The immersed interface method for acoustic wave equations with discontinuous coefficients

Chaoming Zhang\textsuperscript{a}, Randall J. LeVeque\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a} Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005-1892, USA
\textsuperscript{b} Departments of Mathematics and Applied Mathematics, University of Washington, Seattle, WA 98195, USA

Received 10 January 1996; revised 13 August 1996

Abstract

A numerical method of second order accuracy is developed to solve acoustic wave equations in heterogeneous media, whose solutions are discontinuous across the interfaces between different wave speeds. High resolution multi-dimensional flux-limiter methods are used on a Cartesian grid. Near the interface, special formulas are developed using the immersed interface method that incorporate the jump conditions and give pointwise second order accuracy even when the interface is not aligned with the grid.

1. Introduction

We consider the acoustic wave equation in two space dimensions,

\begin{align}
\rho u_t + \rho_x &= 0, \\
\rho v_t + \rho_y &= 0, \\
\rho_x + K(u_x + v_y) &= 0,
\end{align}

(1a)\hfill (1b)\hfill (1c)

where \(\rho(x, y)\), the density, and \(K(x, y)\), the bulk modulus of elasticity, are given functions of \(x\) and \(y\) that may be discontinuous across some interface(s). Our goal is to compute the pressure perturbation \(p(x, y, t)\) and velocities \(u(x, y, t)\) and \(v(x, y, t)\) on a uniform Cartesian grid to second order accuracy, even if the interfaces are curves that are not aligned with the grid.

Away from the interfaces standard finite difference methods can be used, for example the second order Lax–Wendroff method or the high resolution flux-limiter methods from \textsc{clawpack} [12] discussed in Section 5. At grid points near the interface, however, the difference equations must be modified to take into account the discontinuity in the coefficient and resulting non-smoothness in the solution. Across such an interface the tangential velocity will typically have a jump discontinuity, while the normal velocity and pressure will have jumps in their first derivatives.

\textsuperscript{*} Corresponding author. E-mail: rjl@amath.washington.edu.

0165-2125(97)$17.00 © 1997 Elsevier Science B.V. All rights reserved

PH S0165-2125(97)00046-7
In this paper we present an approach to modifying the difference scheme that, for a sufficiently smooth interface, gives second order accuracy at all grid points, including those adjacent to the interface. Numerical experiments presented in Section 6 indicate that the reflection and transmission of waves at the interface is very well modeled.

The techniques we introduce can be used for other linear hyperbolic systems as well. In the present work we consider the acoustic equations in two space dimensions as a model for demonstrating the basic approach. Extension to three space dimensions should also be direct.

Note that Eqs. (1) can be combined (via cross differentiation) to obtain a second order wave equation for the pressure,

\[ p_{tt} = c^2(p_{xx} + p_{yy}), \tag{2} \]

where \( c = \sqrt{K/\rho} \) is the wave speed. Conversely, a second order wave equation of the form (2) can be rewritten as a first order hyperbolic system by introducing new variables corresponding to derivatives. Consequently, the techniques developed here could be applied to general wave equations of the form (2) with discontinuous coefficients. However, our approach requires knowledge about the jump conditions for the solution at the interface, which depend strongly on the particular physical application. (Examples for the simple advection equation are given in Section 2.) For that reason we have chosen to present our results in the context of a particular system that is physically meaningful and has important applications. We also find it easiest to apply our techniques in the context of a first order system of equations rather than a second order equation.

Acoustic equations with discontinuous coefficients arise in many applications. Our original motivation was the study of acoustic waves in the earth. Material properties can change abruptly at interfaces between different geological materials. The acoustic wave equation is sometimes used as a simplified model for seismic modeling of the earth. Here we develop techniques for solving the forward problem of modeling wave propagation in a known medium. Ultimately we hope to use these methods to solve the important inverse problem of determining the structure from boundary measurements. The full equations of linear elasticity theory that are more properly used for modeling seismic waves can also be handled by a similar approach, and work on this extension is underway.

Acoustic equations with discontinuous coefficients also arise in the study of ocean acoustics, where the interfaces between layers of water with different properties can cause reflections as in the seismic case (e.g., [8]). Other applications include the study of acoustic lenses or waveguides, and ultrasound imaging (e.g., [35,36]).

The approach used here relies on the fact that for these problems it is possible to determine jump conditions for the solution and its derivatives at the interface. Standard finite difference methods are typically derived using Taylor series expansions and the assumption that the solution is smooth in order to approximate derivatives by finite differences. These approximations are not valid in the neighborhood of an interface where the solution is not smooth. But if the location of the interface and the jumps in the solution and its derivatives are known at the interface, then it is possible to use this information to derive difference approximations that are accurate.

This basic idea is certainly not new; it dates back at least to Tikhonov and Samarskii [32] who discuss methods for solving one-dimensional ordinary differential equations with discontinuous coefficients by essentially this technique. Other approaches for wave equations with discontinuous coefficients can be found, for example, in [1–6,9–11,25,28–30,34].

We will follow the procedure presented by LeVeque and Li [19,20,22] for Poisson problems with discontinuous coefficients or singular source terms, giving rise to non-smooth solutions across an interface. Some of the basic techniques used in this paper are described more fully there. Similar ideas have been used for the heat equation with discontinuous heat conduction coefficients [20,21,23,24] and for Stokes flow with immersed boundaries or free surfaces [18].

The interface formulas presented here are second order accurate, but the same ideas could be applied to derive higher order formulas near the interface for use in conjunction with higher order methods. It is also possible that the simpler second order formulas developed here may be quite effective in conjunction with higher order methods away
from the interface, based on results of Brown [3]. There it was shown that improvement is achieved by going from a second order method to a fourth order method even when first order accurate interface formulas are used. This is because the overall accuracy is largely determined by the phase errors introduced by the overall method, which are shown to be little affected by the interface formulas. On the other hand, our results show that better accuracy close to the interface can definitely be achieved by using better interface formulas, particularly in the velocity components which can be discontinuous across the interface.

Although we have used second order accurate methods here, we have chosen the high-resolution wave propagation algorithms of CLAWPACK rather than classical centered difference methods as used in [3] (see Section 5). These high-resolution methods using nonlinear limiter functions typically give non-oscillatory solutions that do not suffer from the same dispersive errors as centered formulas. This results in much cleaner wave propagation with very little phase error. This is demonstrated in Fig. 1, where results with the high-resolution methods are compared with results using the centered Lax–Wendroff method. This is also clearly seen in the two-dimensional results of Section 6.

2. One-dimensional advection

To illustrate the ideas most clearly, we begin by considering one-dimensional problems and start with the scalar advection equation (or 1-way wave equation)

\[ U_t + c(x)U_x = 0 \quad \text{for } 0 \leq x \leq 1, \]  \hspace{1cm} (3)

where \( c(x) \) is piecewise constant with a jump at some point \( x = \alpha \in (0, 1) \):

\[ c(x) = \begin{cases} c^- & \text{if } x < \alpha, \\ c^+ & \text{if } x > \alpha, \end{cases} \]

where we assume for concreteness that \( c^+ \) and \( c^- \) are both positive. Then to determine a unique solution we must specify initial data

\[ U(x, 0) = U_0(x), \]

boundary data

\[ U(0, t) = g(t) \]

and also a jump condition at \( x = \alpha \). We could specify, for example, that \( U \) is continuous for all time,

\[ [U] = 0 \quad \text{for all } t, \]  \hspace{1cm} (4)

where \([\cdot]\) represents the jump in a quantity at the point \( \alpha \),

\[ [U] = \lim_{x \to \alpha^+} U(x) - \lim_{x \to \alpha^-} U(x). \]

Then the solution is uniquely determined simply by following characteristics.

Note that from this jump condition we can also determine jump conditions for the derivatives of \( U \). This is extremely important for our approach, which requires a knowledge of jump conditions for the solution and its derivatives. Since \( U \) is continuous for all \( t \), the time derivative of \( U \) must also be continuous and so Eq. (3) then requires that \( cU_x \) be continuous, i.e.,

\[ [cU_x] = 0. \]  \hspace{1cm} (5)

The derivative \( U_x \) must have a jump at \( x = \alpha \) to compensate the jump in \( c \). Plots on the left of Fig. 1 show solutions to the advection equation with the jump \([U] = 0\) at \( \alpha = \frac{3}{5} \). The initial data, a single hump, are plotted on the top. The
Fig. 1. 1-D advection with $c^- = 2$ and $c^+ = 1$. The plots are the initial data at $t = 0$ (top), $t = 0.2$ (second level) and $t = 0.4$ computed with Lax–Wendroff (third level) and with CLAWPACK using the superbee limiter (bottom), respectively. On the left are results using the jump condition $[U] = 0$, and on the right using jump condition $[cU] = 0$. 
wave speeds are $c^- = 2$ for $x < \frac{a}{b}$ and $c^+ = 1$ for $x > \frac{a}{b}$, which results in the slowing down and becomes narrower as it passes the interface. The plot shows the computed (symbols ‘o’) and exact (solid lines) solutions at $t = 0.2$ (second level) and $t = 0.4$ computed with Lax–Wendroff (third level) and with CLAWPACK using the superbee limiter (bottom), respectively. Jumps in higher derivatives can also be calculated directly from the equation. Since $U$ and $U_t$ remain continuous for all $t$, so is $U_{tt}$. By differentiating (3) with respect to $t$ and also $x$ and combining the results we find that

$$U_{tt} = -cU_{xt} = -c(U_t)_x = c(cU_x)_x = cc'U_x + c^2U_{xx}.$$ 

Since $U_{tt}$ is continuous and so is $cc'U_x$ (since $c' = 0$ on both sides of the interface), we conclude that $c^2U_{xx}$ must also be continuous, so

$$[c^2U_{xx}] = 0.$$ 

(6)

This will be used below.

