

Fourier continuation methods for high-fidelity simulation of nonlinear acoustic beams

Nathan Albin

Mathematics, Kansas State University, 138 Cardwell Hall, Manhattan, Kansas 66506

Oscar P. Bruno^{a)}

Applied and Computational Mathematics, California Institute of Technology, 1200 E California Boulevard, Pasadena, California 91125

Theresa Y. Cheung

Department of Mechanical Engineering, Boston University, 110 Cummington Street, Boston, Massachusetts 02215

Robin O. Cleveland

Institute of Biomedical Engineering, University of Oxford, Old Road Campus Research Building, Oxford OX3 7DQ, United Kingdom

(Received 12 November 2010; revised 25 May 2012; accepted 12 July 2012)

On the basis of recently developed Fourier continuation (FC) methods and associated efficient parallelization techniques, this text introduces numerical algorithms that, due to very low dispersive errors, can accurately and efficiently solve the types of nonlinear partial differential equation (PDE) models of nonlinear acoustics in hundred-wavelength domains as arise in the simulation of focused medical ultrasound. As demonstrated in the examples presented in this text, the FC approach can be used to produce solutions to nonlinear acoustics PDEs models with significantly reduced discretization requirements over those associated with finite-difference, finite-element and finite-volume methods, especially in cases involving waves that travel distances that are orders of magnitude longer than their respective wavelengths. In these examples, the FC methodology is shown to lead to improvements in computing times by factors of hundreds and even thousands over those required by the standard approaches. A variety of one- and two-dimensional examples presented in this text demonstrate the power and capabilities of the proposed methodology, including an example containing a number of scattering centers and nonlinear multiple-scattering events.

© 2012 Acoustical Society of America. [http://dx.doi.org/10.1121/1.4742722]

PACS number(s): 43.58.Ta, 43.80.Sh [OAS]

Pages: 2371–2387

I. INTRODUCTION

It is by now well understood that standard, low-order finite-difference (FD), finite-element (FEM), or finite-volume (FV) numerical methods are not well-suited for large-scale acoustic simulations wherein the distance traveled by the highest-frequency waves during the evolution of the solution is orders of magnitude larger than the corresponding wavelength. Specifically, low-order methods are known to exhibit large dispersive errors^{1–5} that cause significant phase errors when acoustic waves are propagated over large distances. These dispersive errors are frequency-dependent, with the smallest errors typically arising in the low-frequency wave modes, and, thus, the associated phase errors are exacerbated in nonlinear applications for which the nonlinearity can generate numerous high-frequency harmonics. In practice, this means that low-order methods are contraindicated for use in problems involving the long-distance propagation of acoustic waves.

As shown in Ref. 4, for numerical solutions to advection-dominated PDE problems there is typically an error relation of the form

$$E \approx \text{const} \times (\Delta x)^p k^{p+1} T, \quad (1)$$

where E is the error in the k th Fourier mode of the numerical solution at time T , Δx is the spatial discretization step-size, and p is the order of accuracy with which the spatial derivatives are approximated. This relation, which is based on periodic Fourier analysis, provides a very accurate prediction of the accuracy with which a numerical method will propagate waves in the interior of the computational domain. Thus, it is a particularly useful tool in analyzing solvers for acoustic problems involving long-distance sound wave propagation. Relation (1) is instrumental in understanding the behavior of a numerical scheme as it is “scaled up” to increasingly larger problems, since it suggests how the spatial discretization should be adjusted with the other parameters in order to retain a given level of accuracy:

$$\Delta x \sim k^{-(p+1)/p} T^{-1/p}. \quad (2)$$

Assuming two numerical methods have the same order of computational complexity (e.g., both scale linearly with the number of discretization points), relation (2) shows that the higher-order method will always outperform the lower-order method for sufficiently large k or T .

^{a)}Author to whom correspondence should be addressed. Electronic mail: obruno@caltech.edu

Of course, whether a high- or low-order method is preferable for a particular problem depends on the constants of proportionality in relation (1) and in the computational complexity relation as well as on the frequencies and time scales involved. As a concrete example, the present contribution concerns problems arising in the simulation of high intensity focused ultrasound (HIFU), which employs sharply focused, high-frequency (~ 1 MHz) transducers to create rapid and extreme temperature rises at a focal point many fundamental wavelengths away from the transducer. The relatively large propagation distances (on the order of a hundred times the fundamental wavelength) and significant nonlinear effects present in HIFU PDE models suggests that such problems could be more efficiently treated by high-order than low-order methods. Indeed, it is demonstrated in what follows that high-order methods are not only preferable for HIFU-type problems, but are likely crucial for their solution. Although FD simulations of HIFU fields with domains on the order of 100 wavelengths have been reported in the literature,^{6–8} numerical accuracies have only been estimated for much smaller problems, on the order of ten fundamental wavelengths or less (Ref. 7, Sec. IV A), or for cases in which significant nonlinear steepening does not occur (Ref. 8, Sec. V C).

The computational difficulties involved in treating full-wave ultrasound models for HIFU and other medical contexts have historically been a driving force toward the development of simplified (and more computationally tractable) nonlinear acoustics models (see, e.g., Refs. 9–16). However, as computing technology has become increasingly powerful, there has been a renewed interest in solving full-wave models (e.g., Navier-Stokes or Westervelt equations).^{7,17–24} One reason for developing improved computational methodologies, therefore, is to enable the use of less restrictive HIFU models in practice. Moreover, the development of advanced computational algorithms is useful in the field of HIFU simulation as a whole. Regardless of the model chosen, replacing a low-order numerical scheme by a high-order one can yield significant resolution improvements and enable accurate solutions of the model equations for larger-scale problems than is currently possible. This point is illustrated in the one-dimensional examples presented in Sec. IV.

Because accurate HIFU simulation requires the faithful advection of acoustic waves across large areas of the domain's interior [which gives rise to large values of k in Eqs. (1) and (2)], the analysis presented at the beginning of this section indicates that use of a numerical method that exhibits high-order accuracy in the interior can be greatly advantageous. In the present work, the recently developed Fourier continuation (FC) algorithm and associated efficient parallelization techniques^{25–27}—which offer essentially spectral ($p \gg 1$) accuracy over the bulk of the domain's interior—are used to produce accurate numerical solutions of the full nonlinear acoustic equations for problems in which such direct simulation was previously impractical or impossible. The method, which can yield accurate solutions using significantly coarser spatial discretization than needed by previous approaches, can lead to improvements in computing times by factors of hundreds or even thousands over those required by other methodologies—thus significantly

increasing the feasibility of the direct solution of HIFU problems.

Although the bulk of the present paper focuses on faithful long-distance nonlinear advection, some brief discussions of potential modifications for treating other difficulties found in HIFU models are also included in Sec. VIII. While the treatment of all aspects of modern HIFU models by FC-based methods has not yet been fully developed, this paper is intended to present a compelling argument that the FC methodology can be used as a powerful tool in the development of large-scale, full-domain simulation of HIFU wave propagation.

II. FOURIER CONTINUATION FUNDAMENTALS: FC(GRAM) ALGORITHM

As is apparent in Sec. III below, the key distinguishing characteristic of the FC solver is the method of spatial differentiation, which is based on a certain FC(Gram) approach. The FC(Gram) algorithm, which was introduced in Refs. 25–27 (cf. Refs. 28–30) enables high-order/spectral convergence of Fourier expansions of non-periodic functions—thus resolving the well known Gibbs ringing phenomenon and associated slow convergence of Fourier series of functions whose periodic extension are discontinuous. In particular, the FC-based Navier-Stokes solver introduced in Ref. 27 (a) enables efficient (linear scaling) *parallel* implementations for general domains, (b) exhibits dispersion relations of spectral character and very high orders of convergence, (c) like all FC methods, does not suffer from the Gibbs phenomenon, and, importantly (d) does not suffer from the severe CFL restrictions arising from polynomial-based spectral methods; see Ref. 27 for details.

The Fourier continuation algorithm proceeds as follows. Given point values $f_j = f(x_j)$ of a function $f: [0, 1] \rightarrow \mathbb{R}$ at N points $x_j = j/(N-1) \in [0, 1]$, $j = 0, 1, \dots, N-1$, the FC algorithm produces, following the prescriptions below, a trigonometric polynomial f^c (a Fourier continuation of f) of periodicity interval $[0, b]$ ($b > 1$),

$$f^c(x) = \sum_{k=-W}^W a_k \exp(2\pi i k x / b) \quad \text{such that } f^c(x_j) = f_j \text{ for } j = 0, 1, 2, \dots, N-1, \quad (3)$$

which approximates f in the interval $[0, 1]$. The bandwidth W and period b are parameters of the FC method, which are defined below. Note that for $b = 1$, Eq. (3) reduces to a discrete Fourier transform of the data which would generally suffer from the Gibbs ringing effect near the points $x = 0$ and $x = 1$. The selection $b > 1$, in contrast, allows the trigonometric polynomial to smoothly transition from the values of f_j near $j = N-1$ back to the values of f_j near $j = 0$ without generating Gibbs ringing.

A primary component of the construction of the FC representation (3) is a discrete periodic extension of the given function values into a longer interval. This discrete extension is obtained by appending to the original N function values an additional $C > 0$ function values that provide a smooth transition from f_{N-1} back to f_0 in the interval $[1, b]$, where

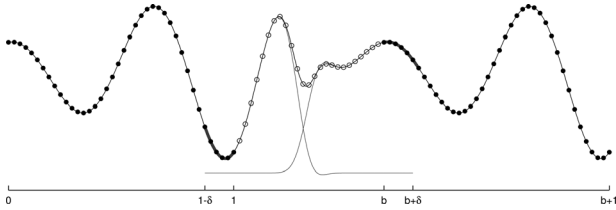


FIG. 1. An illustration of the main elements of the FC method: after producing a translated copy of the given values of f on the interval $[b, 1+b]$, the algorithm proceeds to (1) obtain a smooth function that blends the original function on the interval $[0, 1]$ to zero in the interval $[1, b]$, and, similarly, to obtain a smooth function that blends the original (translated) function on the interval $[b, 1+b]$ to zero in the interval $[1, b]$; and to (2) add these two blending functions to obtain a smooth continuation. The original function values are represented by solid circles and the additional continuation values are represented by open circles. The thick curves denote the polynomial approximations in the matching intervals. The thin gray lines show the blending functions that are summed to produce the continuation values. The Fourier continuation f^c then results by a direct application of the FFT algorithm.

$b = (N + C)/(N - 1)$, as illustrated in Fig. 1. Once this discrete periodic extension is produced, an application of the FFT algorithm in the interval $[0, b]$ yields the coefficients a_k of the Fourier continuation f^c shown in Eq. (3). The resulting approximation has bandwidth $W = \lfloor (N + C)/2 \rfloor$.

To evaluate the necessary discrete periodic extension, the FC(Gram) algorithm first produces a polynomial interpolation (using a small number d of function values) near each one of the endpoints of the interval $[0, 1]$ —where d is a small integer independent of N . As demonstrated in Fig. 1, this gives rise to a high-order representation of the function in the *matching intervals* $[1 - \delta, 1]$ and $[b, b + \delta]$ [$\delta = (\delta - 1)/(N - 1)$], respectively. The method for evaluating the discrete periodic extension is based on a representation of these two polynomials in a particular orthogonal polynomials basis (the Gram polynomials), for each element of which the algorithm possesses a precomputed smooth function which blends the basis polynomial to the zero function over the distance $b - 1$.²⁷ Linear combinations of these “blending to zero” functions provide smooth transitions-to-zero from the left-most and right-most function values in the extension interval $[1, b]$, as shown in Fig. 1. The sum of these two functions provides a smooth transition between the right-most function values to the left-most function values as shown in the figure. The values of this transition function at the points $N/(N - 1)$, $(N + 1)/(N - 1)$, ..., $(N + C - 1)/(N - 1)$ provide the necessary C additional point values; as mentioned above, the continuation function f^c can then be obtained through an application of the FFT algorithm in the interval $[0, b]$.

