LLNL-JRNL-757334

Fourth order finite difference methods for the wave equation with mesh refinement interfaces

Siyang Wang and N. Anders Petersson

August 29, 2018

Submitted to SIAM journal on Scientific Computing

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

October 1, 2007

FOURTH ORDER FINITE DIFFERENCE METHODS FOR THE WAVE EQUATION WITH MESH REFINEMENT INTERFACES

SIYANG WANG [*] AND N. ANDERS PETERSSON [†]

Abstract. We analyze two types of summation-by-parts finite difference operators for solving the two-dimensional wave equation on a grid with a mesh refinement interface. The first type uses ghost points, while the second type does not use any ghost points. A previously unexplored relation between the two types of summation-by-parts operators is investigated. By combining them we develop a new fourth order accurate finite difference discretization for the wave equation with hanging nodes on the mesh refinement interface. Compared to previous approaches using ghost points, the proposed method leads to a smaller system of linear equations that needs to be solved for the ghost point values. An attractive feature of the proposed method is that the explicit time step does not need to be reduced relative to the corresponding periodic problem. Numerical experiments, both for smoothly varying and discontinuous material properties, demonstrate that the proposed method converges to fourth order accuracy. A detailed comparison of the accuracy and the time-step restriction of the simultaneous-approximation-term penalty method is also presented.

Key words. Wave equation, Finite difference methods, Summation-by-parts, Ghost point, Non-conforming, Mesh refinement

AMS subject classifications. 65M06, 65M12

1. Introduction.

Based on the pioneering work by Kreiss and Oliger [8], it is by now well known that high order accurate (≥ 4) numerical methods solve hyperbolic partial differential equations (PDE) more efficiently than low order methods. While Taylor series expansion can easily be used to construct high order finite difference stencils for the interior of the computational domain, it can be more challenging to find stable boundary closures. In this paper we use finite difference operators that satisfy the summation-by-parts (SBP) property, first introduced by Kreiss and Scherer [9], to solve the two-dimensional wave equation with variable coefficients on a grid with a non-conforming mesh refinement interface.

An SBP operator is constructed such that the energy estimate of the continuous PDE can be carried out discretely for the finite difference approximation, with summation-by-parts replacing the integration-by-parts principle. As a consequence, a discrete energy estimate can be obtained to ensure that the discretization is energy stable. When deriving a continuous energy estimate, the boundary terms resulting from the integration-by-parts formula are easily controlled through the boundary conditions. However, for the finite difference approximation, special care is needed to make sure that boundary terms do not lead to unphysical growth of the numerical solution.

When the material properties are discontinuous, one possible approach to ensure high order accuracy is to decompose the domain into multiple subdomains, such that the material is smooth within each subdomain. The governing equation is then discretized by SBP operators in each subdomain, and patched together by imposing interface conditions at the material discontinuity. For computational efficiency it can be desirable to use different mesh sizes in the subdomains, leading to mesh refinement interfaces with hanging nodes.

In the SBP finite difference framework, there are two main approaches to impose boundary conditions. First, we can impose boundary conditions strongly by using ghost points [21]. In this case, the SBP operators also utilize the ghost points for difference approximations. We call this the SBP-GP method. In the second approach, called SBP-SAT, boundary conditions are imposed weakly by adding penalty terms, also known as simultaneous-approximation-terms (SAT) [3], to the discretization. Thus, the SBP-SAT method bears similarities with the discontinuous Galerkin method [2, 5]. For the wave equation with non-conforming mesh refinement interfaces, a high order accurate SBP-SAT finite difference method and a second order accurate SBP-GP method were previously developed in [25] and [16], respectively.

In this paper, we present two ways of generalizing the SBP-GP method in [16] to fourth order

[*] Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, SE-412 96 Gothenburg, Sweden. Email: siyang.wang@chalmers.se

[†] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, USA. Email: petersson1@llnl.gov

accuracy. The first approach is a direct generalization of the second order accurate technique, which uses ghost points from both subdomains for imposing the interface conditions. The second version is based on a previously unexplored relation between SBP operators with and without ghost points. This relation allows for an improved version of the fourth order SBP-GP method, where only ghost points from one side of the interface are used to impose the interface conditions. This approach reduces the computational cost of updating the solution at the ghost points and should also simplify the generalization to three-dimensional problems.

Even though both the SBP-GP and SBP-SAT methods have been used to solve many kinds of PDEs, the relation between them has previously not been explored. An additional contribution of this paper is to connect the two approaches, provide insights into their similarities and differences, and make a comparison in terms of their efficiency.

The remainder of the paper is organized as follows. In Section 2, we introduce the SBP methodology and present the close relation between the SBP operators with and without ghost points. In Section 3, we derive a discrete energy estimate for the wave equation in one space dimension with Dirichlet or Neumann boundary conditions. Both the SBP-GP and the SBP-SAT methods are analyzed in detail and their connections are discussed. In Section 4, we consider the wave equation in two space dimensions, and focus on the numerical treatment of grid refinement interfaces with the SBP-GP and SBP-SAT methods. Numerical experiments are conducted in Section 5, where we compare the SBP-GP and SBP-SAT methods in terms of their time-step stability condition and solution accuracy. Our findings are summarized in Section 6.

2. SBP operators. We begin with preliminaries that will be used in the discussion of SBP finite difference methods. Consider an interval $\Omega = [0, 1]$ and a uniform grid $\boldsymbol{x} = [x_1, \cdots, x_n]^T$, where

$$x_j = (j-1)h, \quad j = 1, \cdots, n.$$

The domain boundaries are at the grid points $j = 1$ and $j = n$, and the grid size is $h = 1/(n-1)$. In addition, there is one ghost point at $x_0 = -h$ and one ghost point at $x_{n+1} = 1 + h$ outside the physical domain Ω.

Let $\boldsymbol{u} = [u_1, \cdots, u_n]$ and $\boldsymbol{v} = [v_1, \cdots, v_n]$ be grid functions on $\boldsymbol{x}$. In the context of SBP identities, the values of the grid functions are arbitrary. However, in the discussion of truncation errors, we assume the grid functions are sufficiently smooth functions evaluated on the grid.

The standard discrete L^2 inner product is defined as

$$(\boldsymbol{u}, \boldsymbol{v})_2 = h \sum_{j=1}^{n} u_j v_j.$$

For SBP operators, we need a weighted inner product

$$(\boldsymbol{u}, \boldsymbol{v})_h = h \sum_{j=1}^{n} w_j u_j v_j, \quad w_j \geq \delta > 0,$$

for some constant δ, where $w_j = 1$ in the interior and $w_j \neq 1$ at a few grid points near each boundary. The norm induced from the inner product $(\cdot, \cdot)_h$ is called a diagonal SBP norm.

The SBP methodology was introduced by Kreiss and Scherer in [9], where the first derivative SBP operator $D \approx \partial/\partial x$ was also constructed. It satisfies the first derivative SBP identity

$$(\boldsymbol{u}, D\boldsymbol{v})_h = -(D\boldsymbol{u}, \boldsymbol{v})_h - u_1 v_1 + u_n v_n. \tag{2.1}$$

Because the weights of an SBP norm equal to one in the interior of the domain, central finite differences with order of accuracy $2p$ can be used in the interior of the domain. To retain the SBP property, special one-sided boundary stencils must be employed for non-periodic problems at a few grid points near each boundary. Kreiss and Scherer also showed in [9] that the order of accuracy of the boundary stencil is lower than the interior stencil. With a diagonal norm and a $2p^{th}$ order accurate interior stencil, the

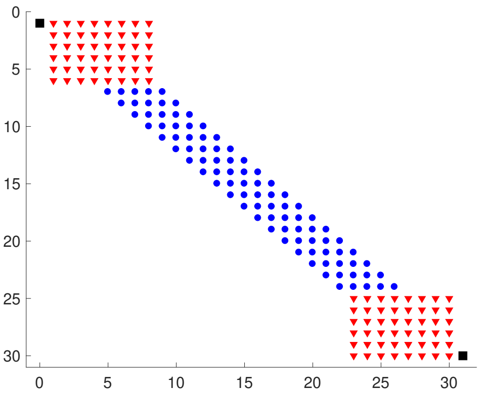

Fig. 1: The structure of the SBP operator $G(\mu)$ on a grid with 30 grid points. Blue circles: standard five-point difference stencil. Red triangles: special boundary stencil. Black squares: ghost point. The structure of $\widetilde{G}(\mu)$ is the same, but without the black squares.

boundary stencil can be at most p^{th} order accurate. Despite this fact, we refer to the accuracy of an SBP operator by its interior order of accuracy $(2p)$.

For second derivative SBP operators, we focus our discussion on the case with variable coefficient $\frac{\partial}{\partial x}(\mu(x)\frac{\partial}{\partial x}u(x))$, where the smooth function $\mu(x) > 0$ often represents a material property. In the following we introduce two different types of second derivative SBP operators. The first type uses one ghost point outside each boundary, while the second type does not use any ghost points. We proceed by explaining the close relation between these two types of SBP operators. To make the presentation concise, we exemplify the relation for the case of fourth order accurate SBP operators $(2p = 4)$.

2.1. Second derivative SBP operators with one ghost point. A fourth order accurate SBP operator $G(\mu)\boldsymbol{u} \approx \frac{\partial}{\partial x}(\mu(x)\frac{\partial}{\partial x}u(x))$ with ghost points was derived by Sjögreen and Petersson [21]. This operator uses a five-point difference stencil of fourth order accuracy in the interior of the domain. At the first six grid points near each boundary, special one-sided stencils of second order accuracy are constructed. Note that $G(\mu)$ only uses the ghost point value at the boundary itself, as is illustrated in Figure 1, where the structure of $G(\mu)$ is shown when the operator is represented by a matrix of size 30×32 on a grid with 30 grid points.

The boundary stencil is constructed such that $G(\mu)$ satisfies the second derivative SBP identity

$$(\boldsymbol{u}, G(\mu)\boldsymbol{v})_h = -S_\mu(\boldsymbol{u},\boldsymbol{v}) - u_1\mu_1\boldsymbol{b}_1^T\boldsymbol{v} + u_n\mu_n\boldsymbol{b}_n^T\boldsymbol{v}, \tag{2.2}$$

where the bilinear form $S_\mu(\cdot,\cdot)$ is symmetric and positive semi-definite. The boundary derivative

$$\boldsymbol{b}_1^T\boldsymbol{v} = \frac{1}{h}\sum_{j=0}^{4}\sigma_j v_j$$

is a fourth order accurate approximation of $V_x(x_1)$, and makes use of the ghost point value v_0. Similarly, $\boldsymbol{b}_{\boldsymbol{n}}^{\boldsymbol{T}}\boldsymbol{v} = V_x(x_n) + \mathcal{O}(h^4)$ uses the ghost point value v_{n+1}. We emphasize that the bilinear form S_μ does not depend on any ghost point values. The SBP operator $G(\mu)$ only uses the ghost point to approximate the second derivative on the boundaries x_1 and x_n.

