Journal of Computational Physics 307 (2016) 333–354

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An FC-based spectral solver for elastodynamic problems in general three-dimensional domains

Faisal Amlani, Oscar P. Bruno [*]

ARTICLE INFO

Article history:
Received 21 March 2015
Received in revised form 27 November 2015
Accepted 28 November 2015
Available online 8 December 2015

Keywords:
Elastic wave equation
Spectral solver
General domain
FC solver
Seismic simulations
Non-destructive testing

ABSTRACT

This paper presents a spectral numerical algorithm for the solution of elastodynamics problems in general three-dimensional domains. Based on a recently introduced "Fourier continuation" (FC) methodology for accurate Fourier expansion of non-periodic functions, the proposed approach possesses a number of appealing properties: it yields results that are essentially free of dispersion errors, it entails mild CFL constraints, it runs at a cost that scales linearly with the discretization sizes, and it lends itself easily to efficient parallelization in distributed-memory computing clusters. The proposed algorithm is demonstrated in this paper by means of a number of applications to problems of isotropic elastodynamics that arise in the fields of materials science and seismology. These examples suggest that the new approach can yield solutions within a prescribed error tolerance by means of significantly smaller discretizations and shorter computing times than those required by other methods.

1. Introduction

This paper introduces a new spectral algorithm for the numerical solution of three-dimensional problems in elastodynamics. Based on a recently introduced FFT-speed Fourier Continuation (FC) methodology for accurate Fourier expansion of non-periodic functions [2,12], the new approach, which entails mild CFL constraints (see e.g. Remark 6.1) and computing times that scale only linearly with the sizes of the underlying spatial discretizations, enables fast, high-order and *essentially dispersionless* solution of the time-dependent elastic wave equation. Explicit and implicit FC-based PDE solvers have previously been introduced for a range of PDE problems, including explicit and implicit solvers for the classical wave and diffusion equations with constant and variable coefficients [12,13,15,26], implicit solvers for the nonlinear Burgers system [11], and explicit solvers for the compressible Navier–Stokes equations [2,3] and the Euler equations [32,33]. The algorithm described in this paper, which extends the applicability of the FC method to the challenging Navier elastic wave equation, includes an FC operator for treatment of Neumann and traction boundary conditions, it allows for treatment of equations with variable coefficients, and it utilizes a block-partition strategy (based on use of overset curvilinear domain-decomposition method [9]) which allows for treatment of general three-dimensional geometries in distributed-memory parallel computing environments with perfect parallel efficiency.

The qualities of spectrally accurate numerical solvers for time-dependent PDEs are well established: spectral methods can produce accurate solutions by means of relatively coarse discretizations, and, crucially, they faithfully preserve the dispersion characteristics of the underlying continuous problems. Classical spectral solvers do present significant challenges, however, including taxing time-stepping CFL restrictions for stability in polynomial spectral methods, as well as geometric and periodicity restrictions for classical Fourier-based methods. In view of its reliance on essentially equispaced Cartesian meshes and

[*] Corresponding author.
E-mail address: obruno@caltech.edu (O.P. Bruno).

http://dx.doi.org/10.1016/j.jcp.2015.11.060
0021-9991/© 2015 Elsevier Inc. All rights reserved.

its ability to effectively produce rapidly convergent Fourier series for discontinuous functions (thereby eliminating the well known Gibbs-ringing phenomenon), the spectral solver presented in this paper does not suffer from any of these difficulties. The broad applicability and beneficial qualities of the proposed FC-based algorithm are demonstrated through a variety of examples, including convergence and scalability studies as well as applications to realistic problems concerning seismic wave motion on three-dimensional topographies and problems arising in the field of non-destructive evaluation—where, for the first time, high-frequency three-dimensional simulations of guided-wave scattering by through-thickness holes in thin plates are presented, including comparisons to experimental data.

This text is organized as follows: after the relevant equations and notations are presented in Section 2, Section 3 briefly describes the FC method and introduces certain extensions of the method that are necessary in the context of the elastic wave problem under consideration. Section 4 then presents a one-dimensional example (including both Dirichlet and Neumann boundary conditions) which demonstrates the character of the proposed solvers in a simple context. Our methods for treatment of complex geometries and parallelization are then described in Section 5. A number of implementation details, such as the time-stepping methods and spectral filters used for the full three-dimensional elastic wave equation, as well as approaches utilized for treatment of associated traction boundary conditions, are presented in Section 6. A variety of performance studies and numerical examples concerning three-dimensional wave scattering problems are then presented in Section 7, including numerical demonstrations of accuracy and limited dispersion. Section 8 then presents illustrative applications, and conclusions, finally, are put forth in Section 9.

2. Governing equations

The problem of propagation of elastic waves in a linear, isotropic, possibly heterogeneous medium contained in a general three-dimensional domain Ω is governed by the Navier equation [19]

$$\rho(\mathbf{x})\mathbf{u}_{tt}(\mathbf{x},t) = \nabla\cdot\left[\mu(\mathbf{x})\left(\nabla\mathbf{u}(\mathbf{x},t)+\nabla\mathbf{u}^T(\mathbf{x},t)\right)+\lambda(\mathbf{x})\left(\nabla\cdot\mathbf{u}(\mathbf{x},t)\right)I\right]+\mathbf{f}(\mathbf{x},t), \quad \mathbf{x}\in\Omega, \tag{1}$$

with initial conditions prescribed at some initial time $t=t_0$,

$$\mathbf{u}(\mathbf{x},t_0)=\mathbf{a}(\mathbf{x}), \qquad \mathbf{u}_t(\mathbf{x},t_0)=\mathbf{b}(\mathbf{x}), \tag{2}$$

and with conditions on displacements or tractions prescribed along the boundary of Ω. Here, using the superscript T to indicate matrix transposition, we have let $\mathbf{x}=(x^1,x^2,x^3)^T$ and $\mathbf{u}=(u^1,u^2,u^3)^T$ denote the position and displacement vectors, I denote the identity matrix, and $\mathbf{f}=\left(f^1(\mathbf{x},t), f^2(\mathbf{x},t), f^3(\mathbf{x},t)\right)^T$ denote a given vector of body forces. The material properties are specified by the Lamé parameters $\mu(\mathbf{x}),\lambda(\mathbf{x})$ and the density $\rho=\rho(\mathbf{x})$; the respective longitudinal and transverse (shear) wave speeds are given by the relations

$$c_L=\sqrt{(\lambda+2\mu)/\rho} \quad\text{and}\quad c_T=\sqrt{\mu/\rho}. \tag{3}$$

On the boundary $\partial\Omega$ various types of boundary conditions may be imposed: typically the domain boundary is partitioned as a union $\partial\Omega=\Gamma_D\cup\Gamma_T$ of two surfaces Γ_D and Γ_T upon which boundary displacements

$$\mathbf{u}=\mathbf{c}(\mathbf{x},t) \quad \text{on} \quad \Gamma_D \tag{4}$$

and boundary tractions

$$\boldsymbol{\sigma}\cdot\mathbf{n}=\mathbf{d}(\mathbf{x},t) \quad \text{on} \quad \Gamma_T \tag{5}$$

are prescribed. In these expressions $\mathbf{n}=(n^1,n^2,n^3)^T$ and $\boldsymbol{\sigma}$ denote the inward unit normal to the surface and the (symmetric) stress tensor

$$\sigma_{ij}=c_{ijk\ell}\frac{\partial u^k}{\partial x^\ell}, \tag{6}$$

respectively, where $c_{ijk\ell}$ is the fourth-order stiffness tensor. As is well known, for an isotropic medium we have

$$\sigma_{ij}=\lambda\epsilon_{kk}\delta_{ij}+2\mu\epsilon_{ij}, \quad \epsilon_{ij}=\frac{1}{2}\left(\frac{\partial u^i}{\partial x^j}+\frac{\partial u^j}{\partial x^i}\right), \quad i,j=1,2,3, \tag{7}$$

where δ_{ij} is the Kronecker delta and $\boldsymbol{\epsilon}$ is the infinitesimal strain tensor.

3. Fast and stable Fourier Continuation: FC(Gram)

A goal to extend the applicability of the classical Fourier-based PDE solvers (together with their inherent excellent qualities, most notably, limited dispersion, high-order accuracy and mild CFL conditions) to problems involving general domains and boundary conditions has led to the development of the Fourier Continuation (FC) method [2,12,13,15,26]. The FC method

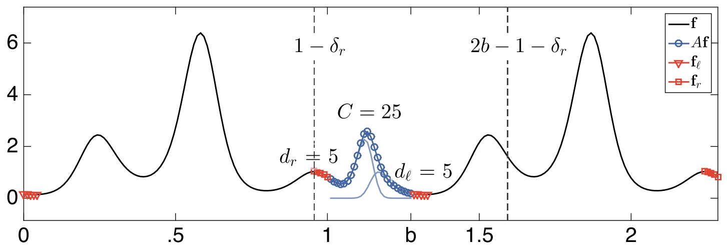

Fig. 1. Fourier continuation of the non-periodic function $f(x) = e^{\sin(5.4\pi x - 2.7\pi) - \cos(2\pi x)}$. Red triangles/squares and blue circles represent $d_\ell = d_r = 5$ matching points and $C = 25$ continuation points, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

enables high-order convergence of Fourier series approximations of non-periodic functions by resolving the well-known Gibbs "ringing" effect. Given point values $f(x_i)$ of a given function $f : [0, 1] \to \mathbb{R}$ on the uniform discretization $x_i = ih$, $i = 0, \ldots, N-1$, $h = 1/(N-1)$, the FC method produces a rapidly-convergent interpolating Fourier series representation $f^c : [0, b] \to \mathbb{R}$ on a region $[0, b]$ larger than the given physical domain $[0, 1]$:

$$f^c(x) = \sum_{k=-M}^{M} a_k e^{\frac{2\pi ikx}{b}} \quad \text{s.t. } f^c(x_i) = f(x_i),\ i = 0, \ldots, N-1 \tag{8}$$

for suitably chosen FC-parameters M (bandwidth) and $b > 1$ (interval length). The continuation function f^c is an approximate *periodic extension* of the given function f: f^c closely approximates f in the original domain $[0, 1]$ and it is periodic on $[0, b]$, $b > 1$. Numerical derivatives necessary in a PDE solver can be produced from the FC series (8) with high-order accuracy by straightforward term-wise series differentiation.

Our method of construction of a continuation series (8) is based on use of a certain "biased-order" FC technique introduced in [2] which is useful in the context of domain decomposition. The biased-order FC method relies on use of numbers $d = d_\ell$ and $d = d_r$ of function values near the left and right endpoints 0 and 1, respectively, together with projections of the corresponding vectors of function values onto a Gram polynomial basis—whose continuations are precomputed by means of high-precision linear algebra methods. An extension of this "FC(Gram)" method to a form suitable for use as part of the elasticity solver proposed in this text is described in what follows.

3.1. Accelerated Fourier continuation: FC(Gram)

Given a column vector $\mathbf{f} = (f_0, \ldots, f_{N-1})^T$ containing point-values of a given function f on the equispaced grid $0 = x_0 < x_1 < \cdots < x_{N-1} = 1$, $f_i = f(x_i)$, the accelerated FC(Gram) method [2,12,26] uses a subset of the given function values on small numbers d_ℓ and d_r of matching points $\{x_0, \ldots, x_{d_\ell - 1}\}$ and $\{x_{N-d_r}, \ldots, x_{N-1}\}$ contained in small subintervals on the left and right ends of the interval $[0, 1]$ (of lengths $\delta_\ell = (d_\ell - 1)h$ and $\delta_r = (d_r - 1)h$) to produce, at first, a discrete periodic extension. Indeed, using such subinterval data points the FC(Gram) algorithm appends a number C of continuation function values in the interval $[1, b]$ to the existing function data, so that the extension transitions smoothly from f_{N-1} back to f_0, as depicted in Fig. 1. The resulting vector $\mathbf{f}^c$ can be viewed as a discrete set of values of a smooth and periodic function which is suitable for high-order approximation by means of the FFT algorithm in an interval of length $(N + C)\Delta x$.

In order to produce the C necessary extension values the FC(Gram) method uses the discrete function defined by the vector $\mathbf{f}$ together with a translation of it by a distance b. In detail, defining the sets $\mathcal{D}_\ell = \{b + x_0, b + x_1, \ldots, b + x_{d_\ell - 1}\}$ and $\mathcal{D}_r = \{x_{N-d_r}, x_{N-(d_r-1)}, \ldots, x_{N-1}\}$, the additional C needed values in the interval $[1, b]$ are obtained as point values of an auxiliary trigonometric polynomial of periodicity interval $[1 - \delta_r, 2b - (1 - \delta_r)]$ (with appropriately selected bandwidth) which closely approximate the function values on $\mathcal{D}_r \cup \mathcal{D}_\ell$. This approximating trigonometric polynomial is obtained as the result of a two-step process, namely 1) Projection onto bases of orthogonal polynomials (Gram bases), and 2) Continuation through use of a precomputed set of continuations-to-zero of each Gram polynomial, as explained in what follows.

The polynomial projection mentioned in step 1) above for the function values on $\mathcal{D}_r$ and $\mathcal{D}_\ell$ (cf. [12]) relies on use of a basis $\mathcal{B}_r$ (resp. $\mathcal{B}_\ell$) of the space of polynomials of degree $< d_r$ (resp. d_ℓ) on the interval $[1 - \delta_r, 1]$ (resp. $[b, b + \delta_\ell]$) which is orthonormal with respect to the discrete scalar product $(\cdot,\cdot)_r$ (resp. $(\cdot,\cdot)_\ell$) defined by the discretization points $\mathcal{D}_r$ (resp. $\mathcal{D}_\ell$):

$$(g, h)_r = \sum_{x_i \in \mathcal{D}_r} g(x_i) h(x_i), \tag{9}$$

with a similar definition for $(g, h)_\ell$. The algorithm also utilizes precomputed extensions, one for each polynomial $p_r \in \mathcal{B}_r$, into a smooth function defined for $x \geq 1 - \delta_r$ which approximates p_r closely in the matching interval $[1 - \delta_r, 1]$, and which blends smoothly to zero for $x \geq b$. Such rightward extensions are constructed as appropriately oversampled least squares approximations by Fourier series of periodicity interval $[1 - \delta_r, 2b - (1 - \delta_r)]$—which are obtained by means of high-precision linear algebra methods, as described in [12,26] (see also Remark 3.1 below). Similarly, the scheme obtains, for each polynomial $p_\ell \in \mathcal{B}_\ell$, a smooth blending function that agrees with p_ℓ in the matching interval $[b, b+\delta_\ell]$ and which vanishes for $x \leq 1$.