Once a continuous continuation function f^c has been obtained from the given point values f_j by means of the FC process, see Eq. (3), numerical derivatives of the original function f can be obtained with high-order/spectral accuracy by term-by-term differentiation of the Fourier series for f^c :

$$\frac{\partial f}{\partial x}(x) \approx \frac{\partial f^c}{\partial x}(x) = \sum_{k=-W}^W (2\pi k/b) a_k \exp(2\pi i k x/b) \quad \text{for } x \in [0, 1]. \quad (4)$$

The pseudocode of the function `diff` (Algorithm 1 below) outlines the overall FC procedure for numerical evaluation of derivatives of a given vector v . This algorithm involves an application of the FC method followed by an application of (4); clearly, derivatives of higher order can be produced similarly. The filtering step, labeled `filter` in Algorithm 1, provides a high-order spectral filter which is sometimes needed for stability of PDE solvers based on the function `diff`; should this filter not be used, energy arising from numerical errors may tend to accumulate and grow rapidly in the highest frequency modes. The effect of this filter is to multiply the k th Fourier coefficient of its argument by the quantity

$$\exp\left(-\alpha(2k/N)^{2p}\right), \quad (5)$$

where α , N , and p are suitably selected parameters. A detailed discussion of the properties of this filter and corresponding parameter selection is presented in Ref. 27; for the simulations presented in this paper, the filter parameters $\alpha = 16 \ln 10$ and $p = 50$ (which were found to give a robust solver while retaining excellent dispersion properties and high accuracies) have been used.

Algorithm 1: The function `diff`, which computes one-dimensional derivatives of a vector of function values.

```

given the function values  $v$ 
//apply the periodic extension (Fig. 1)
 $v_c \leftarrow \text{extend}(v)$ 
//complete the FC construction (3)
 $v_c \leftarrow \text{fft}(v_c)$ 
//apply the spectral filter (5)
 $\hat{v}_c \leftarrow \text{filter}(\hat{v}_c)$ 
//repeated applications of (4)
 $(\hat{v}'_c, \hat{v}''_c, \dots) \leftarrow \text{fft\_diff}(\hat{v}_c)$ 
//evaluate in configuration space
 $(v_c, v'_c, v''_c, \dots) \leftarrow \text{ifft}(\hat{v}_c, \hat{v}'_c, \hat{v}''_c, \dots)$ 
//restrict to original interval
 $(v, v', v'', \dots) \leftarrow \text{restrict}(v'_c, v''_c, \dots)$ 
return the filtered  $v$  and its derivatives.

```

Derivatives in multiple spatial dimensions on a structured mesh are implemented through successive line-by-line applications of the algorithm `diff_2d` in each coordinate direction. Algorithm 2 outlines the procedure `diff_2d` for approximating the gradient of a scalar quantity ϕ on a two-dimensional rectangular grid. The algorithm is straightforward: it simply sweeps through horizontal and vertical lines, applying the one-dimensional differentiation algorithm to each line. The generalization to higher dimensions and higher-order derivatives can be accomplished by suitable generalizations of Algorithms 1 and 2.

Algorithm 2: The function `diff_2d`, which computes the gradient of a two-dimensional scalar function on a rectangular grid.

```

given the array of function values  $\phi$ 
for all rows  $r$  of the grid do

     $v \leftarrow$  row  $r$  of  $\phi$ 

    //note that  $v$  is filtered in this step
     $(v, v') \leftarrow \text{diff}(v)$ 

    row  $r$  of  $\phi \leftarrow v$ 

    row  $r$  of  $\phi_x \leftarrow v'$ 

```

```

end for
for all columns  $c$  of the grid do

```

```

     $v \leftarrow$  column  $c$  of  $\phi$ 

    //note that  $v$  is filtered in this step
     $(v, v') \leftarrow \text{diff}(v)$ 

    column  $c$  of  $\phi \leftarrow v$ 

    column  $c$  of  $\phi_y \leftarrow v'$ 

```

```

end for
return the filtered  $\phi$  and its derivatives.

```

III. FC SOLVERS IN ONE-DIMENSIONAL SPACE

As a first application of the FC method, consider the following FC-based solver for the simple linear advection equation

$$u_t + u_x = 0. \tag{6}$$

In preparation for the examples presented in Sec. IV, periodic boundary conditions are considered, but see Remark 1 in these regards. The interval $[a, b]$ is discretized uniformly and the solution is represented in the interval by its values on the N equispaced discretization points. After the number of discretization points N has been chosen, the PDE (6) can then be represented as a system of ordinary differential equations (ODEs) of the form

$$\frac{dv}{dt} = -Dv, \tag{7}$$

where the vector function $v(t) = \{v_j(t)\}$ is an approximate solution to Eq. (6) in the sense that

$$v_j(t) \approx u(x_j, t), \quad x_j = a + (b - a)j/N, \quad j = 0, 1, 2, \dots, N - 1. \tag{8}$$

Except for a novel implementation of the operator D in Eq. (7), which is the discrete FC approximation to the differential operator $\partial/\partial x$, the solution is obtained as it would be for a classical finite difference scheme: the ODE system is solved by means of a standard explicit time-marching scheme (fourth-order Adams-Bashforth or Runge-Kutta for the examples presented in the present paper).

The remainder of this section is devoted to the description of the FC derivative operator D . In Sec. III A, the details of block domain decomposition are discussed. This decomposition is vital for the parallelization technique described in Sec. VB and is introduced in this simplified, one-dimensional setting in anticipation of that section. In Sec. III B, the application of the FC derivative operator D is described.

A. Block domain decomposition

Figure 2(a) illustrates a particular discretization of the interval $[a, b]$. In anticipation of the parallelization technique described in Sec. VB, the N discretization points are evenly partitioned into M subdomain blocks. Note that for graphical simplicity the figure shows significantly smaller parameter values ($N=30$ and $M=3$) than are typically used in practice. In a parallel implementation, the solution would be evolved almost independently in each of the three 10-point subdomain blocks. Of course, the solution in each subdomain block cannot be computed entirely independently of the solution in other blocks, or it would be impossible for waves to cross subdomain boundaries. However, by the use of the *fringe regions* described below, the solution u is evolved in each subdomain block with a small amount of data communicated from neighboring blocks.

Returning to Fig. 2(a), consider the solution in the right-most of the $M=3$ subdomain blocks. At each time-step, the solution is evolved at the block's $N/M=10$ discretization points, shown as black circles in the figure. The solution at the next time-step must, of course, depend on the current solution at these same discretization points. The evolved solution also depends on the solution values at a few additional points in the domain: the unfilled circles indicate discretization points that are not used to evolve the solution in the right-most block. The remaining gray circles indicate points in the *fringe region*, which are exterior to the block, but which are nonetheless used to evolve the solution within the block. In the FC code used to treat the examples presented in

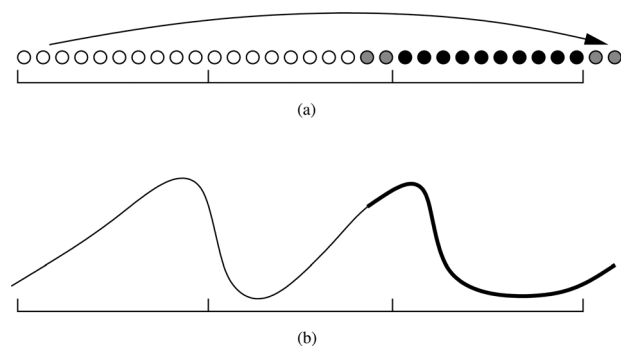


FIG. 2. A simplified illustration of the spatial discretization used by the FC-based solver for the advection Eq. (6). (a) $N=30$ grid points are evenly divided into $M=3$ subdomains. (b) The curve represents a function that is to be differentiated by means of the FC approach. The derivative values at the points in the right-most subdomain [black circles in (a)] are computed using the function values at these points as well as external fringe region points [gray circles in (a)]; the function values on this augmented set of points is represented by the thicker portion of the curve in (b). This (non-periodic) function is differentiated by means of the FC differentiation algorithm. The derivative values in the remaining two subdomains are produced analogously.

this paper, the fringe region typically extends by 4 or 6 points (called a 4-point or 6-point fringe, respectively) beyond the boundary of a block; for the purposes of illustration, however, a two-point fringe is shown in the figure. Due to the periodicity assumed in the current setting, the fringe of the right-most block also extends (periodically) into the left-most block. (Non-periodic, boundary value problems are treated analogously, but without the use of fringe data at the boundary of the domain; see also Remark 1.) Use of fringe regions is crucial in this context, as they allow solution data to pass between neighboring blocks as the solution progresses in time. In a distributed-memory parallel processing environment, the fringe regions require extra care: data in the fringes must be passed between processors controlling neighboring blocks at the beginning of each time step. (In a single-processor or a shared-memory parallel environment, solution values in the fringe regions are immediately accessible to all processors, of course, and no data communication is necessary.)

B. Application of the FC derivative operator

Having discretized the PDE (6) on a uniform N -point grid by means of Eq. (7), and having further subdivided these N points into M subdomain blocks as in Sec. III A, it is now straightforward to describe the application of the FC derivative operator D . The operator is applied to the solution vector v by application of Algorithm 1 to each of the M subdomain blocks; this is illustrated in Fig. 2(b). In the figure, the entire solid curve represents the discrete function v to which the FC derivative operator D is to be applied. The thicker portion of the curve represents the point values used to compute the values of Dv in the right-most block, which include the values at all points within the block as well as the values in the fringe region. The derivative values in the right-most block are computed by applying the FC differentiation procedure defined in Algorithm 1 to this augmented set of function values. The full derivative operator D is applied to v by repeating this process for each of the M subdomain blocks. As the solution is evolved, this block-by-block method for computing Dv is used whenever a spatial derivative is required by the time-stepping algorithm.

IV. FC SOLVERS: ONE-DIMENSIONAL CASE STUDIES

Because it is based on Fourier series expansions, the FC method exhibits essentially spectral accuracy in the interior of the computational domain.²⁷ Thus, as discussed in Sec. I, for wave-propagation problems that require sufficiently long-time propagation of high-frequency wave modes, the FC solvers are significantly more efficient than their lower-order counterparts. In the present section, it is argued that problems of full-domain HIFU simulation are well within the regime where the benefits of the FC method are observable. This is demonstrated through comparisons in the context of two one-dimensional acoustic models: the linear advection equation and the (strongly advecting) viscous Burgers equation. In these one-dimensional studies, the performance of the FC method is of significantly higher quality than those resulting from low-order approaches. Based on these

results, it is then argued that the differences in efficiency would be even more pronounced in higher dimensions.

In order to demonstrate the important performance differences between the FC solver and low-order algorithms, the solution of the one-dimensional equations explored in this section are obtained by means of three different types of solvers: (1) the one-dimensional FC algorithm described in Sec. III, (2) a standard second-order finite difference solver, and (3) the Godunov-type, finite-volume (FV) CLAWPACK solver,³¹ including various flux-limiters provided as part of the CLAWPACK package. While these comparisons are by no means exhaustive, they do demonstrate that use of high-order methods is necessary for long-range, nonlinear advection problems such as arise in HIFU simulation.

It should be mentioned, of course, that a wide variety of high-order finite-difference and finite-element algorithms exist. Examples include high-order centered-difference⁵ methods, summation by parts³² (SBP) methods, Chebyshev-based spectral methods,³³ Padé (or compact) schemes,^{34,35} and discontinuous Galerkin methods.³⁶ The reader is referred to Refs. 25–27 for detailed discussions on the relevance of comparisons of the FC method with various approaches. Briefly, one finds, on one hand, methods that, like the discontinuous Galerkin method, deliver high-order accuracy for general domains in two- and three-dimensional space, but whose computational cost is very high. (In a specific comparison presented in Ref. 27 with a previously published two-dimensional result, for example, the FC method demonstrated a speedup by a factor of 200, for a certain error tolerance, over the computing time required by a highly optimized DG algorithm.) High-order finite difference methods, on the other hand, present difficulties in handling boundary conditions: as discussed in Ref. 26, for example, even for a simple diffusion equation, fourth order of accuracy is the highest order that has been demonstrated for general domains. A similar situation occurs for Padé and SBP methods: these methods rely on use of complex filtering methodologies, and, unlike the FC method,²⁷ require (generally prohibitively complex) perfect grid alignment for stability in the multidomain formulations they use to incorporate complex geometries and parallelization; see, e.g., Refs. 32, 37, and 38. Similarly, Chebyshev methods do not provide a consistent basis for use as solvers on general geometries, unless domain mappings into rectangular domains are used, and, in view of their inherent point clustering, exhibit extremely stiff CFL constraints. In sum, only a few algorithms of order higher than 2, up to order 4 (especially at boundaries) exist. Such methods require challenging special treatments for stability in general domains and do not parallelize easily. The comparisons with second order finite difference methods that are presented in what follows thus provide an indication of the difficulties that arise from finite difference and other methodologies in general contexts. As discussed in detail in Ref. 27, in contrast, the FC method does not give rise to difficulties at boundaries, in view of its connection with Fourier series, the filtering step is straightforward, and, in fact, it gives rise to a nearly exact dispersion relation: as shown in Ref. 27, the FC numerical dispersion relation matches the exact relation *much more closely* than

all other available algorithms, including both the Padé method^{34,35} and the centered-difference method of eighth-order of interior accuracy.⁵ The numerical examples in Sec. VII demonstrate in a two-dimensional context the character of the proposed FC approach.

