs in one space dimension, achieving infinite order accuracy through OTS–NIDC is unique to the advection equation. To illustrate the typical accuracy boost gained by using OTS–NIDC, we apply it to the 2D diffusion equation. This example underscores the need to carefully choose the discretization for the spatial derivatives for problems in more than one space dimension and demonstrates the ease with which OTS–NIDC selection can be applied to problems on irregular domains.

In two space dimensions, the diffusion equation may be written as

$$\frac{\partial u}{\partial t} = D\nabla^2 u + f(x,y,t). \tag{41}$$

The most common finite difference schemes for (41) employ a five-point, second-order central difference stencil for the Laplacian:

$$L^{5pt}u = \frac{u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} - 4u_{i,j}}{\Delta x^2}, \tag{42}$$

which assumes that the grid spacing in the x and y directions are equal. The forward Euler scheme using the five-point stencil is given by

$$u^{n+1}_{i,j} = u^n_{i,j} + \Delta t\left(DL^{5pt}u + f^n_{i,j}\right). \tag{43}$$

This scheme is formally first-order in time and second-order in space. The stability constraint $\Delta t \leqslant \Delta x^2/4D$ implies that this scheme is $O(\Delta x^2)$ accurate overall.

While it is certainly possible to directly apply OTS–NIDC to (43), we would be forced to use additional defect correction terms to deal with the fact that the leading-order error for (42), $\Delta x^2(\tilde{u}_{xxxx} + \tilde{u}_{yyyy})/12\Delta x^2(\tilde{u}_{xxxx} + \tilde{u}_{yyyy})/12$, cannot be directly related to the spatial derivative operator in the diffusion equation. The analysis is more straightforward if we construct our finite difference scheme using the nine-point stencil for the Laplacian [17,27], which possesses a leading-order error directly related to the spatial derivative operator in the PDE (41):

$$L^{9pt}u = \frac{1}{6\Delta x^2}\left(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1} + 4u_{i+1,j} + 4u_{i-1,j} + 4u_{i,j+1} + 4u_{i,j-1} - 20u_{i,j}\right). \tag{44}$$

Observe that the nine-point stencil is simply the standard five-point stencil plus $\Delta x^2/6$ times a second-order, central difference approximation to u_{xxyy}. Therefore, the error in the nine-point stencil is $\frac{\Delta x^2}{12}\nabla^4 u + O(\Delta x^4)$. The forward Euler scheme that uses the nine-point discretization of the Laplacian is also $O(\Delta x^2)$ accurate overall because it has a stability constraint $\Delta t \leqslant 3\Delta x^2/8D = O(\Delta x^2)$. It is worth noting that use of a nine-point stencil for the Laplacian naturally arises when directly applying OTS–NIDC to the five-point scheme if the defect correction term is interpreted as a modification to the original spatial discretization.

Analyzing the finite difference scheme based on the nine-point stencil for the Laplacian, we find that the leading-order terms of the local truncation error is

$$\nabla^4\tilde{u}\left[\frac{\Delta x^2}{12} - \frac{D\Delta t}{2}\right](D\Delta t) - \frac{\Delta t^2}{2}\left(D\nabla^2 f + \frac{\partial f}{\partial t}\right) + O(\Delta t^3) + O(\Delta t\Delta x^4). \tag{45}$$

Therefore, the optimal time step and defect correction terms are $\Delta t_{opt} = \Delta x^2/6D$ and $\Delta t^2(D\nabla^2 f + f_t)$, respectively. The resulting OTS–NIDC scheme that fourth-order accurate. It is interesting to note that the optimal time step for the forward Euler scheme for the diffusion equation is *independent* of the number of spatial dimensions.

The improvement in the accuracy of the forward Euler scheme with OTS–NIDC is shown in Fig. 6(a). Fig. 6(b) shows that the scaling of the computational time with the error is in good agreement with the theoretical estimates given in Table 2. As with the other examples, the forward Euler scheme with OTS–NIDC is much more computationally efficient than the two other schemes.