The fourth order accurate SBP operator $G(\mu)$ has been extensively used in the software package SW4 [19] for the simulation of seismic wave propagation. Prior to SW4, a second order accurate ghost point technique was developed in [15] and implemented in the WPP code [17].

2.2. Second derivative SBP operators without ghost points. The second type of second derivative SBP operators, denoted by $\widetilde{G}_{2p}(\mu)$, does not use any ghost points. This type of operators was constructed by Mattsson [10] for the cases of second, fourth and sixth order accuracy ($2p = 2, 4, 6$). In the following discussion we focus on the fourth order case and define $\widetilde{G}(\mu) = \widetilde{G}_4(\mu)$.

In the interior of the domain, the operator $\widetilde{G}(\mu)$ uses the same five-point wide, fourth order accurate stencil as the operator with ghost points, $G(\mu)$. At the first six grid points near the boundaries, the two operators are similar in that they both define second order accurate stencils that satisfies an SBP identity of the same form as (2.2),

$$(\boldsymbol{u}, \widetilde{G}(\mu)\boldsymbol{v})_h = -\widetilde{S}_\mu(\boldsymbol{u}, \boldsymbol{v})_h - u_1\mu_1\widetilde{\boldsymbol{b}}_{\boldsymbol{1}}^{T}\boldsymbol{v} + u_n\mu_n\widetilde{\boldsymbol{b}}_n^{T}\boldsymbol{v}. \tag{2.3}$$

Similar to (2.2), the bilinear form $\widetilde{S}_\mu(\cdot, \cdot)$ is symmetric and positive semi-definite. The boundary derivative operators $\widetilde{\boldsymbol{b}}_1$ and $\widetilde{\boldsymbol{b}}_n$ are constructed with third order accuracy using stencils that do not use any ghost points. The structure of $\widetilde{G}(\mu)$ is the same as shown in Figure 1, but without the two black squares representing ghost points.

2.3. Relation between SBP operators with and without ghost points. When using the SBP operator $G(\mu)$ with ghost points, boundary conditions are imposed in a strong sense by using the ghost point value as a degree of freedom. On the other hand, for the SBP operator $\widetilde{G}(\mu)$ without ghost points, boundary conditions are usually imposed in a weak sense by using a penalty technique. Though these two types of SBP operators are used in different ways, they are closely related to each other. In fact, an SBP operator with ghost points can easily be modified into a new SBP operator that does not use any ghost points, and vice versa. The new operators preserve the SBP property and the order of accuracy of the original operators. In the following, we demonstrate this procedure on the fourth order accurate version of $G(\mu)$ [21] and $\widetilde{G}(\mu)$ [10]. It is only necessary to consider the left boundary, because the right boundary can be treated in a similar way.

The boundary derivative associated with $G(\mu)$ is in the form

$$\boldsymbol{b}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{v} = \frac{1}{12h}(-3v_0 - 10v_1 + 18v_2 - 6v_3 + v_4) = V_x(x_1) + \mathcal{O}(h^4). \tag{2.4}$$

We define

$$\underline{\boldsymbol{b}}_1^T\boldsymbol{v} = \boldsymbol{b}_1^T\boldsymbol{v} + \beta h^4\boldsymbol{d}_{5+}^T\boldsymbol{v}, \tag{2.5}$$

where

$$\boldsymbol{d}_{5+}^T\boldsymbol{v} = \frac{1}{h^5}(-v_0 + 5v_1 - 10v_2 + 10v_3 - 5v_4 + v_5) = \frac{d^5V}{dx^5}(x_1) + \mathcal{O}(h) \tag{2.6}$$

is a first order accurate approximation of the fifth derivative at the boundary point x_1. Therefore, both the approximations (2.4) and (2.6) are exact at x_1 if $V(x)$ is a polynomial of order at most four. As a consequence, $\underline{\boldsymbol{b}}_1^T\boldsymbol{v}$ is a fourth order accurate approximation of $V_x(x_1)$ for any β. Here and throughout the paper, we use an underbar to indicate operators that have been modified by adding/removing ghost point.

We note that the coefficient of v_0 in $\boldsymbol{b}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{v}$ is -1/4. To eliminate the dependence on v_0 in the approximation $\underline{\boldsymbol{b}}_1^T\boldsymbol{v}$, we choose $\beta = -1/4$ so that

$$\underline{\boldsymbol{b}}_1^T\boldsymbol{v} = \frac{1}{12h}(-25v_1 + 48v_2 - 36v_3 + 16v_4 - 3v_5) = V_x(x_1) + \mathcal{O}(h^4),$$

does not use the ghost point value v_0. Instead, $\underline{\boldsymbol{b}}_1^T\boldsymbol{v}$ uses the value v_5, which is not used by $\boldsymbol{b}_1^T\boldsymbol{v}$.

To retain the SBP property (2.2), the operator $G(\mu)$ must be changed accordingly. Because the bilinear form $S_\mu(\cdot, \cdot)$ is unchanged by the above procedure, the only change in $G(\mu)$ arises from the approximation at the boundary point. The corresponding SBP operator without ghost point becomes

$$\underline{G}(\mu)\boldsymbol{v}_1 = G(\mu)\boldsymbol{v}_1 - \frac{\beta h^4}{hw_1}\mu_1\boldsymbol{d}_{5+}^T\boldsymbol{v} = G(\mu)\boldsymbol{v}_1 + \frac{12}{17}h^3\mu_1\boldsymbol{d}_{5+}^T\boldsymbol{v},$$

and

$$\underline{G}(\mu)\boldsymbol{v}_j = G(\mu)\boldsymbol{v}_j, \quad j = 2, 3, 4, \cdots, n-1,$$

where we have used that $w_1 = 17/48$ is the weight of the SBP norm at the first grid point.

The new operator $\underline{G}(\mu)$ has similar properties as the original operator $G(\mu)$. In particular, it satisfies the SBP property, is fourth order accurate in the interior and second order accurate at the first six grid points near the boundary, and the boundary derivative is approximated to fourth order accuracy. Even though the SBP operator $\underline{G}(\mu)$ does not use ghost point, it is different from the SBP operator $\widetilde{G}(\mu)$ [10], which uses a third order accurate approximation of the boundary derivative.

For the SBP operator $\widetilde{G}(\mu)$ that does not use ghost points, it is straightforward to reverse the above derivation to obtain a new SBP operator that uses a ghost point outside the boundary. The boundary derivative approximation associated with $\widetilde{G}(\mu)$ is

$$\widetilde{\boldsymbol{b}}_1^T\boldsymbol{v} = \frac{1}{6h}(-11v_1 + 18v_2 - 9v_3 + 2v_4) = \frac{dV}{dx}(x_1) + \mathcal{O}(h^3). \tag{2.7}$$

To add a ghost point, we write

$$\underline{\widetilde{\boldsymbol{b}}}_1^T\boldsymbol{v} = \widetilde{\boldsymbol{b}}_1^T\boldsymbol{v} + \gamma h^3 \boldsymbol{d}_{4+}^T\boldsymbol{v}, \tag{2.8}$$

where

$$\boldsymbol{d}_{4+}^T\boldsymbol{v} = \frac{1}{h^4}(v_0 - 4v_1 + 6v_2 - 4v_3 + v_4) = \frac{d^4V}{dx^4}(x_1) + \mathcal{O}(h). \tag{2.9}$$

The boundary stencil (2.7) is exact for polynomials $V(x)$ of order at most three and $\boldsymbol{d}_{4+}^T\boldsymbol{v} = 0$ for such polynomials. Therefore, (2.8) is a third order accurate approximation of $V_x(x_1)$ for any γ. By choosing $\gamma = -1/3$, we find that

$$\underline{\widetilde{\boldsymbol{b}}}_1^T\boldsymbol{v} = \frac{1}{6h}(-2v_0 - 3v_1 + 6v_2 - v_3) = \frac{dV}{dx}(x_1) + \mathcal{O}(h^3).$$

This stencil uses the ghost point v_0 but does not use v_4. Thus it has the minimum stencil width for a third order accurate approximation of a first derivative.

Correspondingly, the new SBP operator $\underline{\widetilde{G}}(\mu)$ that uses a ghost point takes the form

$$\underline{\widetilde{G}}(\mu)\boldsymbol{v}_1 = \widetilde{G}(\mu)\boldsymbol{v}_1 - \frac{\gamma h^3}{w_1 h}\mu_1 \boldsymbol{d}_{4+}^T\boldsymbol{v} = \widetilde{G}(\mu)\boldsymbol{v}_1 + \frac{16}{17}h^2\mu_1\boldsymbol{d}_{4+}^T\boldsymbol{v}$$

and

$$\underline{\widetilde{G}}(\mu)\boldsymbol{v}_j = \widetilde{G}(\mu)\boldsymbol{v}_j, \quad j = 2, 3, 4, \cdots, n-1.$$

3. Boundary conditions.

In this section, we briefly present the techniques of imposing boundary conditions for the wave equation with the SBP operators $G(\mu)$ and $\widetilde{G}(\mu)$, and highlight the relation between the two approaches SBP-GP and SBP-SAT. Our model equation is

$$\rho U_{tt} = (\mu(x)U_x)_x, \quad x \in [0, 1], \tag{3.1}$$

with suitable initial conditions. We assume both ρ and μ are sufficiently smooth. The forcing function is omitted in the right-hand side of (3.1), as it has no influence on how boundary conditions are imposed. We only consider the boundary condition on the left boundary $x = 0$, because the boundary condition at $x = 1$ can be imposed in the same way. Consequently, terms corresponding to the boundary $x = 1$ are omitted in the scheme.