All computations for the examples presented in this section were performed on a 2.5 GHz AMD Opteron processor. Using the notation of Sec. III, the number of blocks M used in the FC solver was chosen to scale with the number of grid points N so that $N/M \approx 1000$ and the solution was integrated in time via the standard fourth-order Runge-Kutta (RK4) method. The finite difference (FD2) results were produced by a solver based on second-order centered differences in space and RK4 in time. The CLAWPACK (CL) results were produced by the linear advection and Burgers equation solvers included in CLAWPACK version 4.3; the notation CL(x) indicates that the CLAWPACK solver was run with the flux limiter selection parameter `methlim=x`. The value $x=0$ indicates that no limiter was used, while the values $x=1, 2, 3, 4$ indicate that the minmod, superbee, van Leer and monotized centered flux limiter, respectively, was enabled. The results thus include runs that incorporate each one of the four flux limiters provided with the CLAWPACK software.

In order to ensure reasonable computing-time comparisons, for each algorithm and each spatial discretization Δx , the temporal discretization Δt was selected in such a way that additional temporal refinement did not significantly reduce the error, but that temporal coarsening either increased the final-time error or violated the CFL constraint

of the method (i.e., the optimal stable choice of Δt was used for each algorithm and for each Δx).

Remark 1. Note that, although the solutions under consideration in this section are periodic, the FC solver does not take undue advantage of this fact—since the solution is not generally periodic on the subdomain blocks used by the FC method (see Sec. III and, in particular, Fig. 2 and its caption). As discussed in Sec. I, Eqs. (1) and (2) are based on periodic Fourier analysis and, thus, provide heuristic predictions of the behavior of numerical methods in regions at a distance from boundaries; the purpose of the one-dimensional examples is to demonstrate the significance of this analysis. For this reason, the solution of boundary-value problems for PDEs is deferred until the two-dimensional examples presented in Secs. VI and VII.

A. Case study I: Linear advection equation

As a first example, consider the solution to the linear advection Eq. (6) on the periodic interval $[-100, 100]$ with initial conditions, shown in the top portion of Fig. 3, given by a smoothed sine wave packet approximately 150 wavelengths across containing approximately 100 full-amplitude waves:

$$u(x, 0) = \exp\left(- (0.0158x)^{10}\right) \sin(2\pi x).$$

This configuration models, for example, the acoustic wave generated by a low-pressure piston transducer in an inviscid

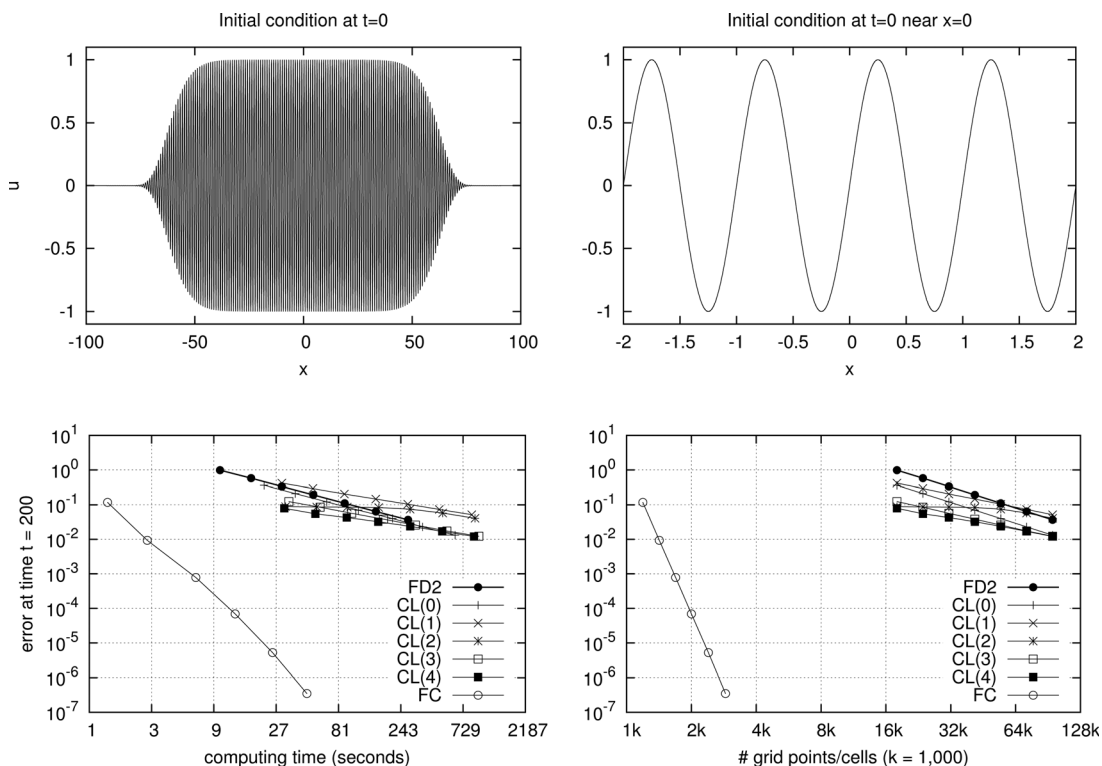


FIG. 3. (Top left) Initial conditions for the velocity u in the one-dimensional advection equation example presented in Sec. IV A. (Top right) Close-up view of the initial condition near $x=0$. (Bottom left and right) Errors vs computing time and discretization size, respectively at time $t = 200$ for the advection equation example presented in Sec. IV A. The graphs indicate that the FC and CL(4) computing-time curves reach the 10^{-2} -accuracy level at about 2.7 s (using 1400 discretization points) and at 1300 s (using 116 000 cells), respectively.

fluid. In the comparisons that follow, the advection equation was solved until time $t=200$, at which point the wave packet should wrap periodically back to its starting position, traveling 200 times the fundamental wavelength—the distance the second or third harmonic might travel in a physically relevant HIFU simulation. For the comparison, the 1D linear advection example distributed with CLAWPACK version 4.3 was modified in four ways. First, the inhomogeneity in the advection coefficient was removed so that the equation solved is (6). Second, the spatial domain was extended from the interval $[-2, 2]$ to $[-100, 100]$. Third, the initial condition was changed from a Gaussian bump to the windowed sine function shown in Fig. 3. Finally, the final time was set to $t=200$.

Figure 3 demonstrates the ability of the FC solver to advect the pulse over long distances with a relatively coarse mesh. For example, the graphs show that the FC solver attains 1% accuracy with approximately 1400 discretization points, requiring approximately 2.7 s of computing time. Extrapolation of the CL(4) (CLAWPACK with the monotonized centered flux limiter) results shows, by comparison, that the CL(4) solver would require approximately 116 000 grid points and 1300 s to achieve the same accuracy. Even larger improvement factors may be expected in higher dimensions. In two dimensions, for example, a simple scaling argument based on the one-dimensional experiments mentioned above suggests that, to produce the 1%-accurate solution in a comparable two-dimensional problem, the CLAWPACK solver would require $116\,000/1400 \approx 83$ times as many grid lines as the FC solver, each of which would entail $1300/2.7 \approx 480$ times as much computing time—so that the two-dimensional CL(4) solution would be $83 \times 480 \approx 40\,000$ times more costly, in terms of computing times, than the corresponding two-dimensional FC solution. In three-dimensions, the corresponding improvement factor would be $83^2 \times 480 \approx 3\,300\,000$. Although these estimates are admittedly rough, they do suggest many orders of magnitude differences in computing times for 3D advection-dominated problems. The use of a high-order FC solver for such a problem, instead of a lower-order method, could reduce to several seconds the computation of a solution that would otherwise require several months.

B. Case study II: Nonlinear viscous Burgers equation

As a second one-dimensional example, consider the viscous Burgers equation model of nonlinear acoustics, which can be used to model acoustic waves produced by high-intensity piston sources in fluids with quadratic attenuation laws. In particular, consider the initial value problem

$$\begin{aligned} u_t + (1 + u)u_x &= \varepsilon u_{xx}, & x \in [-10, 40], \\ u(x, 0) &= u_0 \exp(-(wx)^{10}) \sin(2\pi x) \end{aligned} \quad (9)$$

for the viscous Burgers equation with periodic boundary conditions. Here, the particle velocity u has been normalized with respect to the small signal sound speed and the time and space variables have been normalized with respect to the period and wavelength of the source waveform respectively. Although it is possible to introduce a retarded time frame to make the Burgers equation less numerically challenging, as

is discussed in Remark 2 below, this cannot be done for general full-wave problems. The form employed here allows for the assessment of the FC algorithm's ability to capture the large local wave speed $1 + u \gg u$ present in full-wave problems.

The initial values in Eq. (9) define the smoothly windowed sine wave depicted in the upper left portion of Fig. 4. Over time, this initial pulse advects towards the positive x -direction, steepening appreciably—as can be seen in the upper right portion of Fig. 4. The dimensionless parameters in the equation were selected to be

$$\begin{aligned} u_0 &= 7.96 \times 10^{-3}, & \varepsilon &= 1.5 \times 10^{-5}, \\ w &= 0.179. \end{aligned} \quad (10)$$

The implementation of the FC solver for Eq. (9) is analogous to the solver for the linear advection Eq. (6). The PDE is first discretized into a nonlinear system of ODEs:

$$\frac{dv}{dt} = -f(v) + \varepsilon E v, \quad (f(v))_j = (1 + v_j)(Dv)_j, \quad (11)$$

where D is the (block-by-block) FC derivative operator described in Sec. III and E is the second-order counterpart. As before, the system of ODEs is marched forward in time by a standard integration algorithm (fourth-order Runge-Kutta in this case).

For the present example, the solvers were compared by advancing the solution to final time $t=30$ (at which point the initial signal has traveled approximately 30 times the fundamental wavelength). The problem has been designed so that the solution to Eq. (9) vanishes on the boundary for all times in the range $0 \leq t \leq 30$, in order to allow the boundary conditions to be treated as either periodic (used by the FD and FC solvers) or vanishing (used by the CLAWPACK solver). This allowed CLAWPACK's viscous Burgers solver to be used with minimal modifications; see also Remark 1.

The relative maximum error at the final time was then approximated by comparison with a converged solution produced by the FC solver using 800 points per fundamental wavelength. Comparisons of this solution with the solution on an even finer grid indicate that the maximum error (relative to the initial amplitude u_0) is less than 10^{-5} . Although Eq. (9) was not solved to this level of accuracy with the CLAWPACK solver, the reference solution was further verified to within a relative maximum error of 5×10^{-5} by comparison with a CLAWPACK solution to the time-retarded version of (9), which, in view of its small advection coefficient, is much less challenging than Eq. (9) itself; see Remark 2. It was found by a convergence study that the use of 12 000 points per fundamental wavelength was required to obtain a CLAWPACK solution to the time-retarded problem with a 5×10^{-5} accuracy level.

The lower portion of Fig. 4 summarizes the results of the comparison. As in the case of the linear advection equation, it is apparent that the high-order FC method can have significant advantages in HIFU-related problems, and, as discussed in Sec. IV A, these advantages would be more pronounced in higher spatial dimensions.

