UPPSALA
UNIVERSITET

UPTEC F12 011
Examensarbete 30 hp
Mars 2012

Numerical wave propagation in large-scale 3-D environments

Martin Almquist

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Numerical wave propagation in large-scale 3-D environments

Martin Almquist

High-order accurate finite difference methods have been applied to the acoustic wave equation in discontinuous media and curvilinear geometries, using the SBP-SAT method. Strict stability is shown for the 2-D wave equation with general boundary conditions. The fourth-order accurate method for the 3-D wave equation has been implemented in C and parallelized using MPI. The implementation has been verified against an analytical solution and runs efficiently on a large number of processors.

Handledare: Kristoffer Virta
Ämnesgranskare: Ken Mattsson
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F12 011

Sammanfattning

Inom detta arbete har ett effektivt verktyg för tillförlitlig simulering av akustiska vågutbredningsproblem i realistiska tredimensionella geometrier tagits fram. Verktyget är baserad på finita differensmetoder med fjärde ordningens noggrannhet. Koden har skrivits i C och parallelliserats med MPI. Problemet är väl lämpat för parallellisering och koden uppvisar mycket hög parallell effektivitet, vilket visar på potentialen för parallellberäkningar inom vågutbredningsproblem i allmänhet. Realistiska vågutbredningsproblem är ofta så beräkningsmässigt krävande att en parallell implementering är helt nödvändig för att simuleringar ska kunna genomföras.

Vågutbredningsproblem uppstår inom många tillämpningar såsom allmän relativitetsteori, seismologi, akustik och elektromagnetism. Det går att visa att högre ordningars finita differensmetoder är mycket väl lämpade för sådana problem. Vidare är det önskvärt att använda scheman som inte tillåter icke-fysikalisk tillväxt med tiden, en egenskap som kallas strikt stabilitet. En kombination av högre ordningars noggranna *summation-by-parts (SBP) operatorer* och *the Simultaneous Approximation Term (SAT) method* leder till SBP-SAT-metoden. Med SBP-SAT-metoden är det möjligt att härleda ett energiestimat för den diskretiserade modellen som exakt efterliknar det kontinuerliga energiestimatet. Därmed kan strikt stabilitet bevisas, vilket är en av metodens främsta styrkor. Dessutom är det med SBP-SAT-metoden möjligt att hantera vågutbredning i komplexa geometrier och diskontinuerliga medier noggrant, vilket gör att realistiska problem kan simuleras.

Contents

1 Introduction

Wave propagation problems arise in many applications, such as general relativity, seismology, acoustics and electromagnetics. It can be shown that high-order (higher than second order) spatially accurate finite difference schemes combined with high-order accurate time marching schemes are well suited for such problems. It is also desirable to use schemes which do not allow non-physical growth in time, a property called strict stability. The combination of high-order accurate narrow-stencil summation-by-parts (SBP) operators and the Simultaneous Approximation Term (SAT) method for imposing the physical boundary and interface conditions is here referred to as the SBP-SAT method. The SBP-SAT method makes it possible to derive an energy estimate for the discretized model which exactly mimics the continuous energy estimate. Thus, strict stability can be proved.

In practice, the media in which the waves travel are often discontinuous. Such a discontinuity in the media parameters is here referred to as an interface. In order not to lose accuracy or stability, special care must be taken when treating interfaces.

For wave propagation problems in general, the computational domain is often large compared to the wavelengths, which means that a large number of grid points is required. Thus, wave propagation problems can be computationally demanding, especially when three space dimensions are considered. When solving such large problems, it is imperative to utilize parallel computing, preferrably on a large number of cores.

We have implemented a tool, based on a fourth-order accurate SBP-SAT method, for solving acoustic wave propagation problems in three spatial dimensions. The tool has been constructed to handle curvilinear grids and irregularly shaped media interfaces. In order to speed up the computations, the code has been written in C and parallelized using the Message Passing Interface (MPI).

In this thesis we focus on the following:

1. Showing strict stability for the acoustic wave equation with general boundary conditions on a curvilinear grid.

2. Verifying the parallel implementation against an analytical solution.

3. Evaluating the parallel efficiency of the implementation.

We proceed by introducing some notation and definitions in Section 2. In Section 3 the SBP-SAT method for the 1-D case, including the treatment of media interfaces, is presented. In Section 4 we analyze a model problem with general boundary condiditons in 2-D. We then state the equations that are necessary for the 3-D implementation in Section 5. The details of the parallelization are discussed in Section 6. In Section 7 we verify the parallel implementation by presenting the results of a convergence study. The

parallel efficiency is evaluated by measuring the speedup for three different problem sizes. Conclusions are presented in Section 8.

2 Definitions

Let the inner product for real-valued functions $u, v \in l^2[0,1]$ be defined by $(u,v) = \int_0^1 u\, v\, a(x)\, dx$, $a(x) > 0$, and let the corresponding norm be $\|u\|_a^2 = (u,u)$. The domain $(0 \le x \le 1)$ is discretized using the following $N+1$ equidistant grid points:

$$x_i = i\, h, \quad i = 0, 1, \ldots, N, \quad h = \tfrac{1}{N}\ .$$

The approximate solution at grid point x_i is denoted v_i, and the discrete solution vector is $v^T = [v_0, v_1, \ldots, v_N]$. Similarly, we define an inner product for discrete real-valued vector functions $u, v \in \mathbf{R^{N+1}}$ by $(u,v)_{H_a} = u^T\, HA\, v$, where H is diagonal and positive definite and A is the projection of $a(x)$ onto the diagonal. The corresponding norm is $\|v\|_{H_a}^2 = v^T\, HA\, v$.

Remark The matrix product HA defines a norm if and only if HA is symmetric and positive definite. This can only be guaranteed if H is a diagonal matrix (see [4] for a detailed study on this).

The following vectors will be frequently used:

$$e_0 = [1, 0, \ldots, 0]^T, \quad e_N = [0, \ldots, 0, 1]^T\ . \tag{1}$$

To define the SBP-SAT method, we present the following three definitions (first stated in [3] and [1]):

Definition 2.1 *An explicit pth-order accurate finite difference scheme with minimal stencil width of a Cauchy problem is called a pth-order accurate narrow-stencil.*

Definition 2.2 *A difference operator $D_1 = H^{-1}Q$ approximating $\partial/\partial\, x$, using a pth-order accurate narrow-stencil, is said to be a pth-order accurate narrow-diagonal first-derivative SBP operator if H is diagonal and positive definite and $Q + Q^T = diag\,(-1, 0, \ldots, 0, 1)$.*

Definition 2.3 *Let $D_2^{(b)} = H^{-1}(-M^{(b)} + \bar{B}S)$ approximate $\partial/\partial\, x\,(\, b\, \partial/\partial\, x)$, where $b(x) > 0$, using a pth-order accurate narrow-stencil. $D_2^{(b)}$ is said to be a pth-order accurate narrow-diagonal second-derivative SBP operator, if H is diagonal and positive definite, $M^{(b)}$ is symmetric and positive semi-definite, S approximates the first-derivative operator at the boundaries and $\bar{B} = diag\,(-b_0, 0 \ldots, 0, b_N)$.*

We say that a scheme is explicit if no linear system of equations needs to be solved to compute the difference approximation.