3.1. The Neumann boundary condition. We start by considering the homogeneous Neumann boundary condition $U_x(0,t)=0$. In the SBP-GP method, the semi-discretization of (3.1) is

$$\rho\boldsymbol{u}_{tt} = G(\mu(x))\boldsymbol{u}. \tag{3.2}$$

By using the SBP property (2.2), we obtain

$$\begin{aligned}(\boldsymbol{u}_t, \rho\boldsymbol{u}_{tt})_h &= (\boldsymbol{u}_t, G(\mu(x))\boldsymbol{u})_h \\ &= -S_\mu(\boldsymbol{u}_t, \boldsymbol{u}) - (u_t)_1\mu_1\boldsymbol{b}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u},\end{aligned}$$

which can be written

$$(\boldsymbol{u}_t, \rho\boldsymbol{u}_{tt})_h + S_\mu(\boldsymbol{u}_t, \boldsymbol{u}) = -(u_t)_1\mu_1\boldsymbol{b}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u}. \tag{3.3}$$

We note that the left-hand side of equation (3.3) is the rate of change in the discrete energy in time,

$$\frac{d}{dt}[(\boldsymbol{u}_t, \rho\boldsymbol{u}_t)_h + S_\mu(\boldsymbol{u}, \boldsymbol{u})] = -2(u_t)_1\mu_1\boldsymbol{b}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u}. \tag{3.4}$$

To obtain energy stability, one option is to impose the boundary condition so that the right-hand side of (3.4) is non-positive. The key in the SBP-GP method is to use the ghost point as the additional degree of freedom for the boundary condition. For the Neumann boundary condition $U_x(0,t)=0$, we approximate it by setting $\boldsymbol{b}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u}=0$ at every time step, which determines the solution u_0 on the ghost point x_0. This choice leads to energy conservation, with the energy estimate

$$\frac{d}{dt}[(\boldsymbol{u}_t, \rho\boldsymbol{u}_t)_h + S_\mu(\boldsymbol{u}, \boldsymbol{u})] = 0. \tag{3.5}$$

Next, we consider the semi-discretization of (3.1) by the SBP-SAT method

$$\rho\boldsymbol{u}_{tt} = \widetilde{G}(\mu)\boldsymbol{u} + \boldsymbol{p_n}, \tag{3.6}$$

where $\boldsymbol{p_n}$ is the penalty term. By using the SBP identity (2.3), we obtain

$$\begin{aligned}(\boldsymbol{u}_t, \rho\boldsymbol{u}_{tt})_h &= (\boldsymbol{u}_t, \widetilde{G}(\mu(x))\boldsymbol{u})_h + (\boldsymbol{u}_t, \boldsymbol{p_n})_h \\ &= -\widetilde{S}_\mu(\boldsymbol{u}_t, \boldsymbol{u}) - (u_0)_t\mu_0\widetilde{\boldsymbol{b}}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u} + (\boldsymbol{u}_t, \boldsymbol{p_n})_h,\end{aligned}$$

which can be written

$$\frac{d}{dt}[(\boldsymbol{u}_t, \rho\boldsymbol{u}_t)_h + \widetilde{S}_\mu(\boldsymbol{u}, \boldsymbol{u})] = -2(u_0)_t\mu_0\widetilde{\boldsymbol{b}}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u} + 2(\boldsymbol{u}_t, \boldsymbol{p_n})_h. \tag{3.7}$$

Therefore, we need $(u_1)_t\mu_1\widetilde{\boldsymbol{b}}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u} = (\boldsymbol{u}_t, \boldsymbol{p_n})_h$ to obtain an energy estimate. An obvious choice of the the penalty term is to take $h^{-1}w_1^{-1}\mu_0\widetilde{\boldsymbol{b}}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u}$ as the first component of $\boldsymbol{p_n}$, and 0 elsewhere. This choice leads to an energy conserving discretization with the energy estimate

$$\frac{d}{dt}[(\boldsymbol{u}_t, \rho\boldsymbol{u}_t)_h + \widetilde{S}_\mu(\boldsymbol{u}, \boldsymbol{u})] = 0. \tag{3.8}$$

We note that the energy estimates (3.5) and (3.8) are in exactly the same form. However, $\boldsymbol{b}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u}=0$ is satisfied at every time step, but $\widetilde{\boldsymbol{b}}_{\mathbf{1}}^{\boldsymbol{T}}\boldsymbol{u}=0$ does in general not hold.

3.2. The Dirichlet boundary condition. With the Dirichlet boundary condition $U(0,t)=0$, the semi-discretization in the SBP-GP method remains the same (3.2). From (3.4), by setting $u_1=0$ at every time step, an energy estimate is obtained. At first glance, it seems that the discrete energy is modified by injection, and the ghost point value u_0 is not used. However, we can choose the ghost point value u_0 at the current time step, such that $u_1=0$ is satisfied at the next time step. In this way, the discrete energy is conserved (3.5), and the scheme is stable. We note that $G(\mu)$ only uses ghost point for the approximation on the boundary. As a consequence, it is not necessary to compute u_0 explicitly, because u_1 is injected by the Dirichlet boundary condition in every time step and u_0 is never used.

Therefore, injection at a Dirichlet boundary leads to an energy stable discretization for the SBP operator $G(\mu)$. This is true also for the SBP operator $\widetilde{G}(\mu)$ without ghost point. In [4], energy stability is proved from a different perspective by analyzing the property of the matrix representing the operator $\widetilde{G}(\mu)$.

It is also possible to impose a Dirichlet boundary by the SAT method. The discretized equation is in a more complicated form than the simple injection method, but the technique sheds light on how to impose a grid interface condition, which is the main topic in the next section. Replacing the penalty term in (3.6) by $\boldsymbol{p_d}$, an analogue of (3.7) is

$$(3.9) \qquad \frac{d}{dt}[(\boldsymbol{u}_t, \rho \boldsymbol{u}_t)_h + \widetilde{S}_\mu(\boldsymbol{u}, \boldsymbol{u})] = -2(u_t)_1 \mu_1 \widetilde{\boldsymbol{b}}_{\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{u} + 2(\boldsymbol{u}_t, \boldsymbol{p_d})_h.$$

It is not straightforward to choose $\boldsymbol{p_d}$ such that the right-hand side of (3.9) is non-positive. However, we can choose $\boldsymbol{p_d}$ so that the right-hand side of (3.9) is part of the energy change. One option is to require

$$(3.10) \qquad (u_t, \boldsymbol{p_d})_h = -u_1 \mu_1 \widetilde{\boldsymbol{b}}_{\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{u_t} - \frac{\tau}{h}(u_t)_1 \mu_1 u_1$$

so that (3.9) becomes

$$(3.11) \qquad \frac{d}{dt}[(\boldsymbol{u}_t, \rho \boldsymbol{u}_t)_h + \widetilde{S}_\mu(\boldsymbol{u}, \boldsymbol{u}) + 2u_1 \mu_1 \widetilde{\boldsymbol{b}}_{\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{u} + \frac{\tau}{h} u_1 \mu_1 u_1] = 0.$$

We obtain an energy estimate (3.11) if the quantity in the square bracket is non-negative.

In Lemma 2 of [22], it is proved that the following identity holds

$$(3.12) \qquad \widetilde{S}_\mu(\boldsymbol{u}, \boldsymbol{u}) = \widetilde{S}_{\widetilde{\mu}}(\boldsymbol{u}, \boldsymbol{u}) + h\alpha\mu_m(\widetilde{\boldsymbol{b}}_{\boldsymbol{1}}^{\boldsymbol{T}} \boldsymbol{u})^2,$$

where both the bilinear forms $\widetilde{S}_\mu(\cdot,\cdot)$ and $\widetilde{S}_{\widetilde{\mu}}(\cdot,\cdot)$ are symmetric and positive semi-definite, α is a constant that depends on the order of accuracy of $\widetilde{G}(\mu)$ but not h, and μ_m is the smallest value of μ on the first r_μ grid points. The constant r_μ depends on the order of accuracy of $\widetilde{G}(\mu)$ but not h. As an example, the fourth order accurate SBP operator $\widetilde{G}(\mu)$ constructed in [10] satisfies (3.12) with $r_\mu = 4$ and $\alpha = 0.2505765857$. Any $\alpha > 0.2505765857$ can make $\widetilde{S}_{\widetilde{\mu}}(\cdot,\cdot)$ indefinite.

By using Young's inequality, when the penalty parameter $\tau \geq \mu_1/(\alpha\mu_m)$, equation (3.11) is indeed an energy estimate. The energy estimate (3.11) has more terms than the corresponding energy estimate by the SBP-GP method, but the extra terms vanish when the grid size goes to zero. We note that the penalty parameter τ has a lower bound but no upper bound. Choosing τ to be equal to the lower bound gives large numerical error in the solution [24]. However, an unnecessarily large τ affects the CFL condition negatively and requires a small time step [13]. In computation, we find the increase in τ by 10% to 20% from the lower bound is adequate for accuracy and efficiency.

4. Grid refinement interface. We consider the wave equation in two space dimensions with a discontinuous wave speed. To achieve high order accuracy with a finite difference method, it is important that the difference stencil does not cross the discontinuity. A common strategy for discontinuous parameters is to partition the domain into subdomains, and align the discontinuity with the subdomain boundaries. The finite difference approximation is then carried out in each subdomain, and adjacent subdomains are connected via interface conditions.

As an example, we consider the wave equation in a composite domain $\Omega^f \cup \Omega^c$, where $\Omega^f = [0,1]\times[0,1]$ and $\Omega^c = [0,1]\times[-1,0]$. The governing equation reads

$$(4.1) \qquad \begin{aligned} \rho U_{tt}^f &= \nabla\cdot\mu^f\nabla U^f,\ (x,y)\in\Omega^f,\ t\geq 0,\\ \rho U_{tt}^c &= \nabla\cdot\mu^c\nabla U^c,\ (x,y)\in\Omega^c,\ t\geq 0,\end{aligned}$$

with suitable initial and boundary conditions. We assume that ρ is sufficiently smooth in $\Omega^f \cup \Omega^c$. We also assume μ^f and μ^c are sufficiently smooth in the domain Ω^f and Ω^c, respectively. However, on the interface, μ^f may not equal μ^c, in which case the solution is continuous, but its gradient is discontinuous. The continuous interface conditions

$$\begin{gathered} U^f(x,0,t) = U^c(x,0,t), \\ \mu^f(x,0)U^f_y(x,0,t) = \mu^c(x,0)U^c_y(x,0,t), \end{gathered} \tag{4.2}$$

at $y = 0$ lead to a wellposed problem [12, 16].

Our focus is the numerical treatment of the interface conditions (4.2) when the grids are non-conforming. In particular, we consider periodic boundary conditions in x. For the spatial discretization, we use a Cartesian mesh with mesh size h in the fine domain Ω^f and $2h$ in the coarse domain Ω^c. The number of grid points in the x direction is n in Ω^c, and $2n-1$ in Ω^f. The mesh (x^f, y^f) in Ω^f and (x^c, y^c) in Ω^c are defined as

$$\begin{cases} x^f_i = (i-1)h, & i = 1, 2, \cdots, 2n-1, \\ y^f_j = (j-1)h, & j = 0, 1, 2, \cdots, 2n-1 \end{cases} \quad \text{and} \quad \begin{cases} x^c_i = 2(i-1)h, & i = 1, 2, \cdots, n, \\ y^c_j = 2(j-n)h, & j = 1, 2, \cdots, n+1 \end{cases} \tag{4.3}$$

respectively, where $h = 1/(2n-2)$.

If the wave speed in Ω^c is twice as large as in Ω^f, then the mesh (4.3) is an ideal choice, because the number of grid points per wavelength is constant in the entire domain. However, this leads to a mesh refinement interface with hanging nodes along the interface $y = 0$. In the following, we discuss both the SBP-GP and SBP-SAT method with energy conserving interpolation for the mesh refinement interface.