In presence of such smooth blending functions the algorithm proceeds to step 2): evaluation of an extension from the function values at the set of points $\mathcal{D}_r \cup \mathcal{D}_\ell$. This can be achieved easily since the Gram polynomials of degrees $\leq d_r - 1$ on $\mathcal{D}_r$ and $\leq d_\ell - 1$ on $\mathcal{D}_\ell$ that interpolate the given data can be expressed as linear combinations of the polynomials in the bases $\mathcal{B}_\ell$ and $\mathcal{B}_r$—with coefficients that can be obtained rapidly by means of scalar products. With the extension in hand, an application of the discrete Fourier transform on the interval $[0, b]$ to the vector of function values $\mathbf{f}$ augmented by the C "continuation" values yields the desired trigonometric polynomial (8). An example of the blending procedure is depicted in Fig. 1. For efficiency, the discrete Fourier transform is implemented by means of the Fast Fourier Transform (FFT).

The resulting continuation operation can be expressed in a block matrix form as

$$\mathbf{f}^c = \begin{bmatrix} I \\ A \end{bmatrix} \mathbf{f} = \begin{bmatrix} \mathbf{f} \\ A\mathbf{f} \end{bmatrix}, \tag{10}$$

where $\mathbf{f}^c$ is a vector of the $N + C$ continued function values, I is the $N \times N$ identity matrix and A is the matrix containing the blend-to-zero continuation information. Defining the vector of matching points for the left and right as

$$\mathbf{f}_\ell = \left(f_0, f_1, \ldots, f_{d_\ell - 1}\right)^T, \quad \mathbf{f}_r = \left(f_{N-d_r}, f_{N-d_r+1}, \ldots, f_{N-1}\right)^T, \tag{11}$$

the matrix A can be expressed in the form

$$A\mathbf{f} = A_\ell Q_\ell^T \mathbf{f}_\ell + A_r Q_r^T \mathbf{f}_r, \tag{12}$$

where the columns of Q_ℓ and Q_r contain the d_ℓ, d_r point values of each element of the corresponding Gram polynomial basis, and where the columns of A_ℓ and A_r contain the corresponding C values that blend the polynomials in the left and the right Gram bases to zero. For $d_\ell = d_r$ the matrices A_ℓ and A_r (resp. Q_ℓ and Q_r) differ only by row-ordering (resp. only by column ordering). The matrices $Q = Q_\ell$ and $Q = Q_r$ for given numbers $d = d_\ell$ and $d = d_r$ of matching points may be obtained by orthogonalizing the columns of the $d \times d$ Vandermonde matrix

$$P = \begin{pmatrix} 1 & x_0 & (x_0)^2 & \ldots & (x_0)^{d-1} \\ 1 & x_1 & (x_1)^2 & \ldots & (x_1)^{d-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{d-1} & (x_{d-1})^2 & \ldots & (x_{d-1})^{d-1} \end{pmatrix} \tag{13}$$

of point values of the monomials x^j ($j = 1, \ldots, d-1$) at the discrete points $x_0, x_1, \ldots, x_{d-1}$. (A substitution of x by $(x-1)$ must be used for the right matching problem.) The necessary orthonormalization is performed by applying the stabilized Gram–Schmidt orthogonalization process to produce the QR decomposition

$$P = QR; \tag{14}$$

note that the jth column of Q in (14) contains the d point values of the jth polynomial in the Gram basis. To ensure a good agreement with the constraints described by the least squares problem introduced in what follows, we additionally oversample the monomial basis by a factor of n_{over} (we have used the value $n_{over} = 20$ throughout this paper), which leads to a Vandermonde matrix P^{over} similar to (13) but of size $(n_{over}(d-1)+1) \times d$. The corresponding matrix whose columns are the Gram basis polynomials evaluated on a fine grid of size $d \times n_{over}$ can be produced by means of the expression

$$Q^{over} = P^{over} R^{-1}, \tag{15}$$

where R is the upper triangular matrix in the orthogonalization of the coarse P in (14).

Remark 3.1. The conditioning characteristics of the full Fourier continuation procedure described above derive exclusively from the corresponding conditioning characteristics associated with the orthogonalization of the Vandermonde matrix (13) (such a decomposition is well known to give rise to severe numerical loss of orthogonality) and the subsequent computation of the extensions to zero (which relies on use of Singular Value Decompositions). If accurate QR factorizations and SVDs are somehow obtained and used, then the losses of accuracy arising from ill-conditioning are eliminated. The conditioning difficulties are therefore successfully addressed in our context by relying on high-precision arithmetic (256 digits in Matlab's VPA for the computations used in this paper) throughout the process of orthogonalization and evaluation of precomputed continuations for the polynomial basis elements. This is a computation that needs to be performed only once and stored in file for use each time the Fourier continuation method is invoked. Technical details concerning the implementation of this high-precision precomputation procedure can be found in [4,12].

We provide some details for the rightward blending-to-zero procedure; the corresponding leftward methodology is entirely analogous. Letting Z be the number of zero-blending points and letting E be a certain number of extra points that are used to allow for continuation regions of user-prescribed lengths (cf. Section 5.3.3), we define the intervals

$$\begin{aligned} I_{match} &= [0, (d-1)h], \\ I_{blend} &= [dh, (d+C-1)h], \\ I_{zero} &= [(d+C)h, (d+C+Z-1)h], \\ I_{extra} &= [(d+C+Z)h, (d+C+Z+E-1)h] \end{aligned} \tag{16}$$

which contain d, C, Z and E discretization points, respectively. The function that blends the discrete values of a given Gram polynomial on I_{match} to the zero function values in I_{zero} is obtained as a band-limited trigonometric polynomial

$$f(x) = \sum_{k=-M}^{M} a_k e^{\frac{2\pi ikx}{(d+C+Z+E-1)h}} \tag{17}$$

that matches closely (in a least-squares sense on the oversampled grid) both, the Gram polynomials as well as an interval of vanishing function values, where $M = (d+C+Z+E)/2$. The coefficients $\mathbf{a} = (a_{-M}, \dots, a_M)^T$ are found by solving the minimization problem (posed on the oversampled Gram basis Q^{over} with columns q_k^{over} in (15)) given by

$$\min_{\mathbf{a}=(a_{-M},\dots,a_M)^T} \left\| B^{over}\mathbf{a} - \begin{pmatrix} q_j^{over} \\ \mathbf{0} \end{pmatrix} \right\|_2, \tag{18}$$

where B^{over} is a matrix formed from values of the function (17) at discretizations of I_{match} and I_{zero} of meshsize h/n_{over}, and where $\mathbf{0}$ is the zero vector of dimension $(Z-1)n_{over}+1$. The minimizing Fourier coefficients $\mathbf{a} = (a_{-M}, \dots, a_M)^T$ in (18) are then found via a Singular Value Decomposition and, once determined, the corresponding columns of A_ℓ, A_r are constructed by evaluating (17) on the coarse Δx discretization of the full continuation interval $[0, (d+C+Z+E-1)h]$.

Remark 3.2. For all computations in this paper the parameters $C = 25$, $Z = 12$, $E = 25$ and $n_{over} = 20$ were used to construct continuations for various integer values of $d = d_\ell, d_r$. These selections were made in accordance with those used in previous FC-based algorithms [2,15], and they were otherwise determined empirically to provide an appropriate trade-off between overall computational cost and stability in PDE solvers arising from the FC operator described above.

3.2. A modified accelerated FC operator for Neumann boundary conditions

In order to treat Neumann boundary conditions (and, ultimately, traction boundary conditions) it is necessary for the FC series to match given derivative values in addition to given function values. For example, it may be necessary to produce an FC expansion for a function f whose derivative is known at the end point x_{d-1} together with the values of f at $x_0, \dots, x_{d-2}$. This is easily achieved: following the ideas introduced in Section 3.1, the modified method uses, instead of the polynomial interpolants considered in the previous section, a polynomial interpolant that matches the derivative value $f'(x_{d-1})$ at the endpoint x_{d-1} as well as the function values at $x_0, \dots, x_{d-2}$. Such an interpolant can be obtained by orthonormalizing the columns of the modified Vandermonde matrix

$$P_{der} = \begin{pmatrix} 1 & x_0 & (x_0)^2 & \dots & (x_0)^{d-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{d-2} & (x_{d-2})^2 & \dots & (x_{d-2})^{d-1} \\ 0 & 1 & 2x_{d-1} & \dots & (d-1)(x_{d-1})^{d-2} \end{pmatrix} \tag{19}$$

(instead of the one for the matrix (13)) by means of a high-precision QR-decomposition for the matrix P_{der}:

$$P_{der} = Q_{der}R_{der}. \tag{20}$$

The modified Fourier continuation blend-to-zero information can be obtained by means of two different procedures (which in practice have been found to yield essentially indistinguishable results): 1) by proceeding to form a new continuation basis in a manner analogous to Section 3.1, replacing Q with Q_{der} and the appropriate B^{over} with a corresponding B_{der}^{over} in the least squares formulation; or, 2) by reconstructing the coefficients in the original Gram polynomial basis and simply replacing the operator Q $(= Q_\ell, Q_r)$ in (12) with a new operator $\tilde{Q}$ $(= \tilde{Q}_\ell, \tilde{Q}_r)$ for which the same pre-constructed blend-to-zero Dirichlet operators A_ℓ, A_r obtained in the previous section can be employed. The latter method—which is used in this work and introduced in what follows—carries the advantage that the Dirichlet-boundary-conditions blending-to-zero Gram polynomials described in the previous section may be re-used for treatment of Neumann and traction boundary conditions as well.

The construction of $\tilde{Q}$ is obtained by solving the system

$$P_{der}\mathbf{c} = \left(f_0, f_1, \dots, f_{d-2}, f'(x_{d-1})\right)^T \tag{21}$$

for the coefficients $\mathbf{c} = \left(c_1, c_2, \dots, c_{d-1}\right)^T$ via the decomposition (20) as

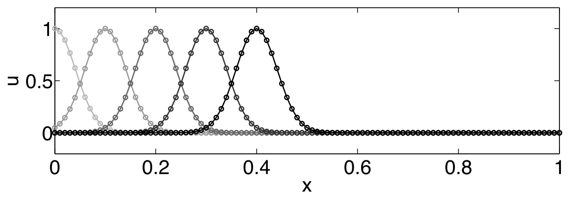

Fig. 2. Numerical solution to the one-dimensional wave equation (28) at times $t = 0.5, 0.6, 0.7, 0.8, 0.9$ s based on use of Fourier continuation ($d_\ell = d_r = 5$, $C = 25$, $\Delta x = 1/100$) and explicit time marching ($\Delta t = \Delta x/32$).

$$\mathbf{c} = R_{der}^{-1} Q_{der}^T \left(f_0, f_1, \ldots, f_{d-2}, f'(x_{d-1})\right)^T . \tag{22}$$

Recalling the Vandermonde matrix P and its QR-decomposition in (14), substitution of these coefficients into

$$P\mathbf{c} = QR\mathbf{c} = \left(f_0, f_1, \ldots, f_{d-2}, f_{d-1}\right)^T \tag{23}$$

yields the expressions

$$QRR_{der}^{-1} Q_{der}^T \left(f_0, f_1, \ldots, f_{d-2}, f'(x_{d-1})\right)^T = \left(f_0, f_1, \ldots, f_{d-2}, f_{d-1}\right)^T ,$$

or, equivalently,

$$\tilde{Q}^T \left(f_0, f_1, \ldots, f_{d-2}, f'(x_{d-1})\right)^T = Q^T \left(f_0, f_1, \ldots, f_{d-2}, f_{d-1}\right)^T \tag{24}$$

for $\tilde{Q} = (RR_{der}^{-1} Q_{der}^T)^T$. Hence the continuation procedure constructed in the previous section and embodied in the formula

$$A\mathbf{f} = A_\ell Q_\ell^T \mathbf{f}_\ell + A_r Q_r^T \mathbf{f}_r \tag{25}$$

is modified by substitution of (24) into (25) to yield

$$A\mathbf{f} = A_\ell \tilde{Q}_\ell^T \tilde{\mathbf{f}}_\ell + A_r \tilde{Q}_r^T \tilde{\mathbf{f}}_r, \tag{26}$$

where $\tilde{\mathbf{f}}_\ell = (f_0, \ldots, f_{d_\ell - 2}, f'(x_{d_\ell - 1}))^T$, $\tilde{\mathbf{f}}_r = (f_{N-d_r}, \ldots, f_{N-2}, f'(x_{N-1}))^T$ and where A_ℓ, A_r are the same blend-to-zero operators. Note that if, for example, one requires the continuation to approximate the value $f(x_0)$ and the derivative $f'(x_{N-1})$ (as is often necessary as a result of the domain decomposition strategy that we introduce later in this paper), one needs only to form the corresponding biased continuation using the appropriate projections on the left and the right as

$$A\mathbf{f} = A_\ell Q_\ell^T \mathbf{f}_\ell + A_r \tilde{Q}_r^T \tilde{\mathbf{f}}_r; \tag{27}$$

a similar procedure can be used to obtain approximations matching $f'(x_0)$ and $f(x_{N-1})$.