5.3.1. Irregular domains

All of the examples we have considered so far share an important feature: they are all defined on simple, rectangular domains. Any of several methods could have been used to obtain solutions with similar or even higher accuracy. When the domain is irregularly shaped, high-order methods are harder to design and implement. In this section, we extend the OTS–NIDC finite difference scheme for the 2D diffusion equation on regular domains to irregular domains. Other researchers have developed fourth-order accurate methods for the diffusion equation on irregular domains [3,4]. However, these earlier methods require a wider ghost cell region [3], use implicit time integration [3,4], or involve a more complex calculation for the boundary conditions [4]. As we shall see, it is possible to achieve fourth-order accuracy for the diffusion equation on irregular domains using only forward Euler integration and a compact, second-order stencil for the Laplacian.

To avoid the need for special stencils at grid points near the irregular boundary, we adopt the ‘ghost cell’ approach for imposing boundary conditions [3,4,15,16]. The main challenge in using the ghost cell method is ensuring ghost cells are filled

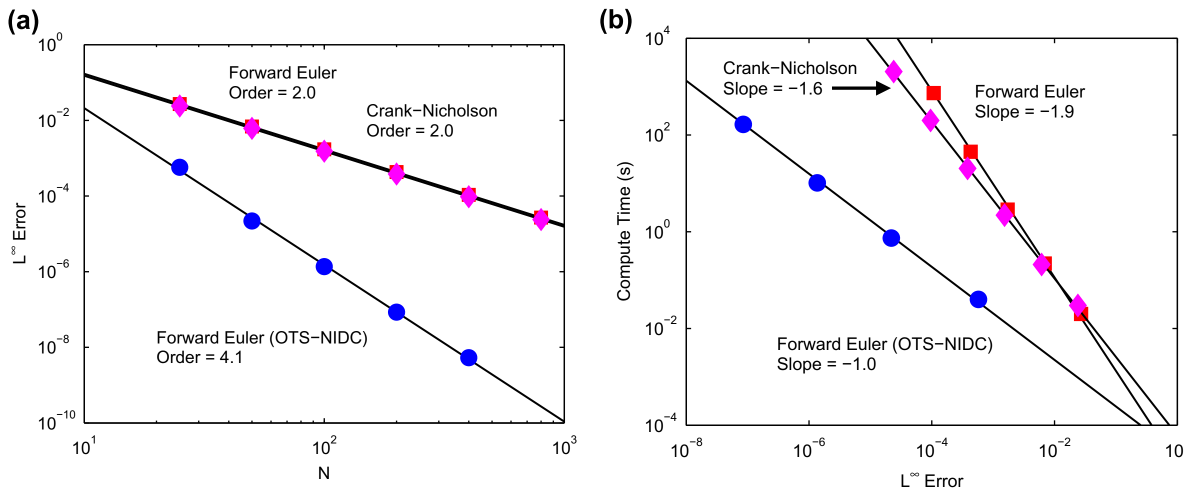

Fig. 6. (a) L^∞ error as a function of number of grid points in each coordinate direction (left) and computation time as a function of L^∞ error (right) for three finite difference schemes that solve the 2D diffusion equation: forward Euler with OTS–NIDC (circles), forward Euler with suboptimal time step (squares), and Crank–Nicholson (diamonds). The forward Euler schemes use a nine-point discretization of the Laplacian; the Crank–Nicholson scheme uses the standard five-point discretization of the Laplacian. Note that the errors for the forward Euler with suboptimal time step and Crank–Nicholson schemes lie almost directly on top of each other at the resolution of the figures. (b) Computation time as a function of L^∞ error. These results were obtained using MATLAB implementations of the finite difference schemes run on a dual-core 2.4 GHz MacBook Pro.