We begin with introducing notations of the SBP properties in two dimensions. Next, we present the SBP-GP method to impose the interface conditions (4.2). A second order accurate method was originally developed in [16], and ghost points from both subdomains are used for the interface conditions. Here, we generalize the technique to fourth order accuracy. After that, we propose a new SBP-GP method that only uses ghost points from the coarse domain, which reduces the computational work for computing numerical solution on the ghost points. We end this section by a discussion of the SBP-SAT method, and its relation with the SBP-GP method.

4.1. SBP properties in two space dimensions. The SBP identity (2.2) and (2.3) are in exactly the same form. In the discussion of SBP properties in two space dimensions, we use the notations of SBP operators with ghost point.

Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be grid functions in Ω^f, $\boldsymbol{p}$ and $\boldsymbol{q}$ be grid functions in Ω^c. We define the two dimensional scalar products

$$(\boldsymbol{u}, \boldsymbol{v})_h = h^2 \sum_{i=1}^{2n-2} \sum_{j=1}^{2n-1} w_j u_{ij} v_{ij}, \quad (\boldsymbol{p}, \boldsymbol{q})_{2h} = (2h)^2 \sum_{i=1}^{n-1} \sum_{j=1}^{n} w_j p_{ij} q_{ij}.$$

Note that we have excluded values on the boundary $x = 1$, because we do not solve them in the numerical scheme. Instead, the numerical solution at $x = 1$ is set to be equal to the numerical solution at $x = 0$ because of the periodic boundary condition. We also define two scalar products for grid functions on the interface

$$\langle \boldsymbol{u}^{\boldsymbol{\Gamma}}, \boldsymbol{v}^{\boldsymbol{\Gamma}} \rangle_h = h \sum_{i=1}^{2n-2} u^\Gamma_i v^\Gamma_i, \quad \langle \boldsymbol{p}^{\boldsymbol{\Gamma}}, \boldsymbol{q}^{\boldsymbol{\Gamma}} \rangle_{2h} = 2h \sum_{i=1}^{n-1} p^\Gamma_i q^\Gamma_i,$$

where the superscripts Γ denotes grid functions on the interface.

We are now ready to state the SBP identity in two space dimensions in the fine domain Ω^f

$$(\boldsymbol{u}, G_x(\mu)\boldsymbol{v})_h = -S_x(\boldsymbol{u}, \boldsymbol{v}), \tag{4.4}$$

$$(\boldsymbol{u}, G_y(\mu)\boldsymbol{v})_h = -S_y(\boldsymbol{u}, \boldsymbol{v}) - \langle \boldsymbol{u}^{\boldsymbol{\Gamma}}, \boldsymbol{v}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}} \rangle_h, \tag{4.5}$$

where the subscripts x and y denote the spatial direction that the operator acts on. The bilinear forms $S_x(\cdot,\cdot)$ and $S_y(\cdot,\cdot)$ are symmetric and positive semi-definite. There is no boundary term in (4.4) for

$G_x(\mu)$ because of the periodic boundary condition. We have also omitted a boundary term at $y = 1$. The last term in (4.5) corresponds to a boundary term on the interface, where

$$(\boldsymbol{v}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}})_i = \mu_{i,1}\boldsymbol{b}_{\boldsymbol{1}}^{\boldsymbol{T}}\boldsymbol{v}_{\boldsymbol{i,:}}, \quad i = 1, 2, \cdots, 2n-1.$$

To condense notation, we define

$$G_f(\mu) = G_x(\mu) + G_y(\mu), \quad S_f = S_x + S_y,$$

so that (4.4)-(4.5) can be written

$$(\boldsymbol{u}, G_f(\mu)\boldsymbol{v})_h = -S_f(\boldsymbol{u}, \boldsymbol{v}) - \langle \boldsymbol{u}^{\boldsymbol{\Gamma}}, \boldsymbol{v}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_h, \tag{4.6}$$

The SBP identity for the operators in the coarse domain Ω^c can be written in a similar way

$$(\boldsymbol{p}, G_c(\mu)\boldsymbol{q})_{2h} = -S_c(\boldsymbol{p}, \boldsymbol{q}) + \langle \boldsymbol{p}^{\boldsymbol{\Gamma}}, \boldsymbol{q}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_{2h}, \tag{4.7}$$

where a boundary term at $y = -1$ is omitted.

4.2. The fourth order accurate SBP-GP method. In [16], a second order accurate SBP-GP method was developed for the wave equation with mesh refinement interfaces. In this section, we generalize the scheme to fourth order accuracy in both space and time.

Equation (4.1) is approximated by

$$\rho \boldsymbol{f}_{tt} = G_f(\mu)\boldsymbol{f}, \quad \rho \boldsymbol{c}_{tt} = G_c(\mu)\boldsymbol{c}, \tag{4.8}$$

where the grid functions $\boldsymbol{f}$ and $\boldsymbol{c}$ are approximated solutions of (4.1) in Ω^f and Ω^c, respectively. At the interface between Ω^f and Ω^c, discrete interface conditions must be imposed to ensure energy stability. Because it is a mesh refinement interface, interpolation between the fine and coarse grids on the interface are needed. We denote $\mathcal{P}$ an interpolation operator that interpolates a grid function on the interface from the coarse domain to the fine domain, and $\mathcal{R}$ a restriction operator that restricts a grid function on the interface from the fine domain to the coarse domain. The stability result is summarized in the theorem below.

THEOREM 4.1. *With the discrete interface conditions*

$$\boldsymbol{f}^{\boldsymbol{\Gamma}}_t = \mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}_t, \tag{4.9}$$

$$\boldsymbol{c}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}} = \mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}, \tag{4.10}$$

where the interpolation and restriction operators satisfy

$$\mathcal{P} = 2\mathcal{R}^T, \tag{4.11}$$

the scheme (4.8) *is energy stable.*

Proof. Applying the SBP identity (4.4) and (4.5), we obtain

$$(\boldsymbol{f}_t, \rho \boldsymbol{f}_{tt})_h = -S_f(\boldsymbol{f}_t, \boldsymbol{f}) - \langle \boldsymbol{f}^{\boldsymbol{\Gamma}}_t, \boldsymbol{f}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_h \tag{4.12}$$

from the approximation in the fine domain Ω^f. Similarly, in Ω^c we have

$$(\boldsymbol{c}_t, \rho \boldsymbol{c}_{tt})_{2h} = -S_c(\boldsymbol{c}_t, \boldsymbol{c}) + \langle \boldsymbol{c}^{\boldsymbol{\Gamma}}_t, \boldsymbol{c}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_{2h}. \tag{4.13}$$

With the discrete energy defined as

$$E = (\boldsymbol{f}_t, \rho \boldsymbol{f}_t)_h + S_f(\boldsymbol{f}, \boldsymbol{f}) + (\boldsymbol{c}_t, \rho \boldsymbol{c}_t)_{2h} + S_c(\boldsymbol{c}, \boldsymbol{c}),$$

we find that the sum of (4.12) and (4.13) can be written as the rate of energy change

$$\frac{1}{2}\frac{d}{dt}E = -\langle \boldsymbol{f}^{\boldsymbol{\Gamma}}_t, \boldsymbol{f}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_h + \langle \boldsymbol{c}^{\boldsymbol{\Gamma}}_t, \boldsymbol{c}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_{2h}. \tag{4.14}$$

To obtain an energy estimate, we need the two terms on the right-hand side of (4.14) to cancel identically through the interface condition (4.2). By (4.11), we have

$$\langle \mathcal{P}\boldsymbol{q}, \boldsymbol{v}\rangle_h = \langle \boldsymbol{q}, \mathcal{R}\boldsymbol{v}\rangle_{2h}, \tag{4.15}$$

for any grid functions $\boldsymbol{v}$ and $\boldsymbol{q}$ on the interface of Ω^f and Ω^c, respectively. We write the right-hand side of (4.14) as

$$\begin{aligned}
&-\langle \boldsymbol{f}_t^{\boldsymbol{\Gamma}}, \boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h + \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{c}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h}\\
=&-\langle \boldsymbol{f}_t^{\boldsymbol{\Gamma}} - \mathcal{P}\boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h - \langle \mathcal{P}\boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h + \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{c}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h}\\
=&-\langle \boldsymbol{f}_t^{\boldsymbol{\Gamma}} - \mathcal{P}\boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h - \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \mathcal{R}\boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h} + \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{c}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h}\\
=&-\langle \boldsymbol{f}_t^{\boldsymbol{\Gamma}} - \mathcal{P}\boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h + \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{c}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}} - \mathcal{R}\boldsymbol{f}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h}.
\end{aligned}$$

The above quantity equals to zero if the numerical solution satisfies the interface conditions (4.9)-(4.10). This completes the proof. □

Remark 4.2. The relation (4.11) and (4.15) are equivalent only in two space dimensions. We also note that relation (4.15) is essential for energy stability. In the case when the boundary condition is non-periodic in x, boundary modifications must be performed for the projection and restriction operator so that (4.11) is satisfied [1, 7, 11].

From the perspective of accuracy, it is desirable to match the order of accuracy of the interpolation and restriction operators to the SBP operators. For the interpolation operator $\mathcal{P}$ in (4.9), it is natural to enforce

$$f_{2i-1}^{\Gamma} = c_i^{\Gamma}, \quad i = 1, 2, \cdots, n-1 \tag{4.16}$$

on the grid points that coincide, and use a fourth order interpolation for the hanging nodes

$$f_{2i}^{\Gamma} = -\frac{1}{16}c_{i-1}^{\Gamma} + \frac{9}{16}c_i^{\Gamma} + \frac{9}{16}c_{i+1}^{\Gamma} - \frac{1}{16}c_{i+2}^{\Gamma}, \quad i = 1, 2, \cdots, n-1. \tag{4.17}$$

With the stencils of $\mathcal{P}$ shown in (4.16)-(4.17), the stencil of $\mathcal{R}$ is completely determined by the condition (4.15). The restriction operator $\mathcal{R}$ in (4.10) can be written

$$(c_{\nabla}^{\Gamma})_i = -\frac{1}{32}(f_{\nabla}^{\Gamma})_{2i-4} + \frac{9}{32}(f_{\nabla}^{\Gamma})_{2i-2} + \frac{1}{2}(f_{\nabla}^{\Gamma})_{2i-1} + \frac{9}{32}(f_{\nabla}^{\Gamma})_{2i} - \frac{1}{32}(f_{\nabla}^{\Gamma})_{2i+2}, \tag{4.18}$$

where $i = 1, 2, \cdots, n-1$. We note that in (4.17) and (4.18), some grid points outside the x-boundary are used by the interpolation and restriction operators. We do not consider them to be unknown ghost point values, because they can be set by the periodic boundary conditions.