4. A simple one-dimensional example

As a simple example to demonstrate the character of the FC methodology for Dirichlet and Neumann problems we consider a 1D wave propagation problem with initial values given by the Gaussian profile

$$u_{tt}(x,t) = u_{xx}(x,t), \quad u(x,0) = e^{-300(x+.5)^2}, \quad u_t(x,0) = 600(x+.5)e^{-300(x+.5)^2}. \tag{28}$$

In order to facilitate evaluation of errors in our Dirichlet (resp. Neumann) test cases, we set up Dirichlet (resp. Neumann) boundary conditions at $x = 0$ and $x = 1$ in such a way that the exact solution is given by the traveling wave function depicted in Fig. 2 and given by

$$u(x) = e^{-300(x-t+.5)^2}, \quad 0 \le x \le 1. \tag{29}$$

Using the FC method for evaluation of spatial derivatives, and utilizing the explicit fourth-order Adams–Bashforth (AB4) method to evolve the resulting set of ODEs, we obtain a PDE solver for the wave equation (28); see also Remark 4.1. Using a small time-step enables us to easily demonstrate the accuracy of the FC based spatial discretization method for various spatial mesh sizes. Table 1 displays, for both the Dirichlet and Neumann problems, the maximum error over all spatial points $x \in [0,1]$ and for all times between 0 and $T = 2$, where T is taken to be large enough that the wave has passed through both interval endpoints. As expected, a 4th-order Gram polynomial basis (using 5 matching points) results in essentially 5th-order convergence; since the derivatives of the continued function are spectrally accurate, the error is dominated by the polynomial approximation used to project the end function values onto a Fourier continuation basis.

Table 1
Convergence results for the maximum error over all space and time of the solution to the 1D wave equation (28).

$1/\Delta x$	L^∞ error (Dirichlet)	$O(L^\infty)$	L^∞ error (Neumann)	$O(L^\infty)$
50	3.60e−2	–	6.15e−2	–
100	1.05e−3	5.10	1.44e−3	5.41
200	3.69e−5	4.83	3.21e−5	5.49
300	4.93e−6	4.96	4.20e−6	5.02
400	1.16e−6	5.03	1.00e−6	4.99

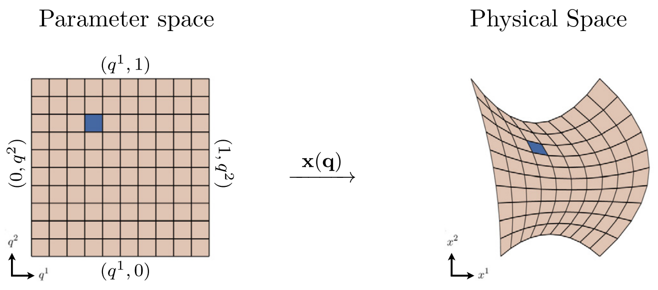

Fig. 3. Curvilinear coordinates.

Remark 4.1. The necessary solution values at the three initial time-steps, which are required by the AB4 method, can be obtained in a variety of ways, including 1) Use of the explicit Runge–Kutta method of order 4, with intermediate-step boundary-conditions enforced as in [27], and 2) Use of a first order (e.g. forward Euler) method in conjunction with Richardson extrapolation of sufficiently high order [10]. In all of the examples considered in this paper the initial solution values used were directly obtained from the context—since for the cases considered the solution either ramps-up from zero, in which case the solution values at the three initial time-steps can be taken to equal zero, or is given by an explicit manufactured solution which can be used even before the initial time $t = 0$.

Although the FC error is of the same order as the corresponding error resulting from, say, a finite-difference (FD) solver based on a six-point stencil, the FC methodology provides a significant advantage: it is essentially dispersionless. As is well known, wave propagation by means of FD schemes leads to an accumulation of errors that compound over the length of the domain. These accumulating "dispersion errors" demand a significant increase in the number of points per wavelength (PPW) to resolve the solution for a given accuracy. This behavior is also found in Finite Element Methods and usually called "pollution errors" in that context [6]. A detailed discussion in these regards, including numerical results and comparisons of the dispersion characteristics of the FC method with a Finite Difference scheme is put forth in Section 7.2.

5. Complex geometries

5.1. Curvilinear coordinate systems

One of the main advantages of the FC method lies in its use of uniform meshes—for which, as demonstrated in [2], the FC differentiation operator has optimal spectral radius, and for which, in particular, the CFL constraints scale linearly with the size of spatial discretizations (and not quadratically, as do certain other spectral algorithms such as those based on the use of Chebyshev polynomials). Cartesian meshes in physical space cannot capture the geometry of an object of interest with high-order accuracy, however, unless the object has particularly simple shapes: any curvature on an object boundary precludes the use of Cartesian meshes in physical space if high-order accuracy is required. In order to enable applicability to general domains Ω the proposed algorithm relies on use of a decomposition of Ω as a union $\Omega = \cup\Omega_i$ of separate but overset patches Ω_i (see Section 5.3), each one of which is in turn mapped via a domain mapping

$$\mathcal{M}_i : [0,1]^3 \to \Omega_i \tag{30}$$

from the Cartesian domain $[0,1]^3$ into Ω_i. (In order to maintain continuity of the solution and its derivatives across patches, an interpolation strategy is used, as detailed in Section 5.3.)

In explicit coordinates the mapping $\mathcal{M}_i$ is given by the position vector of three scalar functions $\mathbf{x}_i = (x_i^1, x_i^2, x_i^3)^T$ of three independent variables $\mathbf{q} = (q^1, q^2, q^3)^T$:

$$\mathcal{M}_i = \mathbf{x}_i : [0,1]^3 \to \Omega_i. \tag{31}$$

A two dimensional example of such a mapping is presented in Fig. 3.

The patches Ω_i may be rectangular (e.g. for certain regions in Ω away from domain boundaries) or truly curvilinear (necessarily so for patches that must conform to curved domain boundaries). The curvilinear boundary-conforming patches can be constructed in a number of ways: either analytically (for simple geometries), algebraically (on the basis of transfinite interpolation [16]), on the basis of PDE solvers (as is the case in elliptic mesh generation [17]), or via combinations thereof [34]. Such patches are endowed with Cartesian-like discretizations: the coordinate lines in a given patch are images of Cartesian coordinate lines in parameter space. Each map is required to be locally invertible with a positive, non-singular Jacobian of transformation that preserves the mathematical type of the PDE [23].

5.2. The governing equations in curvilinear coordinates

In order to obtain the curvilinear formulation of our algorithm we consider the chain rule expression

$$\nabla_{\mathbf{q}} = [J_{\mathbf{x}}(\mathbf{q})]^T \nabla_{\mathbf{x}}, \tag{32}$$

where $(J_{\mathbf{x}}(\mathbf{q}))_{ij} = \partial x^i/\partial q^j$ is the Jacobian matrix of the given mapping $\mathbf{x}(\mathbf{q})$. Provided $\det((J_{\mathbf{x}})(\mathbf{q}))$ does not vanish, inversion of this linear system gives the expression [23]

$$\nabla_{\mathbf{x}} = \left[(J_{\mathbf{x}}(\mathbf{q}))^{-1}\right]^T \nabla_{\mathbf{q}} = \left[J_{\mathbf{q}}(\mathbf{x})\right]^T \nabla_{\mathbf{q}}, \tag{33}$$

where the last equality results from the reverse chain rule formula $\nabla_{\mathbf{x}} = \left[J_{\mathbf{q}}(\mathbf{x})\right]^T \nabla_{\mathbf{q}}$. Thus "metric derivatives" $\partial q^i/\partial x^j$ can be produced in terms of the derivatives of the given mapping $\partial x^i/\partial q^j$. The necessary derivatives, in turn, can be computed directly from analytical expressions for the functions $\mathbf{x}_i$ in (31), if such expressions are available and their differentiation in closed form is not unduly cumbersome. Alternatively, the metric derivatives can obtained from a discrete set of values of the functions $\mathbf{x}_i$ on a Cartesian mesh in parameter space, via a straightforward application of the FC method. In order to maintain the overall order of accuracy of the time-stepping algorithm under the differentiation operation required by the metric derivatives, the FC construction should be based on Gram polynomials of degrees higher, by at least one unit, than the corresponding Gram polynomials used in the FC representation of the solution—since the differentiation process inherent in the metric derivatives reduces the order of accuracy by one unit. For all of the examples considered in this paper analytical expressions of the metric derivatives were used.

The curvilinear form of the Cauchy–Navier equations follows from (33): letting

$$\widetilde{\mathbf{u}}(\mathbf{q},t) = \mathbf{u}(\mathbf{x}(\mathbf{q}),t),\;\; \widetilde{\rho}(\mathbf{q}) = \rho(\mathbf{x}(\mathbf{q})),\;\; \widetilde{\lambda}(\mathbf{q}) = \lambda(\mathbf{x}(\mathbf{q})),\;\; \widetilde{\mu}(\mathbf{q}) = \mu(\mathbf{x}(\mathbf{q})),\;\; \widetilde{J}_{\mathbf{q}}(\mathbf{q}) = J_{\mathbf{q}}(\mathbf{x}(\mathbf{q})), \tag{34}$$

equation (1) can be re-expressed in the form (cf. [23])

$$\sqrt{g}\,\widetilde{\rho}\,\widetilde{\mathbf{u}}_{tt} = \left(\widetilde{J}_{\mathbf{q}}^{\,T}\nabla_{\mathbf{q}}\right)\cdot\left[\widetilde{\mu}\left(\widetilde{J}_{\mathbf{q}}^{\,T}\nabla_{\mathbf{q}}\widetilde{\mathbf{u}} + (\nabla_{\mathbf{q}}\widetilde{\mathbf{u}})^T\widetilde{J}_{\mathbf{q}}\right) + \widetilde{\lambda}\left(\left(\widetilde{J}_{\mathbf{q}}^{\,T}\nabla_{\mathbf{q}}\right)\cdot\widetilde{\mathbf{u}}\right)I\right] + \sqrt{g}\,\widetilde{\mathbf{f}}, \tag{35}$$

where $\sqrt{g} = \det \widetilde{J}_{\mathbf{q}}$ and $\widetilde{\mathbf{f}}(\mathbf{q},t) = \mathbf{f}(\mathbf{x}(\mathbf{q}),t)$. The initial data, in turn, is given by

$$\widetilde{\mathbf{u}}(\mathbf{q},t_0) = \widetilde{\mathbf{a}}(\mathbf{q}), \qquad \widetilde{\mathbf{u}}_t(\mathbf{q},t_0) = \widetilde{\mathbf{b}}(\mathbf{q}), \tag{36}$$

where $\widetilde{\mathbf{a}} = \mathbf{a}(\mathbf{x}(\mathbf{q}))$ and $\widetilde{\mathbf{b}} = \mathbf{b}(\mathbf{x}(\mathbf{q}))$.

In the proposed method, the q^1, q^2, q^3 derivatives required in this formulation are obtained on the basis of the Fourier continuation method (Section 5.2.1). Boundary conditions of the problem can be applied easily in the (q^1, q^2, q^3) coordinates. For instance, a Dirichlet boundary condition (4) on the planar Cartesian square $q^1 = 0$ is given by

$$\widetilde{\mathbf{u}}(0,q^2,q^3,t) = \widetilde{\mathbf{c}}(q^2,q^3,t), \tag{37}$$

where $\widetilde{\mathbf{c}}(q^2,q^3,t) = \mathbf{c}(\mathbf{x}(0,q^2,q^3),t)$. Similarly, the traction boundary conditions (5) at $q^1 = 0, 1$ are expressed in the form

$$\widetilde{\boldsymbol{\sigma}}\cdot\mathbf{n} = \widetilde{\mathbf{d}}(\mathbf{q},t) \quad \text{on } \partial\Omega, \tag{38}$$

where $\widetilde{\sigma}_{ij} = c_{ijk\ell}\sum_{m=1}^{3}\frac{\partial q^m}{\partial x^j}\frac{\partial \widetilde{u}^i}{\partial q^m}$ and where $\mathbf{n}(\mathbf{q}) = (n^1,n^2,n^3)^T$ is the unit inner normal—which is given by $\pm\nabla_{\mathbf{x}}q^1/\|\nabla_{\mathbf{x}}q^1\|$ on the surfaces at $q^1 = 0$ and $q^1 = 1$, respectively. The numerical treatment and application of boundary conditions of the form of (38) is detailed in Section 6.3.

5.2.1. The discrete curvilinear formulation

A discretization of each patch is obtained, simply, by using a Cartesian parameter-space mesh

$$q^i_j = j\Delta q^i, \quad 0 \le j \le N_{q^i} - 1, \quad i = 1,2,3, \tag{39}$$

containing a total of $N_{q^1}\cdot N_{q^2}\cdot N_{q^3}$ discretization points in $[0,1]^3$—so that the resulting uniform parameter-space mesh-sizes in each dimension equal to $\Delta q^i = 1/(N_{q^i} - 1), i = 1,2,3$. The mapping defined by (31) carries curvilinear mesh points (q^1_ℓ, q^2_m, q^3_n) to physical space mesh points

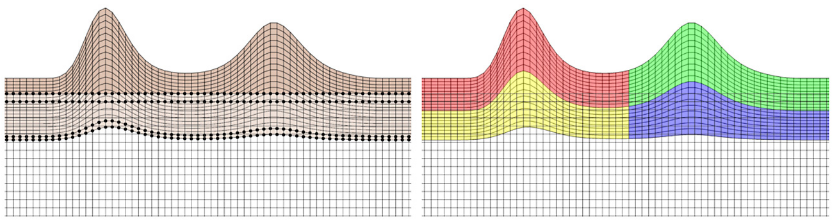

Fig. 4. Left: A portion of a physical domain covered by two patches where, for demonstration, the bold dots indicate two layers of interpolation points. (For all of the numerical examples presented in this paper four layers of interpolation points were used.) Right: One such curvilinear patch decomposed into four sub-patches.

$$\mathbf{x}_{\ell mn} = \mathbf{x}(q^1_\ell, q^2_m, q^3_n), \quad 0 \le \ell \le N_{q^1} - 1, \quad 0 \le m \le N_{q^2} - 1, \quad 0 \le n \le N_{q^3} - 1, \tag{40}$$

and displacement solutions in curvilinear coordinates $\widetilde{\mathbf{u}}_{\ell mn} = \widetilde{\mathbf{u}}(q^1_\ell, q^2_m, q^3_n)$ to physical space solutions

$$\mathbf{u}_{\ell mn} = \mathbf{u}\left(\mathbf{x}\left(q^1_\ell, q^2_m, q^3_n\right), t\right) = \widetilde{\mathbf{u}}_{\ell mn}. \tag{41}$$

The discrete Fourier continuation method detailed in Section 3 can be applied to compute the necessary spatial first and second derivatives for each component of $\widetilde{\mathbf{u}} = (\widetilde{u}^1, \widetilde{u}^2, \widetilde{u}^3)^T$ with respect to each component of the curvilinear coordinates $\mathbf{q} = (q^1, q^2, q^3)^T$ via Fast Fourier Transforms.