Note that (4) is not the only possible jump condition one could impose, and the correct condition must be determined for the physical problem being studied. If $U$ represents the concentration of a tracer in incompressible flow through a pipe with variable cross-sectional area $a(x)$, and hence velocity $c(x) = c_0/a(x)$, then the jump condition used above would be correct at an interface where the pipe suddenly becomes wider with a corresponding drop in velocity. The concentration (measured as a volume fraction) should clearly remain continuous.

On the other hand suppose we are modeling the interface between two conveyor belts moving at different speeds, with $U$ now representing the density in terms of the number of items per unit length on the belt. Then a drop in speed from the first belt to the second leads to an increase in density, so $U$ will be discontinuous. The correct interface condition is now that the flux $cU$ should be continuous. The number of items crossing the interface per unit time must be the same on both sides. So the jump condition we impose is

$$[cU] = 0.$$ 

(7)

This is also enough to specify a unique solution, though it will be different from the previous solution. Plots on the right of Fig. 1 show the solution for the same problem as before with this jump condition imposed.

Again we can compute jump conditions for higher order derivatives from (7). For example, since $cU$ must be continuous for all time, its time derivative must be continuous, so $[cU_t] = 0$. Multiplying (3) by $c$ then shows that

$$[c^2U_x] = 0,$$ 

(8)

giving the jump condition for $U_x$. In a similar manner we find

$$[c^2U_{xx}] = 0.$$ 

(9)

Once we have determined the physically relevant jump conditions and obtained a well-posed problem with a unique solution, we can begin to develop a numerical method to solve it. Of course the one-dimensional problem can be solved by characteristics and there is no need for a numerical method, but we will use this example to illustrate our procedure. In one dimension it would be sensible to choose our grid so that $\alpha$ lies at a grid point, but this will not generally be possible in two dimensions with a Cartesian grid, and so we do not assume this. We use a uniform grid with points $x_i = ih, i = 0, 1, \ldots, N$ where $h = 1/N$ and assume that $x_J \leq \alpha < x_{J+1}$ for some $J$. We also assume that the initial and boundary data are specified in such a way that the solution is smooth away from the interface, so we can use a standard finite difference method for the constant coefficient advection equation at all grid points except $x_J$ and $x_{J+1}$. For example, we might use the second order Lax–Wendroff method

$$U_{i+1}^n = U_i^n - \frac{c_i k}{2h} (U_{i+1}^n - U_{i-1}^n) + \frac{1}{2} \left( \frac{c_i k}{h} \right)^2 (U_{i+1}^n - 2U_i^n + U_{i-1}^n),$$ 

(10)

where $U_i^n \approx U(x_i, t_n), k$ is the time step, and $t_n = nk$. We use $c_i = c(x_i)$. 
The Lax–Wendroff method is based on the Taylor series expansion

\[
U(x_i, t_{n+1}) = U(x_i, t_n) + kU_t(x_i, t_n) + \frac{1}{2} k^2 U_{tt}(x_i, t_n) + \cdots
\]

\[
= U(x_i, t_n) - ckU_x(x_i, t_n) + \frac{1}{2} c^2 k^2 U_{xx}(x_i, t_n) + \cdots
\]  

(11)

The Lax–Wendroff method results from replacing \( U_x \) and \( U_{xx} \) by centered finite difference approximations and dropping the higher order terms. This works well only if \( U(x, t_n) \) is smooth in the interval from \( x_{i-1} \) to \( x_{i+1} \). At the grid points \( i = J \) and \( J + 1 \) this is not the case, and centered approximations to these derivatives yield little or no accuracy.

We will assume that \( \alpha \) is not exactly at a grid point, but lies strictly between these points. Then the Taylor series expansions (11) are still valid – the solution is smooth in both \( x \) and \( t \) at every grid point, even those near the interface. The problem comes in approximating \( U_x \) and \( U_{xx} \) by difference formulas that use values of \( U \) lying on both sides of the interface. The value \( U_x(x_j, t_n) \), for example, cannot be approximated with any accuracy by \( (1/2h)(U_{j+1} - U_{j-1}) \). However, by taking into account the jumps in \( U \) and its derivatives at \( \alpha \), it is possible to find linear combinations of the grid values that do give accurate approximations to the derivatives at the grid points. This is the basic idea in the immersed interface method. (See [19] for more examples.)

Rather than deriving expressions for \( U_x \) and \( U_{xx} \) separately, it is easiest to define a general three-point scheme of the form

\[
U_{j+1}^{n+1} = U_j^n + \frac{k}{h}(\gamma_{j,1}U_{j-1}^n + \gamma_{j,2}U_j^n + \gamma_{j,3}U_{j+1}^n),
\]

\[
U_{j+1}^{n+1} = U_{j+1}^n + \frac{k}{h}(\gamma_{j+1,1}U_{j+2}^n + \gamma_{j+1,2}U_{j+1}^n + \gamma_{j+1,3}U_j^n)
\]  

(12)

and then compute the proper \( \gamma \) coefficients so that the local truncation error gives second order accuracy.

Consider the local truncation error at \( J \):

\[
L(x_j, t_n) = \frac{1}{h}(\gamma_{j,1}U(x_{j-1}, t_n) + \gamma_{j,2}U(x_j, t_n) + \gamma_{j,3}U(x_{j+1}, t_n)) + cU_x(x_j, t_n)
\]

\[
- \frac{1}{2} kc^2 U_{xx}(x_j, t_n) + O(k^2).
\]  

(13)

Taylor series expansion at \( x = \alpha \) gives

\[
L = \frac{1}{h} \gamma_{j,1}(U^- + (x_{j-1} - \alpha)U_x^- + \frac{1}{2}(x_{j-1} - \alpha)^2 U_{xx}^-)
\]

\[
+ \frac{1}{h} \gamma_{j,2}(U^- + (x_j - \alpha)U_x^- + \frac{1}{2}(x_j - \alpha)^2 U_{xx}^-)
\]

\[
+ \frac{1}{h} \gamma_{j,3}(U^+ + (x_{j+1} - \alpha)U_x^+ + \frac{1}{2}(x_{j+1} - \alpha)^2 U_{xx}^+)
\]

\[
+ c(U_x^- + (x_j - \alpha)U_{xx}^-) - \frac{1}{2} kc^2 U_{xx}^- + O(k^2).
\]  

(14)

Here, \( U^- \) means the limiting value of \( U(x, t_n) \) as \( x \to \alpha \) from the left, and \( U^+ \) is the limiting value of \( U(x, t_n) \) as \( x \to \alpha \) from the right, and similarly for derivatives. We can eliminate \( U^+, U_x^+ \) and \( U_{xx}^+ \) by using jump conditions.

For example, with the jump condition (7) given by \( |cU| = 0 \) we also have the jump conditions for \( U_x \) and \( U_{xx} \) given by (8) and (9). These allow us to express \( U^+, U_x^+ \) and \( U_{xx}^+ \) in terms of \( U^- \), \( U_x^- \) and \( U_{xx}^- \) as follows:

\[
U^+ = d_1 U^-, \quad U_x^+ = d_2 U_x^-, \quad U_{xx}^+ = d_3 U_{xx}^-,
\]  

(15)

where,

\[
d_j = (c^- / c^+)^j, \quad (j = 1, 2, 3).
\]  

(16)
Plugging (15) into (14) yields

\[
L = \frac{1}{h} \gamma_{j,1}(U^- + (x_{j-1} - \alpha)U_x^- + \frac{1}{2}(x_{j-1} - \alpha)^2 U_{xx}^-) \\
+ \frac{1}{h} \gamma_{j,2}(U^- + (x_j - \alpha)U_x^- + \frac{1}{2}(x_j - \alpha)^2 U_{xx}^-) \\
+ \frac{1}{h} \gamma_{j,3}(d_1 U^- + d_2(x_{j-1} - \alpha)U_x^- + \frac{1}{2}d_3(x_{j+1} - \alpha)^2 U_{xx}^-) \\
+ c^- (U_x^- + (x_j - \alpha)U_{xx}^-) - \frac{1}{2}k(c^-)^2 U_{xx}^- + O(k^2),
\]

(17)

or,

\[
L = \frac{1}{h} (\gamma_{j,1} + \gamma_{j,2} + d_1 \gamma_{j,3})U^- + \frac{1}{h} ((x_{j-1} - \alpha)^2 \gamma_{j,1} + (x_j - \alpha)^2 \gamma_{j,2}) \\
+ (x_{j+1} - \alpha)\gamma_{j,3} + \frac{1}{2h} ((x_{j-1} - \alpha)^2 \gamma_{j,1} + (x_j - \alpha)^2 \gamma_{j,2}) \\
+ (x_{j+1} - \alpha)^2 d_1 \gamma_{j,3} + 2h(x_j - \alpha)c^- - kh(c^-)^2 U_{xx}^- + O(k^2).
\]

(18)