In (4.18), ghost point values $f_{i,0}$ and $c_{j,n+1}$ for $i = 1, 2, \cdots, 2n-2$ and $j = 1, 2, \cdots, n-1$ are used. The number of unknown ghost point values is $3n-3$. We observe from (4.16)-(4.18) that the number of linear equations is also $3n-3$. Therefore, the number of unknowns equals the number of equations.

To obtain the unknown ghost point values from (4.16)-(4.18), it necessities to consider a fully discrete version of the discretization (4.8), and impose the conditions (4.16)-(4.17) at a different time level than (4.18). Since we have a fourth order accurate spatial discretization, we match the accuracy in time by using a fourth order accurate predictor-corrector time stepping scheme. The fully discrete scheme consists of the predictor step

$$\begin{aligned}
\rho\frac{\boldsymbol{f}^{*} - 2\boldsymbol{f}^{\boldsymbol{k}} + \boldsymbol{f}^{\boldsymbol{k-1}}}{\delta_t^2} &= G_f(\mu)\boldsymbol{f}^{\boldsymbol{k}},\\
\rho\frac{\boldsymbol{c}^{*} - 2\boldsymbol{c}^{\boldsymbol{k}} + \boldsymbol{c}^{\boldsymbol{k-1}}}{\delta_t^2} &= G_c(\mu)\boldsymbol{c}^{\boldsymbol{k}},
\end{aligned} \tag{4.19}$$

and the corrector step

$$\begin{aligned}
\boldsymbol{f}^{\boldsymbol{k+1}} &= \boldsymbol{f}^{*} + \frac{\delta_t^4}{12\rho}G_f(\mu)\boldsymbol{v}^{\boldsymbol{f}},\\
\boldsymbol{c}^{\boldsymbol{k+1}} &= \boldsymbol{c}^{*} + \frac{\delta_t^4}{12\rho}G_c(\mu)\boldsymbol{v}^{\boldsymbol{c}},
\end{aligned} \tag{4.20}$$

where

$$\boldsymbol{v^f} = \frac{\boldsymbol{f^*} - 2\boldsymbol{f^k} + \boldsymbol{f^{k-1}}}{\delta_t^2} \quad \text{and} \quad \boldsymbol{v^c} = \frac{\boldsymbol{c^*} - 2\boldsymbol{c^k} + \boldsymbol{c^{k-1}}}{\delta_t^2}.$$

The superscript denotes the time level, and δ_t is the time step.

Assuming that the numerical solutions $\boldsymbol{f^{k-1}}$, $\boldsymbol{c^{k-1}}$, $\boldsymbol{f^k}$ and $\boldsymbol{c^k}$ are known on all grid points, the numerical solution at $t = t^{k+1}$ can be computed as follows.

1 Compute by (4.19) the predictor $\boldsymbol{f^*}$ and $\boldsymbol{c^*}$ on all points except the ghost points in Ω^f and Ω^c, respectively.

2 Impose (4.18) for the predictor $\boldsymbol{f^*}$, $\boldsymbol{c^*}$, and (4.16)-(4.17) for the corrector $\boldsymbol{f^{k+1}}$, $\boldsymbol{c^{k+1}}$. Together with (4.20), this gives a system of $3n-3$ linear equations. By solving the system, we obtain $\boldsymbol{f^*}$, $\boldsymbol{c^*}$ on all ghost points.

3 Compute by (4.20) the corrector $\boldsymbol{f^{k+1}}$, $\boldsymbol{c^{k+1}}$ on all points except the ghost points in Ω^f and Ω^c, respectively.

4 Impose (4.18) for the corrector solution $\boldsymbol{f^{k+1}}$, $\boldsymbol{c^{n+1}}$, and (4.16)-(4.17) for the solution $\boldsymbol{f^{**}}$, $\boldsymbol{c^{**}}$, where

$$\rho\frac{\boldsymbol{f^{**}} - 2\boldsymbol{f^{k+1}} + \boldsymbol{f^k}}{\delta_t^2} = G_f(\mu)\boldsymbol{f^{k+1}}, \tag{4.21}$$
$$\rho\frac{\boldsymbol{c^{**}} - 2\boldsymbol{c^{k+1}} + \boldsymbol{c^k}}{\delta_t^2} = G_c(\mu)\boldsymbol{c^{k+1}}.$$

By solving the system of $3n-3$ linear equations, we obtain $\boldsymbol{f^{k+1}}$, $\boldsymbol{c^{k+1}}$ on all ghost points.

Remark 4.3. With the above procedure to obtain the ghost point values, the fully discrete energy is conserved [16, 21].

In each time step, we need to solve two different systems of $3n-3$ linear equations. The coefficients in the linear equations are time independent. As a consequence, it is very efficient to LU-factorize the system before the time stepping scheme, and use backward substitution to compute the solutions on the ghost points at each time step. However, for real-world problems, computations are performed on many processors on a parallel machine. It is then not straightforward to perform an LU-factorization in an efficient way. In [16], an iterative block Jacobi relaxation method is used, and works well in large-scale problems.

4.3. An improved SBP-GP method. In the fourth order accurate SBP-GP method presented in Section 4.2, $n-1$ ghost points from the coarse domain Ω^c and $2n-2$ ghost points from the fine domain Ω^f are used to impose interface conditions. As a consequence, we need to solve two systems of linear equations whose coefficients are independent of time. In this section, we present an improved SBP-GP method, where only $n-1$ ghost points from Ω^c are used for interface conditions. This reduces the number of linear equations to $n-1$.

The key in the improved method is to combine SBP operator with ghost point and SBP operator without ghost point. More precisely, in Ω^c we use the SBP operator with ghost point, and $n-1$ ghost points are used in the spatial discretization. The semi-discretized equation in Ω^c is the same as in the original SBP-GP method

$$\rho\boldsymbol{c}_{tt} = G_c(\mu)\boldsymbol{c}. \tag{4.22}$$

In Ω^f, for the grid points on the interface, we obtain the discretized equation from the first interface condition (4.9) by differentiating twice in time

$$\rho\boldsymbol{f}_{tt}^{\boldsymbol{\Gamma}} = \rho\mathcal{P}\boldsymbol{c}_{tt}^{\boldsymbol{\Gamma}} = \rho\mathcal{P}\left(\frac{1}{\rho}G_c(\mu)\boldsymbol{c}^{\boldsymbol{\Gamma}}\right).$$

For all the other grid points in Ω^f, we use the SBP operator without ghost point

$$\rho\boldsymbol{f}_{tt}^{\boldsymbol{\Omega}} = G_x(\mu)\boldsymbol{f}^{\boldsymbol{\Omega}} + \underline{G_y}(\mu)\boldsymbol{f}^{\boldsymbol{\Omega}},$$

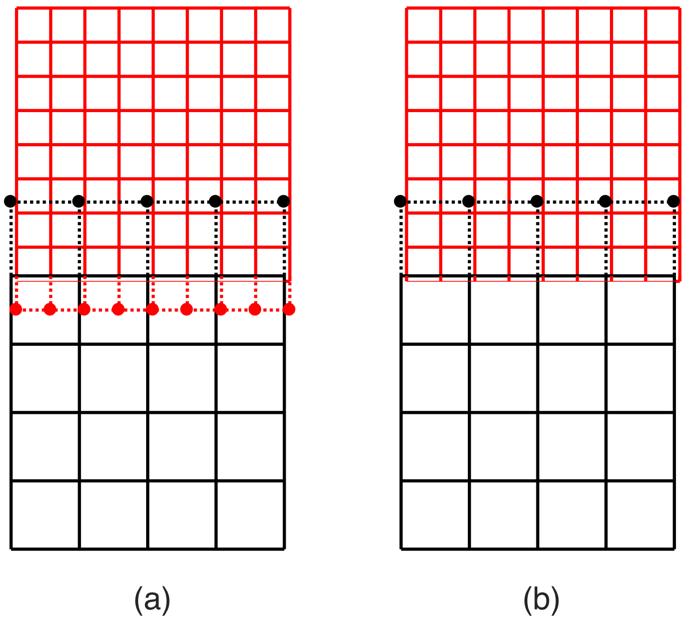

Fig. 2: A mesh refinement interface with ghost points denoted by filled circles. (a) ghost points from both domains. (b) ghost points from the coarse domain.

where the superscript Ω denotes all grid points not on the interface. The complete semi-discretized equation in Ω^f can be written as

$$\rho \boldsymbol{f}_{tt} := L_h \boldsymbol{f} = \begin{cases} G_x(\mu)\boldsymbol{f}^{\boldsymbol{\Gamma}} + \underline{G_y}(\mu)\boldsymbol{f}^{\boldsymbol{\Gamma}} + \boldsymbol{\eta}, & \text{on the interface,} \\ G_x(\mu)\boldsymbol{f}^{\boldsymbol{\Omega}} + \underline{G_y}(\mu)\boldsymbol{f}^{\boldsymbol{\Omega}}, & \text{in the interior,} \end{cases} \tag{4.23}$$

where

$$\boldsymbol{\eta} = \rho \mathcal{P}\left(\frac{1}{\rho} G_c(\mu)\boldsymbol{c}^{\boldsymbol{\Gamma}}\right) - (G_x(\mu)\boldsymbol{f}^{\boldsymbol{\Gamma}} + \underline{G_y}(\mu)\boldsymbol{f}^{\boldsymbol{\Gamma}}).$$

We see two differences when comparing (4.23) with (4.8): the SBP operator $G_y(\mu)$ is replaced by $\underline{G_y}(\mu)$, and there is a penalty-type term $\boldsymbol{\eta}$ for the grid points on the interface. A modified interface condition leads to energy stability.