From (40) we see that the boundary discretization points at the left, right, bottom, top, backward, and forward faces of the parameter cube (where boundary conditions are enforced) are characterized by the indices (ℓ, m, n) respectively as

$$(0, m, n), (N_{q^1}-1, m, n), (\ell, 0, n), (\ell, N_{q^2}-1, n), (\ell, m, 0), (\ell, m, N_{q^3}-1), \tag{42}$$

where $0 \le \ell \le N_{q^1}-1, 0 \le m \le N_{q^2}-1$ and $0 \le n \le N_{q^3}-1$. As intended, in view of the boundary-conforming curvilinear transformation, boundary conditions are applied—with high-order accuracy and for an arbitrarily complex surface—at either the first or last index in each dimension in the corresponding data structure.

5.3. A parallel block decomposed, overlapping grid strategy

For a general physical domain the final discretization strategy adopted by our elasticity solver (see Fig. 4) consists of two key components [2,3,9,21]: 1) A decomposition of the domain Ω into a collection of overlapping curvilinear patches $\Omega_1, \Omega_2, \ldots, \Omega_M$ (introduced in Section 5.1) whose solutions are communicated across overlapping patch boundaries by means of a high-order polynomial interpolation (Section 5.3.1); and 2) A further decomposition of each curvilinear patch Ω_k into mutually disjoint "sub-patches" that are then extended by shared "fringe regions" which are utilized to communicate information between subpatches in a distributed parallel computing environments (Section 5.3.2). To facilitate an efficient implementation of element 2), in Section 5.3.3 we additionally put forth a simple load-balancing algorithm which, as demonstrated in Section 7.3, gives rise to excellent parallel scaling. The two main components 1) and 2) of our overall discretization strategy are described in detail in Sections 5.3.1 and 5.3.2.

5.3.1. Overset meshes and artificial "inter-patch" boundaries

As mentioned in Section 5.1, the proposed algorithm treats general geometries via decomposition of the computational domain Ω as a union of a finite number of overlapping curvilinear patches, $\Omega = \bigcup_j \Omega_j$, within each one of which the PDE (35) is evolved; continuity and smoothness across patches is ensured by exchange of solution values between patches in regions near patch boundaries. Fig. 4 (left) shows an example of part of a computational domain that is covered by two curvilinear patches, with points that are recipient of interpolation data shown as bold dots. In this domain-decomposition method, commonly known as the *overset grid method* [9], a patch $\Omega_k \subset \Omega$ exchanges information with adjacent patches $\Omega_j \subset \Omega$ by means of a suitable interpolation scheme—as described in what follows.

By construction, subsets of a patch boundary $\partial\Omega_k$ that do not coincide with a physical boundary of the computational domain Ω are necessarily contained within one or more of the other patches that make up the overall domain. Along these boundaries, solution values from neighboring patches are interpolated prior to a time-step by means of a tensor-product polynomial. In this paper we employ stencils of size $7 \times 7 \times 7$ points (that is, 6th-order polynomial interpolation in each dimension) and interpolate to a four-point wide layer of points around the overlapping boundary (cf. Fig. 4 and its caption). It is assumed that the overlapping regions between patches are "sufficiently" large, so that interpolation stencils are contained sufficiently far away from the respective donor patch boundary (to minimize differentiation and approximation errors which are largest at patch boundaries). And, to maximize the benefits of the interpolation procedure, given a point $\mathbf{x} = (x^1, x^2, x^3) \in \Omega_k$, interpolation is performed from an adjacent donor patch Ω_j for which $\mathbf{x} \in \Omega_j \cap \Omega_k$ is farthest from the boundary Ω_j amongst all patches Ω_j $(j \ne k)$ that contain the point $\mathbf{x}$. Once such a patch has been obtained, a corresponding seven-point stencil is located in patch Ω_j for which $\mathbf{x}$ lies as close to its center as possible. Interpolation is then performed in the corresponding (q^1, q^2, q^3) parameter space of the patch Ω_j via a straightforward application of Neville's algorithm [28]. Additional details in these regards are provided in [4].

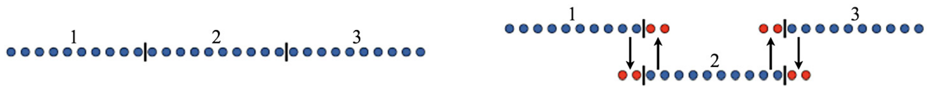

Fig. 5. One dimensional line segmentation analogous to the multidimensional parallel decomposition of a patch.

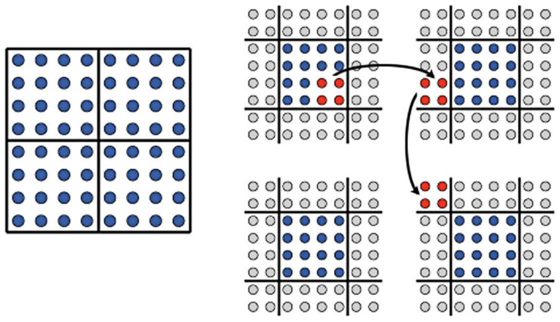

Fig. 6. Left: A two-dimensional example of a square patch decomposed into four disjoint sub-patches. Right: The four sub-patches augmented by fringe regions and an example of the transmission of data to corner patches.

5.3.2. Parallel decomposition and artificial "intra-patch" boundaries

A parallel implementation of the FC solver described in this paper can be produced by means of the overset mesh decomposition described in the previous section. In order to incorporate large numbers of processing computer cores, each curvilinear patch can be further decomposed into a union of disjoint sub-patches within each one of which the PDE (35) is evolved. At each time-step, solution values in each one of the sub-patches are used in conjunction with solution values on a certain number of fringe discretization points of neighboring sub-patches. This concept is illustrated by a simple one-dimensional example in Fig. 5: in this simplified example the center sub-domain (sub-patch) has (only) two shared points (which, following [2] we call "fringe points") on either side; the left and right boundary sub-domains contain fringe points on one side only. In our actual algorithm implementation a total of four fringe points in each dimension are used for each interior boundary.

Remark 5.1. In order to avoid undue degradation of the excellent dispersion characteristics of the FC differentiation scheme, the FC algorithm is set up to use $d_\ell, d_r = 12$ discretization points in the Gram polynomial projection interval at interior boundaries. (At such intra-patch boundaries polynomial interpolation of very high-order, which can be used while maintaining stability, is utilized in our method to accurately communicate the solution across the artificial boundaries and thus preserve the overall global accuracy of the method.) In order to maintain stability, on the other hand, at all other (physical) boundaries the values $d_\ell, d_r = 5$ are used. Numerical experiments have suggested that use of higher orders at physical boundaries leads to instability (an observation which is consistent with [2]).

The multidimensional version of the shared (fringe) region strategy is demonstrated in Fig. 6: the Fourier continuations needed to apply the differentiation algorithm along a vertical or horizontal line in a sub-patch are obtained by applying FC to the combination of discrete values at the light-gray fringe points and the black sub-patch points. As additionally illustrated in the figure, the fringe discretization region associated with a sub-patch may extend not only into the six neighbors with which the sub-patch shares a face in three-dimensions but also into corner-adjacent neighboring sub-patches, for up to twenty-six neighbors. The potential additional communication required for all neighbors can be circumvented by communicating the fringe information in one coordinate direction at a time (as demonstrated for a two-dimensional case in the right-hand figure), so that information with corner-adjacent sub-patches is indirectly passed along, thus reducing the overall communication load: this allows the full exchange of information to be carried out by only six send-and-receive calls from a sub-patch with its (up to six) face-sharing neighbors.

For a parallel implementation in which a processing core is responsible for advancing the solution of a single sub-patch, the solution values for the assigned sub-patch are stored together with the values in the fringe discretization regions. The fringe regions are subsequently updated at each time-step to allow for transmission of information across component sub-patches to take place via neighbor-to-neighbor parallel communication. This procedure is akin to the one used in connection with the "artificial" interpolating inter-patch boundaries considered in Section 5.3.1, but here no interpolation is necessary—since neighboring sub-patches share the same discretization grid of the underlying patch. For all simulations in this paper we have employed four points in the fringe intervals in each dimension.

Remark 5.2. Periodic boundary conditions on curvilinear patches such as annuli can be treated by means of the decomposition method described above. Indeed, it suffices to use fringe points on each end of a periodic interval and a communication strategy analogous to the one described above—as demonstrated in Fig. 7.

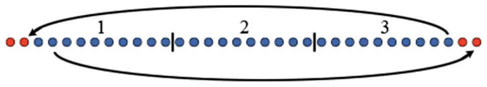

Fig. 7. One dimensional line segmentation for periodic boundary conditions.

5.3.3. *Load balancing*

An efficient parallel decomposition strategy should give rise to minimal communication between computing cores, on one hand, and to a balanced overall workload, on the other. In our approach, one core is assigned to each sub-patch. In order to determine the number of sub-patches (cores) to be used along each dimension of a given patch the algorithm proceeds in two steps: the total number of available cores are distributed to the various patches in direct proportion to the size of the associated discretizations; each patch is then subdivided into a number sub-patches that are balanced in an effort to optimize the execution of FFTs—as indicated in what follows.

Given a number of patches $\Omega_1, \ldots, \Omega_M$ that cover the computational domain, and a number p_{total} of cores assigned to the solver, then each patch Ω_k is assigned at least

$$p^k = \left\lfloor \frac{N^k_{q^1} N^k_{q^2} N^k_{q^3}}{\sum_{j=1}^M N^j_{q^1} N^j_{q^2} N^j_{q^3}} p_{total} \right\rfloor \tag{43}$$

cores ($k = 1, 2, \ldots, M$), where $\lfloor \cdot \rfloor$ is the floor function and where $N^k_{q^1}, N^k_{q^2}, N^k_{q^3}$ are the number of discretization points along each side of the computational domain Ω_k. The remaining $\left(p_{total} - \sum_{k=1}^M p^k \right)$ cores either remain un-utilized or are redistributed over all sub-domains on the basis of some adequate ad-hoc criterion.

The number of cores now assigned to a curvilinear patch Ω_k is then factored as $p^k = p^k_{q^1} p^k_{q^2} p^k_{q^3}$ where $p^k_{q^1}, p^k_{q^2}, p^k_{q^3} \geq 1$ denote the number of cores assigned in the q^1, q^2, q^3 directions, respectively. The sizes $p^k_{q^j}$ of the blocks used in each dimension should be balanced so as to achieve good parallel scaling; this can be accomplished by selecting $(p^k_{q^1}, p^k_{q^2}, p^k_{q^3})$ which minimize the quantity

$$\min_{p^k_{q^1} p^k_{q^2} p^k_{q^3} = p^k} \left| N^k_{q^1}/p^k_{q^1} - N^k_{q^2}/p^k_{q^2} \right| + \left| N^k_{q^2}/p^k_{q^2} - N^k_{q^3}/p^k_{q^3} \right| + \left| N^k_{q^1}/p^k_{q^1} - N^k_{q^3}/p^k_{q^3} \right|. \tag{44}$$

This procedure results in patches that contain approximately the same number of discretization points in each one of the directions q^i: $N^k_{q^1}/p^k_{q^1} \approx N^k_{q^2}/p^k_{q^2} \approx N^k_{q^3}/p^i_{q^3}$. In addition to facilitating load balancing, this strategy results in meshes of approximately equal sizes. Thus, adding a small number E of extra points to each continuation region (see Section 3.1) the added advantage is obtained that FFTs of fixed adequately-selected sizes can be used across the computational domain, as discussed in [2,4].

6. Implementation details

This section completes the full description of the proposed FC-based linear elasticity solver for general domains, including a description of the explicit time evolution (Section 6.1) and a spectral filter which, without unduly deteriorating the solver's accuracy, eliminates growth of high-frequency errors (Section 6.2). Our methodology for treatment of traction boundary conditions, which is based on use of expressions for relevant out-of-plane directional derivatives in terms of in-plane derivatives, is discussed in Section 6.3, and a pseudo-code for the overall algorithm is presented Section 6.4.

6.1. *Explicit treatment of temporal derivatives*

Following [2], our solver utilizes the Adams–Bashforth scheme of order four (AB4) for time-stepping; see also Remark 4.1. A natural alternative, the explicit fourth-order Runge–Kutta (RK4) method, has also been considered in the context of FC solvers. Both methods provide adequate regions of absolute stability [7,20]. The RK4 method only requires initialization of the first step, but each subsequent time-step entails four evaluations of the right-hand-side. Proper enforcement of boundary conditions at intermediate RK steps, further, may be problematic—particularly so for time-dependent boundary conditions [1,14]. The AB4 method, in contrast, requires only one evaluation of the right-hand-side and straightforward application of boundary conditions, but it requires initialization of the first three steps. The latter task is trivially accomplished in most relevant contexts, in which sources ramp up from zero (carrying zero initial displacement)—thus enabling use of solution values identically equal to zero for the first three time-steps. In all, the AB4 approach seems advantageous in the present context, and it was thus used in all of the numerical examples presented in this paper.