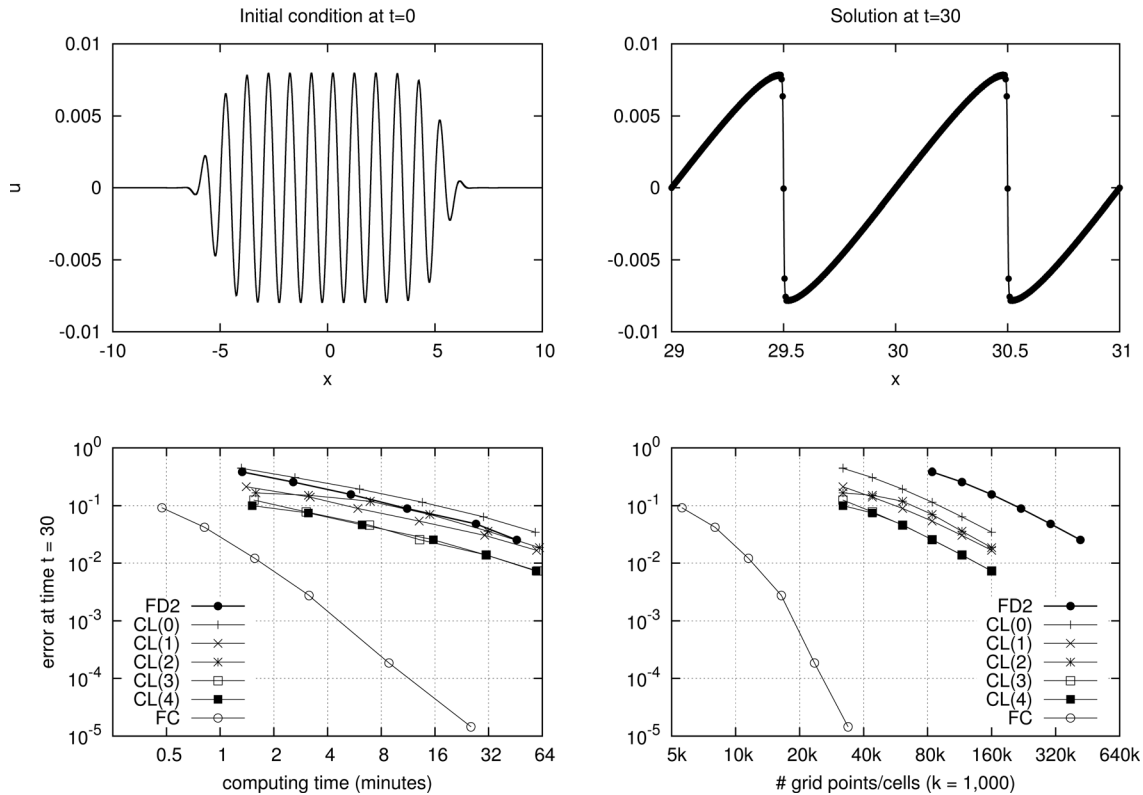


FIG. 4. (Top left) Initial conditions for the velocity u in the one-dimensional (non-linear) Burgers equation example presented in Sec. IV. (Top right) Comparison of the corresponding highly resolved, converged FC solution at time $t = 30$ (solid line, obtained numerically, as indicated in the text, with a relative accuracy of less than 10^{-3}) and the corresponding FC solution (shown in black dots). The relative maximum error in the latter FC solution is 1.2×10^{-2} . (Bottom left and right) Errors vs computing time and discretization size respectively at time $t = 30$ for the Burgers equation example presented in Sec. IV. The FC and CL(4) computing-time curves reach the 10^{-2} -accuracy level at 106 s (using 12 100 discretization points) and at 2700 s (using 138 000 cells), respectively.

Remark 2. In the present simplified context a moving frame of reference $x' = x - t$ can be used to transform the PDE (9) into the classical viscous Burgers equation $u_t + uu_x = \varepsilon u_{xx}$ —greatly reducing the wave speed to the value $u \ll 1 + u$, and thus, essentially stopping wave propagation. This procedure effectively eliminates the computationally challenging dispersion errors that originate both from the approximate spatial derivative (which is multiplied by the wave speed) and from the approximate time-derivative (by greatly reducing time variation). This velocity-reducing change of variables is the reason it was possible to produce the accuracy level 5×10^{-5} CLAWPACK solution to the problem considered in this section. Use of such a moving frame of reference has purposely been avoided for the one-dimensional comparison in this section. The purpose of this one-dimensional study is to demonstrate the performance of various solvers in a simple context that, nonetheless, captures the main difficulties inherent in general nonlinear acoustic simulations—for which use of moving frame of reference cannot slow propagation of all existing waves simultaneously; e.g., see Fig. 12(b). Note that, even the acoustic fields displayed in Figs. 7–11—for which use was made of a time-dependent change of variables to track the acoustic signal in order to reduce the size of the computational domain—exhibit significant velocity gradients in the x -direction [see Figs. 6–12(a)]. Unlike the z -component of the convection velocity in the moving frame of reference, the x -component of the convection velocity in these cases

cannot be made to be small by a frame of reference moving in the z direction, and thus the potential for significant dispersion errors remains. In any case, for configurations for which propagation and scattering occur [as demonstrated, for example, in Fig. 12(b)], use of a moving frame of reference actually generally leads to even *faster* waves speeds (e.g., for backscattered waves) than those occurring in the stationary reference frame: use of a moving frame of reference under such circumstances gives rise to a requirement of even finer resolutions than would be necessary if a moving reference frame were not used.

V. FC SOLVERS IN MULTIDIMENSIONAL SPACE

The generalization of the one-dimensional techniques described in Sec. III to multiple spatial dimensions require relatively simple generalizations of Algorithms 1 and 2 to compute the necessary spatial derivatives, and standard techniques for time evolution. Two key differences between one-dimensional problems and multidimensional problems do exist, however, namely, (1) multidimensional domains can have more complex boundaries and (2) efficient parallelization becomes increasingly important. The present section treats both of these topics in turn.

A. Overlapping curvilinear patches

Complex geometries are accommodated in the context of the FC solver by means of the overlapping grid approach

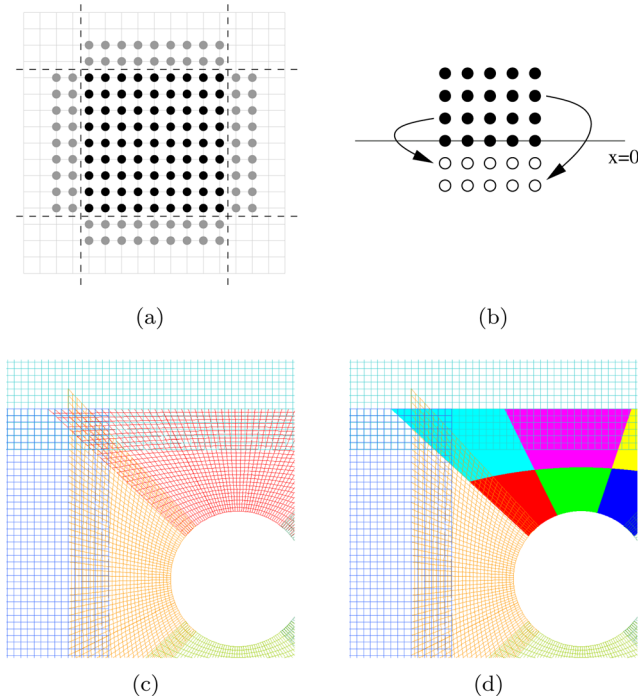


FIG. 5. (Color online) (a) An illustration of the domain decomposition used for a parallel FC solver. The dashed lines represent processor boundaries. In order to update the solution values in the central subdomain (represented by dark circles), the associated processor receives a thin layer of shared values (represented by light circles) from each of four neighboring processors. (b) The treatment of the mirror points across the line of symmetry $x=0$. Values are copied from the mirror image values (on the same processor) above the axis according to the symmetry relations (20)–(22). Additionally, the vertical component of the velocity is set to 0 on the axis as required by (22). (c) A portion of a grid covered by overlapping curvilinear patches. (d) As indicated by solid filled regions, one of the patches subdivided into sub-patches for distribution in a parallel computing environment.

illustrated in Fig. 5(c). (A comprehensive treatment of the history and essential properties of the overlapping grid methodology can be found in Ref. 39.) In this approach (whose description is specialized here to the two-dimensional cases considered throughout this paper) the geometry is divided into logically rectangular curvilinear patches, each one of which is the image of the unit square $[0, 1] \times [0, 1]$ under a certain smooth mapping $(x, y) = [x(\zeta, \eta), y(\zeta, \eta)]$, in such a way that physical boundaries are described as the union of a subset of all patch boundaries. As shown in Fig. 5(c), the patches are designed to overlap neighboring patches by a few grid points. This patch domain decomposition is incorporated into the time-stepping algorithm as follows: at each time step, the solution is first advanced independently on each patch; then the solution values at points near the boundary of a patch that also overlap a neighboring patch are replaced by interpolating the value from solution values on the neighboring patch by means of high-order polynomial interpolation. This interpolation procedure ensures that information is adequately exchanged across the computational domain, and that convergence to the exact solution occurs as mesh sizes and time steps are refined; see Refs. 27 and 39 for additional details.

Spatial derivatives on each curvilinear patch are produced by means of application of the FC differentiation algorithm together with the chain rule relations

$$\frac{\partial}{\partial x} = \frac{\partial \zeta}{\partial x} \frac{\partial}{\partial \zeta} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta} \quad \text{and} \quad \frac{\partial}{\partial y} = \frac{\partial \zeta}{\partial y} \frac{\partial}{\partial \zeta} + \frac{\partial \eta}{\partial y} \frac{\partial}{\partial \eta}. \quad (12)$$

If the derivatives of ζ and η with respect to x and y are not known analytically, they may be approximated by means of the FC method itself. For the examples presented in this paper analytical derivatives were utilized, together with wide 13×13 polynomial interpolation stencils across patch boundaries.

B. Parallelization techniques

In order to divide the work required to produce a numerical solution evenly among a number of computational nodes in a distributed parallel computer, the multidimensional analog of the one-dimensional block domain decomposition described in Sec. III A can be used. In this approach, one proceeds by splitting the reference domain of each curvilinear patch of the computational domain into an array of approximately equally sized rectangular subdomains, assigning the computation on each subdomain to one processor. Where the subdomains abut, the processors share thin layers of fringe values with their neighbors, as illustrated in Fig. 5(a). The dashed lines in the figure denote subdomain boundaries, across which the processors communicate, while the black circles represent the (generally large number of) grid points that are assigned to one particular processor. As in the one-dimensional case, to advance the solution, this processor uses values from the fringe region, which are denoted by light-gray circles. Although the processor uses these values in its computation, it does not update them, but instead receives updated function values from its neighboring processors at every time step. In the simulations presented in this paper, 4-points fringes were used. (For rendering simplicity, the illustration shows small numbers of black circles and fringes that are only 2 points wide.)

In order to retain spectral-like accuracy across subdomain interfaces, it is necessary to introduce a modification to the periodic extension algorithm described in Sec. II; we call this modified algorithm the *biased-order extension method*. The goal of the biased-order extension approach is to allow the number of sampled values d in the extension construction to vary depending on the type of boundary encountered. For boundary points that lie inside the domain (arising from the domain decomposition either from the overlapping grid decomposition or from the block decomposition of a patch), the algorithm uses the value $d=12$ instead: since at non-physical boundaries the algorithm uses 4 points from a neighboring processor, stability can be ensured even when polynomial interpolations of such high orders are used. At physical boundaries, in contrast, only boundary values at a single point (the boundary point) are given; for such cases stability is ensured by the choice $d=5$. The ability to enforce interface conditions of very high order of accuracy at interior boundaries gives rise to both excellent parallelization, and dispersive properties that are very close to those resulting from Fourier expansions of periodic functions. We refer the reader to Ref. 27 for details on the biased-order

extension along with an analysis of the computational efficiency of the method—which exhibits essentially linear scaling for a wide range of configurations—as well as one- and two-dimensional examples which demonstrate its high-order/spectral accuracy.

VI. TWO-DIMENSIONAL APPLICATION: COMPARISON OF THE NAVIER-STOKES AND KZK MODELS

Based on the one-dimensional examples presented in Sec. IV, it should be expected that the high-order FC solver will be much more efficient than standard low-order solvers in treating full-wave models for HIFU-related problems, thus rendering possible previously infeasible full-wave solutions. The present section demonstrates a particular application of the efficient full-wave solver presented in this paper: comparative studies of full-wave and simplified HIFU models. Although simplified models (e.g., those based on the Rayleigh integral or paraxial approximation) can significantly decrease the computational complexity of the numerical simulation of an acoustic field, it is not always clear when the solution of the reduced model should be trusted. A high-order solver with the ability to efficiently produce solutions to a full-wave model, therefore, can be a valuable tool in exploring the validity of a simplified model. The concept is illustrated here using the well-known Navier-Stokes and a KZK models for focusing sound beams. Although the results presented here are not surprising—a common rule of thumb for the KZK equation is that it should not be considered accurate in configurations where acoustic waves approach the axis at angles of more than about 20° —they do demonstrate a particular application of the full-wave solver.