Theorem 4.4. *The scheme* (4.22)-(4.23) *is energy stable with the interface condition*

$$\mathcal{R}(\langle \boldsymbol{f}_t^{\boldsymbol{\Gamma}}, \underline{\boldsymbol{f}}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h - hw_1\langle(\boldsymbol{f}_t^{\boldsymbol{\Gamma}}, \boldsymbol{\eta}\rangle_h) = \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{c}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h}, \tag{4.24}$$

if the projection and restriction operators satisfy (4.11)*. The value w_1 is the weight of the SBP operator $G(\mu)$ on the first grid point.*

Proof. With the discrete energy

$$E = (\boldsymbol{f}_t, \rho \boldsymbol{f}_t)_h + \underline{S_f}(\boldsymbol{f}, \boldsymbol{f}) + (\boldsymbol{c}_t, \rho \boldsymbol{c}_t)_{2h} + S_c(\boldsymbol{c}, \boldsymbol{c}),$$

the energy change in time is

$$\frac{1}{2}\frac{d}{dt}E = -\langle \boldsymbol{f}_t^{\boldsymbol{\Gamma}}, \underline{\boldsymbol{f}}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_h + \langle \boldsymbol{c}_t^{\boldsymbol{\Gamma}}, \boldsymbol{c}_{\boldsymbol{\nabla}}^{\boldsymbol{\Gamma}}\rangle_{2h} + hw_1\langle(\boldsymbol{f}_t^{\boldsymbol{\Gamma}}, \boldsymbol{\eta}\rangle_h. \tag{4.25}$$

With the interface condition (4.24) and the requirement on the interpolation and restriction operators (4.11), the right-hand side of (4.25) vanishes, which proves energy stability. □

When combined with the fourth order accurate predictor-corrector time integration, the fully discrete scheme can be written as the predictor step

$$\rho\frac{\boldsymbol{f}^{*}-2\boldsymbol{f}^{\boldsymbol{k}}+\boldsymbol{f}^{\boldsymbol{k-1}}}{\delta_t^2}=L_h\boldsymbol{f}^{\boldsymbol{k}},\qquad \rho\frac{\boldsymbol{c}^{*}-2\boldsymbol{c}^{\boldsymbol{k}}+\boldsymbol{c}^{\boldsymbol{k-1}}}{\delta_t^2}=G_c(\mu)\boldsymbol{c}^{\boldsymbol{k}}, \tag{4.26}$$

and the corrector step

$$\boldsymbol{f}^{\boldsymbol{k+1}}=\boldsymbol{f}^{*}+\frac{\delta_t^4}{12\rho}L_h\boldsymbol{v}^{\boldsymbol{f}},\qquad \boldsymbol{c}^{\boldsymbol{k+1}}=\boldsymbol{c}^{*}+\frac{\delta_t^4}{12\rho}G_c(\mu)\boldsymbol{v}^{\boldsymbol{c}}, \tag{4.27}$$

where

$$\boldsymbol{v}^{\boldsymbol{f}}=\frac{\boldsymbol{f}^{*}-2\boldsymbol{f}^{\boldsymbol{k}}+\boldsymbol{f}^{\boldsymbol{k-1}}}{\delta_t^2}\quad\text{and}\quad \boldsymbol{v}^{\boldsymbol{c}}=\frac{\boldsymbol{c}^{*}-2\boldsymbol{c}^{\boldsymbol{k}}+\boldsymbol{c}^{\boldsymbol{k-1}}}{\delta_t^2},$$

and the superscript denotes the time level.

Assuming that the solution $\boldsymbol{f}^{\boldsymbol{k-1}}$, $\boldsymbol{c}^{\boldsymbol{k-1}}$, $\boldsymbol{f}^{\boldsymbol{k}}$ and $\boldsymbol{c}^{\boldsymbol{k}}$ are known on all grid points, the solution at $t=t^{k+1}$ can be computed as follows.

1 Impose (4.24) for $\boldsymbol{f}^{\boldsymbol{k}}$ and $\boldsymbol{c}^{\boldsymbol{k}}$. This gives a system of $n-1$ linear equations. By solving the system, we obtain $\boldsymbol{c}^{\boldsymbol{k}}$ on the ghost points.
2 Compute the predictor $\boldsymbol{f}^{*}$ and $\boldsymbol{c}^{*}$ by (4.26) for all grid points excluding the ghost points.
3 Impose (4.24) for the predictor $\boldsymbol{f}^{*}$, $\boldsymbol{c}^{*}$. This gives a system of $n-1$ linear equations, with the same coefficient matrix as the system in Step 1. By solving the system, we obtain $\boldsymbol{c}^{*}$ on the ghost points.
4 Compute the corrector $\boldsymbol{f}^{\boldsymbol{k+1}}$ and $\boldsymbol{c}^{\boldsymbol{k+1}}$ by (4.27) for all grid points excluding the ghost points.

We note that after Step 4, $\boldsymbol{c}^{\boldsymbol{k+1}}$ on the ghost points are not known yet, but will be computed in Step 1 in the next time loop. Step 1 is needed even in the first time loop, when all numerical solutions are given by the initial data. This is to make sure that the ghost point values are compatible with the algorithm to guarantee energy conservation.

The improved method presented in this section is used in numerical experiments in Section 5. The system of $n-1$ linear equations is LU-factorized before the time loop, and backward substitution is used to solve the system in every time step.

4.4. The SBP-SAT method. With stable SBP-SAT schemes for both the Neumann problem in Section 3.1 and the Dirichlet problem in Section 3.2, it is straightforward to derive the penalty terms for the interface conditions (4.2). The semi-discretization can be written

$$\rho\boldsymbol{f}_{tt}=G_x(\mu)\boldsymbol{f}+\widetilde{G}_y(\mu)\boldsymbol{f}+\boldsymbol{p}_{\boldsymbol{f}}, \tag{4.28}$$

$$\rho\boldsymbol{c}_{tt}=G_x(\mu)\boldsymbol{c}+\widetilde{G}_y(\mu)\boldsymbol{c}+\boldsymbol{p}_{\boldsymbol{c}}, \tag{4.29}$$

where

$$(\boldsymbol{f}_t,\boldsymbol{p}_{\boldsymbol{f}})_h=-\frac{1}{2}\langle(\boldsymbol{f}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}})_t,\boldsymbol{f}^{\boldsymbol{\Gamma}}-\mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}\rangle_h-\frac{\tau_f}{h}\langle\boldsymbol{f}^{\boldsymbol{\Gamma}}_t,\boldsymbol{f}^{\boldsymbol{\Gamma}}-\mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}\rangle_h+\frac{1}{2}\langle(\boldsymbol{f}^{\boldsymbol{\Gamma}})_t,\boldsymbol{f}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}-\mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_h, \tag{4.30}$$

and

$$(\boldsymbol{c}_t,\boldsymbol{p}_{\boldsymbol{c}})_{2h}=\frac{1}{2}\langle(\boldsymbol{c}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}})_t,\boldsymbol{c}^{\boldsymbol{\Gamma}}-\mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}\rangle_{2h}-\frac{\tau_c}{2h}\langle\boldsymbol{c}^{\boldsymbol{\Gamma}}_t,\boldsymbol{c}^{\boldsymbol{\Gamma}}-\mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}\rangle_{2h}-\frac{1}{2}\langle(\boldsymbol{c}^{\boldsymbol{\Gamma}})_t,\boldsymbol{c}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}-\mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}_{\boldsymbol{\nabla}}\rangle_{2h}. \tag{4.31}$$

In both (4.30) and (4.31), the first two terms penalize continuity of the solution, and the third term penalizes continuity of the flux.

Energy stability is proved in [25] for the special case when μ is constant. Following the same approach, we find that the scheme (4.28)-(4.31) is energy stable when the penalty parameters satisfy

$$\tau_f = \frac{1}{2}\tau_c \geq \max_{i,j}\left(\frac{(\mu^f_{i,1})^2}{2(\mu^f_m)_i\alpha}, \frac{(\mu^c_{j,n})^2}{2(\mu^c_m)_j\alpha}\right), \tag{4.32}$$

where $i = 1, \cdots, 2n-2$ and $j = 1, \cdots, n-1$.

In the numerical experiments in Section 5, we observe that the scheme with the penalty terms (4.30) and (4.31) leads to a suboptimal convergence rate. To recover the desired rate, we find one remedy is to use four penalty terms in the same way as in [23]. More precisely, we may replace

$$\frac{\tau_f}{h}\langle \boldsymbol{f}^{\boldsymbol{\Gamma}}_t, \boldsymbol{f}^{\boldsymbol{\Gamma}} - \mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}\rangle_h$$

by

$$\frac{\tau_f}{2h}\langle \boldsymbol{f}^{\boldsymbol{\Gamma}}_t, \boldsymbol{f}^{\boldsymbol{\Gamma}} - \mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}\rangle_h + \frac{\tau_f}{2h}\langle \boldsymbol{f}^{\boldsymbol{\Gamma}}_t, \mathcal{P}\mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}} - \mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}}\rangle_h$$

in (4.30), and replace

$$\frac{\tau_c}{2h}\langle \boldsymbol{c}^{\boldsymbol{\Gamma}}_t, \boldsymbol{c}^{\boldsymbol{\Gamma}} - \mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}\rangle_{2h}$$

by

$$\frac{\tau_c}{4h}\langle \boldsymbol{c}^{\boldsymbol{\Gamma}}_t, \boldsymbol{c}^{\boldsymbol{\Gamma}} - \mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}\rangle_{2h} + \frac{\tau_c}{4h}\langle \boldsymbol{c}^{\boldsymbol{\Gamma}}_t, \mathcal{R}\mathcal{P}\boldsymbol{c}^{\boldsymbol{\Gamma}} - \mathcal{R}\boldsymbol{f}^{\boldsymbol{\Gamma}}\rangle_{2h}$$

in (4.31). The motivation of using the four penalty terms in [23] was to stabilize the scheme when using boundary modified interpolation operators. In our case, we do not need to stabilize the scheme, as the interpolation operators are not boundary modified. But the four penalty terms do improve the convergence rate to the desired order.

A second remedy to obtain the optimal convergence rate is to use more accurate interpolation and restriction operators, which is also tested in Section 5.

5. Numerical experiments.

In this section, we conduct numerical experiments to compare the SBP-GP method and the SBP-SAT method in terms of computational efficiency. Our first focus is CFL condition, which is an important factor in solving large-scale problems. We numerically test the effect of different boundary and interface techniques on the CFL condition. We then compare L^2 error and convergence rate of the SBP-GP method and the SBP-SAT method with the same spatial and temporal discretizations. The convergence rate is computed by

$$\log\left(\frac{e_h}{e_{2h}}\right) \Big/ \log\left(\frac{1}{2}\right),$$

where e_{2h} is the L^2 error on a grid $\boldsymbol{x}$, and e_h is the L^2 error on a grid with grid size half of $\boldsymbol{x}$ in each subdomain and spatial direction.

5.1. Time-stepping stability restrictions.

We consider the scalar wave equation in one space dimension

$$u_{tt} = u_{xx} + F, \tag{5.1}$$

in the domain $x \in [-\pi/2, \pi/2]$, and choose a manufactured solution

$$u = \cos(x + 2t),$$

which is also used to obtain initial and boundary data, and the forcing function F.

We discretize equation (5.1) by using the fourth order accurate SBP operator, and use a predictor-corrector time stepping method [21] for the time integration.