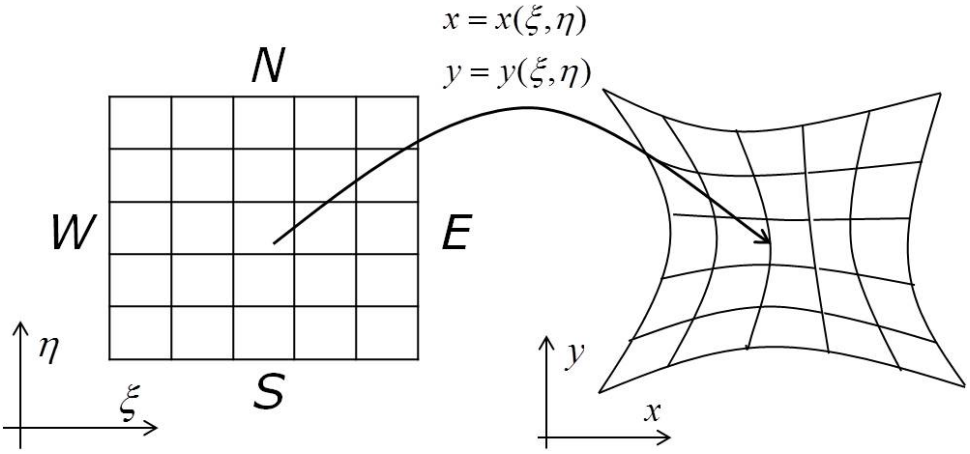

Figure 1: The mapping between cartesian (left) and curvilinear (right) coordinates for the 2-D case.

2.1 The 2-D case

To make the notation in two and three dimensions more compact we introduce the Kronecker product:

$$C \otimes D = \begin{bmatrix} c_{0,0}D & \cdots & c_{0,q-1}D \\ \vdots & & \vdots \\ c_{p-1,0}D & \cdots & c_{p-1,q-1}D \end{bmatrix}, \tag{2}$$

where C is a $p \times q$ matrix and D is an $m \times n$ matrix. We also let I_N be the $N \times N$ identity matrix.

If the problem is given on a curvilinear domain Ω, we transform it to the unit square, Ω'. We will refer to Ω as the physcial domain and Ω' as the logical domain. The logical domain is discretized using the $(N_\xi + 1)(N_\eta + 1)$ grid points:

$$(\xi_i, \eta_j) = \left(\frac{i}{N_\xi}, \frac{j}{N_\eta} \right), \qquad i = 0, 1, \ldots, N_\xi, \quad j = 0, 1, \ldots, N_\eta.$$

The boundaries of Ω' are denoted by W (west), N (north), E (east) and S (south), respectively, as shown in Figure 1. The approximate solution at a grid point (ξ_i, η_j) is denoted by v_{ij}, and the discrete solution vector is $v^T = [v_{00}, ..., v_{0N_\eta}, v_{10}, ..., v_{N_\xi N_\eta}]$. The matrix i_W is defined so that $i_W v$ is a vector with the same length as v and the same elements on the positions corresponding to the west boundary, but zeros everywhere else. The matrices i_N, i_E and i_S are defined similarly for the north, east and south boundaries, respectively.

By $D_{1\xi}$ we denote the 2-D version of the narrow-stencil first-derivative SBP operator D_1, approximating $\frac{\partial}{\partial \xi}$. Similarly, $D_{2\xi}^{(b)}$ approximates $\frac{\partial}{\partial \xi}\left(b \frac{\partial}{\partial \xi}\right)$.

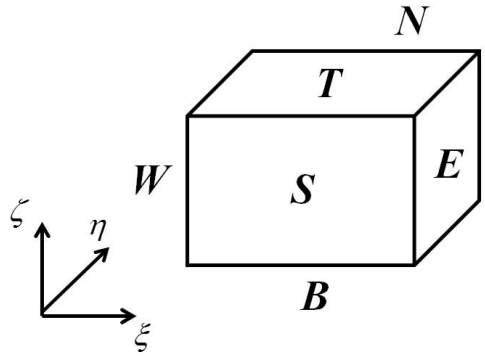

Figure 2: The logical domain Ω' in the 3-D case.

In the same manner, we let H_ξ denote the 2-D version of the diagonal matrix H, applied in the ξ-direction. $D_{1\eta}$, $D_{2\eta}^{(b)}$ and H_η are defined similarly for the η-direction.

To simplify the notation (without any restriction) we here assume $N_\xi = N_\eta = N$. The 2-D operators can then be neatly expressed in terms of the 1-D operators using the Kronecker product:

$$\begin{array}{llll} D_{1\xi} & = D_1 \otimes I_N, & D_{1\eta} & = I_N \otimes D_1 \\ D_{2\xi}^{(b)} & = D_2^{(b)} \otimes I_N, & D_{2\eta}^{(b)} & = I_N \otimes D_2^{(b)} \\ H_\xi & = H \otimes I_N, & H_\eta & = I_N \otimes H \\ i_W & = e_0 \otimes I_N, & i_S & = I_N \otimes e_0 \\ i_E & = e_N \otimes I_N, & i_N & = I_N \otimes e_N, \end{array} \tag{3}$$

where the vectors e_0 and e_N are defined in (1).

2.2 The 3-D case

If the three-dimensional problem is given on a curvilinear domain Ω, we transform it to the unit cube, Ω'. The logical domain Ω' is discretized using the $(N_\xi + 1)(N_\eta + 1)(N_\zeta + 1)$ grid points:

$$(\xi_i, \eta_j, \zeta_k) = \left(\frac{i}{N_\xi}, \frac{j}{N_\eta}, \frac{k}{N_\zeta} \right),$$
$$i = 0, 1, \ldots, N_\xi, \quad j = 0, 1, \ldots, N_\eta, \quad k = 0, 1, \ldots, N_\zeta.$$

The boundaries of Ω' are denoted by W (west), N (north), E (east), S (south), B (bottom) and T (top), respectively, as shown in Figure 2. Extending the notation presented for the 2-D case and assuming $N_\xi = N_\eta =$

$N_\zeta = N$, the 3-D operators can be expressed as follows:

$$\begin{array}{llllll}
D_{1\xi} & = I_N \otimes D_1 \otimes I_N, & D_{1\eta} & = I_N \otimes I_N \otimes D_1, & D_{1\zeta} & = D_1 \otimes I_N \otimes I_N \\
D_{2\xi}^{(b)} & = I_N \otimes D_2^{(b)} \otimes I_N, & D_{2\eta}^{(b)} & = I_N \otimes I_N \otimes D_2^{(b)}, & D_{2\zeta}^{(b)} & = D_2^{(b)} \otimes I_N \otimes I_N \\
H_\xi & = I_N \otimes H \otimes I_N, & H_\eta & = I_N \otimes I_N \otimes H, & H_\zeta & = H \otimes I_N \otimes I_N \\
i_W & = I_N \otimes e_0 \otimes I_N, & i_S & = I_N \otimes I_N \otimes e_0, & i_B & = e_0 \otimes I_N \otimes I_N \\
i_E & = I_N \otimes e_N \otimes I_N, & i_N & = I_N \otimes I_N \otimes e_N, & i_T & = e_N \otimes I_N \otimes I_N.
\end{array}$$

Remark We are using for example $D_{1\xi}$ to denote both the 2-D and 3-D operator, which might seem confusing. However, it will always be clear from context whether we are referring to the 2-D or 3-D version of the operator.