A. Model and equations

1. Ultrasound transducer model

The configuration considered in this example is illustrated in Fig. 6. The source is taken to be an infinite focusing cylindrical transducer parallel to the y -axis, whose cross-section in the xz -plane is a circular arc. The transducer is parametrized by the aperture radius a , and the radius of curvature R . The focal length F , which is the distance from the mouth of the aperture to the geometrical focus is given by $F^2 = R^2 - a^2$. For both the Navier-Stokes and KZK solvers,

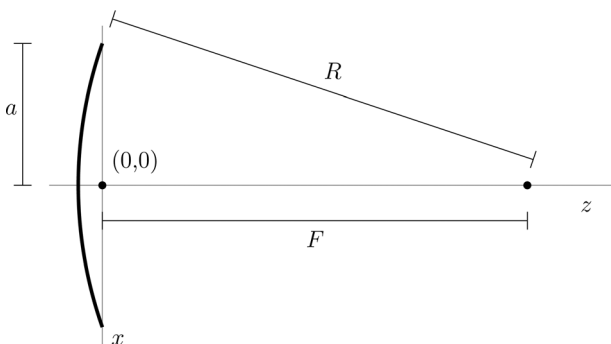


FIG. 6. A circular arc transducer with focal length F , aperture radius a and radius of curvature $R = \sqrt{a^2 + F^2}$.

the transducer surface is migrated to a boundary condition at the mouth of the source, i.e., the plane $z = 0$, as described in Secs. VIC2 and VIB.

2. Nonlinear acoustics Navier-Stokes equations

The Navier-Stokes model used for the present demonstration is expressed in the form [Ref. 11, Eqs. (1) and (2)]

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0, \quad (13)$$

$$\rho \frac{D\mathbf{u}}{Dt} + \nabla P = \mu \nabla^2 \mathbf{u} + \frac{1}{3} \mu \nabla (\nabla \cdot \mathbf{u}), \quad (14)$$

where ρ is the density, \mathbf{u} is the fluid velocity vector, P is the total pressure, μ is the coefficient of shear viscosity and $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$ is the material or convective derivative. The system is closed with the equation of state

$$p = c_0^2 \tilde{\rho} + \frac{c_0^2 B}{\rho_0 2A} \tilde{\rho}^2, \quad (15)$$

where $p = P - P_0$ is the acoustic pressure and $\tilde{\rho} = \rho - \rho_0$ is the acoustic fluctuation in density. Here P_0 and ρ_0 denote the ambient values of the pressure and density respectively, c_0 is the small-signal sound speed, and the quotient B/A is the parameter of nonlinearity of the medium (see, e.g., Ref. 40). For simplicity of presentation, thermal conductivity and bulk viscosity have been neglected in the present demonstration, but, as discussed in Sec. VIII, incorporation of these effects poses no difficulties.

3. KZK equation

The Khokhlov-Zabolotskaya-Kuznetsov (KZK) equation is a model of finite-amplitude wave propagation that is appropriate for sound beams for which the paraxial approximation is sufficiently accurate. A detailed derivation of the KZK equation is presented in Ref. 11. Briefly, Eqs. (13)–(15) are expanded in terms of the acoustic fluctuations and only linear and quadratic terms are retained. Combining these equations under the assumption that sound propagates in accordance with the paraxial approximation, the KZK equation

$$\frac{\partial p}{\partial z} = \frac{c_0}{2} \int_{-\infty}^{\tau} \nabla_{\perp}^2 p d\tau' + \frac{\beta}{2\rho_0 c_0^3} \frac{\partial p^2}{\partial \tau} + \frac{\delta}{2c_0^3} \frac{\partial^2 p}{\partial \tau^2} \quad (16)$$

results. [Equation (16) follows, by integration with respect to τ , from Ref. 11, Eq. (65)]. Here we utilize the notations used in Ref. 11: $\beta = 1 + B/(2A)$ denotes the coefficient of nonlinearity,

$$\delta = \frac{4\mu}{3\rho_0} \quad (17)$$

is the diffusivity of sound, $\tau = t - z/c_0$ is the retarded time variable, and the operator ∇_{\perp}^2 is the Laplacian perpendicular to the direction of propagation; for Cartesian coordinates and propagation in the z direction ∇_{\perp}^2 is given by $\nabla_{\perp}^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$.

B. KZK solver

For this example, the KZK Eq. (16) was solved by means of the KZK Texas code.⁴¹ The two dimensional computational domain for the KZK simulations is the rectangle $(x, z) \in [0, 3a] \times [0, 2F]$ where a is the aperture radius of the transducer source and F is the focal length. The KZK Texas code only solves for $x \geq 0$ and uses a reflecting boundary condition at $x=0$. At the computational boundary $x=3a$, the boundary condition is pressure-release, which produces a large reflection (Ref. 42, Sec. III C). The choice $x=3a$ is sufficiently large, however, that this reflection arrives too late to affect the results. As stated in the previous section, the curved surface of the true source is mapped to the plane $z=0$ by retarding the phase of the waveforms according to Ref. 24, where the approximate values $p_s = \rho_0 c_0 u_s$ were used [cf. Eq. (25)], with u_s defined by Eq. (23).

C. Navier-Stokes solver

The Navier-Stokes solver was implemented using the FC approach described in Sec. V. For added efficiency, the solver was specialized to exploit certain properties of the solution, namely reflection symmetry and the limited temporal duration of the waveform implicit in Eq. (23). (Note, however, that in Sec. VII, a full-wave FC solver is presented that does not make use of these specializations.) The implementation of the symmetry condition described in Sec. VIC 1 provides a decrease in computational costs of approximately 50% by restricting the computation to the half-plane $x \geq 0$. In view of the aforementioned finite temporal duration of the pulse, the signal from the transducer can be effectively localized in space by using a temporal windowing function as described in Sec. VIC 2. (Notice that simplification of the problem through use of such temporal windowing functions is not a requirement of the FC method: for example, the test cases presented in Sec. VII do not use any such temporal windowing.) While the entire domain is over 100λ in length [where, in this context, the fundamental wavelength λ is defined according to Eq. (30) below], the signal stays essentially bounded inside a narrow strip as it progresses into the domain. To take advantage of this localization, a moving frame of reference is introduced so that the horizontal coordinate becomes $z = z' + c_0 t$. The resulting equations are transformed as

$$\frac{D\rho}{Dt} - c_0 \frac{\partial \rho}{\partial z'} + \rho \nabla \cdot \mathbf{u} = 0, \quad (18)$$

$$\rho \frac{D\mathbf{u}}{Dt} - c_0 \rho \frac{\partial \mathbf{u}}{\partial z'} + \nabla P = \mu \nabla^2 \mathbf{u} + \frac{1}{3} \mu \nabla (\nabla \cdot \mathbf{u}), \quad (19)$$

where the spatial derivatives now refer to derivatives in the transformed coordinates (x, y, z') .

1. Reflection symmetry

The symmetry of the problem across the z axis implies the following properties of the solution:

$$\tilde{\rho}(x, z, t) = \tilde{\rho}(-x, z, t), \quad (20)$$

$$u_x(x, z, t) = -u_x(-x, z, t), \quad (21)$$

and

$$u_z(x, z, t) = u_z(-x, z, t), \quad (22)$$

where u_x and u_z are the velocity components in the x and z directions, respectively. Note, in particular, that $u_x(0, z, t) \equiv 0$. To take advantage of the symmetry with respect to the z axis, nearby mirror points are used; see Fig. 5(b). A layer of these mirror points are added to each subdomain block adjacent to the z axis. They are used in a similar manner as fringe points, but rather than being updated from a neighboring subdomain block, they are updated from the block's own solution values above the z axis by means of Eqs. (20)–(22). For all points which lie on the line $x=0$, the condition $u_x=0$ is enforced.

2. Spatial and temporal windowing

a. Stage I: Near the transducer. The simulation is initiated with a rectangular computational domain whose left edge lies on the line $z=0$. Thus, the transducer lies just outside the domain, as illustrated in Fig. 7. Equations (13)–(15) are solved within this domain until the signal is essentially centered within the computational window. Spatial derivatives are obtained by means of the FC method and time stepping is produced by means of a standard fourth-order Adams-Bashforth time-marching scheme. A sinusoidal source velocity waveform u_s (smoothly windowed in time to reduce numerical artifacts,⁴¹

$$u_s(t) = u_0 \exp(-(2f_0 t / n_{\text{cyc}})^{2m_{\text{env}}}) \sin(2\pi f_0 t), \quad (23)$$

is assumed at the transducer surface, where u_0 is the source velocity amplitude and f_0 is the frequency. The parameter n_{cyc} determines the number of cycles in the pulse and m_{env} controls the taper with $m_{\text{env}}=1$ producing a Gaussian envelope. For these simulations the values $n_{\text{cyc}}=8$ and $m_{\text{env}}=5$ were used, which give rise to an 8 cycle pulse with a sharp transition and thus produce a finite-duration pulse as required by the Texas KZK code—while containing sufficiently many cycles that time-harmonic behavior is observed

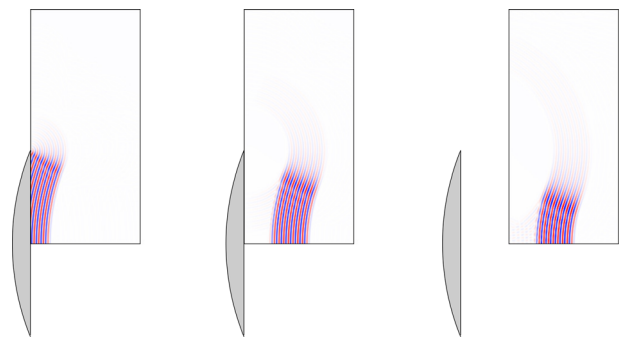


FIG. 7. (Color online) Stages of the FC focused transducer solver. In Stage I, the computational window is fixed in space next to the transducer (left image). The computation proceeds in Stage I until the signal is centered within the computational window (center image). At this point, the solver enters Stage II computation and the computational window begins moving away from the transducer (right image).

in the focal region. As the purpose of this example is a comparison with the Texas KZK solution, a consistent transducer model is required. In the Texas KZK code, the pressure waveform on the plane $z=0$ is time shifted by a parabolic time delay:

$$p(x, 0, t) = p_s(t + x^2/(2c_0F))g(x/a), \quad (24)$$

where p_s is the source pressure signal and g is the step function taking the value 1 for $|x/a| < 1$ and 0 elsewhere [see Ref. 42, Eq. (2.30)]. Although more accurate transducer models are possible by using curvilinear grids (similar to the construction in Fig. 5), the primary point of interest here lies in producing a model for the Navier-Stokes equations which is consistent with (24) so that adequate comparisons of FC results and KZK results can be made. Excluding the nonlinear and viscous terms in (13)–(15) gives the relations⁴³

$$p = c_0^2 \tilde{\rho} = \rho_0 c_0 \mathbf{u} \cdot \boldsymbol{\nu}, \quad (25)$$

between the density, pressure and normal velocity at $z=0$, where

$$\boldsymbol{\nu} = (\nu_x, \nu_z)^T = \frac{1}{\sqrt{x^2 + F^2}} (-x, F)^T \quad (26)$$

is the unit vector normal to the transducer pointing towards the focus. Based on (23)–(25), the boundary conditions

$$\tilde{\rho}(x, 0, t) = \frac{\rho_0}{c_0} u_s(t + x^2/(2c_0F))g(x/a), \quad (27)$$

$$u_x(x, 0, t) = u_s(t + x^2/(2c_0F))g(x/a)\nu_x, \quad (28)$$

$$u_z(x, 0, t) = u_s(t + 0.5c_0^{-1}x^2/F)g(x)\nu_z, \quad (29)$$

follow. On all other boundary points the conditions $\tilde{\rho} = \mathbf{u} = 0$ are used. The simulation begins at time $t < 0$ sufficiently small, so that, at that time, the right-hand-sides of (27)–(29) are zero to machine precision for all x .