Expressing the second-order-in-time elastic wave equation (1) in the form

$$\widetilde{\mathbf{u}}_{tt} = \widetilde{\mathbf{F}}\left(\widetilde{\mathbf{u}}, \nabla_{\mathbf{q}} \widetilde{\mathbf{u}}, \nabla_{\mathbf{q}} \cdot \widetilde{\mathbf{u}}, \mathbf{q}, t\right) \tag{45}$$

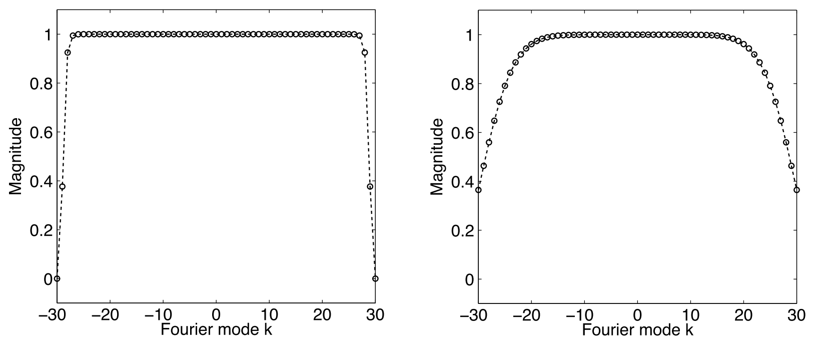

Fig. 8. Depictions of the exponential filter with parameter values given by $(p, \alpha) = (3N/5, 16\log(10))$ (left) and $(p, \alpha) = (4, -c_L\Delta t/h_{min}\ln(10^{-2}))$ (right).

(cf. Section 5.2) and letting

$$\widetilde{\mathbf{g}}(\mathbf{q}, t) = \widetilde{\mathbf{u}}_t(\mathbf{q}, t),$$

the PDE under consideration can be recast as the first-order system

$$\begin{cases} \widetilde{\mathbf{u}}_t(\mathbf{q}, t) = \widetilde{\mathbf{g}}(\mathbf{q}, t), \\ \widetilde{\mathbf{g}}_t(\mathbf{q}, t) = \widetilde{\mathbf{F}}\left(\widetilde{\mathbf{u}}, \nabla_{\mathbf{q}}\widetilde{\mathbf{u}}, \nabla_{\mathbf{q}} \cdot \widetilde{\mathbf{u}}, \mathbf{q}, t\right) \end{cases} \tag{46}$$

for unknowns $\widetilde{\mathbf{u}} = (u^1, u^2, u^3)^T$ and $\widetilde{\mathbf{g}} = (g^1, g^2, g^3)^T$. Integration of this system in time on the basis of the fourth-order explicit Adams–Bashforth discretization with uniform time-step $\Delta t > 0$ then yields the fully discrete equations we use. With an easy-to-understand notational license the corresponding time-step is given by

$$\begin{aligned} \widetilde{\mathbf{g}}(t + \Delta t) &= \widetilde{\mathbf{g}}(t) + \frac{\Delta t}{24}\left[55\widetilde{\mathbf{F}}(t) - 59\widetilde{\mathbf{F}}(t - \Delta t) + 37\widetilde{\mathbf{F}}(t - 2\Delta t) - 9\widetilde{\mathbf{F}}(t - 3\Delta t)\right], \\ \widetilde{\mathbf{u}}(t + \Delta t) &= \widetilde{\mathbf{u}}(t) + \frac{\Delta t}{24}\left[55\widetilde{\mathbf{g}}(t) - 59\widetilde{\mathbf{g}}(t - \Delta t) + 37\widetilde{\mathbf{g}}(t - 2\Delta t) - 9\widetilde{\mathbf{g}}(t - 3\Delta t)\right]. \end{aligned} \tag{47}$$

Dirichlet and traction boundary conditions (cf. Section 6.3) are injected after each time-step $t_n \to t_{n+1}$; the resulting normal derivatives and function boundary values at time t_{n+1} are then used to produce the FC approximations needed to evaluate the right hand side $\widetilde{\mathbf{F}}$ at time t_{n+1} and the time-stepping iteration is thus complete.

6.2. Frequency space filters

Like other spectral and finite-difference solvers, our algorithm relies on use of a spatial filter to control error growth in unresolved modes; cf. [2] and references therein. In keeping with the character of the FC method the filter we utilize is applied in Fourier space: using the function

$$\sigma\,(2k/N) = \exp\left(-\alpha\,(2k/N)^{2p}\right), \tag{48}$$

the filtering operation on a function $u(x)$ with Fourier coefficients $\hat{u}_k$ is given by

$$\sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \hat{u}_k \exp(ikx) \quad \longrightarrow \quad \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \sigma\,(2k/N))\,\hat{u}_k \exp(ikx). \tag{49}$$

The positive integer p controls the rate of decay of the filter coefficients, and the real parameter α determines the level of suppression: the highest-frequency modes are multiplied by $e^{-\alpha}$.

The filter σ with parameters $(p, \alpha) = (3N/5, 16\log(10))$ (which were used in [2] as part of a Navier–Stokes FC solver) is displayed in the left portion of Fig. 8. We have found that, in absence of the viscous terms inherent in the Navier–Stokes equations, error discontinuities arising in neighboring patches and sub-patches can give rise to instability—as the Fourier series expansions in segments from different overlapping patches may be made overly inconsistent by the filtering procedure. The impact of such inconsistencies, which is also mentioned in [15], is demonstrated in the left portion of Fig. 9.

A "milder" choice of parameters (for which the highest frequency terms aren't entirely eliminated) is used in this paper, namely

$$(p, \alpha) = (4, -c_{max}\Delta t/h_{min}\ln(10^{-2})), \tag{50}$$

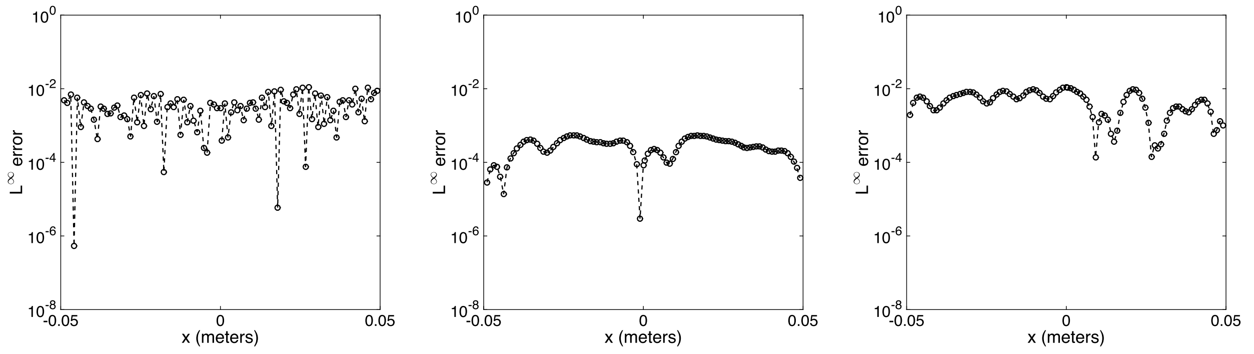

Fig. 9. Numerical errors along the x-axis for plane waves in a thin plate of dimensions 100 mm × 20 mm × 20 mm. Left: Errors after 10,000 time-steps for filter parameters values $(p, \alpha) = (3N/5, 16\ln(10))$; by 15,000 time-steps, the magnitude of these errors is higher by several orders of magnitude. Middle: Errors after 10,000 time-steps for filter parameters values $(p, \alpha) = (4, -c_L\Delta t/h_{min}\ln(10^{-2}))$. Right: Errors after 500,000 time-steps for the latter filter parameter values.

where $c_{max} = \max\{c_L, c_T\}$ is the maximum wave speed in the material, and where h_{min} is the finest spatial step size throughout the computational domain. In particular, this choice of parameters, inspired by a similar choice in [15], ensures that the filter approaches unity as $\Delta t \to 0$ for a fixed Δx. As seen in the right graph of Fig. 8, this filter does not eliminate the highest modes completely but more smoothly forces the decay of the Fourier coefficients. This filter was used in all of the numerical examples presented in this paper. The usefulness of this filter is illustrated in Fig. 9, which results from a simulation of plane waves $\sin(25\pi(x^1 - c_L t))$ in a plate of dimensions 100 mm × 20 mm × 20 mm with Dirichlet boundary conditions, and with material parameters $\rho = 1, \lambda = 2.173 \times 10^4$ and $\mu = 9.630 \times 10^3$. The domain is constructed as a union of two overlapping patches of dimensions 50 mm × 20 mm × 20 mm ($N = 48 \times 16 \times 16$) and 70 mm × 20 mm × 20 mm ($N = 72 \times 16 \times 16$) whose discretizations do not coincide on the overlap (so that interpolation is necessary) and which are distributed into six and ten parallel sub-patches, respectively. After 10,000 time-steps at $\Delta t = .1$ µs, the left image shows some significant errors (that occur mainly at patch and subpatch boundaries). By 15,000 time-steps the magnitude of these errors is higher by several orders of magnitude. With the filter parameters used to produce the middle portion of the figure and at the same Δt value, on the other hand, the errors remain as shown in the figure for 15,000 time-steps—and, indeed, at least up to 500,000 time-steps, as demonstrated in the rightmost portion of Fig. 9.

Remark 6.1. The Δt value used above in this section coincides with the value given by the expression $\Delta t_{\text{stab}} = \frac{\Delta x_{min}}{32(c_L + c_T)}$ for the parameters of the problem under consideration. The solver presented in this paper has consistently demonstrated stability, over a wide range of material parameters and geometries, for all $\Delta t \le \Delta t_{\text{stab}}$ and for runs including several hundreds of thousands of time-steps. Naturally, smaller values of Δt may be needed for accuracy under certain circumstances and, conversely, larger values of Δt could be used to advantage for some configurations.

6.3. Treatment of traction boundaries in curvilinear coordinates

Traction boundary conditions can be enforced by solving for the relevant normal derivatives in terms of tangential derivatives, upon which the Neumann FC operator introduced in Section 3.2 can be applied. For example, the traction boundary condition (38) on the face $\{(q^1, q^2, q^3) : q^1 = 0\} \subset [0, 1]^3$ can be re-expressed as an identity between the out-of-plane derivatives $\partial\widetilde{u}^1/\partial q^1$, $\partial\widetilde{u}^2/\partial q^1$ and $\partial\widetilde{u}^3/\partial q^1$ and the in-plane derivatives $\partial\widetilde{u}^1/\partial q^2$, $\partial\widetilde{u}^1/\partial q^3$, $\partial\widetilde{u}^2/\partial q^2$, $\partial\widetilde{u}^2/\partial q^3$, $\partial\widetilde{u}^3/\partial q^2$ and $\partial\widetilde{u}^3/\partial q^3$ at $(0, q^2, q^3)$. Once they have been determined, every invocation of the FC algorithm for the relevant *first* derivatives in the right hand side of (35) at each time-step, i.e. in the computation of $\partial\widetilde{u}^1/\partial q^1(\mathbf{q}, t)$, $\partial\widetilde{u}^2/\partial q^1(\mathbf{q}, t)$ and $\partial\widetilde{u}^3/\partial q^1(\mathbf{q}, t)$ throughout the domain, employs the Neumann operator of Section 3.2 using, for example, the d matching points given by

$$\left(\widetilde{u}^1(q^1_d, q^2, q^3), \widetilde{u}^1(q^1_{d-1}, q^2, q^3), \ldots, \widetilde{u}^1(q^1_1, q^2, q^3), \frac{\partial\widetilde{u}^1}{\partial q^1}(0, q^2, q^3)\right)^T. \tag{51}$$

This portion of the algorithm is placed in the context of the overall FC methodology in the algorithm pseudo-code presented in Section 6.4.

Remark 6.2. The use of the filtering procedure described in the previous section is restricted in our algorithm to the evaluation of the right hand side of the elastic wave equation (35), and it is not used in the computation of the in-plane derivatives in the treatment of traction boundary conditions. In fact, use of filtering for evaluation of the traction boundary conditions, which is not necessary for stability, decreases the consistency of the Neumann boundary values and the solution values in the interior of the computational domain, and thus gives rise to larger errors near the traction boundaries than can otherwise be obtained.

6.4. Overall algorithm pseudo-code

The brief pseudo-code below (Algorithm 1) summarizes the proposed FC-based elasticity solver.

Algorithm 1 Summary of the full FC-based numerical PDE solver for the elastic wave equation.

```
1: for Each time-step $n$ do
2:    for All patches $\Omega_i$ do
3:       for All sub-patches $\omega_j \in \Omega_i$, each assigned to a core do
4:          Evaluate the directional derivatives needed to enable enforcement of the traction boundary conditions as indicated in Section 6.3
            (if they exist in a given problem) at time $t = t_n$
5:          Advance the solution throughout the domain $\omega_j$ and its boundary $\partial\omega_j$ to time $t_{n+1}$ via Equations (47). (Traction conditions
            are enforced by using the directional derivatives with the FC Neumann operator in the computation of the right hand sides
            $F^1, F^2, F^3$.)
6:          Update physical Dirichlet boundary values at $t_{n+1}$ (enforced by direct injection of the given Dirichlet boundary values.)
7:          if $\omega_j$ has neighboring sub-patches $\{\omega_k\} \in \Omega_m,\ m = i$ then
8:             Exchange fringe points solution values with $\{\omega_k\}$ via MPI
9:          end if
10:         if $\omega_j$ has neighboring (sub)patches $\{\omega_k\} \in \Omega_m,\ m \neq i$ then
11:            Interpolate and exchange values with $\{\omega_k\}$ via MPI
12:         end if
13:      end for
14:   end for
15: end for
```

7. Performance studies

This section presents a variety of numerical experiments that demonstrate the qualities of the elasticity solver presented above in this text, including the algorithm's convergence and dispersion properties, its stability and its parallel performance.