In general, we do not have a closed form expression for the CFL condition. Instead, we can estimate the CFL condition by considering periodic boundary conditions and Fourier methods. More precisely, the Fourier transform of the fourth order accurate central finite difference stencil is

$$\widehat{Q} = -\frac{4}{h^2}\sin^2\frac{\omega h}{2}\left(1+\frac{1}{3}\sin^2\frac{\omega h}{2}\right),$$

where ω is the wave number and h is the grid size [6, pp. 9]. In [21], it is proved that for the predictor-corrector time stepping method, the time step constraint by the CFL condition is

$$\delta_t \leq \frac{2\sqrt{3}}{\sqrt{\kappa}}, \tag{5.2}$$

where κ is the spectral radius of the spatial discretization matrix. Taking $\kappa = \max_\omega |\widehat{Q}| = 16/(3h^2)$, we find that the estimated CFL condition is $\delta_t \leq 1.5h$, which is used as a reference when comparing CFL conditions in the following numerical tests.

First, we consider the Neumann boundary condition at $x = \pm\pi/2$, and use the SBP-GP and the SBP-SAT method to solve the equation (5.1) until $t = 200$. For the SBP-GP method with the fourth order SBP operator derived in [21], we find that the scheme is stable when $\delta_t \leq 1.44h$. In other words, the time step needs to be reduced by about 4% when comparing with the reference CFL condition. For the SBP-SAT method with the fourth order SBP operator derived in [14], the scheme is stable up to the reference CFL condition $\delta_t \leq 1.5h$.

Next, we consider the equation with the Dirichlet boundary condition at $x = \pm\pi/2$. To test the injection method and the SAT method, we use the fourth order accurate SBP operator without ghost point [14]. When using the injection method to impose the Dirichlet boundary condition, the scheme is stable with $\delta_t \leq 1.5h$. However, when using the SAT method to weakly impose the Dirichlet boundary condition and choosing the penalty parameter 20% larger than its stability-limiting value, the scheme is stable with $\delta_t \leq 1.16h$. This amounts to a reduction in time step by 23%. If we decrease the penalty parameter so that it is only 0.1% larger than its stability-limiting value, then the scheme is stable with $\delta_t \leq 1.25h$, i.e. the time step needs to be reduced by 17% comparing with the injection method.

In conclusion, for the Neumann boundary condition, both the SBP-GP and the SBP-SAT method can be used with a time step comparable to that given by the reference CFL condition. This is not surprising, given the similarity in the methods and in the discrete energy. For the Dirichlet boundary condition, we need to reduce the time step by 23% in the SAT method. If we instead inject the Dirichlet data, then the scheme is stable with the time step given by the reference CFL condition derived from Fourier analysis for the periodic boundary problem.

5.2. Discontinuous material properties.

We now investigate the SBP-GP and SBP-SAT method for the wave equation with a mesh refinement interface. The model problem is

$$\rho u_{tt} = \nabla\cdot(\mu\nabla u) + f \tag{5.3}$$

in a two dimensional domain $\Omega = [0, 4\pi]\times[-4\pi, 4\pi]$, where $\rho(x, y) > 0$, $\mu(x, y) > 0$, and the wave speed is $c = \sqrt{\mu}$. Equation (5.3) is augmented with Dirichlet boundary conditions at $y = \pm 4\pi$, and periodic boundary conditions at $x = 0$ and $x = 4\pi$.

The domain Ω is divided into two subdomains $\Omega_1 = [0, 4\pi]\times[-4\pi, 0]$ and $\Omega_2 = [0, 4\pi]\times[0, 4\pi]$ with an interface Γ at $y = 0$. The material parameter μ is a smooth function in each subdomain, but may be discontinuous across the interface. In particular, we consider two cases: μ is piecewise constant in Section 5.2, and μ is a smooth function in Section 5.3. In each case, we test the fourth order accurate SBP-GP method and the SBP-SAT method, in terms of CFL condition and convergence rate.

When μ is piecewise constant, an analytical solution can be constructed by Snell's law. We choose a unit density $\rho = 1$ and denote the piecewise constant μ as

$$\mu(x,y) = \begin{cases}\mu_1, & (x,y)\in\Omega_1,\\ \mu_2, & (x,y)\in\Omega_2,\end{cases}$$

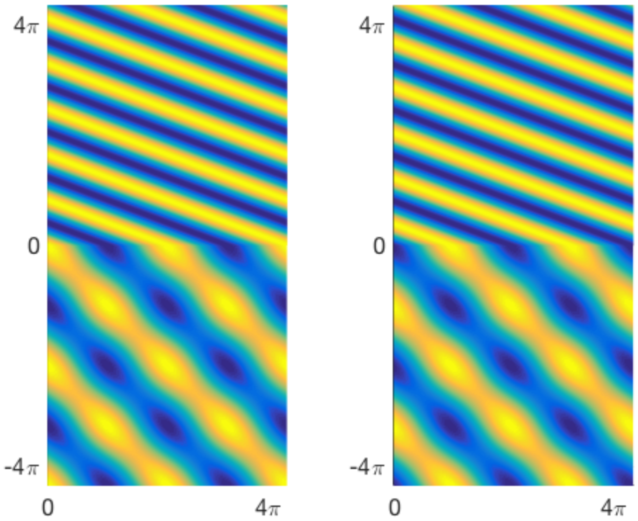

Fig. 3: The exact solution at time $t = 0$ (left) and $t = 11$ (right). The solution is continuous at the material interface $x = 0$ but the normal derivative is discontinuous due to the material discontinuity.

where $\mu_1 \neq \mu_2$.

Let an incoming plane wave u_I travel in Ω_1 and impinge on the interface Γ. The resulting field consists of the incoming wave u_I, as well as a reflected field u_R and a transmitted field u_T. With the ansatz

$$\begin{aligned} u_I &= \cos(x + y - \sqrt{2\mu_1}t),\\ u_R &= R\cos(-x + y + \sqrt{2\mu_1}t),\\ u_T &= T\cos(x + ky + \sqrt{2\mu_1}t), \end{aligned}$$

where $k = \sqrt{2\mu_1/\mu_2 - 1}$, the two parameters R and T are determined by the interface conditions

$$\begin{aligned} u_I + u_R &= u_T,\\ \mu_1 \frac{\partial}{\partial x}(u_I + u_R) &= \mu_2 \frac{\partial}{\partial x} u_T, \end{aligned}$$

yielding $R = (\mu_1 - \mu_2 k)/(\mu_1 + \mu_2 k)$ and $T = 1 + R$.

In the following experiments, we choose $\mu_1 = 1$ and $\mu_2 = 0.25$. As a consequence, the wave speed is $c_1 = 1$ in Ω_1 and $c_2 = 0.5$ in Ω_2. To keep the number of grid points per wavelength the same in two subdomains, we use a coarse grid with grid spacing $2h$ in Ω_1, and a fine grid with grid spacing h in Ω_2. We let the wave propagate from $t = 0$ until $t = 11$. The exact solution at these two time points are shown in Figure 3.

5.2.1. CFL condition. To derive an estimated CFL condition, we perform a Fourier analysis in each subdomain Ω_1 and Ω_2. Assuming periodicity in both spatial directions, the spectral radius of the spatial discretization in Ω_1 and Ω_2 is the same $\kappa = 4/(3h^2)$. By using (5.2), we find that the estimated CFL condition is

$$\delta_t \leq \frac{1}{\sqrt{2}} \frac{2\sqrt{3}}{\sqrt{4/(3h^2)}} = \frac{3}{\sqrt{2}} h \approx 2.12h. \tag{5.4}$$

We note that the restriction on time step is the same in both subdomains. The factor $1/\sqrt{2}$ in (5.4), which is not present in (5.2), comes from (5.3) having two space dimensions.

$2h$	L^2 error (rate)
1.57×10^{-1}	1.6439×10^{-3}
7.85×10^{-2}	1.0076×10^{-4} (4.02)
3.93×10^{-2}	6.2738×10^{-6} (4.01)
1.96×10^{-2}	3.9193×10^{-7} (4.00)
9.81×10^{-3}	2.4344×10^{-8} (4.01)

Table 1: L^2 errors (convergence rates) of the fourth order SBP-GP method for piecewise constant μ.

$2h$	L^2 error (rate) SAT3	L^2 error (rate) SAT4	L^2 error (rate) INT6
1.57×10^{-1}	3.0832×10^{-3}	2.1104×10^{-3}	2.1022×10^{-3}
7.85×10^{-2}	3.4792×10^{-4} (3.15)	1.1042×10^{-4} (4.26)	1.1014×10^{-4} (4.25)
3.93×10^{-2}	4.4189×10^{-5} (2.98)	6.6902×10^{-6} (4.04)	6.6815×10^{-6} (4.04)
1.96×10^{-2}	5.6079×10^{-6} (2.98)	4.0374×10^{-7} (4.05)	4.0346×10^{-7} (4.05)
9.81×10^{-3}	7.0745×10^{-7} (2.99)	2.4659×10^{-8} (4.03)	2.4651×10^{-8} (4.03)

Table 2: L^2 errors (convergence rates) of the fourth order SBP-SAT method for piecewise constant μ.

For the SBP-GP method, we have found numerically that the method is stable when the time step $\delta_t \leq 2.09h$. However, for the SBP-SAT method, the stability limit appears to be $\delta_t \leq 1.18h$, which represents approximately 45% reduction in time step. This indicates that the non-periodic boundary condition and the non-conforming grid interface do not affect time step restriction of the SBP-GP method, but the time step in the SBP-SAT method must be reduced significantly.

5.2.2. Convergence rate. We now perform a convergence study for the SBP-GP method and the SBP-SAT method. We choose the time step $\delta_t = h$ so that both methods are stable. The L^2 errors in the numerical solution with the SBP-GP method are shown in Table 1. Though the dominating truncation error is $\mathcal{O}(h^2)$ at grid points near boundaries, the numerical solution converges to fourth order, i.e. two orders are gained in convergence rate [24].

For the SBP-SAT method with three penalty terms (4.28)-(4.31), the L^2 errors labeled as SAT3 in Table 2 only converge at a rate of three. Because the dominating truncation error is $\mathcal{O}(h^2)$ at grid points close to boundaries, we gain only one order of accuracy in the numerical solution. This suboptimal convergence behavior has also been observed in other settings [24].

The proof of the suboptimal convergence behavior is out of scope of this paper. Instead, we present two simple remedies to obtain a fourth order convergence rate. First, we note that for the sixth order SBP-SAT method, energy stability requires four penalty terms when the grid interface is non-conforming [23]. When using the same type of penalty terms in the fourth order method, we obtain a fourth order convergence, as shown in the third column of Table 2 labeled as SAT4. Alternatively, we can use three penalty terms but employ a sixth order interpolation and restriction at the non-conforming interface, which also leads to a fourth order convergence, see the fourth column of Table 2 labeled as INT6. In both approaches, the dominating truncation error is still $\mathcal{O}(h^2)$ at grid points close to boundaries.

We find that the L^2 errors of the SBP-GP method is almost identical to that of the SBP-SAT method (SAT4 and INT6) with the same mesh.