It will suffice for second order accuracy if we force the coefficients of $U^-, U_x^-$ and $U_{xx}^-$ to be zero, which leads to the $3 \times 3$ linear system for the three $\gamma$'s

\[
a_{11} \gamma_{j,1} + a_{12} \gamma_{j,2} + a_{13} \gamma_{j,3} = 0,
\]

\[
a_{21} \gamma_{j,1} + a_{22} \gamma_{j,2} + a_{23} \gamma_{j,3} = -c^-.
\]

\[
a_{31} \gamma_{j,1} + a_{32} \gamma_{j,2} + a_{33} \gamma_{j,3} = -2a_{22}c^- + \frac{k}{h}(c^-)^2.
\]

(19)

where

\[
a_{ij} = \left(\frac{(x_{j+(i-2)} - \alpha)}{h}\right)^{j-1}, \quad i, j = 1, 2, 3.
\]

(20)

Similarly, at the point $J + 1$ we obtain the linear equations

\[
\hat{a}_{11} \gamma_{J+1,1} + \hat{a}_{12} \gamma_{J+1,2} + \hat{a}_{13} \gamma_{J+1,3} = 0,
\]

\[
\hat{a}_{21} \gamma_{J+1,1} + \hat{a}_{22} \gamma_{J+1,2} + \hat{a}_{23} \gamma_{J+1,3} = -c^+.
\]

\[
\hat{a}_{31} \gamma_{J+1,1} + \hat{a}_{32} \gamma_{J+1,2} + \hat{a}_{33} \gamma_{J+1,3} = -2\hat{a}_{22}c^+ + \frac{k}{h}(c^+)^2
\]

(21)

with the coefficients

\[
\hat{a}_{ij} = \left(\frac{(x_{J+(i-2)} - \alpha)}{h}\right)^{j-1}, \quad i, j = 1, 2, 3.
\]

(22)

For different jump conditions, we only need to make slight modifications. Consider for example the jump condition $[U] = 0$ from (4), from which we obtained (5) and (6): $[cU_x] = [c^2U_{xx}] = 0$. Expressions (15) for $U^+, U_x^+$ and $U_{xx}^+$ are still valid, with the appropriate modification of $d_1, d_2, d_3$ as

\[
d_j = \left(\frac{c^-}{c^+}\right)^{j-1} \quad (j = 1, 2, 3).
\]

The linear equations for the $\gamma$'s remain the same, with these new values of $d_1, d_2, d_3$. 
Here we show two examples. Fig. 1 shows solutions to the advection equation both with the jump \([U] = 0\) (in the plots on the left) and \([cU] = 0\) (on the right) at \(\alpha = \frac{\pi}{5}\). The initial data, a single hump given by

\[
U(x, 0) = \begin{cases} 
\frac{1}{2}(1 + \cos((x - 0.28)\pi/0.24)) & \text{if } 0.04 \leq x \leq 0.52, \\
0 & \text{otherwise},
\end{cases}
\]

are plotted on the top. The wave speeds are \(c^- = 2\) for \(x < \frac{5}{9}\) and \(c^+ = 1\) for \(x > \frac{5}{9}\), which results in the slowing down and becomes narrower as it passes the interface. The plot shows the computed (symbols ‘o’) and exact (solid lines) solutions at \(t = 0.2\) (second level) and \(t = 0.4\) computed with Lax–Wendroff (third level) and with CLAWPACK (bottom), respectively. The computations are on grid with \(h = \frac{1}{30}\).

For the case of the jump condition \([U] = 0\), the solution value is continuous, though \(U_x\) is discontinuous, and the magnitude remains the same, while for the case of the jump condition \([cU] = 0\), the magnitude changes (amplified by a factor of \(c^-/c^+\)) and there is a jump discontinuity in the solution. The oscillations visible near the hump at time \(t = 0.4\) are due to the use of the dispersive Lax–Wendroff method. When CLAWPACK is used instead, this dispersive error is greatly reduced (bottom plots). Grid refinement studies confirm that second order accuracy is achieved at all grid points.

3. One-dimensional acoustic wave equations

The acoustic wave equation in one space dimension is given by

\[
\rho u_t + p_x = 0, \quad p_t + K u_x = 0,
\]

(23)

where \(p(x, t)\) is the perturbation in pressure from some uniform state \(p_0\), and \(u(x, t)\) is the velocity. The density of the elastic material or gas is \(\rho(x)\) and \(K(x)\) is the bulk modulus of elasticity for a solid, or \(\sqrt{\gamma p_0}\) for an ideal gas, where \(\gamma\) is the ratio of specific heats. See, e.g., [27,31] for discussions of these equations.

We consider the case where \(\rho\) and \(K\) are piecewise constant with jump discontinuities at some interface \(x = \alpha\):

\[
(\rho, K) = \begin{cases} 
(\rho^-, K^-) & \text{if } x < \alpha, \\
(\rho^+, K^+) & \text{if } x > \alpha.
\end{cases}
\]

The proper jump conditions for acoustic waves in a heterogeneous medium are given by

\[
[u] = 0, \quad [p] = 0
\]

(24)

at the interface \(x = \alpha\).

Though we could eliminate one variable to get a second order wave equation, we find it easier to use the system of first order equations directly. Let

\[
U = \begin{pmatrix} u(x, t) \\
p(x, t) \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1/\rho \\
K & 0 \end{pmatrix},
\]

then we get a system of first order equations

\[
U_t + AU_x = 0.
\]

(25)

In the region where \(A\) is smooth, the standard Lax–Wendroff method can be used:

\[
U_j^{n+1} = U_j^n - \frac{k}{2h} A(U_{j+1}^n - U_{j-1}^n) + \frac{1}{2} \left( \frac{k}{h} \right)^2 A^2(U_{j+1}^n - 2U_j^n + U_{j-1}^n).
\]

(26)

This can be used at all points \(j \neq J, J + 1\), where again \(J\) is such that \(x_J \leq \alpha < x_{J+1}\).
At the two points \(J\) and \(J+1\), we can no longer use the Lax–Wendroff method since the solution is not smooth. Instead, we will use a more general 3-point difference scheme of the form:

\[
\begin{align*}
U_{n+1}^{J} &= U_{n}^{J} + \frac{k}{h} (\Gamma_{J,1} U_{n-1}^{J} + \Gamma_{J,2} U_{n}^{J} + \Gamma_{J,3} U_{n+1}^{J}), \\
U_{n+1}^{J+1} &= U_{n+1}^{J+1} + \frac{k}{h} (\Gamma_{J+1,1} U_{n+2}^{J+1} + \Gamma_{J+1,2} U_{n+1}^{J+1} + \Gamma_{J+1,3} U_{n}^{J}).
\end{align*}
\]  
(27)

These have the same form as (12), but the \(\Gamma\)’s are now \(2 \times 2\) matrices. We want to choose the \(\Gamma\)’s so that second order accuracy is maintained. We proceed exactly as before, expanding the local truncation error and using the jump conditions, which now become

\[
U^+ = D_1 U^-, \quad U_x^+ = D_2 U_x^-, \quad U_{xx}^+ = D_3 U_{xx}^-,
\]  
(28)

where \(D_1, D_2, D_3\) are now \(2 \times 2\) matrices given by

\[
D_1 = I, \quad D_2 = \text{diag} \left( \frac{K^-}{K^+}, \frac{\rho^+}{\rho^-} \right), \quad D_3 = \left( \frac{c^-}{c^+} \right)^2 I.
\]  
(29)

These jump conditions are derived in Appendix A.

We obtain an expression for the local truncation error that is exactly analogous to (18), and from this a linear system of equations for the \(\Gamma\)’s analogous to (19),

\[
\begin{align*}
\tilde{a}_{11} \Gamma_{J,1} + \tilde{a}_{12} \Gamma_{J,2} + \tilde{a}_{13} \Gamma_{J,3} D_1 &= 0, \\
\tilde{a}_{21} \Gamma_{J,1} + \tilde{a}_{22} \Gamma_{J,2} + \tilde{a}_{23} \Gamma_{J,3} D_2 &= -A^-, \\
\tilde{a}_{31} \Gamma_{J,1} + \tilde{a}_{32} \Gamma_{J,2} + \tilde{a}_{33} \Gamma_{J,3} D_3 &= -2\tilde{a}_{22} A^- + \frac{k}{h} (A^-)^2
\end{align*}
\]  
(30)

with \(\tilde{a}_{ij}\) (\(i, j = 1, 2, 3\)) defined in (20). Similarly, at the point \(J+1\) we obtain the system of equations

\[
\begin{align*}
\hat{a}_{11} \Gamma_{J+1,1} + \hat{a}_{12} \Gamma_{J+1,2} + \hat{a}_{13} \Gamma_{J+1,3} D_1^{-1} &= 0, \\
\hat{a}_{21} \Gamma_{J+1,1} + \hat{a}_{22} \Gamma_{J+1,2} + \hat{a}_{23} \Gamma_{J+1,3} D_2^{-1} &= -A^+, \\
\hat{a}_{31} \Gamma_{J+1,1} + \hat{a}_{32} \Gamma_{J+1,2} + \hat{a}_{33} \Gamma_{J+1,3} D_3^{-1} &= -2\hat{a}_{22} A^+ + \frac{k}{h} (A^+)^2
\end{align*}
\]  
(31)

where \(\hat{a}_{ij}\) (\(i, j = 1, 2, 3\)) are as shown in (22). Eqs. (30) and (31) can be used to solve for \(\Gamma\)’s. Since each matrix \(\Gamma\) has four components, this is a system of 12 equations for 12 unknowns. However, since the matrices \(D_1, D_2\) and \(D_3\) are diagonal, each system can be decoupled into four disjoint \(3 \times 3\) systems of equations for the four components of the \(\Gamma\) matrices, each of which is \(3 \times 3\). These are easily solved for the \(\Gamma\)’s. Second order accuracy has been verified in numerical tests.