3 The SBP-SAT method

In this section we introduce the simple, yet powerful, SBP-SAT method for model problems in 1-D. We start by assuming that the media parameters are continuous and then move on to the case of discontinuous media.

3.1 Continuous media

Consider the following second-order hyperbolic equation:

$$\begin{array}{llll}
au_{tt} = (bu_x)_x, & & 0 \le x \le 1, & t \ge 0, \\
\alpha u_t - bu_x = g, & & x = 0, & t \ge 0, \\
\alpha u_t + bu_x = g, & & x = 1, & t \ge 0, \\
u = f_1, & u_t = f_2, & 0 \le x \le 1, & t = 0,
\end{array} \tag{4}$$

where $a(x) > 0$ and $b(x) > 0$. Multiplying the first equation in (4) by u_t, integrating by parts (referred to as "the energy method") and imposing the boundary conditions leads to

$$\frac{d}{dt}\left(\|u_t\|_a^2 + \|u_x\|_b^2\right) = -2\left(\alpha u_t - g\right)u_t|^{x=1} - 2\left(\alpha u_t - g\right)u_t|_{x=0}\,. \tag{5}$$

An energy estimate is obtained if $\alpha \ge 0$. The discrete approximation of (4) using the SBP-SAT method is

$$\begin{array}{ll}
Av_{tt} = D_2^{(b)}v & -H^{-1}\tau e_0\left\{(\alpha v_t - BSv)_0 - g\right\} \\
& -H^{-1}\tau e_N\left\{(\alpha v_t + BSv)_N - g\right\}\,,
\end{array} \tag{6}$$

where e_0 and e_N are defined in (1). (We assume the same initial conditions $v = f_1, \quad v_t = f_2$ as in the continuous case). The matrices A and B have the values of $a(x)$ and $b(x)$ injected on the diagonal.

Applying the energy method by multiplying (6) by $v_t^T H$ and adding the transpose leads to

$$\begin{aligned}\tfrac{d}{dt}\left(\|v_t\|_{H_a}^2 + v^T M^{(b)} v\right) = \; & -(v_t^T)_0 \,(2-2\tau)\,(BSv)_0 + (v_t^T)_N \,(2-2\tau)\,(BSv)_N \\ & +2\tau\left(v_t^T(g-\alpha v_t)\right)_0 + 2\tau\left(v_t^T(g-\alpha v_t)\right)_N \; .\end{aligned}$$

Setting $\tau = 1$ leads to

$$\frac{d}{dt}\left(\|v_t\|_{H_a}^2 + v^T M^{(b)} v\right) = -2\left(v_t^T(\alpha v_t - g)\right)_0 - 2\left(v_t^T(\alpha v_t - g)\right)_N \; . \qquad (7)$$

Equation (7) is a semi-discrete version of (5).

3.2 Discontinuous media

The following result is important in the present study:

Lemma 3.1 *The dissipative part $M^{(b)}$ of a narrow-diagonal second-derivative SBP operator has the following property:*

$$v^T M^{(b)} v = h\frac{\alpha}{b_0}(\bar{B}Sv)_0^2 + h\frac{\alpha}{b_N}(\bar{B}Sv)_N^2 + v^T \widetilde{M}^{(b)} v, \qquad (8)$$

where $\widetilde{M}^{(b)}$ is symmetric and positive semi-definite, and α a positive constant, independent of h.

For a proof of this lemma, see [1].

Second-order	Fourth-order	Sixth-order
$\alpha = 1$	$\alpha = 0.2508560249$	$\alpha = 0.1878715026$

Table 1: The value of α in Eq. (8) for the second-, fourth- and sixth-order accurate narrow-diagonal second-derivative SBP operators.

When deriving the interface conditions it is useful to consider the 1-D wave equation,

$$au_{tt} = (bu_x)_x, \quad -1 \le x \le 1, \qquad (9)$$

where the coefficients $a(x), b(x) > 0$ are discontinuous at $x = 0$. Applying the energy method leads to

$$\begin{aligned}\int_{-1}^{1} au_{tt}u_t\,dx &= \lim_{\epsilon\to 0}\left(\int_{-1}^{-\epsilon}(bu_x)_x u_t\,dx + \int_{\epsilon}^{1}(bu_x)_x u_t\,dx\right) \\ &= \lim_{\epsilon\to 0}\left(bu_x u_t|_{-1}^{1} - bu_x u_t|_{-\epsilon}^{\epsilon} - \int_{-1}^{-\epsilon} bu_x u_{xt}\,dx - \int_{\epsilon}^{1} bu_x u_{xt}\,dx\right).\end{aligned}$$

Obtaining an energy estimate requires that u_t and bu_x are continuous across the interface, in which case we have $\lim_{\epsilon\to 0}(bu_x u_t|_{-\epsilon}^{\epsilon}) = 0$ and we obtain the energy estimate

$$\frac{d}{dt}\left(\|u_t\|_a^2 + \|u_x\|_b^2\right) = 2bu_x u_t|_{-1}^{1}. \qquad (10)$$

We now consider the following problem:

$$\begin{aligned}
&a_1 u_{tt}^{(1)} = (b_1 u_x^{(1)})_x, && -1 \le x \le 0, && t \ge 0,\\
&a_2 u_{tt}^{(2)} = (b_2 u_x^{(2)})_x, && 0 \le x \le 1, && t \ge 0,\\
&u_x^{(1)} = 0, && x = -1, && t \ge 0,\\
&u_x^{(2)} = 0, && x = 1, && t \ge 0,\\
&u = f_1, \quad u_t = f_2, && -1 \le x \le 1, && t = 0,
\end{aligned} \tag{11}$$

where $a_1(0) \neq a_2(0)$, $b_1(0) \neq b_2(0)$. Here $u^{(1,2)}$ denote the solutions corresponding to the left and right domains, respectively. Note that we have chosen homogeneous Neumann conditions in order to minimize the number of boundary terms in the coming energy estimate. Other types of boundary conditions, like Dirichlet and radiation conditions, can also be used, but the main focus here is on the interface treatment.

We have to impose the interface conditions

$$u_t^{(1)} = u_t^{(2)}, \quad b_1 u_x^{(1)} = b_2 u_x^{(2)}, \tag{12}$$

at the interface ($x = 0$). Note that the first condition in (12) holds if we impose $u^{(1)} = u^{(2)}$ at the interface.

