Seismic modeling with radial basis function-generated finite differences (RBF-FD)

(Seismic modeling with RBF-FD)

Presented as a poster at the 2014 SEG Annual Meeting in Denver, Colorado

Authors:

Bradley Martin (University of Colorado-Boulder) Email: brma7253@colorado.edu

Bengt Fornberg (University of Colorado-Boulder) Email: Fornberg@colorado.edu

Amik St-Cyr (Shell International Exploration and Production) Email: Amik.St-Cyr@shell.com

ABSTRACT

Seismic exploration is the primary tool for finding and mapping out hydrocarbon deposits. We consider here the 2-D forward modeling problem. The subsurface structures are assumed to be known, and the task is to simulate elastic wave propagation throughout the medium. RBF-FD (radial basis function-generated finite difference) spatial discretizations have previously been shown to offer high accuracy and algebraic simplicity also when using scattered layouts of computational nodes. We have now developed this method further, to provide third-order accuracy not only throughout smooth regions, but also for wave reflections and transmissions at arbitrarily curved material interfaces. The key step is to supplement the RBFs which underlie the RBF-FD approximation with polynomials, and then customize the latter to incorporate the interface requirements. The method is illustrated on a couple of test problems for the 2-D isotropic elastic wave equation.

INTRODUCTION

The two dominant error sources in solving elastic wave equations in domains featuring discontinuous material parameters are

- Dispersive errors in regions with constant or smoothly variable material properties
- Reflection/transmission errors at interfaces

The former error type causes traveling wave fronts to develop spurious trailing wave trains, which are best overcome by using approximations of high orders of accuracy (Fornberg, 1987).

Many simulations have been carried out under the assumption that the use of a high-order FD scheme sufficiently reduces the latter error as well, and that point-to-point changes in material parameter values provide enough information to reach a satisfactory solution (Robertsson, et al., 2012). However, without more rigorous treatment near interfaces, the application of high-order

methods to such a domain yields a solution accurate to only first order (Symes, et al., 2009), (Vishnevsky, et al., 2014).

Some investigators have improved the second type of error (differentiation across interfaces) by employing curvilinear transformations such that grid lines follow interfaces (Fornberg, 1988). Moving from structured to unstructured meshes, as used for example with finite element methods (FEM), improves geometric flexibility further still. We focus here on an altogether mesh-free discretization, placing scattered nodes suitably with respect to interfaces, but without forming any triangles or tetrahedra, etc. A web search on 'Meshfree methods' will reveal a long list of attempted variations on this theme. However, the recent literature has increasingly focused on RBF based approximations (Fasshauer, 2007). Still more recently, RBF-FD has emerged as a particularly good way to obtain highly accurate 'local' derivative approximations using scattered node sets. The first RBF-FD application in the context of seismic modeling was given in (Martin, et al., 2013).

The present RBF-FD approach shares certain concepts with immersed interface methods, described for ex. in (Zhang, et al., 1997). In including appropriate basis functions directly into stencils that cross the interface, it also shares some features with the FEM approaches in (Li, 1998) and (Zhebel, et al., 2014). As with the finite difference approach in (Lombard, et al., 2004), the present approach maintains flexibility in the choice of spatial discretization and time integration techniques, while presenting a framework that allows relatively simple preprocessing routines, and avoids the need to construct smooth (or modified) structures of data through interfaces.

RBF-FD METHODOLOGY

This section introduces the present method in a simple 2-D elastic wave equation case. Figure 1 shows the unit square divided into two regions by a mildly curved interface. Across the interface, the density ρ and the Lamé parameters λ and μ of the medium can be discontinuous. The figure also illustrates a node distribution that follows the interface but transitions to a regular Cartesian lattice for standard FD usage a short distance away from it. The three cases we need to handle are:

- *Case* 1: Stencils that are located sufficiently far away from the interface that they only contain Cartesian grid type nodes
- *Case* 2: Stencils that are not intersected by the interface, but with some or all nodes deviating from perfect lattice arrangement
- *Case* 3: Stencils that are intersected by the interface.

In *Case* 1, we use standard FD weights and integration techniques to model wave propagation. In *Case* 2, we instead obtain stencil weights by the RBF-FD formalism, permitting arbitrarily scattered nodes. Finally, in *Case* 3, we additionally invoke continuity conditions (on velocity and traction) at the interface to create specially modified RBF-FD weights that preserve high-order accuracy for waves interacting with the interface.