4. Two-dimensional acoustic wave equations

We now consider the two-dimensional acoustic wave equations (1). Let

\[
U = \begin{pmatrix} u(x, y, t) \\ v(x, y, t) \\ p(x, y, t) \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 1/\rho \\ 0 & 0 & 0 \\ K & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1/\rho \\ 0 & K & 0 \end{pmatrix}.
\]

We have the first order system

\[
U_t + AU_x + BU_y = 0.
\]  
(32)
In regions where $K$ and $\rho$ are smooth, we can use the standard Lax–Wendroff method, for example. (In practice we use the high-resolution method described in Section 5, but for simplicity we again first describe our approach in conjunction with Lax–Wendroff.) For a general system (32) the two-dimensional Lax–Wendroff method takes the form

$$U_{ij}^{n+1} = U_{ij}^n - \frac{k}{2h}(A(U_{i+1,j}^n - U_{i-1,j}^n) + B(U_{i,j+1}^n - U_{i,j-1}^n))$$
$$+ \frac{1}{2} \left(\frac{k}{h}\right)^2 (A^2(U_{i+1,j}^n - 2U_{ij}^n + U_{i-1,j}^n) + B^2(U_{i,j+1}^n - 2U_{ij}^n + U_{i,j-1}^n))$$
$$+ (AB + BA)(U_{i+1,j+1}^n - U_{i-1,j+1}^n - U_{i+1,j-1}^n + U_{i-1,j-1}^n)).$$  (33)

This is valid at regular points, where all nine points in the stencil lie on the same side of any interface.

For the acoustic equations we can simplify the Lax–Wendroff method to use a 5-point stencil if we assume that $u_y - v_x \equiv 0$, i.e., the deformation is irrotational as is generally assumed. Note that differentiating (1a) with respect to $y$, (1b) with respect to $x$, and subtracting gives $(u_y - v_x) \equiv 0$, so this condition remains true provided it is true of the initial data. Then we can eliminate the cross derivative terms since we find by simple algebra that

$$U_{tt} = A^2 U_{xx} + (AB - BA)U_{xy} + B^2 U_{yy} = c^2(U_{xx} + U_{yy}),$$

where again $c = \sqrt{K/\rho}$. This allows us to use the 5-point Lax–Wendroff formula

$$U_{ij}^{n+1} = U_{ij}^n - \frac{k}{2h}(A(U_{i+1,j}^n - U_{i-1,j}^n) + B(U_{i,j+1}^n - U_{i,j-1}^n))$$
$$+ \frac{1}{2} \left(\frac{k}{h}\right)^2 c^2(U_{i+1,j}^n + U_{i-1,j}^n + U_{i,j+1}^n + U_{i,j-1}^n - 4U_{ij}^n).$$  (34)

This is now valid at regular points where all five points in the stencil lie on the same side of the interface. This both simplifies the algebra and also reduces the number of irregular points where special treatment is needed. Moreover, our computations indicate that it gives better results as well.

Now suppose we have an interface, a smooth curve $C$ across which the coefficients are discontinuous. We can define a general interface by the equation

$$\mathcal{F}_{\text{int}}(x, y) = 0.$$  

For example, a discontinuity along the line $x = \alpha$ could be specified by taking $\mathcal{F}_{\text{int}}(x, y) = x - \alpha$.

We consider the case of piecewise constant coefficients,

$$(\rho, K) = \begin{cases} (\rho^-, K^-) & \text{if } \mathcal{F}_{\text{int}}(x, y) < 0, \\ (\rho^+, K^+) & \text{if } \mathcal{F}_{\text{int}}(x, y) > 0. \end{cases}$$

At irregular grid points, where some of the points of the 5-point stencil are on the other side of the interface, the Lax–Wendroff method is not valid anymore. At these points a 6-point difference scheme of the form

$$U_{ij}^{n+1} = U_{ij}^n + \frac{k}{h} \sum_{l=1}^{6} \Gamma_{ij,l} U_{ij,l}^{n}$$  (35)

is used. The $\Gamma$’s are $3 \times 3$ matrices, and the $U_{ij,l}^n$ for $l = 1, 6$ are some set of grid points neighboring the point $(i, j)$. The standard 5-point stencil together with one additional corner point is used, as indicated in Fig. 2. We want to choose the $\Gamma$’s so that second order accuracy is maintained. Consider the local truncation error

$$L = \frac{1}{h} \sum_{l=1}^{6} \Gamma_{ij,l} U_{l} - (U_{l}(x_i, y_j) + \frac{1}{2} k U_{tt}(x_i, y_j)) + O(k^2).$$  (36)
There will be a different set of six $\Gamma$ matrices for each point $(i, j)$. These must be derived at each point, based on the geometry of the interface near this point. We will now concentrate on a particular point $(i, j)$ and discuss the derivation of the corresponding $\Gamma$ matrices. Let $P_0 = (x_0, y_0)$ be a point on the interface near $(i, j)$ as shown in Fig. 2. We use the coordinate transformation

$$
\begin{pmatrix}
x
y
\end{pmatrix} = \begin{pmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{pmatrix} \begin{pmatrix}
\xi
\eta
\end{pmatrix} + \begin{pmatrix}
x_0
y_0
\end{pmatrix},
$$

(37)

where $\xi$ and $\eta$ are in the directions normal and tangential to the interface at $(x_0, y_0)$ respectively, and $\theta$ is the rotation angle. We assume the interface is smooth in a neighborhood of the point $(x_0, y_0)$ and can be described locally by the relation $\xi = \chi(\eta)$, with $\chi(0) = \chi'(0) = 0$ since $\xi = \eta = 0$ corresponds to the point $(x_0, y_0)$. We assume that $\chi''(0)$, the curvature, is finite since this value is needed in determining the jump conditions after the coordinate transformation.

We can then write the system (1) in $\xi$-$\eta$ coordinates and the rotated equations have the same form due to isotropy:

$$
\begin{align*}
\rho \ddot{u} + p_\xi &= 0, \quad \text{(38a)} \\
\rho \ddot{v} + p_\eta &= 0, \quad \text{(38b)} \\
p_t + K(\ddot{u}_\xi + \ddot{v}_\eta) &= 0, \quad \text{(38c)}
\end{align*}
$$

where $\mathbf{u}(\xi, \eta, t)$ and $\mathbf{v}(\xi, \eta, t)$ are the components of velocity in the $\xi$ and $\eta$ directions, respectively. Let

$$
\mathbf{U}(\xi, \eta, t) = \begin{pmatrix}
\mathbf{u}(\xi, \eta, t) \\
\mathbf{v}(\xi, \eta, t) \\
p(\xi, \eta, t)
\end{pmatrix}, \quad Q_0 = \begin{pmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{pmatrix}.
$$

(39)

Then we have

$$
U(x, y, t) = Q_0 \mathbf{U}(\xi, \eta, t),
$$

(40)

and (38) gives the transformed equations

$$
\ddot{\mathbf{U}} + A\dot{\mathbf{U}} + B\ddot{\mathbf{U}} = 0.
$$

(41)

We also want to transform (36) to $\xi - \eta$ coordinates. Expressing the derivatives of $U$ with respect to $t$ in terms of those of $\mathbf{U}$ with respect to $\xi$ and $\eta$, we obtain

$$
U_t = Q_0 \ddot{U} = -Q_0(A\ddot{U}_\xi + B\ddot{U}_\eta), \quad U_{tt} = Q_0 \dddot{U} = Q_0 c^2(\dddot{U}_{\xi\xi} + \dddot{U}_{\eta\eta}).
$$

(42)
Also, we have
\[ \sum_{l=1}^{6} \Gamma_{ij,l} U_{ij,l} = \sum_{l=1}^{6} \Gamma_{ij,l} Q_0 \bar{U}^n_{ij,l}. \]  
(43)

Since we are concentrating on a particular point \((i, j)\) we drop these subscripts, and also incorporate the necessary rotations into the \(\Gamma\)'s to further simplify the notation. We define the following \(\Gamma\)'s and \(\bar{U}\)'s as shorthand:
\[ \Gamma_{ij} = Q_0^{-1} \Gamma_{ij,l} Q_0, \quad \bar{U}_{ij} = \bar{U}^n_{ij,l}. \]  
(44)

Then with (42)-(44) the local truncation error (36) at point \((i, j)\) has the form
\[ L = Q_0 \left\{ \frac{1}{h} \sum_{l=1}^{6} \Gamma_{ij} \bar{U}_{ij} - \left( \bar{U}_i(x_i, y_j) + \frac{1}{2} k \bar{U}_{ii}(x_i, y_j) \right) \right\} + O(k^2). \]  
(45)