Applying the energy method to (11) with the interface conditions (12) leads to

$$\frac{d}{dt} E^{(2)} = 0, \tag{13}$$

where the energy of the two subdomains is defined as

$$E^{(2)} = \|u_t^{(1)}\|_{a_1}^2 + \|u_t^{(2)}\|_{a_2}^2 + \|u_x^{(1)}\|_{b_1}^2 + \|u_x^{(2)}\|_{b_2}^2. \tag{14}$$

The left and right domains are discretized using $(N+1)$ grid points and $v^{(1,2)}$ denote the solution vectors corresponding to the left and right domains, respectively. The semi-discrete approximation of (12) can be written

$$\begin{aligned}
&I_1 = v_N^{(1)} - v_0^{(2)} = 0\\
&I_2 = (v_t^{(1)})_N - (v_t^{(2)})_0 = 0\\
&I_3 = (\bar{B}_1 S v^{(1)})_N + (\bar{B}_2 S v^{(2)})_0 = 0,
\end{aligned} \tag{15}$$

where all conditions (also $v_N^{(1)} = v_0^{(2)}$) are written out. A semi-discretization of the homogeneous Neumann boundary conditions in (11) is given by

$$L_1 v^{(1)} = (\bar{B}_1 S v^{(1)})_0 = 0, \quad L_2 v^{(2)} = (\bar{B}_2 S v^{(2)})_N = 0. \tag{16}$$

A semi-discretization of the complete problem (11), using narrow-diagonal SBP operators and the SAT method to impose the semi-discrete interface

conditions (15) and boundary conditions (16) can be written as

$$\begin{array}{lll} A_1 v_{tt}^{(1)} & = D_2^{(b_1)} v^{(1)} & A_2 v_{tt}^{(2)} \;\; = D_2^{(b_2)} v^{(2)} \\ & +\tau H^{-1} e_N(I_1) & -\tau H^{-1} e_0(I_1) \\ & +\beta(\bar{B}_1 S)^T e_N H^{-1}(I_1) & -\beta(\bar{B}_2 S)^T e_0 H^{-1}(I_1) \\ & +\gamma H^{-1} e_N(I_3) & -\gamma H^{-1} e_0(I_3) \\ & +\sigma H^{-1} e_N(I_2) & -\sigma H^{-1} e_0(I_2) \\ & -H^{-1} e_0(L_1 v^{(1)}) & +H^{-1} e_N(L_2 v^{(2)}). \end{array} \tag{17}$$

Lemma 3.2 *The scheme (17) is strictly stable if $D_2^{(b_{1,2})}$ are narrow-diagonal SBP operators, $\sigma \leq 0$, $\gamma = -\frac{1}{2}$, $\beta = \frac{1}{2}$ and $\tau \leq -\frac{b_1+b_2}{4h\alpha}$ hold.*

Proof Applying the energy method by multiplying (17) by $(v^{(1)})_t^T H$ and $(v^{(2)})_t^T H$, respectively, and adding the transpose leads to

$$\frac{d}{dt} E_H^{(2)} = 2 w_t^T D w_t + \frac{d}{dt} x^T R x,$$

where

$$w = \begin{bmatrix} v_N^{(1)} \\ v_0^{(2)} \end{bmatrix}, \quad D = \sigma \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

and

$$x = \begin{bmatrix} v_N^{(1)} \\ v_0^{(2)} \\ (\bar{B}_1 S v^{(1)})_N \\ (\bar{B}_2 S v^{(2)})_0 \end{bmatrix}, \quad R = \begin{bmatrix} -\tau & \tau & -\frac{1}{2} & \frac{1}{2} \\ \tau & -\tau & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{\alpha}{b_1} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & -\frac{\alpha}{b_2} \end{bmatrix}.$$

Here we have used Lemma 3.1 and the fact that $D_2^{(b_{1,2})}$ are narrow-diagonal SBP operators. The discrete energy is given by

$$E_H^{(2)} = \|v_t^{(1)}\|_{HA_1}^2 + \|v_t^{(2)}\|_{HA_2}^2 + (v^{(1)})^T \widetilde{M}^{(b_1)} v^{(1)} + (v^{(2)})^T \widetilde{M}^{(b_2)} v^{(2)}.$$

Strict stability follows if D and R are negative semi-definite, which leads to the following conditions:

$$\sigma \leq 0, \quad \tau \leq -\frac{b_1 + b_2}{4h\alpha},$$

where $b_{1,2}$ denote the local values of $b_{1,2}$ at the interface. □

4 Analysis

In this section we analyze the scalar 2-D wave equation with general boundary conditions. To allow for complex domains, we transform the equation given on a curvilinear domain to an equation on a rectangular domain. We

then derive an energy estimate for the continuous case. After discretizing the model in space with the SBP-SAT method, we prove strict stability by exactly mimicking the continuous energy estimate in the semi-discrete case.

The method for dealing with discontinuous media presented in Section 3.2 can be extended to two and three dimensions, but in this section we limit ourselves to the case were the media parameters are continuous. For a detailed analysis of media interfaces in 2-D see [5].

4.1 The continuous problem

We consider the following problem:

$$\begin{array}{lll} u_{tt} = (bu_x)_x + (bu_y)_y & (x,y) \in \Omega, & t \geq 0 \\ \gamma_1 u + \gamma_2 b \nabla u \cdot \mathbf{n} + \gamma_3 u_t = 0 & (x,y) \in \partial\Omega, & t \geq 0 \\ u = f_1, \quad u_t = f_2, & (x,y) \in \Omega, & t = 0, \end{array} \tag{18}$$

where $b(x,y) > 0$. We have chosen homogeneous boundary conditions to avoid unnecessary notation in the analysis, but the analysis holds for inhomogeneous conditions as well. We also limit our present study to the case $\gamma_2 \neq 0$, which includes the important case of Neumann conditions $(\gamma_1 = 0, \gamma_2 = 1, \gamma_3 = 0)$.

Remark Dirichlet conditions $(\gamma_1 = 1, \gamma_2 = \gamma_3 = 0)$ form an important category of boundary conditions which are not included in the case $\gamma_2 \neq 0$. Treating Dirichlet conditions with the SAT method is more complicated than treating the conditions of the present study. For a detailed analysis of this matter see for example [2].

Assume that there is a smooth one-to-one mapping

$$\begin{cases} x = x(\xi, \eta) \\ y = y(\xi, \eta), \end{cases} \tag{19}$$

from Ω' to Ω. The Jacobian J of the transformation is

$$J = x_\xi y_\eta - x_\eta y_\xi. \tag{20}$$

The *scale factors* η_1 and η_2 of the transformation are defined as

$$\eta_1 = \sqrt{x_\xi^2 + y_\xi^2}, \quad \eta_2 = \sqrt{x_\eta^2 + y_\eta^2}. \tag{21}$$

Since the mapping is one-to-one, the Jacobian is everywhere non-zero. By the chain rule, we have

$$\begin{cases} u_\xi = u_x x_\xi + u_y y_\xi \\ u_\eta = u_x x_\eta + u_y y_\eta, \end{cases} \tag{22}$$

which is equivalent to

$$\begin{cases} u_x = \frac{1}{J}\left(u_\xi y_\eta - u_\eta y_\xi\right) = \frac{1}{J}\left((uy_\eta)_\xi - (uy_\xi)_\eta\right) \\ u_y = \frac{1}{J}\left(u_\eta x_\xi - u_\xi x_\eta\right) = \frac{1}{J}\left((ux_\xi)_\eta - (ux_\eta)_\xi\right). \end{cases} \tag{23}$$

Replacing u with bu_x and bu_y in (23) yields

$$\begin{aligned} (bu_x)_x &= \tfrac{1}{J}\left(\tfrac{b}{J}\left(u_\xi y_\eta - u_\eta y_\xi\right) y_\eta\right)_\xi - \tfrac{1}{J}\left(\tfrac{b}{J}\left(u_\xi y_\eta - u_\eta y_\xi\right) y_\xi\right)_\eta \\ (bu_y)_y &= \tfrac{1}{J}\left(\tfrac{b}{J}\left(u_\xi x_\eta - u_\eta x_\xi\right) x_\eta\right)_\xi - \tfrac{1}{J}\left(\tfrac{b}{J}\left(u_\xi x_\eta - u_\eta x_\xi\right) x_\xi\right)_\eta. \end{aligned} \tag{24}$$