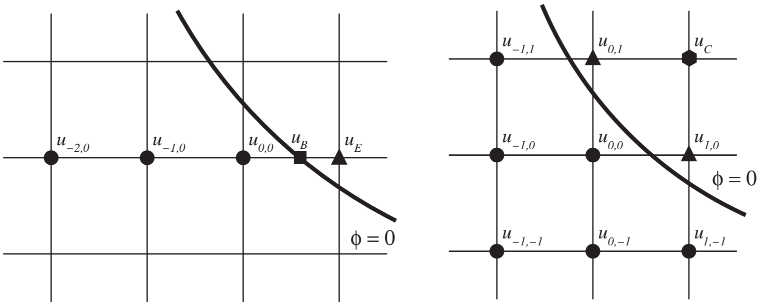

Fig. 7. Illustrations of edge (left) and corner (right) ghost cells. The edge ghost cell, u_E, is filled by using cubic extrapolation of the value on the boundary, u_B, and values at three interior grid points. The corner ghost cell, u_C, is filled by using a fourth-order accurate Taylor series expansion centered at $u_{0,0}$, which uses the values of u at the specified interior grid points and neighboring edge ghost cells to approximate the partial derivatives in the expansion.

using a sufficiently high-order extrapolation scheme. Fig. 7 shows examples of ghost cells that lie in the vicinity of an irregular boundary. Notice that there are two types of ghost cells: edge and corner. Edge and corner ghost cells are distinguished by the relative position of the nearest interior grid point. An edge ghost cell is linked to its nearest interior neighbor by an edge of the computational grid whereas a corner ghost cell and its nearest interior neighbor lie on the opposite corners of a grid cell.

To set the value of u in each edge ghost cell, we use 1D cubic (*i.e.* fourth-order accurate) extrapolation of the values of u across the interface. A cubic Lagrange interpolant is constructed for each edge ghost cell using a point on the domain boundary and three interior grid points that lie along the line parallel to the edge connecting the ghost cell to its nearest interior neighbor (see Fig. 7)[6]. Because the quality of the interpolant deteriorates if the boundary point u_B is too close to any of the interior grid points used to construct the Lagrange extrapolant, we follow [3] and choose the interior grid points so that the nearest interior grid point is sufficiently far from the boundary point. Without this procedure, the errors introduced in the values of the ghost cells lead to instability of the numerical method. Gibou et al. suggest shifting the interpolation points by one grid point when the distance between the boundary point and the nearest interior grid point is less than Δx^2 [3]. For the numerical scheme presented in this article, this threshold was too low. Better stability was obtained if an $O(\Delta x)$ threshold was used.

To set the value of u in a corner ghost cell, we use a fourth-order accurate Taylor series expansion of u centered at the nearest interior neighbor. The partial derivatives in the Taylor series expansion are approximated using finite differences

[6] If a level set representation of the boundary [16] is used, the location of the point u_B can be calculated by inverting the linear interpolant of ϕ that passes through the ghost cell and its nearest interior neighbor. We found that using higher-order interpolants to locate boundary points was unnecessary for achieving high-order accuracy in the computed solution.

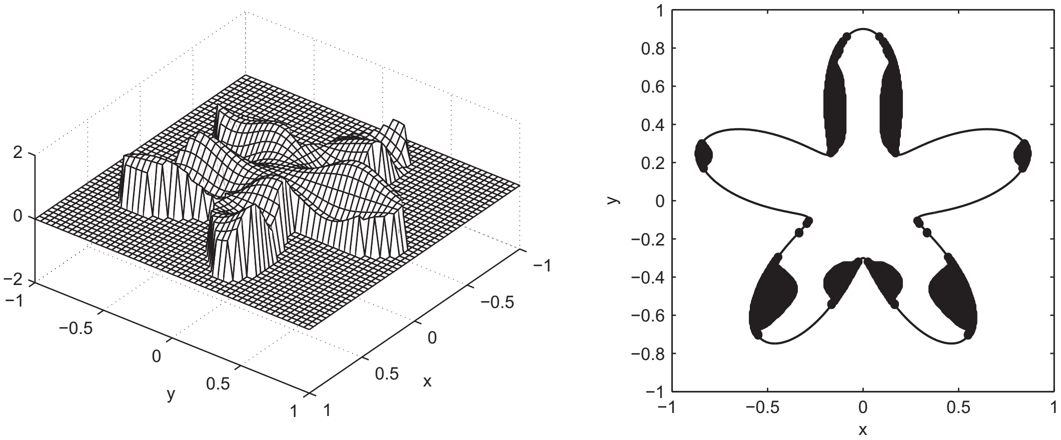

Fig. 8. Numerical solutions (left) and dominant error (right) for the 2D diffusion equation on a starfish-shaped domain. The solution is computed using forward Euler time stepping with OTS-NIDC. In the error plot, the dark regions represent points where the error in the solution is larger than *25%* of the L^∞ error of the solution.