Let \(U^-\) and \(U^+\) represent the limiting values of \(\bar{U}(x, y)\) as we approach the point \((x_0, y_0)\) on the interface from the \(-\) and \(+\) sides, respectively. Expanding \(\bar{U}_l (l = 1, 2, \ldots, 6)\) about the point \((x_0, y_0)\), we get
\[ \bar{U}_l = U^* + (\xi_l U_{\xi}^* + \eta_l U_{\eta}^*) + \frac{1}{2} (\xi_l^2 U_{\xi\xi}^* + 2 \xi_l \eta_l U_{\xi\eta}^* + \eta_l^2 U_{\eta\eta}^*) + O(k^2), \]  
(46)

where \(\xi_l\) and \(\eta_l\) are coordinates of point \((x_l, y_l)\) in the \(\xi - \eta\) coordinates, and \(* = -\) or \(+\) depending on whether the \(l\)th point is on the \(-\) or on the \(+\) side of the interface. The point at the center of the stencil is denoted by \(l = 2\) (see Fig. 2) and at this point we can compute the time derivatives (using (42)) to be
\[ \bar{U}_t = -AU_{\xi}^* - BU_{\eta}^* - (A \xi_2 U_{\xi\xi}^* + (A \eta_2 + B \xi_2) U_{\xi\eta}^* + B \eta_2 U_{\eta\eta}^*) + O(k^2), \]
\[ \bar{U}_{tt} = c^2(U_{\xi\xi}^* + U_{\eta\eta}^*) = c^2(U_{\xi\xi}^* + U_{\eta\eta}^*) + O(k). \]  
(47)

We must now use the jump conditions to replace limiting values on one side of the interface by values on the other side, just as we did in one dimension. The jump conditions are similar to (24) from the one-dimensional case
\[ [n \cdot u] = [p] = 0, \]  
(48)

where \(n\) is the unit vector in the direction normal to the interface. More generally specifying \([n \cdot u]\) and \([p]\) as given functions of distance along the interface defines a unique solution. The jump condition for the velocity component in the direction tangential to the interface can be deduced from these jump conditions, together with the initial conditions.

From the second equation of (38), we have
\[ [p] = 0 \implies [p_\eta] = 0 \implies [\rho \vec{v}_t] = 0 \implies [\rho \vec{v}] = 0. \]

Note that \([\rho \vec{v}]\) is constant in time and hence maintains the same value for all time at each point on the interface, and is completely determined by the initial conditions. For simplicity, we assume that \([\rho \vec{v}] = 0\), noting that the procedure used here can be carried over for a general jump. Then our jump conditions are given by
\[ [\vec{u}] = 0, \quad [\rho \vec{v}] = 0, \quad [p] = 0. \]  
(49)

With these jump conditions we have the following relations between \(+\) and \(-\) values at the interface (which are derived in Appendix A):
\[ U^+ = Q_1 U^-, \]
\[ U^+ = Q_1 U^-, \]
\[ U^+ = Q_2 U^+ + Q_3 U^-, \]
\[ U^+ = Q_4 \left( \frac{c^-}{c^+} \right)^2 Q_1 U^- + \left( \frac{c^-}{c^+} \right)^2 - 1 \]
\[ \left( \frac{c^-}{c^+} \right) Q_1 U^+ + \chi''(0) \left( (Q_2 - Q_1) U^+ + Q_3 U^- \right), \]
\[ U^+ = Q_2 U^+ + Q_3 U^+ + \chi''(0) Q_4 \left( (Q_2 - Q_1) U^+ + Q_3 U^- \right), \]

where the \( Q \)’s are given as follows:

\[ Q_1 = \text{diag}(1, (\rho^+ / \rho^-), 1), \quad Q_2 = \text{diag}((K^- / K^+), 1, (\rho^+ / \rho^-)), \]
\[ Q_3 = \frac{\rho^-}{\rho^+} \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad Q_4 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \]

and \( \chi''(0) \) is the curvature of the interface at the point \((x_0, y_0)\). Exchanging + and - signs in the superscripts of all terms (including the \( Q \)’s) in (50), we can express \( U^-, U^- \), ..., \( U^+ \), in terms of \( U^+ \), \( U^+ \), ..., \( U^+ \). As in the one-dimensional case, plugging (46) and (47) into (45), and replacing \( U^+ \), \( U^+ \), ..., \( U^+ \) with jump condition (50), we will get the local truncation error expressed in terms of \( U^- \), \( U^- \), ..., \( U^- \). It will suffice for second order accuracy if we can make all the coefficients of \( U^- \), \( U^- \), \( U^- \), \( U^- \), \( U^- \), \( U^- \) be zero in the local truncation error \( L \). This gives the following six equations:

\[ \sum_{l=1}^{6} \gamma_1 \hat{Q}_{1l} = 0, \quad \sum_{l=1}^{6} \gamma_1 \hat{Q}_{4l} = \frac{k}{h} c^2 - 2 \xi_2^2 A, \]
\[ \sum_{l=1}^{6} \gamma_1 \hat{Q}_{2l} = -A, \quad \sum_{l=1}^{6} \gamma_1 \hat{Q}_{5l} = -\frac{2}{h} (\eta_2 A + \xi_2 B), \]
\[ \sum_{l=1}^{6} \gamma_1 \hat{Q}_{3l} = -B, \quad \sum_{l=1}^{6} \gamma_1 \hat{Q}_{6l} = \frac{k}{h} c^2 - 2 \eta_2^2 B, \]

where \( c = c^- \) or \( c^+ \) depending on which side the center point of the stencil sits, and the \( \hat{Q} \)’s are \( 3 \times 3 \) matrices. If point \( l \) is on the same side as the center point, then

\[ \hat{Q}_{1l} = I, \quad \hat{Q}_{2l} = \frac{\xi_l}{h} I, \quad \hat{Q}_{3l} = \frac{\eta_l}{h} I, \quad \hat{Q}_{4l} = \left( \frac{\xi_l}{h} \right)^2 I, \quad \hat{Q}_{5l} = 2 \frac{\xi_l \eta_l}{h^2} I, \quad \hat{Q}_{6l} = \left( \frac{\eta_l}{h} \right)^2 I. \]

Otherwise,

\[ \hat{Q}_{1l} = Q_1, \quad \hat{Q}_{2l} = \frac{\xi_l}{h} Q_2 + \frac{1}{2 h^2} \chi''(0) (\xi_l^2 - \eta_l^2 + 2 \xi_l \eta_l Q_4) (Q_2 - Q_1), \]
\[ \hat{Q}_{3l} = \frac{\eta_l}{h} Q_1 + \frac{\xi_l}{h} Q_3 + \frac{1}{2 h^2} \chi''(0) (\xi_l^2 - \eta_l^2 + 2 \xi_l \eta_l Q_4) Q_3, \quad \hat{Q}_{4l} = \left( \frac{\xi_l c^-}{h c^+} \right)^2 Q_1, \]
\[ \hat{Q}_{5l} = 2 \frac{\xi_l \eta_l}{h^2} Q_2, \quad \hat{Q}_{6l} = \left( \frac{\eta_l}{h} \right)^2 Q_1 + 2 \frac{\xi_l \eta_l}{h^2} Q_3 + \left( \frac{\xi_l}{h} \right)^2 \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) Q_1. \]

Note that the \( \Gamma \)’s for one-dimensional problem arise as a special case of those for the two-dimensional problem. The six matrix equations can be used to solve for \( \Gamma \) (\( l = 1, \ldots, 6 \)). Since each \( \Gamma \) is \( 3 \times 3 \), we have 54 equations for
54 unknowns. Normally we do not need to solve for the 54 unknowns simultaneously, but instead we can decouple it into smaller systems of equations. Note that \( \hat{Q} \)'s in (53) are always diagonal, therefore how we can decouple Eq. (52) depends on the structure of \( \hat{Q} \)'s in (54), which relies on the application including the interface and the jump conditions.

In the special case of a straight line interface, since \( \chi''(0) = 0 \), and \( \hat{Q}_{21}, \hat{Q}_{22} \) and \( \hat{Q}_{33} \) in (54) are diagonal or upper-triangular matrices, the system can be decoupled into nine systems of equations, each of which is 6 x 6. In this case, \( Q_1 \) and \( Q_2 \) are diagonal. But since \( Q_3(1, 2) \) is non-zero, and therefore so are \( \hat{Q}_{31}(1, 2) \) and \( \hat{Q}_{66}(1, 2) \), we see that \( \Gamma(1, j) \) and \( \Gamma(2, j) \) \( (j = 1, 2, 3) \) are coupled with each other. We need to solve for \( \Gamma(1, j) \) before we solve for \( \Gamma(2, j) \) \( (j = 1, 2, 3) \).

If not all the \( Q \)'s are diagonal, instead are triangular matrices, the nine systems of equations are coupled with each other. So they have to be solved sequentially. More typically, as in the case of curved interface, the \( Q \)'s will not be of any particular form, such as diagonal or triangular form. In our case, \( Q_4 \) is not of any particular form, therefore nor is \( \hat{Q}_{22} \) or \( \hat{Q}_{33} \) due to non-zero \( \chi''(0) \). Then we will need to solve a system of equations, from (52), of the form

\[
\sum_{l=1}^{6} \Gamma_{jl} \hat{Q}_{jl} = F_j, \quad j = 1, 2, \ldots, 6,
\]

where \( F_j \in \mathbb{R}^{3 \times 3} \) \( (j = 1, 2, \ldots, 6) \) are the right-hand sides of Eqs. (52). Taking the transpose of the system, we get

\[
\sum_{l=1}^{6} \hat{Q}^t_{jl} \Gamma^t_l = F^t_j, \quad j = 1, 2, \ldots, 6,
\]

which is equivalent to

\[
\hat{Q} \hat{\Gamma} = \hat{F}, \tag{55}
\]

where \( \hat{Q} \) is a 6 x 6 block matrix, with block element \( (j, l) \) being \( \hat{Q}^t_{jl} \), and \( \hat{\Gamma} \) and \( \hat{F} \) are block matrices of dimension 6 x 1, with block element \( l \) of \( \hat{\Gamma} \) and \( j \) of \( \hat{F} \) being \( \Gamma^t_l \) and \( F^t_j \), respectively. Solving for the system of equation (55) is equivalent to solving three systems of 18 linear equations, with the three systems each having the same coefficient matrix \( \hat{Q} \). Also note that the calculation of the \( \Gamma \)'s for each irregular point is independent of all other points, lending itself to parallel computing. More importantly, the procedure of determining the finite difference formulas at the irregular points needs to be done only once as a preprocessing stage. The same formulas are then used in every time step in solving the wave equation.