Remark 3. It is tempting to initialize the FC solver with the same initial conditions as for the KZK solver, thereby removing the need for Stage I of the solver. However, the dramatic (and non-physical) drop in pressure and velocity at the edge of this initial signal implies the existence of high-frequency spatial signal components traveling in the radial direction. Since the KZK equation is relatively insensitive to radially directed waves, this does not cause any visible side-effects in the KZK solver. However, these high-frequency waves are apparent in the output of the Navier-Stokes solver. To avoid this the FC solver is initialized with motion-free initial conditions, and the transducer signal is directly evolved across the line $z=0$ and into the computational domain before introducing the moving reference frame, as described in the previous section.

b. Stage II: Away from the transducer. Once the signal has progressed into the domain and away from the transducer, a moving frame of reference is introduced and Eqs. (18) and (19) are solved. This allows the computational window to move along with the wave front as it progresses

into the domain, as illustrated in Fig. 7. For time-marching, a standard fourth-order Runge-Kutta method is used, and the conditions $\tilde{\rho} = \mathbf{u} = 0$ are enforced on all window boundaries. For simplicity, reflected acoustic waves are prevented from interfering with the solution in the FC solver as they are in the KZK solver: the computational window is made sufficiently large in the x direction that reflected waves remain outside the domain of interest. (See, Ref. 27, however, for examples of FC-based solvers that utilize absorbing boundary conditions to prevent strong reflections from computational boundaries.)

c. A note on time integration. Different time integration methods are used in Stage I and Stage II. In the first stage, the boundary conditions are time-dependent. This is known to be problematic for high-order Runge-Kutta methods:^{44,45} if Runge-Kutta methods are used, special care must be taken to apply the correct boundary conditions at intermediate steps. The Adams-Bashforth integration scheme does not give rise to such complications. Although the Adams-Bashforth method requires the solution at three previous time steps, the initial conditions are simple so that this presents no difficulty at the initial step. Upon entering the second stage, the boundary conditions no longer vary in time. For this stage the Runge-Kutta method is used, as with this choice the simulation can be restarted more easily—should that prove necessary (in a parallel computation in which there is a computing time limit, say)—since the Runge-Kutta solver can be initialized from the solution at a given time, rather than requiring the solution at four consecutive time steps.

D. Results of Navier-Stokes/KZK comparisons

This section presents comparisons between the numerical solutions to the KZK and Navier-Stokes model equations produced by the solvers described in Secs. VIB and VIC, respectively. The acoustic properties of the medium were chosen to be close to those of soft tissue: $\rho_0 = 1000 \text{ kg/m}^3$, $c_0 = 1540 \text{ m/s}$, $\beta = 4.8$ and $\delta = 6.4117 \times 10^{-4} \text{ m}^2/\text{s}$. The transducer was driven at a frequency $f_0 = 1.1 \text{ MHz}$, producing a signal with fundamental wavelength

$$\lambda = c_0/f_0 = 1.4 \text{ mm}. \quad (30)$$

For these comparisons, the focal length of the circular arc transducer was fixed at $F = 50\lambda$, and four different transducer aperture radii were used: $a = 5\lambda$, 10λ , 20λ and 30λ . For the KZK solver the transducer source pressure was set to $p_0 = 1 \text{ MPa}$, which corresponds to a velocity source in the Navier-Stokes equations of $u_0 = p_0/(\rho_0 c_0) \approx 0.64935 \text{ m/s}$. In the accompanying figures, the label “KZK” refers to the solution produced by the KZK solver and “DNS” refers to the solution produced by direct numerical simulation of the full Navier-Stokes equations by means of the FC solver. The pressures produced by the two solvers are compared using (25) to approximate the pressure in the Navier-Stokes equation. According to Eq. (15) this approximation is accurate—since, in this case, $|\tilde{\rho}| \ll \rho_0$.

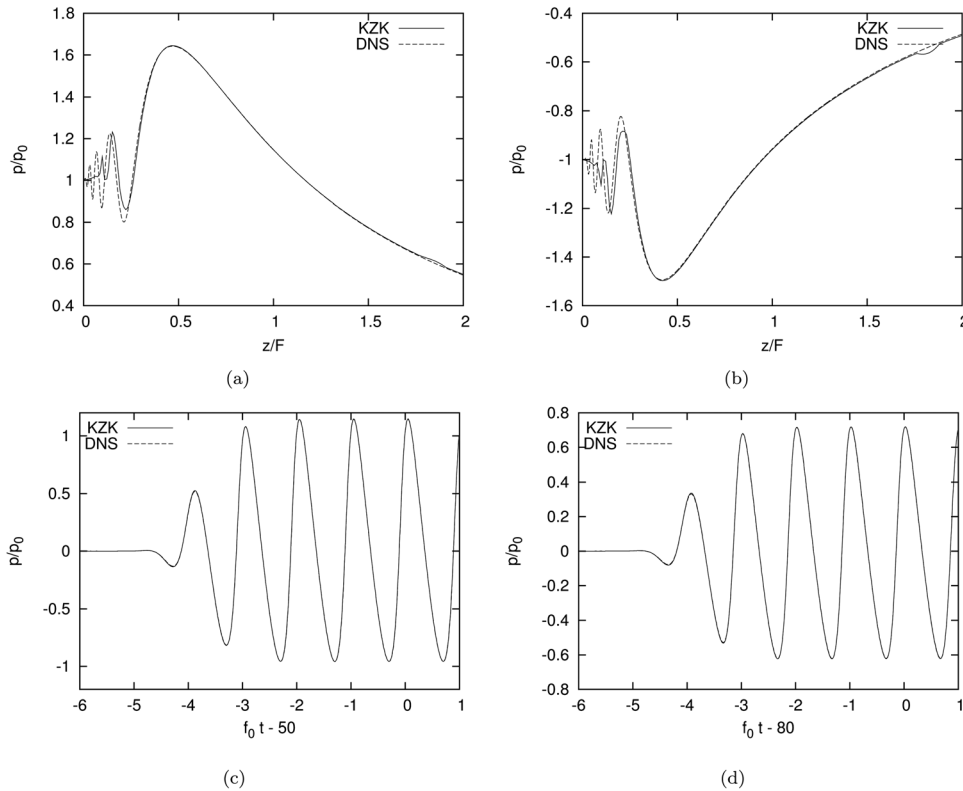


FIG. 8. Comparisons of the KZK and Navier-Stokes (DNS) solutions for the 5λ aperture radius transducer. (a) Peak positive on-axis pressure. (b) Peak negative on-axis pressure. Axial pressure waveform (c) 50λ and (d) 80λ from the transducer.

For the Navier-Stokes simulations, a spatial grid discretization of approximately $\Delta x = \Delta z = \lambda/21$ and a time discretization satisfying $c_0\Delta t \approx \Delta x/40$ in Stage I and $c_0\Delta t \approx \Delta x/10$ in Stage II were used; in view of comparisons with the solution with $\Delta x = \Delta z = \lambda/15$, it was found that the latter parameters produce solutions with an error of less than 1%, and thus, we conclude, so do the former (see Remark 4). For the KZK simulations, the time discretization varied in the range of 500 points per cycle to 2000 points per cycle—that is, a sampling rate of 550 MHz to 2.2 GHz. The nominal step size in the marching direction is $F/500$. The KZK code automatically reduces this if the nonlinearity becomes too strong. In the x -axis the discretization was chosen so that further refinement did not significantly improve the solution and varied in the range $\lambda/2.5 \leq \Delta x \leq \lambda/40$.

For the 5λ transducer, excellent agreement between the KZK and FC solutions was observed. The peak positive and negative pressures on the z -axis are plotted for both solvers in Figs. 8(a) and 8(b) and the temporal waveforms sampled at two on-axis points are plotted in Figs. 8(c) and 8(d), both of which show almost perfect agreement except in the near field where, as is known, the KZK approximation is inaccurate. In order to compare the full acoustic field, Fig. 9 displays the amplitudes of the first five harmonics generated by the transducer. Although the fundamental shows good agreement away from the near field, some differences are noticeable in the higher harmonics. (Note, for example, the differences on the z -axis near the 0.5 tick mark.)

As the transducers are taken to be more strongly focused—thereby deviating to a larger degree from a unidirectional sound beam—a departure of the KZK approximation from the converged FC Navier-Stokes solution is

observed. Figures 10(a) and 10(b), for example, displays the on-axis peak positive pressures for the 10λ and 20λ aperture radius transducers. Although the 10λ case also shows good agreement, in the 20λ case the two solvers have started to differ noticeably near the acoustic focus. The peak positive pressure plot for the 30λ transducer is shown in Fig. 10(c); the solvers in this case predict significantly different locations for the acoustic focus and for the peak pressure at the focus. The waveforms at the geometric focus are plotted in Fig. 10(d). The harmonic plots for the 30λ transducer in Fig. 11 also show significant differences throughout the domain.

Remark 4. In view of the high-order convergence of the numerical solvers used, evidenced, e.g., by the results presented in the bottom right portion of Fig. 4, a small increase in the discretization density suffices to produce an adequate basis for comparison and error determination.

VII. TWO-DIMENSIONAL APPLICATION: COMPLEX GEOMETRIES AND FULL-DOMAIN SIMULATIONS

This section explores a second application of the new high-order FC approach: the full-domain solution of the nonlinear acoustic field generated by a transducer both in a homogeneous medium and in a medium containing an array of rigid cylinders at whose boundaries the velocity is assumed to vanish. In the latter case the medium contains five circular scatterers centered at the points of (x, z) -coordinates $(-21.4\lambda, 20.2\lambda)$, $(-11.2\lambda, 19.4\lambda)$, $(2.3\lambda, 18.1\lambda)$, $(8.1\lambda, 19.6\lambda)$ and $(20.4\lambda, 20.9\lambda)$ (see Fig. 6), and with radii 1.1λ , 1.5λ , 1.4λ , 1.3λ , and 1.2λ , respectively. The circular inclusions are

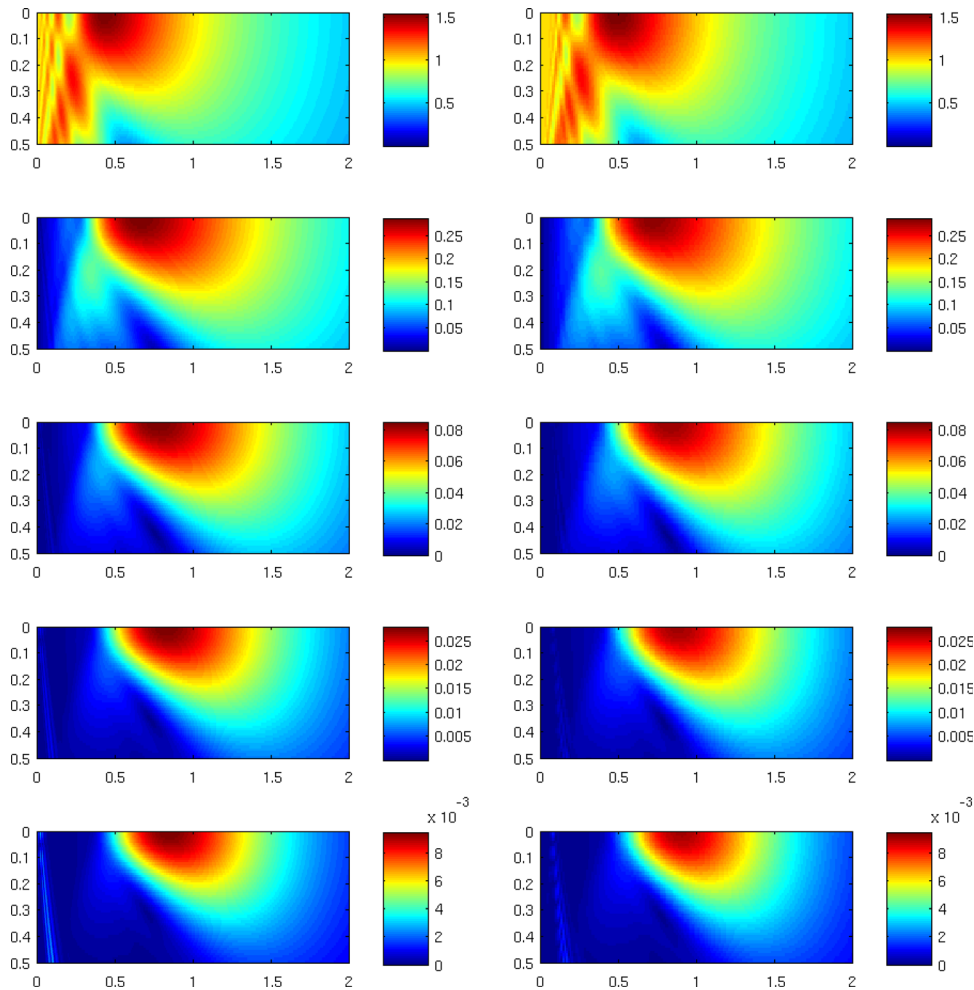


FIG. 9. (Color online) Comparisons of the amplitudes of the first five harmonics (top to bottom) for the 5λ aperture radius transducer. (Left) Results obtained from the Navier-Stokes FC simulation. (Right) Results obtained from the KZK model. Horizontal axes are measured in focal lengths. Vertical axes are measured in aperture radii.