By adding $(bu_x)_x$ and $(bu_y)_y$ and rearranging terms, the first equation in (18) can be written as

$$Ju_{tt} = (\alpha_1 u_\xi)_\xi + (\beta u_\xi)_\eta + (\beta u_\eta)_\xi + (\alpha_2 u_\eta)_\eta, \quad (\xi,\eta) \in \Omega' \tag{25}$$

where

$$\alpha_1 = \frac{b}{J}\left(y_\eta^2 + x_\eta^2\right), \beta = -\frac{b}{J}\left(y_\eta y_\xi + x_\eta x_\xi\right), \alpha_2 = \frac{b}{J}\left(y_\xi^2 + x_\xi^2\right). \tag{26}$$

Using equation (23) to transform $\nabla u \cdot \mathbf{n}$ in the second equation in (18) yields the transformed boundary condition:

$$\begin{cases} \gamma_1\eta_2 u - \gamma_2\left(\alpha_1 u_\xi + \beta u_\eta\right) + \gamma_3\eta_2 u_t = 0, & (\xi,\eta) \in W \\ \gamma_1\eta_2 u + \gamma_2\left(\alpha_1 u_\xi + \beta u_\eta\right) + \gamma_3\eta_2 u_t = 0, & (\xi,\eta) \in E \\ \gamma_1\eta_1 u - \gamma_2\left(\alpha_2 u_\eta + \beta u_\xi\right) + \gamma_3\eta_1 u_t = 0, & (\xi,\eta) \in S \\ \gamma_1\eta_1 u + \gamma_2\left(\alpha_2 u_\eta + \beta u_\xi\right) + \gamma_3\eta_1 u_t = 0, & (\xi,\eta) \in N. \end{cases} \tag{27}$$

The complete transformed problem is given by (25), (27) and the initial conditions stated in (18). Applying the energy method leads to

$$\frac{d}{dt}E = -\int_W \frac{\gamma_3}{\gamma_2}\eta_2 u_t^2 dr - \int_E \frac{\gamma_3}{\gamma_2}\eta_2 u_t^2 dr - \int_N \frac{\gamma_3}{\gamma_2}\eta_1 u_t^2 dr - \int_S \frac{\gamma_3}{\gamma_2}\eta_1 u_t^2 dr \tag{28}$$

where

$$E = \frac{1}{2}\left(\int_{\Omega'} J u_t^2 d\Omega' + \int_{\Omega'} \begin{bmatrix} u_\xi & u_\eta \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta \\ \beta & \alpha_2 \end{bmatrix} \begin{bmatrix} u_\xi \\ u_\eta \end{bmatrix} d\Omega' + BT\right) \tag{29}$$

and

$$BT = \int_W \frac{\gamma_1}{\gamma_2}\eta_2 u^2 dr + \int_E \frac{\gamma_1}{\gamma_2}\eta_2 u^2 dr + \int_N \frac{\gamma_1}{\gamma_2}\eta_1 u^2 dr + \int_S \frac{\gamma_1}{\gamma_2}\eta_1 u^2 dr. \tag{30}$$

The matrix $\begin{bmatrix} \alpha_1 & \beta \\ \beta & \alpha_2 \end{bmatrix}$ is positive definite since $\alpha_1 > 0$ and $\alpha_1\alpha_2 - \beta^2 = (x_\xi y_\eta - x_\eta y_\xi)^2 = J^2 > 0$. Thus, the problem has an energy estimate if the relations

$$\frac{\gamma_1}{\gamma_2} \geq 0, \quad \frac{\gamma_3}{\gamma_2} \geq 0 \tag{31}$$

hold.

4.2 The semi-discrete problem

The discrete version of (27) is given by

$$\begin{cases} L^W v = i_W \left\{ \gamma_1 \eta_2 v + \gamma_2 \left(\bar{B}^{(\alpha_1)} S_\xi v - \beta D_{1\eta} v \right) + \gamma_3 \eta_2 v_t \right\} = 0 \\ L^E v = i_E \left\{ \gamma_1 \eta_2 v + \gamma_2 \left(\bar{B}^{(\alpha_1)} S_\xi v + \beta D_{1\eta} v \right) + \gamma_3 \eta_2 v_t \right\} = 0 \\ L^S v = i_S \left\{ \gamma_1 \eta_1 v + \gamma_2 \left(\bar{B}^{(\alpha_2)} S_\eta v - \beta D_{1\xi} v \right) + \gamma_3 \eta_1 v_t \right\} = 0 \\ L^N v = i_N \left\{ \gamma_1 \eta_1 v + \gamma_2 \left(\bar{B}^{(\alpha_2)} S_\eta v + \beta D_{1\xi} v \right) + \gamma_3 \eta_1 v_t \right\} = 0. \end{cases} \tag{32}$$

The semi-discrete approximation of (25) and (27) using the SBP-SAT method is

$$\begin{aligned} J v_{tt} = & D_{2\xi}^{(\alpha_1)} v + D_{1\xi} \beta D_{1\eta} v + D_{1\eta} \beta D_{1\xi} v + D_{2\eta}^{(\alpha_2)} v \\ & + \tau_1 H_\xi^{-1} L^W v + \tau_1 H_\xi^{-1} L^E v + \tau_2 H_\eta^{-1} L^S v + \tau_2 H_\eta^{-1} L^N v. \end{aligned} \tag{33}$$

One of the main results of the present study is stated in the following lemma:

Lemma 4.1 *The scheme (33) is strictly stable if* $\tau_1 = \tau_2 = -\frac{1}{\gamma_2}$ *and the conditions (31) hold.*

Proof Applying the energy method by multiplying (33) by $v_t^T H_\xi H_\eta$ and adding the transpose leads to

$$\begin{aligned} \tfrac{d}{dt} E = \; & (1 + \tau_1 \gamma_2) v_t^T H_\eta \bar{B}^{(\alpha_1)} S_\xi v + (1 + \tau_2 \gamma_2) v_t^T H_\xi \bar{B}^{(\alpha_2)} S_\eta v \quad + \\ & (1 + \tau_1 \gamma_2) \, v_t^T H_\eta \left(-i_W + i_E \right) \beta D_{1\eta} v \quad + \\ & (1 + \tau_2 \gamma_2) \, v_t^T H_\xi \left(-i_S + i_N \right) \beta D_{1\xi} v \quad + \\ & \tau_1 \gamma_3 v_t^T H_\eta \eta_2 \left(i_W + i_E \right) v_t + \tau_2 \gamma_3 v_t^T H_\xi \eta_1 \left(i_S + i_N \right) v_t, \end{aligned}$$

where

$$\begin{aligned} E = \; & \tfrac{1}{2} v_t^T H_\xi H_\eta J v_t \quad + \\ & \tfrac{1}{2} \left(v^T H_\eta M_\xi^{(\alpha_1)} v + v^T H_\xi M_\eta^{(\alpha_2)} v + 2 \left(D_{1\xi} v \right)^T \beta H_\xi H_\eta \left(D_{1\eta} v \right) \right) + \\ & \tfrac{1}{2} \left(-\tau_1 \gamma_1 v^T H_\eta \eta_2 (i_W + i_E) v - \tau_2 \gamma_1 v^T H_\xi \eta_1 (i_S + i_N) v \right). \end{aligned}$$

By choosing $\tau_1 = \tau_2 = -\frac{1}{\gamma_2}$ we obtain an energy estimate completely analogous to (28). If (31) holds, we have a non-growing energy. □

5 The 3-D problem

The analysis for the 3-D case is completely analogous to that for the 2-D case shown in Section 4, but includes more terms and notation. We therefore omit the details of the analysis in this section, but for completeness we state the transformed version of the scalar 3-D wave equation with general boundary conditions, and show how to discretize the problem in space using the SBP-SAT method.