Let us compare three computational costs: the cost for the preprocessing step to obtain the \( \Gamma \)'s (one-time cost), the cost for updating values at irregular points near the interface, and the cost for the entire computation. Suppose we are working on an \( m \times m \) grid, with \( m = 1/h \) where \( h \) is the grid spacing. Since the interfaces are only curves in the plane, when we refine the grid the number of irregular points near the interface will grow as \( O(m) \). Thus the cost to compute the \( \Gamma \)'s is \( O(m) \). The cost to update \( U \) values at all irregular points near the interface is \( O(mn) \), where \( n \) is the number of time steps. The cost to use a standard solver over all grid points is \( O(m^2n) \), where \( O(m^2) \) comes from the total number of grid points. The total number of time steps \( n \) used in the computation also grows like \( O(m) \) as we refine the grid.

In comparing these we see that the cost of using the immersed interface method is low compared to the cost of advancing the solution on main portion of the grid. Table 1 in Section 6 confirms the above analysis and shows that, with CLAWPACK being our standard solver, the cost of using the immersed interface method is very low compared with the total computational cost.
5. High-resolution methods

The procedure described above yields a 6-point stencil which can be used at the irregular points where a standard finite difference method would break down. Away from the interface, any standard method with second order accuracy can be used. Above we have discussed the simple Lax–Wendroff method, but in the examples given in Section 6 we have instead used a high-resolution multi-dimensional flux-limiter method of the sort developed for multi-dimensional nonlinear systems of conservation laws, particularly in the field of gas dynamics (see, e.g., [14]). The particular algorithm used is the wave propagation method described in [15,17] and implemented in the CLAWPACK (Conservation LAWS Package) software available from netlib [12]. A description of these algorithms, including applications to acoustic equations, can be found in [17], which includes a discussion of acoustic equations with discontinuous coefficients. See also the CLAWPACK User Notes [13].

For more general interfaces, CLAWPACK is coupled with the special methods developed in this paper. The CLAWPACK algorithm is a finite volume method, meaning that each value $U_{ij}$ is viewed as a cell average over a grid cell of size $h^2$ centered at the point $(x_i, y_j)$. Each cell average is updated due to waves that propagate into the cell, as determined by the solution of Riemann problems at the four edges of these cells, and also the eight edges between cells that are adjacent to cell $(i, j)$. Hence the method has at least a 9-point stencil. In addition, a flux limiter can be used to eliminate oscillations in the solution due to the dispersive nature of the second order method. The use of these limiters extends the stencil out one more point in each coordinate direction, giving a 13-point stencil.

In practice, however, we have found that it is necessary to use the special stencils derived in this paper only at grid points for which the 9-point stencil crosses the interface. If the limiter uses a point on the wrong side of the interface it is not catastrophic provided the limiter is not allowed to increase the magnitude of a jump based on information from the neighboring Riemann problem. In other words the limiter function must lie between 0 and 1 and so we have used the minmod limiter in these two-dimensional experiments. (See [16,17] for discussions of limiters.) A limiter such as superbee, which often gives sharper discontinuities in other problems, can lead to oscillations and poor performance near the interface in the present context.

Moreover, at a point where the 9-point stencil crosses the interface but the 5-point stencil used by the Lax–Wendroff formula (34) does not, we can use (34). Thus the special 6-point formulas developed in Section 4 are needed only at points where the standard Lax–Wendroff method cannot be used.

Using CLAWPACK rather than the standard Lax–Wendroff method over most of the grid has several advantages. It is more stable than Lax–Wendroff and allows a time step $k$ for which $ck/h \leq 1$, i.e., up to Courant number 1. Lax–Wendroff is stable only for $ck/h < 0.7$ roughly (see [17]). The wave propagation approach is also slightly more accurate than Lax–Wendroff when no limiters are used, and gives a great reduction in dispersive oscillations and phase errors with the use of limiters. Finally, the wave propagation methods allow one to simply extrapolate the solution at computational boundaries where non-reflecting computational boundary conditions are desired. Though not perfect, this gives a very reasonable solution with very little work. This is discussed further in [13].

In practice we use the following approach to advance the solution by one-time step. The time step is first taken with CLAWPACK, updating the solution at all points on the grid based on assuming that the coefficients are piecewise constant in each grid cell. Then, for any irregular cells as defined above, we recompute a value based on either (34) or the special 6-point stencil and overwrite the value computed by CLAWPACK.

One natural concern with this hybrid approach is the stability of the resulting method, which is always a concern with hyperbolic problems having interfaces. For the one-dimensional problem it should be possible to analyze this using the stability theory of [7,26,33], etc. For two-dimensional problems with general curved interfaces there seems to be little hope of proving stability rigorously. Numerous computational experiments have shown no instabilities whatsoever, but this remains an interesting open question.

We also present some results below obtained with CLAWPACK alone, without incorporating the modifications near the interface that are the subject of this paper. These wave propagation algorithms work surprisingly well on their own, simply by setting the density $\rho_{ij}$ and bulk modulus $K_{ij}$ in the $(i, j)$ cell to be the average of the functions $\rho(x, y)$ and $K(x, y)$ over this grid cell. In general this will smear the interface to the nearby grid cell interfaces,
and so we might expect this would lead to spurious reflections and poor results. In fact the results typically look quite clean, though only first order accuracy is observed in the pressure, which is continuous but not smooth at the interface. The discontinuous velocities $u$ and $v$ naturally exhibit even greater errors and the discontinuities are smeared over a few grid cells, about what one would expect from a high-resolution flux-limiter method designed to handle discontinuous solutions such as shock waves.

Using CLAWPACK alone is certainly much simpler than incorporating the corrections described here, and for many applications this may give very acceptable results with minimal effort. The methods developed here are designed to deliver second order accuracy at all grid points, even in the discontinuous velocity components. Although the theory and derivation of the modified stencils are somewhat complicated, this needs to be done only once as a preprocessing step before beginning the time integration, and hence may be well worthwhile in applications where better accuracy is required.

6. Numerical results

In this section we present some numerical results to illustrate this method and verify second order accuracy in some simple cases.

Example 6.1. As a first example we consider a plane wave striking an interface at an angle. The wave consists of a single pulse with

$$p(x, y, t) = \begin{cases} \frac{1}{2}A(1 - \cos(2\pi \omega \xi)) & \text{for } 0 \leq \xi \leq 1/\omega, \\ 0 & \text{otherwise}, \end{cases}$$

where

$$\xi = a(x-x_0) + b(y-y_0) - ct$$

with $(a, b)$ a unit vector giving the direction of propagation, $A$ is the amplitude, and $c$ the sound speed. The parameter $\omega$ determines the width of the pulse. The incident plane-wave velocities $u$ and $v$ are then given by

$$u(x, y, t) = \left( \frac{a}{pc} \right) p(x, y, t), \quad v(x, y, t) = \left( \frac{b}{pc} \right) p(x, y, t).$$

After hitting the interface the incident wave is split into a reflected wave and a transmitted wave, and their directions and magnitudes are easily determined from standard theory.

Fig. 3 shows contour plots of the exact pressure at $t = 0$ and $t = 0.45$. Fig. 4 shows the solution at $t = 0.45$ as computed using CLAWPACK alone. Contour plots of both $p$ and $u$ are shown, along with slices of the solutions along $x = 0.35$ in the case of $p$ and along $x = 0.6$ for $u$. The slice of $p$ shows that both the transmitted and reflected wave are well captured. The slice of $p$ is chosen to include both the transmitted and the reflected waves, while the slice of $u$ is chosen to go through the point where the wave is currently interacting with the interface since it is only here that $u$ is discontinuous.

Fig. 5 shows log–log plots of the max-norm errors in the computed solution along these slices on a series of different grids. The maximum error on this slice with no limiter (plotted in o) and with the minmod limiter (plotted in *) are shown. We see that first order accuracy is observed in pressure while the discontinuous velocity has worse errors.

We consider the error only along these slices rather than over the full domain to minimize contamination from the effect of computational boundary conditions, which is not our intent to study here. With CLAWPACK we have simply specified the exact solution along all boundaries, which is stable in spite of the apparent over-specification because of the fact that the wave propagation algorithms used in CLAWPACK are based on solving Riemann problems at
Fig. 3. Plane wave hitting an interface that is not aligned with the grid. Pressure contours of the exact solution at $t = 0$ and $t = 0.45$. The upward moving incident wave is split at the interface into a slower transmitted wave and a weaker reflected wave.