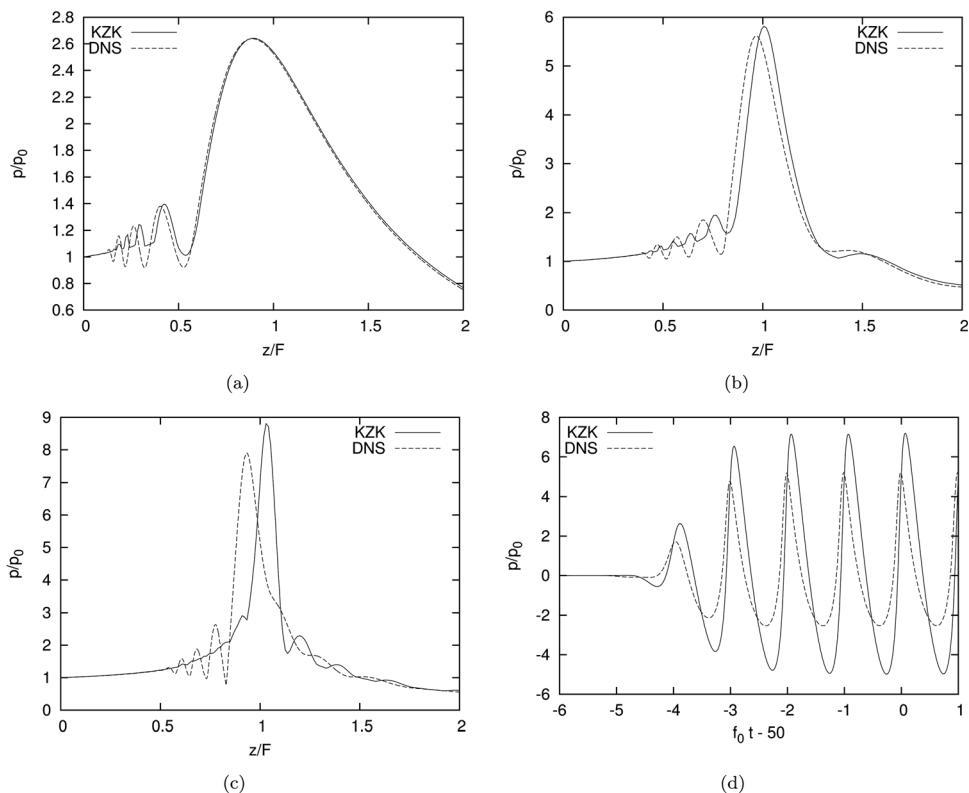


FIG. 10. Additional comparisons of the KZK and Navier-Stokes (DNS) solutions for transducers with various aperture radii. Peak positive on-axis pressure for the (a) 10λ , (b) 20λ , and (c) 30λ aperture radius transducers. (d) Axial pressure waveform 50λ from the transducer for the 30λ transducer.

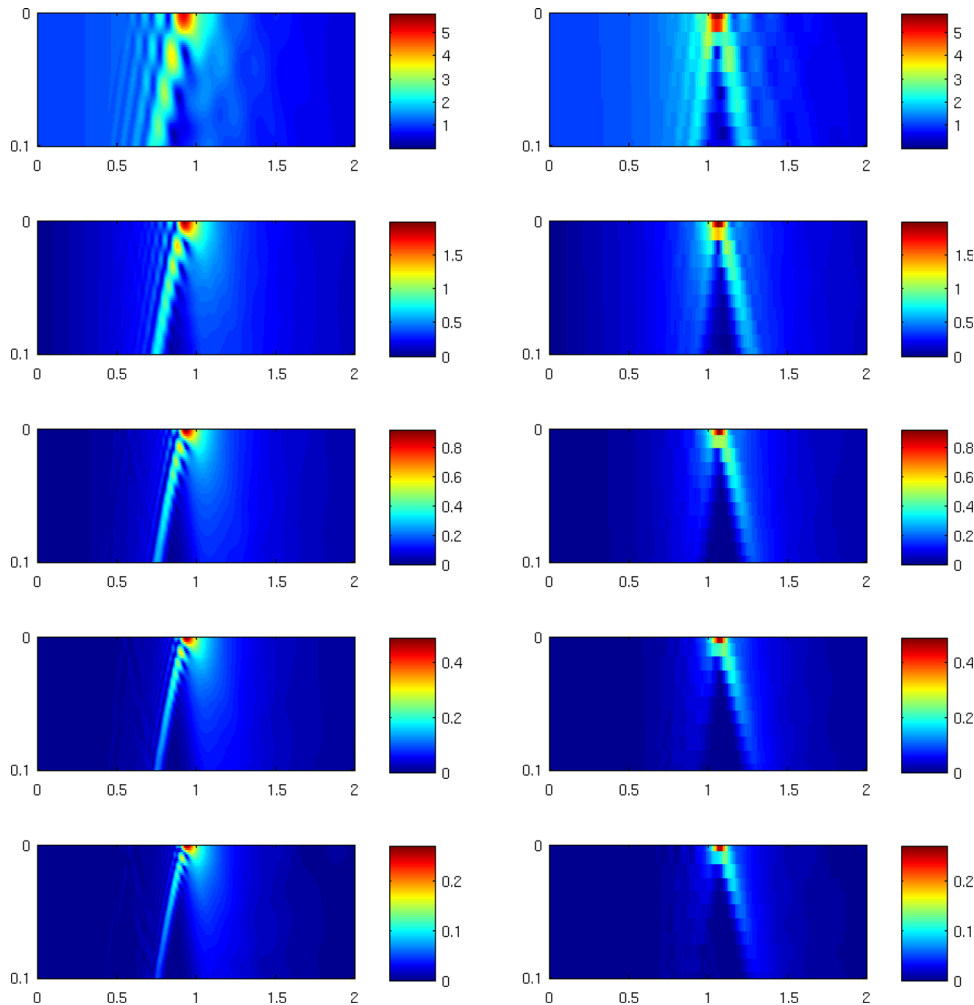


FIG. 11. (Color online) Comparisons of the amplitudes of the first five harmonics (top to bottom) for the 30λ aperture radius transducer. (Left) Results obtained from the Navier-Stokes FC simulation. (Right) Results obtained from the KZK model. Horizontal axes are measured in focal lengths. Vertical axes are measured in aperture radii.

incorporated into the geometry by means of overlapping curvilinear patches as described in Sec. V A.

Figure 12 displays solutions produced by the FC solver for a transducer with aperture radius $a = 20\lambda$ and focal length $F = 100\lambda$, with model parameters as in the previous examples. In these simulations, spatial windowing (Sec. VIC 2) was not used; instead, the sinusoidal transducer signal was smoothly ramped-up over five cycles [using a windowing function akin to (23) but which equals 1 for $t \geq 5/f_0$], and the solution was then evolved throughout the computational domain up the final time. For these examples a geometric optics transducer model was used to migrate the transducer signal to the line $z = 0$. By comparing the solutions produced by spatial discretizations of 21 and 25 points per fundamental wavelength, we have estimated the maximum error (relative to the maximum pressure at the acoustic focus) at the final

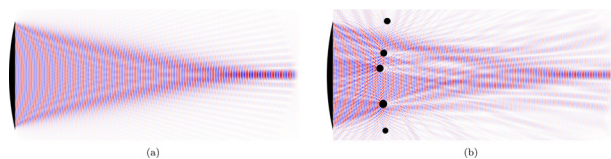


FIG. 12. (Color online) Pressure field for focusing transducer simulations. (a) Propagation and focusing in a homogeneous medium. (b) Propagation and focusing across an array of scatterers.

time in both solutions depicted in Fig. 12 to be on the order of 1% (see Remark 4).

In the case of the geometry with inclusions, the nature of the grids [see Fig. 5(c)] is such that the discretization is somewhat finer near the circles. The computing time required for the homogeneous problem was about 1.1 h, while the inhomogeneous problem completed in 5.6 h. (The additional computing cost in the latter case arises as a result of the small time step that is required, in view of the CFL condition for the finer discretization used near the circular scatterers. This additional cost can be reduced significantly through use of implicit/explicit FC solvers, wherein fine discretizations near circles are treated by means of an alternating-direction implicit algorithm.) Both simulations were performed on a total of 128 AMD Opteron cores, running at 2.2 GHz, at Caltech's Shared Heterogeneous Cluster, SHC. (For comparison, the full Navier-Stokes windowed simulations presented in Sec. VID required 7 min for the 5λ aperture radius transducer and 14 min for the 30λ aperture radius transducer in 128 cores of the same cluster.)

In light of the one-dimensional examples presented in Sec. IV, a second-order solver would be expected to require several orders of magnitude more time to obtain these solutions. Even adopting a conservative estimate of 1000 times (the estimate presented in Sec. IVA suggests a factor of 40 000), one finds that a second-order method would require

at least 45 days and 7 months, respectively, to produce the solutions to the examples in this section to within 1% accuracy.

VIII. DISCUSSION AND CONCLUSIONS

The examples presented in this work were motivated by the problem of HIFU simulation, which requires the solution of nonlinear acoustic advection equations in large (relative to the fundamental wavelength) domains. In order to efficiently treat the large-scale simulation of HIFU via a full-wave model, it is essential to use a numerical algorithm with very low dispersion errors; as demonstrated through the one- and two-dimensional examples in this text, the FC method exhibits the required exceptionally low dispersion errors and, therefore, can be a powerful tool in efficiently and accurately solving the types of problems under consideration. Such a tool could have a number of uses for HIFU researcher and practitioners, including, as demonstrated in Sec. VI, the validation of simplified wave models. Moreover, in highly complex tissue configurations, the assumptions inherent in the paraxial or one-way propagation models are expected to fail and, thus, it is unlikely that one can completely avoid the use of a full-wave model, such as the Navier-Stokes or Westervelt equations. Therefore, the development of efficient high-order solvers will be crucial for the advancement of full-scale, multitissue HIFU simulation. In view of this suggestion, the remainder of this section is devoted to a discussion of the ways in which the flexible FC approach outlined in this paper can be adapted to meet the challenging demands of complex HIFU simulations. In particular, the topics of three-dimensional simulations, advanced tissue models, heterogeneous media, adaptive mesh refinement, and solutions containing shocks are discussed.

A. Three-dimensional simulation

Axisymmetric three-dimensional problems can be treated by solving the three-dimensional PDE in axisymmetric form. The axial symmetry is implemented in three dimensions analogously to reflection symmetry in two dimensions [see, e.g., Sec. VIC 1 and Fig. 5(b)]. The extension from two to three dimensions is also straightforward; the basic construction necessary is the three-dimensional analog of Algorithm 2. The concepts of overlapping grid decomposition and block domain decomposition generalize trivially into three dimensions. Several axially symmetric and fully three-dimensional tests have provided promising results, and research in this direction is ongoing.

B. Advanced tissue models

The inclusion of thermal conductivity and bulk viscosity is trivial: the FC method is not restricted to any particular PDE and, in fact, has been used to solve a number of different advection-diffusion PDE systems.^{25–27,46} Power-law attenuation can be modeled by adding relaxation processes to the PDE model,^{14,47,48} and several preliminary tests of an axisymmetric piston transducer have been able to recover the correct attenuation curve efficiently using three to four relaxation processes.