7.1. Accuracy and convergence order

A verification of the correctness and numerical accuracy of the proposed numerical scheme for the full elasticity equations (35) and its C++ implementation was performed by the method of manufactured solutions (MMS); similar verification procedures have been used extensively in the literature, see e.g. [29,31,35]. In the MMS a smooth solution is postulated and a right-hand forcing term and boundary conditions are then determined that indeed make the postulated function an exact solution of the problem. For our example we consider the solution

$$u^i(x^1, x^2, x^3, t) = \sin\left[2\pi f_{x^1} \cdot (x^1 - ct)\right] \sin\left[2\pi f_{x^2} \cdot (x^2 - ct)\right] \sin\left[2\pi f_{x^3} \cdot (x^3 - ct)\right] \tag{52}$$

for $i = 1, 2, 3$ and with parameters $f_{x^1} = f_{x^2} = f_{x^3} = 12$ and $c = \sqrt{5}$ m/s on a cylinder of radius $r = 10$ cm and length of 30 cm filled with a solid of material constants $\rho = 1$, $\lambda = 1$ and $\mu = 2$. For the numerical computations the cylindrical domain is viewed as the union of two overlapping patches Ω_1 (a rectangular box) and Ω_2 (an annular portion of the cylinder)—as depicted in Fig. 10, which also contains a snapshot of the displacement component in the vertical x^2 direction of the solution $\mathbf{u}(\mathbf{x}, t)$. Considering integer multiples of the coarsest spatial discretization used, which contains a total of $N = 30 \times 30 \times 30$ points in Ω_1 and $N = 30 \times 30 \times 80$ points in Ω_2, we advance the simulation on up to 512 cores for five thousand time-steps at a step size of $\Delta t = 0.1$ µs, employing either Dirichlet and traction boundary conditions. The maximum absolute errors among all components of the vector solution (u^1, u^2, u^3) for all time-steps are displayed in Table 2—where the fifth-order accuracy of the algorithm can easily be appreciated.

7.2. Dispersion and stability experiments

In order to demonstrate the dispersion and stability characteristics of the proposed solver by means of wave-propagation test problems for which waves propagate over long distances, we consider a 3D aluminum plate of dimensions 400 mm × 10 mm × 100 mm and Poisson's ratio $\nu = .35$ (corresponding to $\rho = 2700$ kg/m^3, and Lamé parameters $\lambda = 6.049 \times 10^{10}$ N/m^2 and $\mu = 2.593 \times 10^{10}$ N/m^2). The proposed FC algorithm is applied to advance the MMS elasticity solution given by

$$\mathbf{u}(\mathbf{x}, t) = (u^1, u^2, u^3)^T = \left(\sin\left[2\pi n\left(x^1 - c_L t\right)\right], 0, 0\right)^T \tag{53}$$

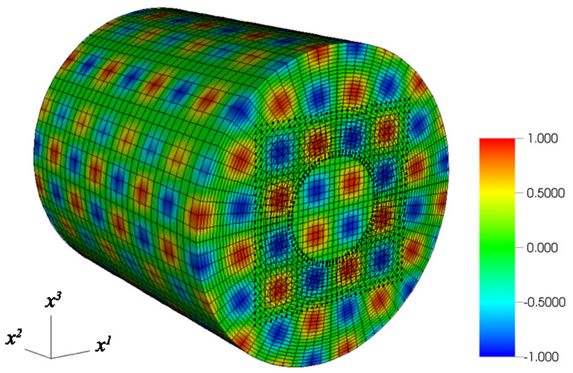

Fig. 10. Numerical values of the vertical displacement produced by the right-hand-side source (52). The bold dots indicate the layers of interpolation points used.

Table 2
Maximum errors in the elastic displacement resulting from use of the proposed FC elasticity solver for the MMS problem considered in Section 7.1. Maximum errors over the full computational domain and over 5000 time-steps, which were calculated by comparison with the MMS exact solution, are displayed for elasticity problems under Dirichlet and Traction boundary conditions (L^{∞}_{err} Dirichlet and L^{∞}_{err} Traction, respectively). The corresponding convergence orders in the infinity norm are presented as well ($O(L^{\infty})$ Dirichlet and $O(L^{\infty})$ Traction). A fine temporal discretization was used so that the overall error is dominated by errors arising from the spatial discretization.

N (patch Ω_1)	N (patch Ω_2)	L^{∞}_{err} Dirichlet	$O(L^{\infty})$ Dirichlet	L^{∞}_{err} Traction	$O(L^{\infty})$ Traction
$30 \times 30 \times 30$	$30 \times 30 \times 80$	2.34e−03	–	2.82e−03	–
$60 \times 60 \times 60$	$60 \times 60 \times 160$	7.89e−05	4.89	8.29e−05	5.09
$90 \times 90 \times 90$	$90 \times 90 \times 240$	9.41e−06	5.24	9.37e−06	5.38
$120 \times 120 \times 120$	$120 \times 120 \times 320$	2.00e−06	5.38	2.00e−06	5.37
$150 \times 150 \times 150$	$150 \times 150 \times 400$	6.51e−07	5.03	6.51e−07	5.03

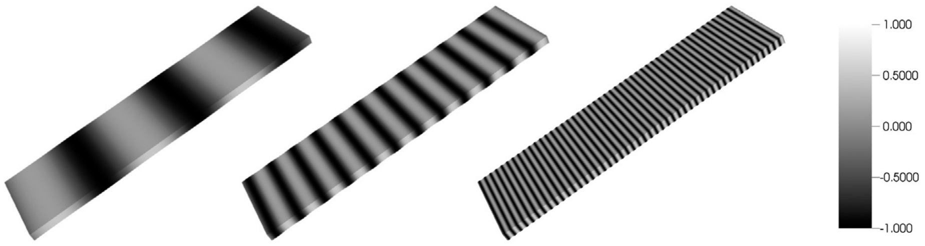

Fig. 11. Numerical solution of equation (53) at time $t = 0.62$ μs for configurations involving $W = 3$, 12 and 48 wavelengths (from left to right) along a thin aluminum plate.

(where $c_L = \sqrt{(\lambda + 2\mu)/\rho}$ is the longitudinal wave speed in meters-per-second), under Dirichlet and traction boundary conditions, and for various wavenumbers n (or, equivalently, for various numbers W of wavelengths along the plate). Illustrations of the computational domain and corresponding solutions are presented in Fig. 11 for $W = 3$, 12 and 48. As shown in what follows, the FC numerical errors are virtually independent of frequency as long as the numbers PPW of points per wavelength in the spatial discretization are kept constant.

The two left graphs in Fig. 12 present maximum numerical errors resulting from applications of the FC solver to the problem described above in this section—under Dirichlet and traction boundary conditions, respectively, over the complete plate and for $0 < t < 62$ μs (the time required for any one crest to travel the length of the plate)—as functions of the number W of wavelengths along the plate (cf. Fig. 11) and for PPW = 15 and PPW = 20. The time-step for these runs was chosen to be sufficiently small ($\Delta t = .45$ ns) to guarantee that the overall numerical errors are dominated by those arising from the spatial discretization. (With reference to Remark 6.1 we have $\Delta_{stab} = 0.436$ ns for the finest discretization considered in this section.) The rightmost graph in Fig. 12, in turn, compares maximum errors that result as a single-patch FC-based solver and a second-order finite-difference (FD) solver are used to evolve the MMS solution $u(x,t) = \sin(2\pi n(x - t))$ of the 1D wave equation $u_{tt} = u_{xx}$ in the domain $0 \le x \le 1$ under Neumann boundary conditions. Clearly, for each fixed value of PPW the accuracy resulting from the FC algorithms remains essentially constant as W grows. In contrast, the accuracy of the FD solution degrades as W grows, and, even for low W values, large numbers of PPW are required to achieve reasonable engineering accuracies. The 3D accuracies demonstrated in the two left figures, which result from simulations that use up to 512 cores (and, thus, 512 patches) for the highest values of W, were obtained by means of the domain decomposition and parallel implementation method described in Section 5.3.2. The use of patch decomposition represents a key difference between the 3D FC runs leading to the left and center graphs and those associated with the 1D example presented in the

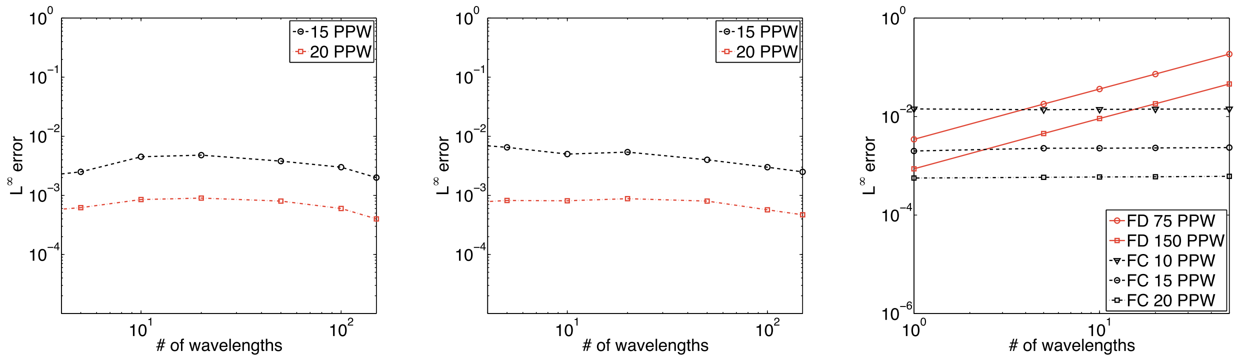

Fig. 12. Maximum numerical errors over all space and over one full temporal cycle (defined as the time required for any one crest to travel the length of the plate) of a plane wave solution with increasing number of wavelengths for the 3D elastic wave equation with Dirichlet and traction boundary conditions (left and center, respectively), and the 1D wave equation (right) using the FC method and a second-order-accurate finite difference method. A single time-step size (under AB4 time-stepping) were used for all the experiments considered in this figure.

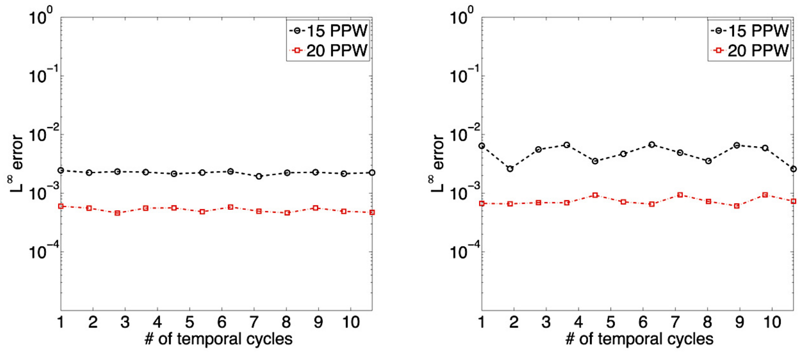

Fig. 13. Numerical errors for FC solutions considered in Section 7.2 over many temporal cycles (where one cycle spans 60,000 time-steps) for a plate ten-wavelengths in length, demonstrating long-time stability for Dirichlet and Traction boundary conditions (left and right images, respectively).

right graph: in view of the particular geometrical structure of the three-d problem under consideration (for which only one core is used on the full thickness and width of the plate) in the two former cases, for a number p_{total} of computing cores, waves that travel along the length of the plate are subject to a total of $2p_{total}$ Gram-polynomial projections—as opposed to the 2 projections in the 1D case. Consideration of Fig. 12 suggests that the use of large numbers of sub-domains and interior sub-patches does not give rise to significant dispersion. Additionally, errors over many temporal cycles of a ten-wavelength solution in Fig. 13 (which resulted from a run involving more than 600,000 time-steps at $\Delta t = 1$ ns) demonstrates the long-term stability of the FC method even in presence of large numbers of sub-patches. (Here a cycle is defined as the time required for any one crest to travel the length of the plate; for example, a cycle is covered in 60,000 time steps for the test cases considered in Fig. 13.)

The aforementioned rightmost graph in Fig. 12 compares error curves that result from an FC scheme discretized by fixed numbers 10, 15 and 20 of PPW and a second-order-in-space FD scheme using 75 and 150 PPW. (It is important to note that even though the given solution is periodic, the FC algorithm does not exploit this property—since it still extends the domain and creates a new periodic extension.) Clearly the number of PPW required to provide fixed accuracies for longer and longer wave-trains remains essentially fixed for the FC approach (that is, the method is essentially dispersionless) but it does not for the FD method. In particular, for large two- and three-dimensional problems, low-order FD approximations tend to require prohibitively large discretizations. Similar comparisons of FC algorithms to fourth- and eighth-order FD schemes, to "spectral-like" high-order Padé schemes, to finite-volume methods of various orders, and to high- and low-order hybrids of Discontinuous Galerkin and Finite Volume methods can be found in [2,3,15,26]; corresponding comparisons for the present FC-based elasticity solver are presented in Section 8.1 below. In each of these studies the FC algorithms demonstrate significant advantages over the alternatives—either as a result of their favorable dispersion characteristics, or their benign CFL time-step restrictions, or both [2,3]. The ability of the FC representations to maintain accuracy over long distances by accurately approximating the dispersion characteristics of the underlying continuous problems—at FFT speeds and without severe CFL constraints—makes the FC solver methodology a highly competitive approach for solution of general Partial Differential Equations in the time domain.

Table 3
CPU-seconds per million unknowns and errors for a domain consisting of a single curvilinear patch (no interpolation), indicating excellent scalability up to at least 480 cores. Left table: weak convergence test, for which both the number of discretization points and the numbers of cores are increased simultaneously. Right table: strong convergence test, wherein for a fixed number of grid points, the number of cores used is increased. Clearly, essentially perfect parallel efficiency is obtained under both weak and strong convergence tests.