5.1 The continuous problem

We consider the problem:

$$\begin{array}{lll} u_{tt} = (bu_x)_x + (bu_y)_y + (bu_z)_z & (x,y,z) \in \Omega, & t \geq 0 \\ \gamma_1 u + \gamma_2 b \nabla u \cdot \mathbf{n} + \gamma_3 u_t = 0 & (x,y,z) \in \partial\Omega, & t \geq 0 \\ u = f_1, \quad u_t = f_2, & (x,y,z) \in \Omega, & t = 0, \end{array} \tag{34}$$

where $b(x,y,z) > 0$. Assuming that there is a smooth one-to-one mapping from the unit cube Ω' to Ω,

$$\begin{cases} x = x(\xi,\eta,\zeta) \\ y = y(\xi,\eta,\zeta) \\ z = z(\xi,\eta,\zeta), \end{cases} \tag{35}$$

we can transform the problem (34) into a problem on Ω'. The transformed partial differential equation reads

$$\begin{aligned} Ju_{tt} = &\left(\alpha_1 u_\xi + \beta_1 u_\eta + \beta_2 u_\zeta\right)_\xi \\ &+ \left(\alpha_2 u_\eta + \beta_1 u_\xi + \beta_3 u_\zeta\right)_\eta \\ &+ \left(\alpha_3 u_\zeta + \beta_2 u_\xi + \beta_3 u_\eta\right)_\zeta \end{aligned} \tag{36}$$

where

$$\begin{array}{llll} \alpha_1 = \frac{b}{J}(t_{11}^2 + t_{21}^2 + t_{31}^2), & \beta_1 = \frac{b}{J}(t_{11}t_{12} + t_{21}t_{22} + t_{31}t_{32}), \\ \alpha_2 = \frac{b}{J}(t_{12}^2 + t_{22}^2 + t_{32}^2), & \beta_2 = \frac{b}{J}(t_{11}t_{13} + t_{21}t_{23} + t_{31}t_{33}), \\ \alpha_3 = \frac{b}{J}(t_{13}^2 + t_{23}^2 + t_{33}^2), & \beta_3 = \frac{b}{J}(t_{12}t_{13} + t_{22}t_{23} + t_{32}t_{33}), \end{array} \tag{37}$$

and

$$\begin{array}{lll} t_{11} = y_\eta z_\zeta - y_\zeta z_\eta, & t_{12} = y_\zeta z_\xi - y_\xi z_\zeta, & t_{13} = y_\xi z_\eta - y_\eta z_\xi, \\ t_{21} = x_\zeta z_\eta - x_\eta z_\zeta, & t_{22} = x_\xi z_\zeta - x_\zeta z_\xi, & t_{23} = x_\eta z_\xi - x_\xi z_\eta, \\ t_{31} = x_\eta y_\zeta - x_\zeta y_\eta, & t_{32} = x_\zeta y_\xi - x_\xi y_\zeta, & t_{33} = x_\xi y_\eta - x_\eta y_\xi. \end{array} \tag{38}$$

The Jacobian J of the transformation is

$$J = x_\xi y_\eta z_\zeta + x_\eta y_\zeta z_\xi + x_\zeta y_\xi z_\eta - z_\xi y_\eta x_\zeta - z_\eta y_\zeta x_\xi - z_\zeta y_\xi x_\eta. \tag{39}$$

The boundary condition in (34) transforms into

$$\begin{cases} L_W[u] &= 0, \quad (\xi,\eta,\zeta) \in W \\ L_E[u] &= 0, \quad (\xi,\eta,\zeta) \in E \\ L_S[u] &= 0, \quad (\xi,\eta,\zeta) \in S \\ L_N[u] &= 0, \quad (\xi,\eta,\zeta) \in N \\ L_B[u] &= 0, \quad (\xi,\eta,\zeta) \in B \\ L_T[u] &= 0, \quad (\xi,\eta,\zeta) \in T, \end{cases} \tag{40}$$

where

$$\begin{aligned}
L_W[u] &= \gamma_1\sqrt{\alpha_1}u - \gamma_2\left(\alpha_1 u_\xi + \beta_1 u_\eta + \beta_2 u_\zeta\right) + \gamma_3\sqrt{\alpha_1}u_t \\
L_E[u] &= \gamma_1\sqrt{\alpha_1}u + \gamma_2\left(\alpha_1 u_\xi + \beta_1 u_\eta + \beta_2 u_\zeta\right) + \gamma_3\sqrt{\alpha_1}u_t \\
L_S[u] &= \gamma_1\sqrt{\alpha_2}u - \gamma_2\left(\beta_1 u_\xi + \alpha_2 u_\eta + \beta_3 u_\zeta\right) + \gamma_3\sqrt{\alpha_2}u_t \\
L_N[u] &= \gamma_1\sqrt{\alpha_2}u + \gamma_2\left(\beta_1 u_\xi + \alpha_2 u_\eta + \beta_3 u_\zeta\right) + \gamma_3\sqrt{\alpha_2}u_t \\
L_B[u] &= \gamma_1\sqrt{\alpha_3}u - \gamma_2\left(\beta_2 u_\xi + \beta_3 u_\eta + \alpha_3 u_\zeta\right) + \gamma_3\sqrt{\alpha_3}u_t \\
L_T[u] &= \gamma_1\sqrt{\alpha_3}u + \gamma_2\left(\beta_2 u_\xi + \beta_3 u_\eta + \alpha_3 u_\zeta\right) + \gamma_3\sqrt{\alpha_3}u_t.
\end{aligned} \tag{41}$$