C. Heterogeneous media

Because smoothly varying parameter inhomogeneities can be easily incorporated into the PDE coefficients, the main challenge with heterogeneous tissues arises from discontinuities in parameters at tissue interfaces. Overlapping grid techniques already exist for heterogeneous and multi-material problems and have been used with finite-difference solvers.⁴⁹ Based on the successful use of overlapping grids for the FC solver presented in this paper, it seems reasonable to expect that a similar approach is suitable for the FC method.

D. Adaptive mesh refinement

In HIFU simulation, steep, spatially localized pressure gradients often occur, in such a way that use of a uniform spatial discretization is inefficient. The standard approach for dealing with a situation such as this involves the use of adaptive mesh refinement (AMR). Although AMR has not been used in the examples in the present paper, it can be implemented by any standard technique currently used for grid-based solvers. Several successful demonstrations have indeed indicated that AMR can greatly increase the efficiency of the FC solver for HIFU-type problems, and research in this direction is ongoing. The comparisons presented in Sec. IV, which do not include use of AMR, nonetheless demonstrate the capabilities that AMR versions of the FV, FD, and FC solvers would exhibit in each one of the quasi-uniform portions of an adaptively refined grid. These results are therefore suggestive of the overall behavior that may be expected from AMR-augmented versions of the solvers under consideration.

E. Solutions containing shocks

Due to its utilization of Fourier methods for spatial differentiation, the FC method presented in the present paper is not well-suited for problems involving moving shock discontinuities. (Note, however, that Sec. IV B shows that the method is very efficient in resolving viscous shocks.) Problems with shocks can be treated by means of an adaptive hybrid approach wherein regions of smooth solution are treated by the FC method and regions containing shocks are treated by a high-resolution shock-tracking scheme. Such a hybrid, which uses the weighted essential non-oscillatory (WENO) differentiation operator near shocks, is discussed in Ref. 46.

ACKNOWLEDGMENTS

N.A. gratefully acknowledges the support of the NSF postdoctoral fellowship. N.A. and O.P.B. gratefully acknowledge support from NSF and AFOSR. T.Y.C. and R.O.C. gratefully acknowledge support from NSF through grant 0835795.

¹K. Roberts and N. Weiss, "Convective difference schemes," *Math. Comput.* **20**, 272–329 (1966).

²R. Alford, K. Kelly, and D. Boore, "Accuracy of finite-difference modeling of the acoustic wave equation," *Geophysics* **39**, 834–842 (1974).

- ³R. Vichnevetsky and B. Peiffer, "Error waves in finite element and finite difference methods for hyperbolic equations," in *Advances in Computer Methods for Partial Differential Equations*, edited by R. Vichnevetsky (AICA, New Brunswick, NJ, 1975), pp. 53–58.
- ⁴L. Jameson, "High order schemes for resolving waves: Number of points per wavelength," *J. Sci. Comput.* **15**, 417–439 (2000).
- ⁵B. Gustafsson, *High Order Difference Methods for Time Dependent PDE* (Springer-Verlag, Berlin, 2008), Chap. 1.2.
- ⁶C. Connor and K. Hynynen, "Bio-acoustic thermal lensing and nonlinear propagation in focused ultrasound surgery using large focal spots: A parametric study," *Phys. Med. Biol.* **47**, 1911–1928 (2002).
- ⁷S. Ginter, M. Liebler, E. Steiger, T. Dreyer, and R. Ridlinger, "Full-wave modeling of therapeutic ultrasound: Nonlinear ultrasound propagation in ideal fluids," *J. Acoust. Soc. Am.* **111**, 2049–2059 (2002).
- ⁸M. Liebler, S. Ginter, T. Dreyer, and R. Ridlinger, "Full wave modeling of therapeutic ultrasound: Efficient time-domain implementation of the frequency power-law attenuation," *J. Acoust. Soc. Am.* **116**, 2742–2750 (2004).
- ⁹E. Zabolotskaya and R. Khokhlov, "Quasi-plane waves in the nonlinear acoustics of confined beams," *Sov. Phys. Acoust.* **15**, 35–40 (1969).
- ¹⁰V. Kuznetsov, "Equations of nonlinear acoustics," *Sov. Phys. Acoust.* **16**, 467–470 (1971).
- ¹¹M. F. Hamilton and C. L. Morfey, "Model equations," in *Nonlinear Acoustics*, edited by M. F. Hamilton and D. T. Blackstock (Academic Press, San Diego, CA, 1998), Chap. 3, pp. 41–63.
- ¹²T. Christopher, "Algorithm for the nonlinear propagation of acoustic beams from phased arrays and nonplanar sources," *IEEE T. Ultrason. Ferr.* **53**, 2188–2192 (2006).
- ¹³T. Varslot and S.-E. Måsøy, "Forward propagation of acoustic pressure pulses in 3D soft biological tissue," *Model. Ident. Control* **27**, 181–190 (2006).
- ¹⁴Y. Jing and R. Cleveland, "Modeling the propagation of nonlinear three-dimensional acoustic beams in inhomogeneous media," *J. Acoust. Soc. Am.* **122**, 1352–1364 (2007).
- ¹⁵M. Canney, M. Bailey, L. Crum, V. Khokhlova, and O. Sapozhnikov, "Acoustic characterization of high intensity focused ultrasound fields: A combined measurement and modeling approach," *J. Acoust. Soc. Am.* **124**, 2406–2420 (2008).
- ¹⁶F. Dagrau, M. Rnieri, R. Marchiano, and F. Coulouvrat, "Acoustic shock wave propagation in a heterogeneous medium: A numerical simulation beyond the parabolic approximation," *J. Acoust. Soc. Am.* **130**, 20–32 (2011).
- ¹⁷Y. Pishchal'nikov, O. Sapozhnikov, and V. Khokhlova, "A modification of the spectral description of nonlinear acoustic waves with discontinuities," *Acoust. Phys.* **42**, 362–367 (1996).
- ¹⁸G. F. Pinton, J. Dahl, S. Rosenzweig, and G. E. Trahey, "A heterogeneous nonlinear attenuating full-wave model of ultrasound," *IEEE T. Ultrason. Ferr.* **56**, 474–488 (2009).
- ¹⁹A. Karamalis, W. Wein, and N. Navab, "Fast ultrasound image simulation using the westervelt equation," in *Medical Image Computing and Computer-Assisted Intervention MICCAI 2010, Lecture Notes in Computer Science*, edited by T. Jiang, N. Navab, J. Pluim, and M. Viergever (Springer, Berlin, 2010), Vol. 6361, pp. 243–250.
- ²⁰L. Demi, K. W. A. van Dongen, and M. D. Verweij, "A contrast source method for nonlinear acoustic wave fields in media with spatially inhomogeneous attenuation," *J. Acoust. Soc. Am.* **129**, 1221–1230 (2011).
- ²¹Y. Jing and G. T. Clement, "On the use of Gegenbauer reconstructions for shock wave propagation modeling," *J. Acoust. Soc. Am.* **130**, 1115–1124 (2011).
- ²²Y. Jing, D. Shen, and G. Clement, "Verification of the Westervelt equation for focused transducers," *IEEE T. Ultrason. Ferr.* **58**, 1097–1101 (2011).
- ²³K. Okita, K. Ono, S. Takagi, and Y. Matsumoto, "Development of high intensity focused ultrasound simulator for large-scale computing," *Int. J. Numer. Meth. Fl.* **65**, 43–66 (2011).
- ²⁴P. Yuldashev and V. Khokhlova, "Simulation of three-dimensional nonlinear fields of ultrasound therapeutic arrays," *Acoust. Phys.* **57**, 334–343 (2011).
- ²⁵O. P. Bruno and M. Lyon, "High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic elements," *J. Comput. Phys.* **229**, 2009–2033 (2010).
- ²⁶M. Lyon and O. P. Bruno, "High-order unconditionally stable FC-AD solvers for general smooth domains II. Elliptic, parabolic and hyperbolic PDEs; theoretical considerations," *J. Comput. Phys.* **229**, 3358–3381 (2010).
- ²⁷N. Albin and O. P. Bruno, "A spectral FC solver for the compressible Navier-Stokes equations in general domains I: Explicit time-stepping," *J. Comput. Phys.* **230**, 6248–6270 (2011).
- ²⁸J. P. Boyd, "A comparison of numerical algorithms for Fourier extension of the first, second, and third kinds," *J. Comput. Phys.* **178**, 118–160 (2002).
- ²⁹O. P. Bruno, "Fast, high-order, high-frequency integral methods for computational acoustics and electromagnetics," in *Topics in Computational Wave Propagation, Lect. Notes Comput. Sci. Eng.* (Springer-Verlag, Berlin, 2003), Vol. 31, pp. 43–82.
- ³⁰O. P. Bruno, Y. Han, and M. M. Pohlman, "Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis," *J. Comput. Phys.* **227**, 1094–1125 (2007).
- ³¹CLAWPACK website, <http://depts.washington.edu/clawpack/> (Last viewed February 14, 2012).
- ³²J. Nordström, J. Gong, E. van der Weide, and M. Svärd, "A stable and conservative high order multi-block method for the compressible Navier-Stokes equations," *J. Comput. Phys.* **228**, 9020–9035 (2009).
- ³³J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed. (Dover Publications, Mineola, NY, 2001), Chaps. 4 and 9.
- ³⁴S. Lele, "Compact finite difference schemes with spectral-like resolution," *J. Comput. Phys.* **103**, 16–42 (1992).
- ³⁵G. Ashcroft and X. Zhang, "Optimized prefactored compact schemes," *J. Comput. Phys.* **190**, 459–477 (2003).
- ³⁶J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods, Texts in Applied Mathematics* (Springer, New York, 2008), Vol. 54, Chaps. 1–3, 5–6.
- ³⁷M. R. Visbal and D. V. Gaitonde, "On the use of higher-order finite-difference schemes on curvilinear and deforming meshes," *J. Comput. Phys.* **181**, 155–185 (2002).
- ³⁸P. Morgan, M. Visbal, and D. Rizzetta, "A parallel overset grid high-order flow solver for large eddy simulation," *J. Sci. Comput.* **29**, 165–200 (2006).
- ³⁹W. Henshaw and D. Schwendeman, "Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement," *J. Comput. Phys.* **227**, 7469–7502 (2008).
- ⁴⁰R. T. Beyer, "The Parameter B/A," in *Nonlinear Acoustics*, edited by M. F. Hamilton and D. T. Blackstock (Academic Press, San Diego, CA, 1998), Chap. 2, pp. 25–39.
- ⁴¹Y.-S. Lee and M. F. Hamilton, "Time-domain modeling of pulsed finite-amplitude sound beams," *J. Acoust. Soc. Am.* **97**, 906–917 (1995).
- ⁴²Y.-S. Lee, "Numerical solution of the kzk equation for pulsed finite amplitude sound beams in thermoviscous fluids," Ph.D. thesis, University of Texas, Austin, TX, 1993.
- ⁴³H. O'Neil, "Theory of focusing radiators," *J. Acoust. Soc. Am.* **21**, 516–526 (1949).
- ⁴⁴M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W. S. Don, "The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: A study of the boundary error," *SIAM J. Sci. Comput.* **16**, 1241–1252 (1995).
- ⁴⁵S. Abarbanel, D. Gottlieb, and M. H. Carpenter, "On the removal of boundary errors caused by Runge-Kutta integration of nonlinear partial differential equations," *SIAM J. Sci. Comput.* **17**, 777–782 (1996).
- ⁴⁶K. Shahbazi, N. Albin, O. P. Bruno, and J. S. Hesthaven, "Multi-domain Fourier-Continuation/WENO hybrid solver for conservation laws," *J. Comput. Phys.* **230**, 8779–8796 (2011).
- ⁴⁷A. I. Nachman, J. F. Smith III, and R. C. Waag, "An equation for acoustic propagation in inhomogeneous media with relaxation losses," *J. Acoust. Soc. Am.* **88**, 1584–1595 (1990).
- ⁴⁸X. Yuan, D. Borup, J. Wiskin, M. Berggren, and S. Johnson, "Simulation of acoustic wave propagation in dispersive media with relaxation losses by using FDTD method with PML absorbing boundary condition," *IEEE T. Ultrason. Ferr.* **46**, 14–23 (1999).
- ⁴⁹W. Henshaw, "A high-order accurate parallel solver for Maxwell's equations on overlapping grids," *SIAM J. Sci. Comput.* **28**, 1730–1765 (2006).