5.3. Smooth material parameters. In this section, we test the two methods when the material parameters are smooth functions in the whole domain Ω. More precisely, we use material parameters

$$\begin{aligned}\rho &= -\cos(x)\cos(y)+3,\\ \mu &= \cos(x)\cos(y)+2.\end{aligned} \tag{613}$$

$2h$	L^2 error (rate)
1.57×10^{-1}	2.7076×10^{-4}
7.85×10^{-2}	1.6000×10^{-5} (4.08)
3.93×10^{-2}	9.7412×10^{-7} (4.04)
1.96×10^{-2}	6.0183×10^{-8} (4.02)
9.81×10^{-3}	3.7426×10^{-9} (4.01)

Table 3: L^2 errors (convergence rates) of the SBP-GP method for smooth μ.

$2h$	L^2 error (rate) SAT3	L^2 error (rate) SAT4	L^2 error (rate) INT6
1.57×10^{-1}	3.8636×10^{-3}	1.8502×10^{-3}	1.8503×10^{-3}
7.85×10^{-2}	4.3496×10^{-4} (3.15)	9.4729×10^{-5} (4.29)	9.4736×10^{-5} (4.29)
3.93×10^{-2}	5.3152×10^{-5} (3.03)	3.7040×10^{-6} (4.68)	3.7043×10^{-6} (4.68)
1.96×10^{-2}	6.6271×10^{-6} (3.00)	2.0778×10^{-7} (4.16)	2.0779×10^{-7} (4.16)
9.81×10^{-3}	8.2783×10^{-7} (3.00)	1.3372×10^{-8} (3.96)	1.3372×10^{-8} (3.96)

Table 4: L^2 errors (convergence rates) of the fourth order SBP-SAT method for smooth μ.

The forcing function and initial conditions are chosen so that the manufactured solution is

$$u(x, y, t) = \sin(x + 2)\cos(y + 1)\sin(t + 3).$$

We use the same grid as in Section 5.2 with grid size $2h$ in Ω_1 and h in Ω_2. The parameters $\rho_{\min} = 2$ and $\mu_{\max} = 3$ take the extreme values at the same grid point. Therefore, a Fourier analysis to the corresponding periodic problem gives a time step restriction

$$\delta_t \leq \frac{1}{\sqrt{2}} \frac{2\sqrt{3}}{\sqrt{16/(3h^2)}\sqrt{\mu_{\max}/\rho_{\min}}} = \frac{\sqrt{3}}{2} h \approx 0.86h.$$

Numerically, we have found that the SBP-GP method is stable when $\delta_t \leq 0.86h$. This shows again that the non–periodicity and interface coupling do not affect the CFL condition in the SBP-GP method. The SBP-SAT method is stable with $\delta_t \leq 0.77h$, which means that the time step needs to be reduced by approximately 10%.

To test convergence, we choose the time step $\delta_t = 0.7h$ so that both the SBP-GP method and SBP-SAT method are stable. The L^2 errors at $t = 11$ are shown in Table 3 for the SBP-GP method. We observe a fourth order convergence rate.

Similar to the case with piecewise constant material property, the standard SBP-SAT method only converges to third order accuracy, see the second column of Table 4 labeled as SAT3. We have tested the SBP-SAT method with four penalty terms, or with a sixth order interpolation and restriction operator. Both methods lead to a fourth order convergence rate, see the third and fourth column in Table 4. However, the L^2 error is about three times large as the L^2 error of the SBP-GP method with the same mesh size.

6. Conclusion.

We have analyzed two different types of SBP finite difference operators for solving the wave equation with variable coefficients; operators with ghost points, $G(\mu)$, and operators without ghost points, $\widetilde{G}(\mu)$. The close relation between the two operators has been analyzed and we have presented a way of adding or removing the ghost point dependence in the operators. Traditionally, the two operators have been used within different approaches for imposing the boundary conditions. Based on their relation, we have in this paper devised a scheme that combines both operators for satisfying the interface conditions at a non-conforming grid refinement interface.

We first used the SBP operator with ghost points to derive a fourth order accurate SBP-GP method for the wave equation with a grid refinement interface. This method uses ghost points from both sides of the refinement interface to enforce the interface conditions. Accuracy and stability of the method are ensured by using a fourth order accurate interpolation stencil and a compatible restriction stencil. Secondly, we presented an improved method, where only ghost points from the coarse side are used to impose the interface conditions. This is achieved by combining the operator $G(\mu)$ in the coarse grid and the operator $\widetilde{G}(\mu)$ in the fine grid. Compared to the first SBP-GP method, the improved method leads to a smaller system of linear equations for the ghost points. In addition, we have made improvements to the traditional fourth order SBP-SAT method, which only exhibits a third order convergence rate for the wave equation with a grid refinement interface. Two remedies have been presented and both result in a fourth order convergence rate.

We have conducted numerical experiments to verify that the proposed methods converge with fourth order accuracy, both for smooth and discontinuous material properties. We have also found numerically that the proposed SBP-GP method is stable under a CFL time-step condition that is very close to the von Neumann limit for the corresponding periodic problem. Being able to use a large time step is essential for solving practical large-scale wave propagation problems, because the computational complexity grows linearly with the number of time steps. We have found that the SBP-SAT method requires a smaller time step for stability, probably due to the penalization of the interface coupling conditions. In the case of smooth material properties, the SBP-SAT method was also found to yield to a slightly larger solution error compared to the SBP-GP method, for the same grid sizes and time step.

One disadvantage of the SBP-GP method is that a system of linear equations must be solved to obtain the numerical solutions at the ghost points. However, previous work has demonstrated that the system can be solved very efficiently by an iterative method [18, 20]. Furthermore, the proposed method only uses ghost points on one side of the interface and therefore leads to a linear system with fewer unknowns and a more regular structure than previously. In future work we plan to implement the proposed method for the elastic wave equation in three space dimensions on a distributed memory machine and evaluate its efficiency.

Acknowledgments. S. Wang would like to thank Division of Scientific Computing at Uppsala University for the support of this project. Part of the work was conducted when S. Wang was on a research visit at Lawrence Livermore National Laboratory. The authors thank B. Sjögreen for sharing his unpublished work on the SBP-GP method with ghost points on both sides of the grid refinement interface. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. This is contribution LLNL-JRNL-757334.

REFERENCES

[1] M. Almquist, S. Wang, and J. Werpers. Order preserving interpolation for summation-by-parts operators at nonconforming grid interfaces. *arXiv:1806.01931*.

[2] D. Appelö and T Hagstrom. A new discontinuous Galerkin formulation for wave equations in second–order form. *SIAM J. Numer. Anal.*, 53:2705–2726, 2015.

[3] M. H. Carpenter, D. Gottlieb, and S. Abarbanel. Time–stable boundary conditions for finite–difference schemes solving hyperbolic systems: methodology and application to high–order compact schemes. *J. Comput. Phys.*, 111:220–236, 1994.

[4] K. Duru, G. Kreiss, and K. Mattsson. Stable and high–order accurate boundary treatments for the elastic wave equation on second–order form. *SIAM J. Sci. Comput.*, 36:A2787–A2818, 2014.

[5] M. J. Grote, A. Schneebeli, and D. Schötzau. Discontinuous Galerkin finite element method for the wave equation. *SIAM. J. Numer. Anal.*, 44:2408–2431, 2006.

[6] B. Gustafsson. *High Order Difference Methods for Time Dependent PDE*. Springer, 2008.

[7] J. E. Kozdon and L. C. Wilcox. Stable coupling of nonconforming, high–order finite difference methods. *SIAM J. Sci. Comput.*, 38:A923–A952, 2016.

[8] H. O. Kreiss and J. Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24:199–215, 1972.

[9] H. O. Kreiss and G. Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. *Mathematical Aspects of Finite Elements in Partial Differential Equations, Symposium Proceedings*, pages 195–212, 1974.

[10] K. Mattsson. Summation by parts operators for finite difference approximations of second–derivatives with variable coefficient. *J. Sci. Comput.*, 51:650–682, 2012.

[11] K. Mattsson and M. H. Carpenter. Stable and accurate interpolation operators for high–order multiblock finite difference methods. *SIAM J. Sci. Comput.*, 32:2298–2320, 2010.

[12] K. Mattsson, F. Ham, and G. Iaccarino. Stable and accurate wave–propagation in discontinuous media. *J. Comput. Phys.*, 227:8753–8767, 2008.

[13] K. Mattsson, F. Ham, and G. Iaccarino. Stable boundary treatment for the wave equation on second–order form. *J. Sci. Comput.*, 41:366–383, 2009.

[14] K. Mattsson and J. Nordström. Summation by parts operators for finite difference approximations of second derivatives. *J. Comput. Phys.*, 199:503–540, 2004.

[15] S. Nilsson, N. A. Petersson, B. Sjögreen, and H. O. Kreiss. Stable difference approximations for the elastic wave equation in second order formulation. *SIAM J. Numer. Anal.*, 45:1902–1936, 2007.

[16] N. A. Petersson and B. Sjögreen. Stable grid refinement and singular source discretization for seismic wave simulations. *Commun. Comput. Phys.*, 8:1074–1110, 2010.

[17] N. A. Petersson and B. Sjögreen. User's guide to WPP version 2.2. Technical Report LLNL-SM-487431, Lawrence Livermore National Laboratory, 2011. (Source code available from computation.llnl.gov/casc/serpentine).

[18] N. A. Petersson and B. Sjögreen. Wave propagation in anisotropic elastic materials and curvilinear coordinates using a summation–by–parts finite difference method. *J. Comput. Phys.*, 299:820–841, 2015.

[19] N. A. Petersson and B. Sjögreen. User's guide to SW4, version 2.0. Technical Report LLNL-SM-741439, Lawrence Livermore National Laboratory, 2016. (Source code available from `geodynamics.org/cig`).

[20] N. A. Petersson and B. Sjögreen. High order accurate finite difference modeling of seismo-acoustic wave propagation in a moving atmosphere and a heterogeneous earth model coupled across a realistic topography. *J. Sci. Comput.*, 74:290–323, 2018.

[21] B. Sjögreen and N. A. Petersson. A fourth order accurate finite difference scheme for the elastic wave equation in second order formulation. *J. Sci. Comput.*, 52:17–48, 2012.

[22] K. Virta and K. Mattsson. Acoustic wave propagation in complicated geometries and heterogeneous media. *J. Sci. Comput.*, 61:90–118, 2014.

[23] S. Wang. An improved high order finite difference method for non–conforming grid interfaces for the wave equation. *accepted in J. Sci. Comput.*, 2018.

[24] S. Wang and G. Kreiss. Convergence of summation–by–parts finite difference methods for the wave equation. *J. Sci. Comput.*, 71:219–245, 2017.

[25] S. Wang, K. Virta, and G. Kreiss. High order finite difference methods for the wave equation with non–conforming grid interfaces. *J. Sci. Comput.*, 68:1002–1028, 2016.