5.2 The semi-discrete problem

The semi-discrete version of the boundary conditions (40) is given by

$$\begin{cases}
L^W v &= 0 \\
L^E v &= 0 \\
L^S v &= 0 \\
L^N v &= 0 \\
L^B v &= 0 \\
L^T v &= 0,
\end{cases} \tag{42}$$

where

$$\begin{aligned}
L^W v &= i_W\left\{\gamma_1\sqrt{\alpha_1}v + \gamma_2\left(\bar{B}^{(\alpha_1)}S_\xi v - \beta_1 D_{1\eta}v - \beta_2 D_{1\zeta}v\right) + \gamma_3\sqrt{\alpha_1}v_t\right\} \\
L^E v &= i_E\left\{\gamma_1\sqrt{\alpha_1}v + \gamma_2\left(\bar{B}^{(\alpha_1)}S_\xi v + \beta_1 D_{1\eta}v + \beta_2 D_{1\zeta}v\right) + \gamma_3\sqrt{\alpha_1}v_t\right\} \\
L^S v &= i_S\left\{\gamma_1\sqrt{\alpha_2}v + \gamma_2\left(-\beta_1 D_{1\xi}v + \bar{B}^{(\alpha_2)}S_\eta v - \beta_3 D_{1\zeta}v\right) + \gamma_3\sqrt{\alpha_2}v_t\right\} \\
L^N v &= i_N\left\{\gamma_1\sqrt{\alpha_2}v + \gamma_2\left(\beta_1 D_{1\xi}v + \bar{B}^{(\alpha_2)}S_\eta v + \beta_3 D_{1\zeta}v\right) + \gamma_3\sqrt{\alpha_2}v_t\right\} \\
L^B v &= i_B\left\{\gamma_1\sqrt{\alpha_3}v + \gamma_2\left(-\beta_2 D_{1\xi}v - \beta_3 D_{1\eta}v + \bar{B}^{(\alpha_3)}S_\zeta v\right) + \gamma_3\sqrt{\alpha_3}v_t\right\} \\
L^T v &= i_T\left\{\gamma_1\sqrt{\alpha_3}v + \gamma_2\left(\beta_2 D_{1\xi}v + \beta_3 D_{1\eta}v + \bar{B}^{(\alpha_3)}S_\zeta v\right) + \gamma_3\sqrt{\alpha_3}v_t\right\}.
\end{aligned}$$

The semi-discrete approximation of (36) with boundary conditions (40) using the SBP-SAT method is

$$\begin{aligned}
Jv_{tt} =& D_{2\xi}^{(\alpha_1)}v + D_{1\xi}\beta_1 D_{1\eta}v + D_{1\xi}\beta_2 D_{1\zeta}v \\
&+ D_{2\eta}^{(\alpha_2)}v + D_{1\eta}\beta_1 D_{1\xi}v + D_{1\eta}\beta_3 D_{1\zeta}v \\
&+ D_{2\zeta}^{(\alpha_3)}v + D_{1\zeta}\beta_2 D_{1\xi}v + D_{1\zeta}\beta_3 D_{1\eta}v \\
&+ \tau_1 H_\xi^{-1}L^W v + \tau_1 H_\xi^{-1}L^E v \\
&+ \tau_2 H_\eta^{-1}L^S v + \tau_2 H_\eta^{-1}L^S v \\
&+ \tau_3 H_\zeta^{-1}L^B v + \tau_3 H_\zeta^{-1}L^T v.
\end{aligned} \tag{43}$$

By applying the energy method it can be shown that the scheme (43) is strictly stable if $\tau_1 = \tau_2 = \tau_3 = -\frac{1}{\gamma_2}$ and the conditions

$$\frac{\gamma_1}{\gamma_2} \geq 0, \quad \frac{\gamma_3}{\gamma_2} \geq 0 \tag{44}$$

hold.

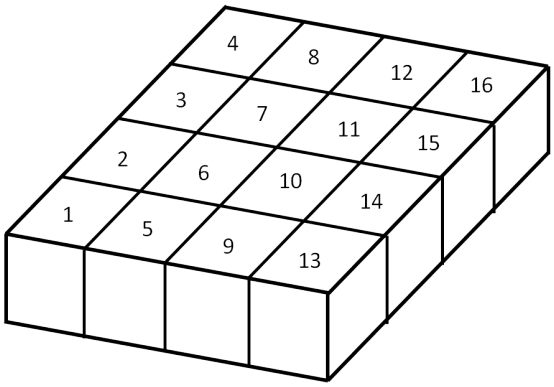

Figure 3: Decomposition of the computational domain into 16 subdomains.

6 Implementation

The fourth-order accurate finite difference method for the 3-D wave equation, i.e. (43), has been implemented in C and parallelized using the Message Passing Interface (MPI). The fourth-order accurate Runge-Kutta scheme (RK4) is used for stepping in time. The computational domain is decomposed into rectangular, non-overlapping subdomains. This is illustrated in Figure 3, where the computational domain has been decomposed into 16 subdomains. To minimize load imbalance, the subdomains should be of similar size. Each such subdomain is then assigned to one processor. Thus, the finite difference approximations can be computed in parallel. Each processor requires border data from its neighbors to compute the finite difference approximations across the subdomains. How much data each processor needs from its neighbours depends on the width of the finite difference stencil. In the case of fourth-order accurate narrow-stencils, data from two layers of grid points in all directions is needed. The concept is illustrated in Figure 4, where we have zoomed in on subdomains 1, 2, 5 and 6 in Figure 3 and separated the subdomains for illustrative purposes. The lines represent grid lines. To compute the finite difference approximations close to the borders with its neighbors, processor 1 requires data from the grid points marked in Figure 4. Therefore, the processors need to communicate with their neighbors every time the spatial finite difference approximations are computed. In the RK4 time marching scheme, this happens four times in every time step.

Since the processors require data from the same grid points in every iteration, the communication pattern is always the same. Thus, the message size, location, tag, communicator and data type in the communication calls remain the same each iteration. We can therefore use *persistent communication*, where the communication is initailized *once* and activated repeatedly, to reduce the overhead associated with redundant message setup. Note also

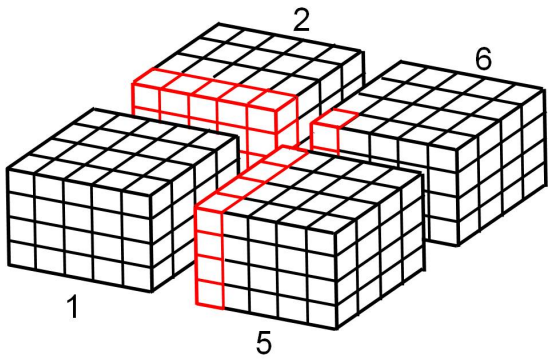

Figure 4: The communication pattern. The four large blocks are the subdomains 1, 2, 5 and 6 in Figure 3. The lines in each block are grid lines. To compute the finite difference approximations, processor 1 requires data corresponding to the grid points marked in red.

that the finite difference approximations in the interior of each subdomain do not depend on data from the neighbors. Thus, each processor can update the interior points while waiting for data from its neighbors. This allows us to reduce communication and synchronization overhead by using *non-blocking* send/receive calls and overlapping the communication with the interior computations.

7 Experiments

7.1 Convergence study

To verify the parallel implementation, a convergence study was performed against an analytical solution. The convergence rate q is calculated as

$$q = \log_{10}\left(\frac{\|u - v^{(N_2)}\|_h}{\|u - v^{(N_1)}\|_h}\right) / \log_{10}\left(\frac{N_1}{N_2}\right)^{1/d}, \tag{45}$$

where d is the dimension ($d = 3$ in the 3-D case), u is the analytical solution, $v^{(N)}$ is the corresponding numerical solution with N grid points and $\|u - v^{(N)}\|_h$ is the discrete l^2 norm of the error.

In this study, we let the physical domain Ω be a cube of side 5 with a curved interface as shown in Figure 5, where the two blocks have been separated for illustrative purposes. The computational grid was constructed using transfinite interpolation in each block. A coarse such grid is shown in Figure 5. To allow for a simple analytic solution while still testing the parallel implementation of the interface treatmenat on a curvilinear grid, we let the coefficient b be constant and continuous across the interface.