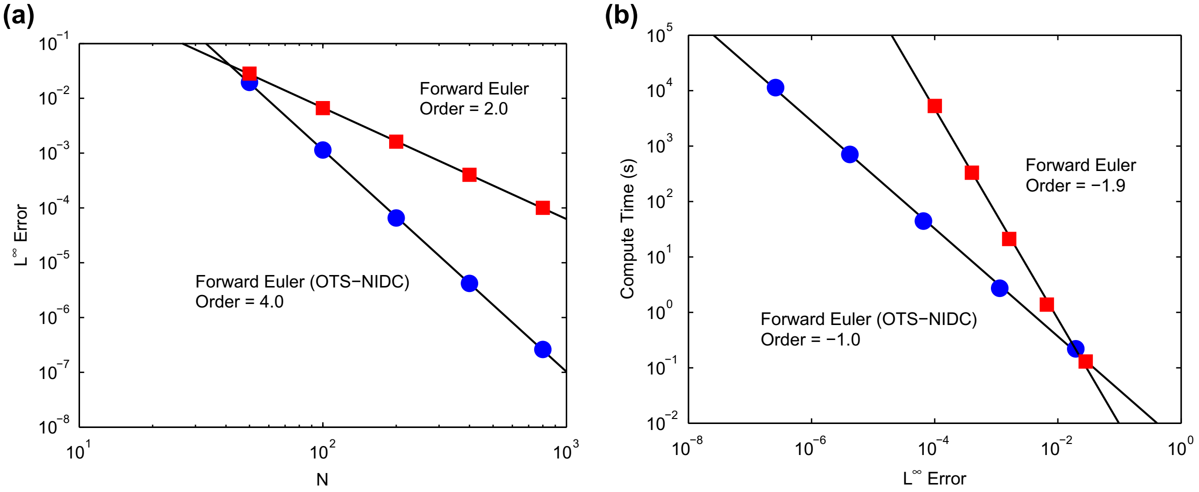

Fig. 9. (a) L^∞ error as a function of number of grid points in each coordinate direction for two finite difference schemes that solve the 2D diffusion equation on an irregular domain: forward Euler with OTS–NIDC (circles) and forward Euler with suboptimal time step (squares). (b) Computation time as a function of L^∞ error. These results were obtained using MATLAB implementations of the finite difference schemes run on a dual-core 2.4 GHz MacBook Pro.

constructed from the values of u at interior grid points and the neighboring edge ghost cells (see Fig. 7). For the corner ghost cell shown in Fig. 7, the extrapolation stencil is given by

$$u_C = -4u_{0,0} - u_{-1,-1} + 2u_{1,0} + 2u_{0,1} - u_{1,-1} - u_{-1,1} + 2u_{0,-1} + 2u_{-1,0}, \tag{46}$$

where u_C and $u_{i,j}$ are indicated in the figure. The stencils for corner ghost cells that are in other positions relative to the interior of the domain are obtained by rotations of Fig. 7 and remapping of the indices in the above stencil.

Using these methods for filling edge and corner ghost cells, it is straightforward to compute high-order accurate solutions of the 2D diffusion equation on irregular domains. We simply use the OTS–NIDC forward Euler scheme designed for regular domains for grid points in the interior of the domain after filling the ghost cells with high-order accurate values. Fig. 8 shows an example of a solution computed on a starfish shaped domain. It also shows that the error in the solution is concentrated near the boundaries of the irregular domain. The improved accuracy and performance of the OTS–NIDC solution compared to a simple forward Euler scheme with a suboptimal time step are illustrated in Fig. 9(a) and (b).