Fig. 4. Plane wave hitting an interface that is not aligned with the grid, as computed by CLAWPACK alone. Top: pressure contours and slice along $x = 0.35$. Bottom: velocity $u$ contours and slice along $x = 0.6$. The solid lines in the slices are the exact solution.
each cell interface and using only the incoming waves, which is tantamount to specifying the correct incoming characteristic variables (see [13]).

Figs. 6 and 7 show analogous results for the case where the immersed interface method developed here is used near the interface. In the plots an improvement is seen primarily in the velocity $u$ near the discontinuity. The observed orders of accuracy with no limiter for $p$ and $u$ are approximately 1.7 and 1.8, respectively. Using the limiter helps to dampen out the oscillation, but it may reduce the order of accuracy.

Table 1 shows three computational costs, $C_{\text{prep}}$, $C_{\text{irr}}$ and $C_{\text{total}}$, for this problem with different grid sizes. All are measured in seconds on a DEC 3000/650 workstation with 32 MB main memory. Here $C_{\text{prep}}$ is the cost for the preprocessing step (the one-time cost of computing the $\Gamma$'s), $C_{\text{irr}}$ is the cost for updating solutions at irregular points near the interface using the special finite difference formula, and $C_{\text{total}}$ is the total cost for the entire computation. We compute the solutions at the same stopping time with different grid sizes.

This table confirms our analysis of the computation cost in Section 4. It shows that both $C_{\text{prep}}$ and $C_{\text{irr}}$ are very small compared with the total cost of the entire computation. When we refine the grid, the computational cost for using the immersed interface method is relatively negligible.

Note that if we multiply $C_{\text{prep}}$ by the number of time steps $n$, which is $O(m)$ as we refine the grid, this will still be very small compared with $C_{\text{total}}$. This means that the relative cost to compute the $\Gamma$'s for each time step, which would be needed in a moving interface problem, will still be small. The low cost should make the immersed interface method practical and efficient also in solving moving interface problems. For inverse problems, we need
Fig. 7. Plane wave hitting an interface that is not aligned with the grid, as computed by CLAWPACK with modified stencils near the interface based on the IIM. Top: pressure contours and slice along $x = 0.35$. Bottom: velocity $u$ contours and slice along $x = 0.6$. The solid lines in the slices are the exact solution.

### Table 1
Computational cost comparison

<table>
<thead>
<tr>
<th>Grid $m$</th>
<th>$C_{\text{prep}}$ (s)</th>
<th>$C_{\text{err}}$ (s)</th>
<th>$C_{\text{total}}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.028</td>
<td>0.14</td>
<td>5.9</td>
</tr>
<tr>
<td>100</td>
<td>0.067</td>
<td>0.60</td>
<td>43.0</td>
</tr>
<tr>
<td>200</td>
<td>0.150</td>
<td>2.30</td>
<td>356.0</td>
</tr>
<tr>
<td>400</td>
<td>0.390</td>
<td>9.70</td>
<td>2482.0</td>
</tr>
</tbody>
</table>

to solve the full forward problem with each guess at the interface, so $C_{\text{prep}}$ again occurs only at the beginning of each forward solve.

**Example 6.2.** As a second example we consider a radially symmetric pressure disturbance moving through a circular interface. The solution remains radially symmetric and hence a highly accurate solution can be computed by solving a one-dimensional problem with appropriate source terms on a very fine grid. This is a good test problem because the interface is curved so that the curvature terms come into the jump conditions. Also the interface cuts through the grid at every possible angle as we move around the circle. The computational results show that the
method works well at all angles and indeed the results are quite isotropic. Contour lines of the computed solution are nearly circular and appear indistinguishable from circles on the finer grid used below. Instead of displaying these plots, we show scatter plots of the solution that are more revealing of the isotropy and accuracy at the interface.

The initial data has \( u = v = 0 \) and

\[
p(x, y, t) = \begin{cases} 
\frac{1}{2}(1 + \cos(15\pi r)) & \text{if } r < 0.2, \\
0 & \text{otherwise},
\end{cases}
\]

where \( r = \sqrt{(x - 0.5)^2 + (y - 0.5)^2} \). This hump moves out radially and is partially reflected at the interface, which we set at \( r = 0.25 \). A one-dimensional simulation on a fine grid produces a reference solution \( P(r, t) \).

The scatter plots in Fig. 8 show the function \( P(r, t) \) at \( t = 0.46 \) plotted as a function of \( r \), as a solid line. The dots in the figures are the values of \( p_{ij} \) from the two-dimensional calculation, plotted against \( r_{ij} = \sqrt{(x_i - 0.5)^2 + (y_j - 0.5)^2} \). Ideally all of these points would fall on the curve, and the degree of scatter shows both the magnitude of the error at different radii, and also the degree of anisotropy in the computed results.

Figs. 8(a) and (b) show results on two different grids in the case where the CLAWPACK algorithms and with no limiters are used away from the interface.

Fig. 9 shows log–log plots of the 1-norm (plotted with \( o \)) and max-norm (plotted with \( * \)) errors in the pressure for a mesh refinement study that includes both of these grids and intermediate grids. The observed orders of accuracy in \( p \) on the finest grids are 2.1 and 1.7, respectively. Note that the velocity is continuous at the interface in this problem since the wave motion is orthogonal to the interface.

**Example 6.3.** As a final example we show a more interesting test case. We consider a plane oscillation striking a half-circular lens which leads to a focusing of the wave. The material parameters are \( \rho = 1 \), \( c = 1 \) in Region 1 and \( \rho = 0.8 \), \( c = 1.25 \) in Region 2. Here Region 2 consists of the set \((x - 0.3)^2 + (y - 0.5)^2 < 0.15\) \( \cup \{0.35 < y < 0.65 \text{ and } x > 0.3\}\) and Region 1 is the exterior of this lens. The solution is forced by an oscillatory solid wall at \( x = 0 \), where we set

\[
u(0, y, t) = 1 - \cos(\omega t)
\]
Fig. 9. Log–log plots of L1-norm (plotted in o) and of max-norm (plotted in *) errors in $p$ for Example 6.2.

Fig. 10. Oscillatory plane wave hitting a lens-like region inside of which the sound speed is larger. Shown on the left is the contour plot of the pressure and on the right is the mesh plot (on a coarser grid than the computation).

with $\omega = 11.17$. Figs. 10 (left) and 11 (left) show contours of $p$ and $u$ at $t = 2$, after a steady oscillatory solution has been reached. This solution was calculated on a $400 \times 400$ grid. Fig. 10 (right) shows a mesh plot of the pressure $p$. We can clearly see the focusing of the wave. Fig. 11 (right) shows a cross-section of $u$ along $x = 0.5$ (the solid line is for the same computed solution). Note that $u$ is discontinuous at the interfaces but smooth on either side, as expected. This is also visible on the contour plot of $u$. 
7. Conclusions

We have presented a technique for deriving accurate finite difference methods at uniform grid points near an interface in material properties. These special formulas have been coupled together with a high-resolution multidimensional algorithm for solving the acoustic equations in regions away from the interface. Test calculations have confirmed that the special finite difference formulas are second order accurate.

These same ideas should apply to other linear systems of hyperbolic equations, as has already been confirmed by the second author in preliminary calculations for the equations of elasticity [37]. In principle it would also be possible to derive higher order accurate interface methods for use with more accurate solvers.

Appendix A. Derivation of interface conditions

We first derive the one-dimensional interface conditions (28). The jump conditions (24) gives

\[ [U] = 0 \implies U^+ = D_1 U^- \tag{A.1} \]

From (23) we get

\[ [u] = 0 \implies [u_r] = 0 \implies \left[ \frac{1}{\rho} \rho_x \right] = 0, \]

\[ [p] = 0 \implies [p_t] = 0 \implies [K u_x] = 0, \]

or,

\[ U^+_x = D_2 U^-_x \tag{A.2} \]

We can deduce \( U_{tt} = c^2 U_{xx} \) from (23), and so
\[ [U] = 0 \implies [U_{tt}] = 0 \implies [c^2U_{xx}] = 0 \implies U_{xx}^+ = D_3 U_{xx}^- . \quad (A.3) \]

The combination of (A.1)–(A.3) gives (28), with \( D_1, \ D_2 \) and \( D_3 \) given in (29).