Execution of a code based on these *Cases* 1-3 stencils becomes highly efficient (especially on GPU-type hardware) since all relevant information about the interfaces (their locations and transmission / reflection properties) become embedded into the node set and into the stencil weights already during the pre-processing stage. During the time stepping, all stencils are then applied in exactly the same way, whether interfaces are present or absent.

We discuss next the Cases 2 and 3 in more detail.

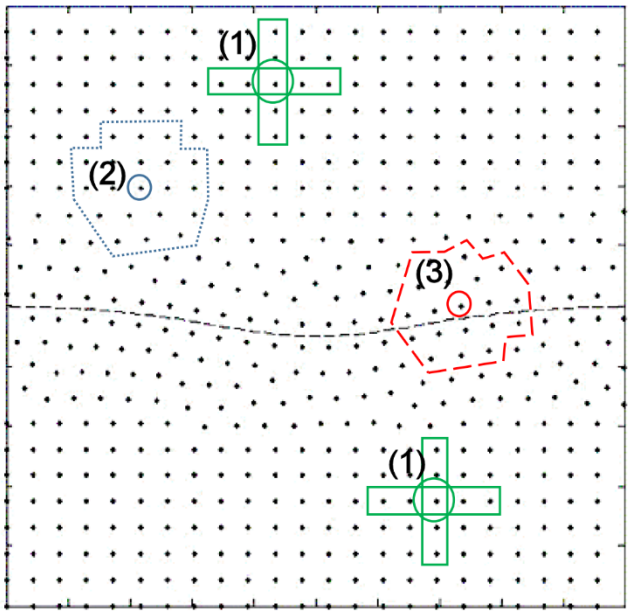


Figure 1. Schematic node layout in the vicinity of a curved interface. Colored regions show *Case* 1 (solid green lines), *Case* 2 (dotted blue lines) and *Case* 3 stencils (dashed red lines). The stencils approximate the PDE at the circled nodes.

Case 2: RBF-FD stencils in regions with smoothly variable medium properties, but without completely Cartesian node structure

Standard FD approximations in 1-D use weights that are chosen such that the resulting approximation becomes exact for monomials (1, x, x^2, etc.) of as high degree as possible. FD approximations for higher-D are then typically tensor-like composites of 1-D approximations. Such approximations can become very accurate when they are applied to data that is locally very well-represented by a Taylor expansion. This approach fails however for scattered nodes in 2-D (and higher), since the linear systems that need to be solved to find the FD weights then frequently become very ill-conditioned, or outright singular.

In the case of RBF-based FD methods, polynomials are replaced by RBFs, and there is no longer any need to consider tensor type extensions from 1-D. Data near a stencil center $\mathbf{x}_c$ is instead

directly represented by a superposition of radially symmetric functions $\phi(\|\mathbf{x}-\mathbf{x}_i\|), i=1,2,\ldots,n$ that are centered at the stencil node locations $\mathbf{x}_i$, i.e.:

$$f(\mathbf{x}) \approx \sum_{i=1}^{n} a_i \phi(\|\mathbf{x}-\mathbf{x}_i\|) \tag{1}$$

If node data $f(\mathbf{x}_i)$ is given, we can determine the expansion coefficients a_i by solving the linear system:

$$\begin{bmatrix} \phi(\|\mathbf{x}_1-\mathbf{x}_1\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_1\|) \\ \phi(\|\mathbf{x}_1-\mathbf{x}_2\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_2\|) \\ \vdots & \ddots & \ddots & \vdots \\ \phi(\|\mathbf{x}_1-\mathbf{x}_n\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_n\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_n\|) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \tag{2}$$