5.3.2. Diffusion equation in higher dimensions

Using OTS–NIDC to boost the accuracy of finite difference schemes for the diffusion equation in higher space dimensions is straightforward. As for the 2D problem, the key step is identification of a finite difference stencil for the Laplacian that has an *isotropic* discretization error. For 3D problems, a catalog of several such stencils is available in [27]. For higher dimensional problems, one simple approach for deriving an isotropic finite difference stencil for the Laplacian would be to start with the generalization of the standard five point stencil for the 2D Laplacian and add finite difference approximations for the missing cross terms that are required to make the leading-order error isotropic (*i.e.* proportional to the bilaplacian of the solution). Because the missing terms are all of the form

$$\frac{\Delta x^2}{6}\left(\frac{\partial^4 u}{\partial x_i^2 \partial x_j^2}\right), \tag{47}$$

this approach is relatively easy to use. To derive finite difference approximations with more general or complex properties, symbolic algebra software can be helpful [27,28].

Once an acceptable finite difference approximation for the Laplacian has been chosen, the usual OTS analysis yields an optimal time step of $\Delta t_{opt} = \Delta x^2/6D$ for forward Euler time integration and a boost of the order of accuracy from two to four. Above three dimensions, the optimal time step is no longer stable, and the forward Euler scheme should be replaced with the DuFort–Frankel scheme (see Section 4.3.1).

Problems on irregular domains can be handled using the ghost cell approach. While the derivation of high accuracy stencils for ghost cells becomes more tedious, the general approach is the same as for the 2D problem. The ghost cells can first be grouped into separate classes based on the number of grid cell edges that must be traversed to reach the nearest interior neighbor. The values in the ghost cells can then be set in order starting with those directly connected to their nearest interior neighbor and ending with those whose nearest interior neighbor is on the opposite corner of a grid cell. Here, too, symbolic algebra programs can be helpful when deriving methods for extrapolating interior and boundary values to ghost cells.

6. Conclusions

In this article, we have presented OTS–NIDC, a simple approach for constructing high-order finite difference methods for time dependent PDEs *without* requiring the use of high-order stencils or high-order time integration schemes. The essential idea underlying OTS selection is that leading order spatial and temporal discretization errors with the highest order derivatives can be simultaneously eliminated by optimally selecting numerical discretization parameters. Any terms in the truncation error that remain are handled in a straightforward manner using non-iterative defect correction. The boost in the order of accuracy achieved by using OTS–NIDC is well worth the extra effort required to derive the optimal time step and defect correction terms. In fact, the amount of analysis required to compute the OTS and correction terms is comparable to traditional defect/deferred correction and modified equation methods.

Through many examples, we have demonstrated the utility of OTS–NIDC to several types of finite difference schemes for a wide range of problems. We showed how OTS–NIDC can be used to very easily obtain high-order accurate solutions to many linear and semilinear PDEs in any number of space dimensions on both regular and irregular domains. The examples illustrate the high-order of accuracy that can be achieved using only low-order discretizations for spatial derivatives and simple forward Euler time integration. They also demonstrate the applicability of OTS–NIDC to more exotic finite difference schemes, such as the Kreiss–Petersson–Yström discretization of the second-order wave equation and the DuFort–Frankel scheme for the diffusion equation.

The work presented in this article suggest several potential areas for future research. First, because explicit time integration schemes tend to be easier to implement and appear to be more suited to modern graphics processing unit (GPU) hardware architectures, it would be interesting to explore application of OTS–NIDC to GPU implementations of numerical algorithms for time dependent PDEs. Second, while it is straightforward to derive of the optimal time step and defect correction terms by using Taylor series expansions, the process can be tedious and error-prone. It would be valuable to use modern symbolic algebra software to develop tools to help automate the process. Finally, the accuracy results for the fourth-order parabolic equation suggest that it may be prudent to empirically revisit the question of accuracy of numerical algorithms based on implicit time integration for problems involving high-order spatial derivatives.