Now we come to two-dimensional jump conditions, starting with the case of an interface that is a straight line. For a more general interface, curvature terms arise as discussed below. Given the interface conditions (49) for the scalar variable \( \bar{u}, \ \bar{v}, \ \text{and} \ \rho \), we can get the corresponding interface conditions for the vector variable \( \bar{U} \) (or equivalently, \( U \)). We get interface jump conditions for \( \bar{U} \) immediately from Eq. (49) and find that

\[ \bar{U}^+ = Q_1 \bar{U}^- \]

with \( Q_1 \) as in (51). Dropping the bar notation for simplicity, noting that the definition and derivation are based on the local coordinate system \( \xi - \eta \), we have

\[ U^+ = Q_1 U^- . \quad (A.4) \]

Differentiating this twice with respect to \( \eta \) gives

\[ U_{\eta}^+ = Q_1 U_{\eta}^- , \quad U_{\eta\eta}^+ = Q_1 U_{\eta\eta}^- . \quad (A.5) \]

Next, we will derive the jump conditions for \( U_{\xi}^+ , \ U_{\xi\eta}^+ \) and \( U_{\xi\xi}^+ \). We will also drop the bar notation in \( \bar{u} \) and \( \bar{v} \) for simplicity. The first equation of (1) gives

\[ [u_{\xi}] = 0 \implies \left[ \frac{1}{\rho^+} p_{\xi}^{+} \right] = 0 \implies p_{\xi}^{+} = \frac{\rho^+}{\rho^{-}} p_{\xi}^{-} . \quad (A.6) \]

Combining Eqs. (38a) and (38b) we get

\[ u_{\eta\eta} - v_{\xi} = 0 \implies (u_{\eta} - v_{\xi})_{\xi} = 0 \]

and therefore

\[ u_{\eta} - v_{\xi} \equiv 0 \]

is true provided that it is true of the initial data. Physically, this means that for acoustic waves, the deformation is irrotational, as is generally assumed. Then we have

\[ [v_{\xi}] = [u_{\eta}] \equiv 0 \implies v_{\xi}^{+} = v_{\xi}^{-} . \quad (A.7) \]

From (38c) it follows that

\[ [K u_{\xi}] = -[K v_{\eta}] \implies u_{\xi}^{+} = \frac{K^-}{K^+} u_{\xi}^{-} - v_{\eta}^{+} + \frac{K^-}{K^+} v_{\eta}^{-} . \quad (A.8) \]

Expressing \( v_{\eta}^{+} \) in terms \( v_{\eta}^{-} \) gives

\[ [\rho v] = 0 \implies [\rho v_{\eta}] = 0 \implies v_{\eta}^{+} = \frac{\rho^{-}}{\rho^{+}} v_{\eta}^{-} . \quad (A.9) \]

Then we have

\[ u_{\xi}^{+} = \frac{K^-}{K^+} u_{\xi}^{-} - \frac{\rho^{-}}{\rho^{+}} v_{\eta}^{-} + \frac{K^-}{K^+} v_{\eta}^{-} = \frac{K^-}{K^+} u_{\xi}^{-} + \frac{\rho^{-}}{\rho^{+}} \left( \frac{c^-}{c^+} \right)^2 v_{\eta}^{-} . \quad (A.10) \]

Combining (A.6), (A.7), and (A.10) we obtain an expression for \( U_{\xi}^{+} \) in terms of \( U_{\xi}^{-} \) and \( U_{\eta}^{-} \):

\[ U_{\xi}^{+} = Q_2 U_{\xi}^{-} + Q_3 U_{\eta}^{-} . \quad (A.11) \]
It follows by differentiating this with respect to \( \eta \) that
\[
U^+_{\xi \eta} = Q_2 U^-_{\xi \eta} + Q_3 U^-_{\eta \eta}. \tag{A.12}
\]

We can obtain a system of decoupled second order equations from (1):
\[
u_{tt} = c^2 (u_{\xi \xi} + u_{\eta \eta}), \quad v_{tt} = c^2 (v_{\xi \xi} + v_{\eta \eta}), \quad p_{tt} = c^2 (p_{\xi \xi} + p_{\eta \eta}). \tag{A.13}
\]

Since \([u] = [\rho v] = [p] = 0\), we have
\[
[c^2(u_{\xi \xi} + u_{\eta \eta})] = 0, \quad [K(v_{\xi \xi} + v_{\eta \eta})] = 0, \quad [c^2(p_{\xi \xi} + p_{\eta \eta})] = 0. \tag{A.14}
\]

With \([u_{\eta \eta}] = [\rho v_{\eta \eta}] = [p_{\eta \eta}] = 0\), and noting that \(K = \rho c^2\), we get
\[
u_{\xi \xi}^+ = \left( \frac{c^-}{c^+} \right)^2 u_{\xi \xi}^- + \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) u_{\eta \eta}^-,
\]
\[
v_{\xi \xi}^- = \frac{K^-}{K^+} v_{\xi \xi}^- + \rho^+ \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) v_{\eta \eta}^-,
\]
\[
p_{\xi \xi}^- = \left( \frac{c^-}{c^+} \right)^2 p_{\xi \xi}^- + \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) p_{\eta \eta}^-,
\]
which is
\[
U^+_{\xi \xi} = \left( \frac{c^-}{c^+} \right)^2 Q_1 U^-_{\xi \xi} + \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) Q_1 U^-_{\eta \eta}. \tag{A.15}
\]

The jump conditions (A.4), (A.5), (A.11), (A.12) and (A.16) form (50), with all the \(Q\)'s being those defined in (51). We next consider the case of a curved interface, expressed locally by \(\xi = \chi(\eta)\). The jump conditions are given by (49), which in matrix form will be
\[
[SU] = 0, \tag{A.17}
\]
where \(S = \text{diag}(1, \rho, 1)\), or equivalently, as we have in (A.5)
\[
U^+ = Q_1 U^- \tag{A.18}
\]

The jump conditions for \(U_{\xi} \) and \(U_{\eta} \) are the same as in the case of straight line interface. Moreover the previous derivations of \(U_{\xi} \) and \(U_{\eta} \) are still valid here and so
\[
U^+_{\eta} = Q_1 U^-_{\eta}, \quad U^+_{\xi} = Q_2 U^-_{\xi} + Q_3 U^-_{\eta}. \tag{A.19}
\]

Differentiating (A.17) with respect to \(\eta\), we get
\[
[S(U_{\xi} \chi' + U_{\eta})] = 0,
\]
from which (A.19) follows naturally by letting \(\eta = 0\) and using \(\chi'(0) = 0\). One more differentiation yields
\[
[S(U_{\xi \xi}(\chi')^2 + 2U_{\xi \eta} \chi' + U_{\xi} \chi'' + U_{\eta}')] = 0.
\]

At \(\eta = 0\) we have
\[
[S(U_{\xi} \chi''(0) + U_{\eta})] = 0,
\]
which gives us

\[ U_{\eta\eta}^+ = Q_1 U_{\eta\eta}^- - \chi''(0)((Q_2 - Q_1)U_{\xi}^- + Q_3 U_{\eta}^-). \]  

(A.20)

As for \( U_{\xi\xi}^+ \), we start with (A.13) and (A.14) from which we can deduce

\[
\begin{align*}
 u_{\xi\xi}^+ &= \left( \frac{c^-}{c^+} \right)^2 (u_{\xi\xi}^- + u_{\eta\eta}^-) - u_{\eta\eta}^+ , \\
v_{\xi\xi}^+ &= \frac{K^-}{K^+} (v_{\xi\xi}^- + v_{\eta\eta}^-) - v_{\eta\eta}^+ , \\
p_{\xi\xi}^+ &= \left( \frac{c^-}{c^+} \right)^2 (p_{\xi\xi}^- + p_{\eta\eta}^-) - p_{\eta\eta}^+ ,
\end{align*}
\]

i.e.,

\[ U_{\xi\xi}^+ = \left( \frac{c^-}{c^+} \right)^2 Q_1 (U_{\xi\xi}^- + U_{\eta\eta}^-) - U_{\eta\eta}^+. \]

But now we must use (A.20) instead of the simpler formula \( U_{\eta\eta}^+ = Q_1 U_{\eta\eta}^- \) that holds for an interface that is a straight line, and so (A.16) becomes

\[ U_{\xi\xi}^+ = \left( \frac{c^-}{c^+} \right)^2 Q_1 U_{\xi\xi}^- + \left( \left( \frac{c^-}{c^+} \right)^2 - 1 \right) Q_1 U_{\eta\eta}^- + \chi''(0)((Q_2 - Q_1)U_{\xi}^- + Q_3 U_{\eta}^-) \]  

(A.21)

in the general case where \( \chi'' \neq 0 \). Now we come to \( U_{\xi\eta}^+ \). From \( u_{\eta} - v_{\xi} = 0 \), we have

\[ (u_{\eta} - v_{\xi})_\eta = 0 \implies v_{\xi\eta}^+ = v_{\xi\eta}^- + u_{\eta\eta}^+ - u_{\eta\eta}^- . \]  

(A.22)

From (38c), using \([p_{\eta\eta}] = 0\), we get

\[ p_{\xi\eta} + \rho c^2 (u_{\xi\eta} + v_{\eta\eta}) = 0 \implies \quad u_{\xi\eta}^+ = \left( \frac{\rho^+}{\rho^-} \right) \left( \frac{c^-}{c^+} \right)^2 (u_{\xi\eta}^- + v_{\eta\eta}^-) - v_{\eta\eta}^+. \]  

(A.23)

Also from (38a) we have

\[ u_{\xi\eta} + \frac{1}{\rho} p_{\xi\eta} = 0 \implies \quad p_{\xi\eta}^+ = \frac{\rho^+}{\rho^-} p_{\xi\eta}^- . \]  

(A.24)

Eqs. (A.22)–(A.24) give us

\[ U_{\xi\eta}^+ = Q_2 U_{\xi\eta}^- + Q_3 U_{\eta\eta}^- + \chi''(0) Q_4 ((Q_2 - Q_1)U_{\xi}^- + Q_3 U_{\eta}^-) . \]  

(A.25)

Combining (A.18)–(A.21) and (A.25), we get (50).

Acknowledgements

We are grateful to Kenneth Bube for numerous suggestions and discussions concerning acoustics, elastic wave equations, and jump conditions. This work was supported in part by NSF grants DMS-9204329 and DMS-9303404, DOE grant DE-FG06-93ER25181, and the Scientific Computing Division at NCAR.
References