# grid pts	# cores	L^∞_{err}	$O(L^\infty)$	S
799,200	120	2.83e−3	–	1.20 s
2,589,408	360	3.00e−4	5.54	1.23 s
4,111,884	480	1.38e−4	5.04	1.30 s

# grid pts	# cores	L^∞_{err}	S
4,111,884	120	2.90e−4	1.34 s
–	360	3.00e−4	1.23 s
–	480	3.06e−4	1.28 s

7.3. Parallel performance and spatial cost asymptotics

The overset patch/sub-patch parallelization and load balancing methods introduced in Sections 5.3.2 and 5.3.3 give rise to an efficient and well-balanced parallel implementation of the FC solver. By augmenting each sub-patch with a fringe region and assigning it to a single core, the proposed methodology enables efficient computation of the needed Fourier transforms using fixed FFT sizes. To demonstrate the parallel scalability, we conducted tests on a computing cluster using 16 cores per node for up to 480 processing cores. The elasticity solver was advanced for 1000 time-steps and each core recorded the maximum absolute error in the solution in the corresponding sub-patch as well as the time spent in local computation—including the time spent in interpolation and communication but excluding the (extremely small) start-up and initialization times.

To analyze the scalability of the solver we first consider a thin-plate test problem with traction boundary conditions, and we record the number S of CPU-seconds required for the solver to advance a million spatial unknowns for one time-step; it is easy to check that S is given by the expression

$$S = \frac{(\text{\# of cores}) \times (\text{total computation time})}{3 \times [(\text{\# of spat. discret. pts.}) + (\text{\# of traction bdry. pts.})]/10^6}, \tag{54}$$

where the factor of 3 stems from the total number of unknowns at each point, i.e. the displacements u_1, u_2 and u_3. Clearly, perfect scaling arises whenever S remains (essentially) constant as the discretization is refined and the number of assigned cores is increased. The S values for an experiment concerning the aforementioned thin plate containing a single patch are given in Table 3; these results illustrate the perfect scalability of the single-patch solver (while maintaining fixed accuracy) up to at least 480 cores/sub-patches.

Remark 7.1. For typical applications of the FC elasticity solver, the expected computational load on each core outweighs the communication load of the fringe regions, and hence this contribution to the overall cost may be safely neglected in a core-allocation analysis.

A second test was conducted to determine the scalability properties of our FC elasticity solver for configurations containing multiple patches—for which an additional computational cost associated with inter-patch interpolation occurs. Use of an appropriate inter-patch interpolation strategy increases the computing cost per million unknowns per core *by a fixed amount*, that is, by an amount that is independent of the number of patches used. This useful property results from the fact that each donor and receiving sub-patch contains a number of interpolation points that does not grow with the overall discretization (as long as the sub-patch discretization sizes remain fixed, which is the recommended strategy for our solver), and, therefore, the total time spent by each core in interpolation procedures is independent of the number of cores used. Furthermore, to mitigate the possible imbalance that may arise from the additional load incurred by sub-patches that require interpolation versus those that don't, all interpolation weights are precomputed on the donor patch during initialization so that all interpolations can be performed locally. This additionally ensures that no stencils are communicated: each boundary sub-patch simply receives, at each time-step, the interpolated u^1, u^2, u^3 values directly from each donor sub-patch and replaces those values in its solution. The corresponding results of an experiment conducted on a plate with a circular through-hole, whose geometry is composed of the six different curvilinear patches detailed in Section 8.2, is displayed in Table 4. Clearly, our methodology maintains the excellent parallel scaling and consistent accuracy previously observed in Table 3 for the single patch configuration, and only a small additional amount of computing time is required by the necessary inter-patch interpolation.

It is interesting to note that the data presented in Tables 3 and 4 also demonstrate the linear growth of the computational cost as a function of the size of the spatial discretization—in both single-patch and multi-patch contexts: consideration of equation (54) shows that a constant value of the parameter S, as demonstrated in these tables, implies, in particular, that the computational cost grows linearly with the size of the discretization mesh.

Table 4
CPU-seconds per million unknowns and errors for a domain of a plate with a circular-through hole consisting of six different curvilinear meshes which interpolate from each other (see Section 8.2). Left table: weak convergence test, for which both the number of discretization points and the numbers of cores are increased simultaneously. Right table: strong convergence test, wherein for a fixed number of grid points, the number of cores used is increased. Clearly, essentially perfect parallel efficiency is obtained under both weak and strong convergence tests. The somewhat increased computational costs in the present table vis-a-vis those reported in Table 3 reflect costs arising from the inter-patch interpolation procedure.

# grid pts	# cores	L^∞_{err}	$O(L^\infty)$	S
377,460	120	1.70e−1	–	1.61 s
3,033,360	360	7.98e−3	4.41	1.54 s
10,252,980	480	1.04e−3	5.03	1.65 s

# grid pts	# cores	L^∞_{err}	S
3,033,360	240	7.89e−3	1.51 s
–	360	7.98e−3	1.55 s
–	480	8.32e−3	1.45 s

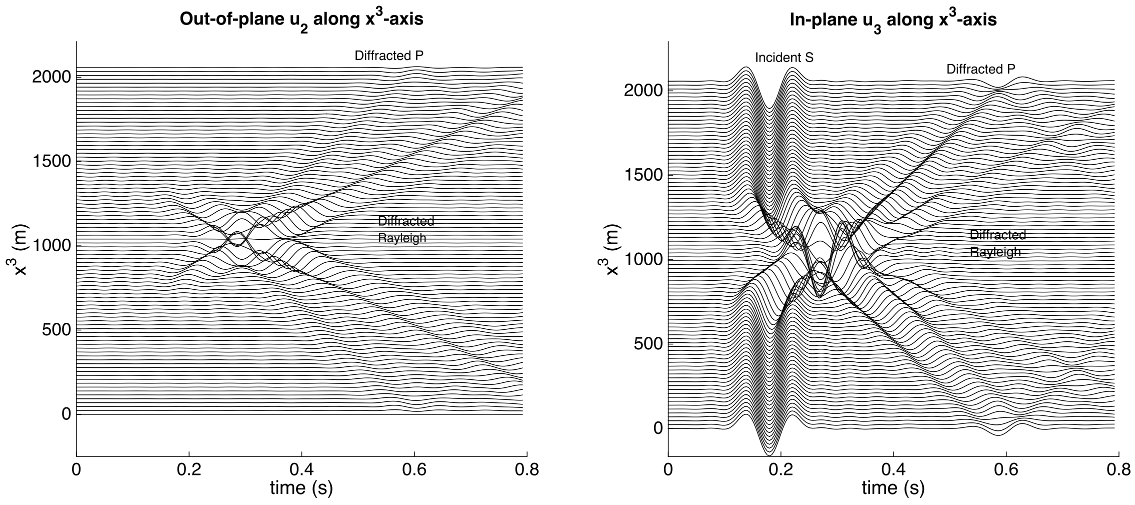

Fig. 14. Time responses of various displacement components on the surface for receivers placed along the x^3-axis (the minor axis of the mountain). The incident S-wave is polarized along the minor (x^3-) axis of the 180 m mountain. Note the existence of reflections from the x^3 equal to 0 and 2.08 km lateral faces, which result from the periodicity of the structure that is assumed in this classical problem.

8. Numerical examples and applications

8.1. Seismic response in 3D topographies

This section presents results of applications of the elastic wave equation solver introduced in this paper, including its overset grid methodology, to 3D geological structures impacted by incident S-waves (shear waves). The effects arising from three-dimensional topographies have been repeatedly studied via consideration of a Gaussian hill of height 180 m whose profile is parameterized by [5,8,22,24]

$$x^2 = x^2(x^1, x^3) = 1050\,\text{m} + 180 \cdot \exp\left[-\frac{x^1 - 1040}{2 \cdot 250^2} - \frac{x^3 - 1040}{2 \cdot 125^2}\right]\text{m} \tag{55}$$

for $(x^1, x^3) \in [0\text{ km}, 2.08\text{ km}]$, and whose computational domain extends to a depth of 1.05 km. (The bottom of the computational patch lies on the plane $x^2 = 0$.) For our treatment, this computational domain is a single patch containing $N = 143{,}871$ discretization points. A homogeneous ground medium is assumed, with a P-wave velocity of 3.2 km/s, an S-wave velocity of 1.8475 km/s and a density of 2200 kg/m^3. A vertically incident S-wave of fundamental wavelength $\lambda = 180$ m polarized along the short axis of the hill is considered; the time-dependent source is a Ricker wavelet of frequency $f_0 = 10.2$ Hz centered at time $t_0 = .1$ s that is described by the Dirichlet boundary condition

$$\mathbf{u}(x^1, 0, x^3, t) = \left(0, 0, .5[2\pi^2 f_0^2 (t - t_0)^2 - 1]e^{-\pi^2 f_0^2 (t-t_0)^2}\right)^T \tag{56}$$

at the bottom surface of the computational domain ($x^2 = 0$); cf. e.g. [8,24]. Traction-free boundary conditions were imposed on the top surface (equation (55)) and periodic conditions were assumed along the lateral edges corresponding to x^1 and x^3 equal to 0 and 2.08 km. The simulation was evolved up to time $t = 1.24$ s at a step size of $\Delta t = 8.0 \times 10^{-4}$ s.

The time responses (seismograms) for the various displacements measured at receivers along the minor axis of the hill are given in Fig. 14, where the early-time incident S-wave trace is so labeled, as are the corresponding diffracted P- and Rayleigh-wave traces. Snapshots of the solution for the in-plane ground displacement $u^3(\mathbf{x}, t)$ over the entire surface are additionally presented in Fig. 15 where, following previous contributions, it can be observed that most of the waves diffracted

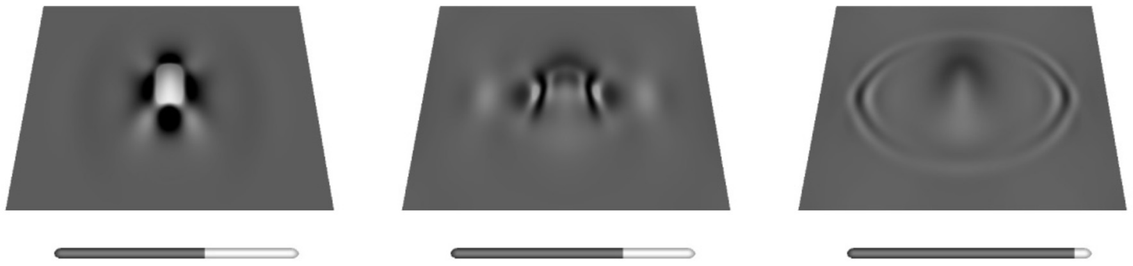
Fig. 15. Horizontal displacement $u^3(\mathbf{x}, t)$ at various snapshots in time, where the x^3-coordinate direction contains the minor axis of the hill. An intense field concentration at the summit is clearly visible, as is the preferential propagation direction (in the direction of the short axis of the mountain) of the resulting diffracted waves. A time slider is given underneath each snapshot.

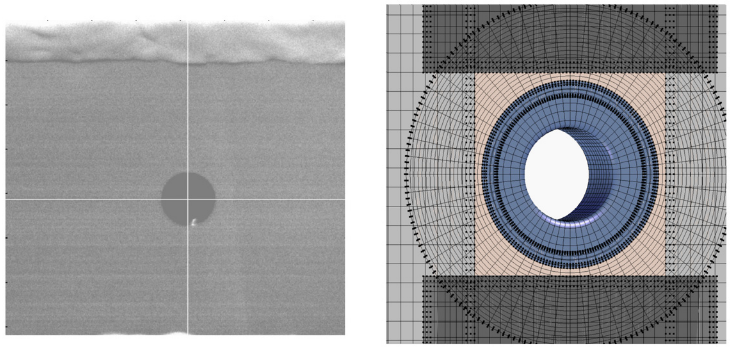
Fig. 16. (Left) Photograph of an experimental sample with a circular hole. (Right) A close up of the corresponding computational model composed of six overlapping patches.

away from the hill are generated at the summit. The results of this simulation are in excellent agreement with previous solutions, including those produced by means of 1) The high-order predictor–corrector spectral element method [24] using a total of $N = 4{,}935{,}953$ discretization points; 2) The stable FD method [5] using a total of $N = 109{,}808{,}412$ discretization points; and 3) The time-domain boundary element method [22] on the basis of a reduced 3721 point surface discretization and a relatively large time step $\Delta t = 4.0 \cdot 10^{-3}$ (approximately five times larger than the time-step $\Delta t = 8.0 \cdot 10^{-4}$ we used), but whose computing time remains high on account of the required large number of Green function evaluations. For comparison the diffracted P- and Rayleigh waves were accurately evaluated in our simulations using a total of $N = 143{,}871$ volumetric discretization points for a run-time of 58 seconds on 96 cores of an infiniband Poweredge cluster consisting of 32 nodes each one of which contains two eight-core Intel Xeon 2.4 GHz processors and 64 GB of RAM.

8.2. Non-destructive evaluation: scattering of waveguides in plates with defects

The thin-plate scattering applications considered in this section are motivated by a collaboration with our co-authors in [25]; corresponding laboratory experiments were performed by these colleagues at the University of Vigo in Spain. The contribution [25] presents a quantitative characterization of defects in aluminum plates by comparison between experimental and simulated (numerical) scattering patterns of narrow-band guided waves with circular and rectangular holes—albeit with a simplified mathematical model based on use of scalar waves on a two-dimensional domain.

In the present section we revisit this problem, but we tackle the numerical simulation problem by means of our full three-dimensional elastic wave solver. The incident waves in these laboratory experiments were generated by means of the wedge method described in [25]: the longitudinal wave emitted by a piezoelectric transducer was coupled to the surface of a thin plate through a prismatic coupling block. In the experiments under consideration the piezoelectric source was excited in such a way that the guided wave trains had a frequency of $f = 1$ MHz and were quasi-monochromatic with wavelength $\lambda = 2.96$ mm, yielding a Rayleigh phase velocity $c = \lambda f = 2960$ m/s. The instantaneous out-of-plane displacement field $u^2(\mathbf{x}, t)$ due to the propagation of the guided wave-train was obtained by means of a novel double-pulsed TV holography system—details can be found in [18,25,30]. For our numerical experiments we use analytical representations provided by our collaborators for the experimentally observed Dirichlet boundary condition on the left edge of the computational domain, representing small wave-trains incident on a circular hole of diameter 12 mm centered in a plate of thickness 10 mm. The corresponding computational geometries were constructed, to the physical specifications of the experimental regions of interest shown on the left portion of Fig. 16, by means of our overlapping grid strategy—the corresponding computational domains including an illustration of the overset patches are given in the right portion of this figure.