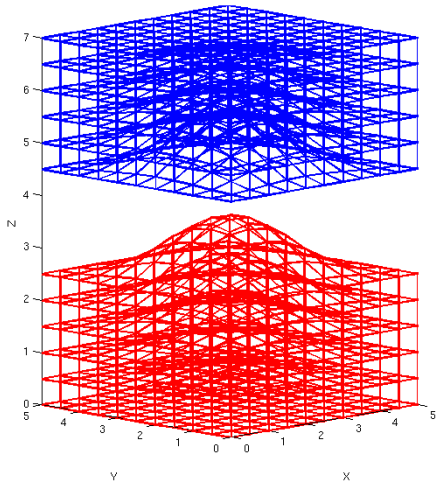

Figure 5: The domain used in the convergence study: a cube with a curved interface between two blocks. The grid was constructed with transfinite interpolation in each block.

Neglecting the interface, the test problem can be written as

$$\begin{array}{lll} u_{tt} = (bu_x)_x + (bu_y)_y + (bu_z)_z & (x,y,z) \in \Omega, & t \geq 0 \\ \nabla u \cdot \mathbf{n} = 0 & (x,y,z) \in \partial\Omega, & t \geq 0 \\ u = \cos(4\pi x)\cos(3\pi y)\cos(2\pi z), & (x,y,z) \in \Omega, & t = 0 \\ u_t = 0, & (x,y,z) \in \Omega, & t = 0. \end{array} \tag{46}$$

For $b = \frac{4}{29}$ the analytical solution to the test problem is the standing wave

$$u = \cos(4\pi x)\cos(3\pi y)\cos(2\pi z)\cos(2\pi t). \tag{47}$$

In the simulations we stepped in time until $t = 1$, using a time step $dt = 5 \cdot 10^{-4}$, and compared the result with the analytical solution (47). The results of this convergence study are presented in Table 2. It is clear that the implementation achieves the expected fourth order convergence rate.

7.2 Speedup measurements

The speedup S is defined as

$$S(p) = \frac{T(1)}{T(p)}, \tag{48}$$

where p is the number of processors and $T(p)$ is the computational time required when using p processors.

N	P	e_{l^2}	$\log_{10}(e_{l^2})$	q
100^3	18	$1.27 \cdot 10^{-3}$	-2.90	0.00
200^3	32	$4.72 \cdot 10^{-5}$	-4.33	4.75
300^3	32	$7.07 \cdot 10^{-6}$	-5.15	4.68
400^3	50	$2.43 \cdot 10^{-6}$	-5.63	3.84
500^3	50	$7.99 \cdot 10^{-7}$	-6.10	4.82
600^3	72	$3.52 \cdot 10^{-7}$	-6.45	4.49

Table 2: l^2-errors, $\log_{10}(l^2$-errors) and convergence rates. P is the number of processors used in the computations.

N_ξ	N_η	N_ζ	N
200	200	20	$8 \cdot 10^5$
400	400	40	$6.4 \cdot 10^6$
800	800	80	$5.12 \cdot 10^7$

Table 3: Problem sizes used in the speedup measurements. N is the total number of grid points.

To test the parallel efficiency of our code, we measured the time required to take 100 time steps, using different numbers of processors. All measurements were performed on resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under Project p2011136. In these tests we used the grid point configurations and problem sizes shown in Table 3. Figure 6 shows the measured speedup. One might expect to see superlinear speedup due to the fact that higher percentages of the data fit in the caches as the number of processors increase. In our implementation however, each processor uses cache-blocking, even in the serial algorithm, and thus superlinear speedup is impossible. Note that for the smallest problem size, performance decreases for $p \gtrsim 100$. This happens when the subdomains are becoming so small that the gain of making them even smaller is negated by the overhead of extra communication and synchronization between processors. The larger the problem size, the more processors one can use before performance starts to drop due to this effect. For the two larger problem sizes we do not suffer from this effect when using 144 processors or less. Since all three curves are close to the ideal case of linear speedup (until the speedup starts to decrease for the solid curve), we can conclude that the implementation is efficient.

8 Conclusions

In this thesis we have studied the acoustic wave equation in discontinuous media and curvilinear geometries in one, two and three spatial dimensions. We have described how to solve the equations with high-order accurate finite

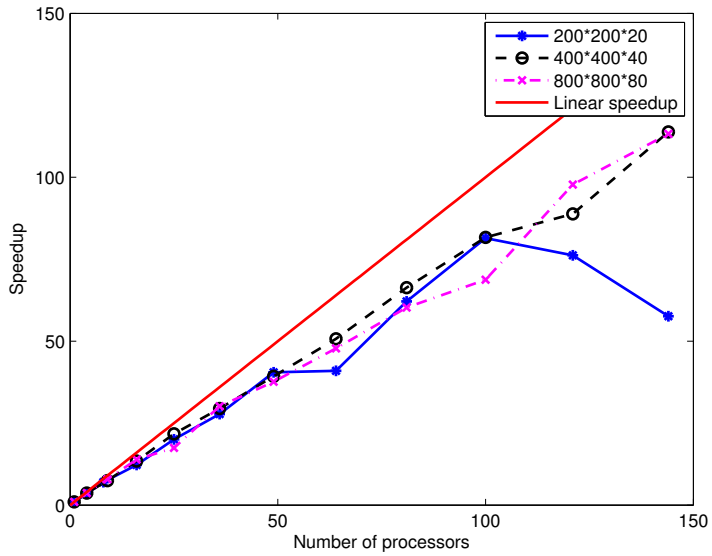

Figure 6: Speedup as a function of the number of processors. The different curves show the speedup for different problem sizes.

difference methods, using the SBP-SAT method. For the 2-D equation with general boundary conditions, we have used the energy method to prove strict stability of the finite difference scheme. We have also outlined how to make a parallel implementation of the 3-D scheme using MPI, and verified such an implentation in a convergence study against an analytical solution. Furthermore, speedup measurements have shown that the parallel efficiency of the implementation is high. For example, for the problem with $6.4 \cdot 10^6$ grid points a speedup factor of 114 was observed when using 144 cores. Since this is a considerable gain, it is apparent that parallel computing is imperative for efficent simulation of wave propagation in 3-D with high-order accurate finite difference methods.

Acknowledgements

I would like to express my heartfelt gratitude towards my supervisor, Kristoffer Virta, for introducing me to the subject and supporting me at all times. I would also like to thank Associate Professor Ken Mattsson for offering invaluable comments on a draft of this report.

References

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

[2] K. Mattsson, F. Ham, and G. Iaccarino. Stable boundary treatment for the wave equation on second-order form. *Journal of Scientific Computing*, 41:366–383, 2009.

[3] K. Mattsson, M. Svärd, and M. Shoeybi. Stable and accurate schemes for the compressible navier-stokes equations. *J. Comput. Phys.*, 227(4):2293–2316, 2008.

[4] M. Svärd. On coordinate transformation for summation-by-parts operators. *Journal of Scientific Computing*, 20(1), 2004.

[5] K. Virta. A high order finite difference scheme for wave propagation in complex geometries and heterogeneous media. Technical report, Dept. of Information Technology, Uppsala University, To appear.