Acknowledgments

The author gratefully acknowledges the support of Vitamin D, Inc. and the Institute for High-Performance Computing (IHPC) in Singapore. He would like to thank P. Fok and J. Lambers for many enlightening discussions and P. Fok, M. Prodanović, A. Chiu and the manuscript's anonymous reviewers for helpful suggestions on the manuscript.

References

[1] W.F. Spotz, G.F. Carey, Extension of high-order compact schemes to time-dependent problems, Numer. Meth. Part. D E 17 (2001) 657–672.

[2] A. Brüger, B. Gustafsson, P. Lötstedt, J. Nilsson, High order accurate solution of the incompressible Navier-Stokes equations, J. Comput. Phys. 203 (2005) 49–71.

[3] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.

[4] K. Ito, Z. Li, Y. Kyei, Higher-order, cartesian grid based finite difference schemes for elliptic equations on irregular domains, SIAM J. Sci. Comput. 27 (2005) 346–367.

[5] R.K. Shukla, X. Zhong, Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation, J. Comput. Phys. 204 (2005) 404–429.

[6] E.A. Heidenreich, J.F. Rodríguez, F.J. Gaspar, M. Doblaré, Fourth-order compact schemes with adaptive time step for monodomain reaction-diffusion equations, J. Comput. Appl. Math. 216 (2007) 39–55.
[7] R.K. Shukla, M. Tatineni, X. Zhong, Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations, J. Comput. Phys. 224 (2007) 1064–1094.
[8] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, UK, 2002.
[9] V. Pereyra, Iterated deferred corrections for nonlinear boundary value problems, Numer. Math. 11 (1968) 111–125.
[10] H.J. Stetter, The defect correction principle and discretization methods, Numer. Math. 29 (1978) 425–443.
[11] B. Gustafsson, L. Hemmingsson-Frïdén, Deferred correction in space and time, J. Sci. Comput. 17 (2002) 541–550.
[12] W. Kress, B. Gustafsson, Deferred correction methods for initial boundary value problems, J. Sci. Comput. 17 (2002) 241–251.
[13] W. Kress, Error estimates for deferred correction methods in time, Appl. Numer. Math 57 (2006) 335–353.
[14] D.F. Griffiths, J.M. Sanz-Serna, On the scope of the method of modified equations, SIAM J. Sci. Stat. Comput. 7 (1986) 994–1008.
[15] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[16] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, NY, 2003.
[17] A. Iserles, A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, Cambridge, UK, 1996.
[18] L.F. Shampine, Error estimation and control for ODEs, J. Sci. Comput. 25 (2005) 3–16.
[19] B. Gustafsson, H.-O. Kreiss, J. Oliger, Time Dependent Problems and Difference Methods, John Wiley & Sons, Inc., New York, NY, 1995.
[20] J.R. Cash, Y. Psihoyios, The MOL solution of time dependent partial differential equations, Comput. Math. Appl. 31 (1996) 69–78.
[21] G. Psihoyios, Solving time dependent pdes via an improved modified extended bdf scheme, Appl. Math. Comput. 184 (2007) 104–115.
[22] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhäuser Verlag, Berlin, 1992.
[23] H.-O. Kreiss, N.A. Petersson, J. Yström, Difference approximations for the second order wave equation, SIAM.J. Numer. Anal 40 (2002) 1940–1967.
[24] G.B. Whitham, Linear and Nonlinear Waves, John Wiley & Sons, Inc., New York, NY, 1999.
[25] T. Chan, Stability analysis of finite difference schemes for the avection–diffusion equation, SIAM J. Numer. Anal. 21 (1984) 272–284.
[26] J.B. Greer, A.L. Bertozzi, G. Sapiro, Fourth order partial differential equations on general geometries, J. Comput. Phys. 216 (2006) 216–246.
[27] M. Patra, M. Karttunen, Stencils with isotropic discretization error for differential operators, Numer. Meth. Part. Diff. Eq. 22 (2005) 936–953.
[28] M.M. Gupta, J. Kouatchou, Symbolic derivation of finite difference approximations for the three-dimensional poisson equation, Numer. Meth. Part. Diff. Eq. 14 (1998) 593–606.