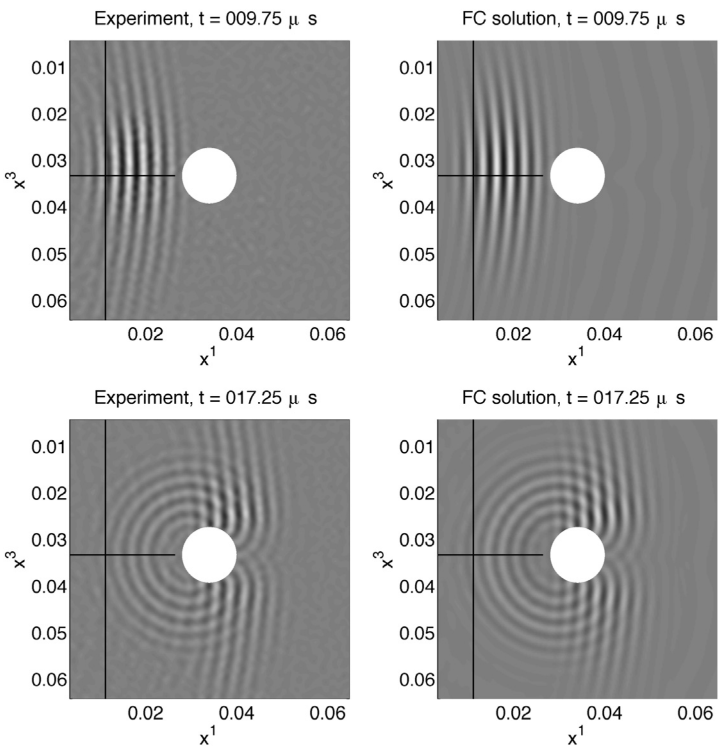

Fig. 17. Experimental (left) and 3D simulated (right) snapshots of the real part of the out-of-plane displacement $u^2(\mathbf{x}, t)$ on the top plate surface for a plate containing a 12 mm diameter circular hole. In the top figures, which correspond to a time before interactions with the hole take place, the field equals the driving incident field. The bottom figures demonstrate the wave scattering caused by the hole.

As indicated in Fig. 16, modeling the hole-edges requires particular care—edges must be smooth to avoid singularities in our high-order numerical method, but they must be sufficiently sharp so as to provide an adequate approximation of the experimental sample. To obtain such slightly rounded edges we utilize a portion of a superellipse given by the equation

$$(x/a)^\eta + (y/a)^\eta = 1, \quad \eta \in \mathbb{R} \tag{57}$$

together with three additional curves to make up a quadrilateral with curved sides, and we then construct, by means of transfinite interpolation, a two-dimensional slice of a domain around the hole boundary that is subsequently rotated to produce the corresponding three-dimensional hole geometry. We have found that use of the rounding parameter $\eta = 24$ in (57) provides adequate approximations: this approximation yields three correct digits in the back-scattering coefficient [30] (as verified by means of a convergence study in the parameter η), and it leads to solutions that agree very well with the experimentally measured fields, as demonstrated in what follows.

The propagation domain was discretized by means of six patches and a total of $N = 5{,}933{,}561$ discretization points. Traction boundary conditions were enforced at the top and bottom of the plate as well as the hole boundaries (as befits the experimental configuration), and the solution was advanced at a time-step of $\Delta t = 0.5$ ns for a total number of 50,000 steps. Simulations were carried out on a computing cluster using 464 cores and using a number of four fringe points in each dimension for all sub-patches for a total run-time of just under two hours per run.

A depiction of the solution values for the scattering by a 12 mm hole is given in Fig. 17 as snapshots of the out-of-plane displacement solution $u^2(\mathbf{x}, t)$ for both the experiments and our numerical simulations. The images displayed demonstrate very good agreement between measured field values and the three-dimensional simulation—and they thus provide a mutual validation of our numerical methodology and the measurement techniques employed by the Vigo group. Fig. 18 presents profiles of the out-of-plane displacement u^2 along the vertical and horizontal lines that overlay Fig. 17 to the left of the hole. Agreement between the experimental and numerical profiles is observed even at points very close to the hole edge—just short of the edge-rounding region. The oscillations in the experimental curves are attributed to experimental error: cf. the left images in Fig. 17 which indeed seem to suggest existence of experimental error of relatively high frequency.

9. Conclusions

This paper introduces a new high-order methodology, based on the Fourier continuation method for the resolution of the Gibbs phenomenon, for the numerical solution of elastodynamics problems in complex three-dimensional geometries. Through combination of several key elements, including a new modified FC operator for the treatment of elastic boundary

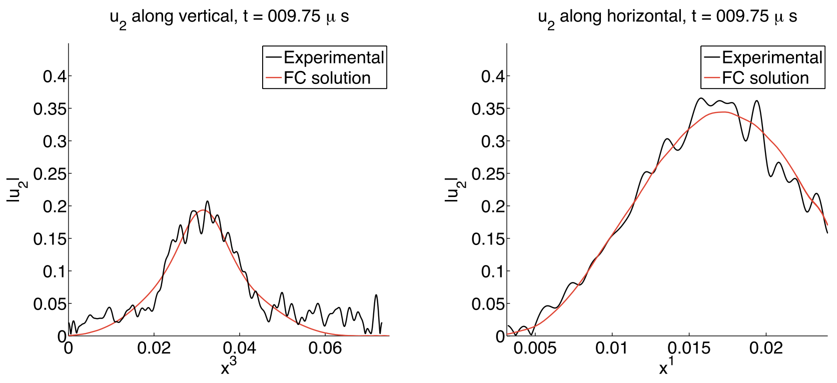

Fig. 18. Field cross-sections along the horizontal and vertical lines shown in Fig. 17.

conditions and an overset strategy for the treatment of general three-dimensional geometries, including possibly absorbing boundary conditions and unbounded media (cf. [4]), the solver developed in this work effectively resolves linear elastic wave propagation problems in physically-relevant three-dimensional computational domains. Our new solver provides fast, accurate solutions with essentially nil dispersion, it only requires mild CFL conditions for stability (Remark 6.1), and it has been implemented on distributed-memory computing clusters by means of a parallelization strategy that has excellent scalability properties. The versatility of the proposed algorithm was demonstrated by means of numerical tests as well as a variety of realistic applications in seismology and non-destructive evaluation of materials with defects—including, in particular, comparisons with experimental data.

Acknowledgements

The authors gratefully acknowledge support from NSF (Grants DMS-1008631 and DMS-1411876) and AFOSR (Grants FA9550-11-1-0193 and FA9550-15-1-0043). Our interactions with Prof. J. Carlos López-Vázquez and his collaborators at the University of Vigo provided one of the main motivations for the efforts leading to the present contribution; we thank these colleagues for profitable discussions and for providing data on through-hole plate-scattering experiments. Valuable conversations with Prof. D. Appelo are also gratefully acknowledged.

References

[1] S. Abarbanel, D. Gottlieb, M.H. Carpenter, On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations, SIAM J. Sci. Comput. 17 (3) (1996) 777–782.
[2] N. Albin, O.P. Bruno, A spectral FC solver for the compressible Navier–Stokes equations in general domains I: explicit time-stepping, J. Comput. Phys. 230 (16) (July 2011) 6248–6270.
[3] N. Albin, O.P. Bruno, T.Y. Cheung, R.O. Cleveland, Fourier continuation methods for high-fidelity simulation of nonlinear acoustic beams, J. Acoust. Soc. Am. 132 (4) (2012) 2371–2387.
[4] F. Amlani, A new high-order Fourier continuation-based elasticity solver for complex three-dimensional geometries, PhD thesis, California Institute of Technology, 2014.
[5] D. Appelö, N.A. Petersson, A stable finite difference method for the elastic wave equation on complex geometries with free surfaces, Commun. Comput. Phys. 5 (1) (2009) 84–107.
[6] I.M. Babuska, S.A. Sauter, Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?, SIAM Rev. 42 (3) (09 2000) 451–484.
[7] F. Bashforth, J. Adams, An Attempt to Test the Theories of Capillary Action: By Comparing the Theoretical and Measured Forms of Drops of Fluid. With an Explanation of the Method of Integration Employed in Constructing the Tables which Give the Theoretical Forms of Such Drops, Cambridge University Press, Cambridge, UK, 1883.
[8] M. Bouchon, C.A. Schultz, M.N. Toksöz, Effect of three-dimensional topography on seismic motion, J. Geophys. Res. 101 (B3) (1996) 5835–5846.
[9] D.L. Brown, W.D. Henshaw, D.J. Quinlan, Overture: an object-oriented framework for solving partial differential equations on overlapping grids, in: Object Oriented Methods for Interoperable Scientific and Engineering Computing, SIAM, 1999, pp. 245–255.
[10] O. Bruno, M. Cubillos, Higher-order in time "quasi-unconditionally stable" ADI solvers for the compressible Navier–Stokes equations in 2D and 3D curvilinear domains, arXiv preprint, arXiv:1509.09262, 2015.
[11] O.P. Bruno, E. Jimenez, Higher-order linear-time unconditionally stable alternating direction implicit methods for nonlinear convection–diffusion partial differential equation systems, J. Fluids Eng. 136 (6) (2014) 060904.
[12] O.P. Bruno, M. Lyon, High-order unconditionally stable FC–AD solvers for general smooth domains I. Basic elements, J. Comput. Phys. 229 (6) (2010) 2009–2033.
[13] O.P. Bruno, A. Prieto, Spatially dispersionless, unconditionally stable FC–AD solvers for variable-coefficient PDEs, J. Sci. Comput. 58 (2) (2014) 1–36.
[14] M.H. Carpenter, D. Gottlieb, S. Abarbanel, 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 (6) (1995) 1241–1252.
[15] T. Elling, GPU-accelerated Fourier-continuation solvers and physically exact computational boundary conditions for wave scattering problems, PhD thesis, California Institute of Technology, 2012.
[16] L. Eriksson, Practical three-dimensional mesh generation using transfinite interpolation, SIAM J. Sci. Stat. Comput. 6 (3) (1985) 712–741.
[17] M. Farrashkhalvat, J.P. Miles, Basic Structured Grid Generation: With an Introduction to Unstructured Grid Generation, Elsevier Science, 2003.

[18] J. Fernández, Á.F. Doval, C. Trillo, J.L. Deán, J. López, Video ultrasonics by pulsed TV holography: a new capability for non-destructive testing of shell structures, Int. J. Optomechatronics 1 (2) (2007) 122–153.
[19] K.F. Graff, Wave Motion in Elastic Solids, Dover Publications, Inc, 1975.
[20] I. Saidu, M.Y. Waziri, A simplified derivation and analysis of fourth order Runge–Kutta method, Int. J. Comput. Appl. 9 (8) (November 2010) 51–55. Published By Foundation of Computer Science.
[21] W.D. Henshaw, D.W. Schwendeman, Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement, J. Comput. Phys. (ISSN 0021-9991) 227 (16) (2008) 7469–7502.
[22] F. Janod, O. Coutant, Seismic response of three-dimensional topographies using a time-domain boundary element method, Geophys. J. Int. 142 (2) (2000) 603–614.
[23] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, Ann Arbor, MI, 1994.
[24] D. Komatitsch, J.-P. Vilotte, The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, Bull. Seismol. Soc. Am. 88 (2) (1998) 368–392.
[25] J.C. López-Vázquez, X.L. Deán-Ben, C. Trillo, Á.F. Doval, J. Fernández, F. Amlani, O.P. Bruno, Numerical modeling and measurement by pulsed television holography of ultrasonic displacement maps in plates with through-thickness defects, Opt. Eng. 49 (9) (09 2010) 095802.
[26] M. Lyon, 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 (9) (May 2010) 3358–3381.
[27] D. Pathria, The correct formulation of intermediate boundary conditions for Runge–Kutta time integration of initial boundary value problems, SIAM J. Sci. Comput. 18 (5) (1997) 1255–1266.
[28] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing, 3rd edition, Cambridge University Press, New York, NY, USA, 2007.
[29] P.J. Roache, Code verification by the method of manufactured solutions, J. Fluids Eng. 124 (1) (2002) 4–10.
[30] P. Rodríguez-Gómez, J.C. López-Vázquez, C. Trillo, Á.F. Doval, J.L. Fernández, Transient elastic wave propagation and scattering in plates: comparison between pulsed TV-holography measurements and finite element method predictions, Opt. Eng. 52 (10) (2013) 101911.
[31] C.J. Roy, T.M. Smith, C.C. Ober, Verification of a compressible CFD code using the method of manufactured solutions, Energy 2 (2002) s2.
[32] K. Shahbazi, N. Albin, O.P. Bruno, J.S. Hesthaven, Multi-domain Fourier-continuation/WENO hybrid solver for conservation laws, J. Comput. Phys. 230 (24) (2011) 8779–8796.
[33] K. Shahbazi, J.S. Hesthaven, X. Zhu, Multi-dimensional hybrid Fourier continuation–WENO solvers for conservation laws, J. Comput. Phys. 253 (2013) 209–225.
[34] S.P. Spekreijse, Elliptic grid generation based on Laplace equations and algebraic transformations, J. Comput. Phys. 118 (1) (1995) 38–61.
[35] J.M. Vedovoto, A.D. Silveira Neto, A. Mura, L.F. Figueira da Silva, Application of the method of manufactured solutions to the verification of a pressure-based finite-volume numerical scheme, Comput. Fluids 51 (1) (2011) 85–99.