In the RBF-FD approach, the approximation of an operator L (such as $\partial/\partial x$, etc.) at the central node $\mathbf{x}_c$ is obtained as a weighted sum of function values at the n stencil nodes; $Lf(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_c} \approx \sum_{i=1}^{n} w_i f(\mathbf{x}_i)$, but with the weights now required to give the exact result not for a polynomial interpolant, but when applied to RBF interpolant (1). The system we solve for each RBF-FD stencil therefore becomes:

$$\begin{bmatrix} \phi(\|\mathbf{x}_1-\mathbf{x}_1\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_1\|) \\ \phi(\|\mathbf{x}_1-\mathbf{x}_2\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_2\|) \\ \vdots & \ddots & \ddots & \vdots \\ \phi(\|\mathbf{x}_1-\mathbf{x}_n\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_n\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_n\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} L\phi(\|\mathbf{x}-\mathbf{x}_1\|)|_{\mathbf{x}=\mathbf{x}_c} \\ L\phi(\|\mathbf{x}-\mathbf{x}_2\|)|_{\mathbf{x}=\mathbf{x}_c} \\ \vdots \\ L\phi(\|\mathbf{x}-\mathbf{x}_n\|)|_{\mathbf{x}=\mathbf{x}_c} \end{bmatrix} \tag{3}$$

or, in abbreviated notation:

$$A\mathbf{w} = L\boldsymbol{\phi}. \tag{4}$$

Here A is the same (n,n) symmetric matrix as in (2), $\mathbf{w}$ is the column vector of RBF-FD weights to be applied at the nodes of the stencil, L is the linear operator we are approximating, and $\boldsymbol{\phi}$ is the column vector of the operator L applied to the different RBFs and then evaluated at $\mathbf{x}_c$.

Remarkably, many choices of radial functions $\phi(r)$ lead to guaranteed non-singularity of these *A*-matrices, no matter how any number of distinct nodes are scattered in any number of dimensions (Schoenberg, 1938). In the present work, we have used the inverse multiquadric (IMQ) radial function $\phi(r) = 1/\sqrt{1+(\varepsilon r)^2}$. If the nodes were located on a lattice, letting $\varepsilon \to 0$ will usually (but not always) recover the standard FD approximation for the same stencil size (Driscoll, et al., 2002), (Wright, et al., 2006). However, small ε will cause the linear systems (3), (4) (as well as (5), (6), described below) to become highly ill-conditioned. A very simple, yet effective strategy in the present context, is to choose ε so that the coefficient matrices have condition numbers in the 10^6 to 10^8-range. This RBF-FD discretization approach has proven highly successful for PDEs in nontrivial geometric settings (Shan, et al., 2008). Following the recent introduction of 'hyperviscosity' (Fornberg, et al., 2010), relatively large stencils (often in the n = 30-70 range in 2-D, giving high accuracies) can be used together with explicit time stepping also for purely convective (wave-type) PDEs (Flyer, et al., 2011), (Flyer, et al., 2014).

As mentioned in the introduction, a key feature of RBF-FD is that one can include additional functions (typically polynomials) with the RBF basis underlying each stencil, and then add matching constraints, as surveyed for ex. in (Fasshauer, 2007). This will be especially crucial in *Case* 3, where we add support of piecewise polynomials in order to represent the non-smooth solutions across material interfaces. For example, when including up to linear 2-D terms (1, *x*, *y*) to the counterparts of (3) and (4), the corresponding system of equations becomes (Larsson, et al., 2013), (Fornberg, et al., 2015):

$$\begin{bmatrix} \phi(\|\mathbf{x}_1-\mathbf{x}_1\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_1\|) & 1|_{\mathbf{x}_1} & x|_{\mathbf{x}_1} & y|_{\mathbf{x}_1} \\ \phi(\|\mathbf{x}_1-\mathbf{x}_2\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_2\|) & 1|_{\mathbf{x}_2} & x|_{\mathbf{x}_2} & y|_{\mathbf{x}_2} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \phi(\|\mathbf{x}_1-\mathbf{x}_n\|) & \phi(\|\mathbf{x}_2-\mathbf{x}_n\|) & \cdots & \phi(\|\mathbf{x}_n-\mathbf{x}_n\|) & 1|_{\mathbf{x}_n} & x|_{\mathbf{x}_n} & y|_{\mathbf{x}_n} \\ 1|_{\mathbf{x}_1} & 1|_{\mathbf{x}_2} & \cdots & 1|_{\mathbf{x}_n} & 0 & 0 & 0 \\ x|_{\mathbf{x}_1} & x|_{\mathbf{x}_2} & \cdots & x|_{\mathbf{x}_n} & 0 & 0 & 0 \\ y|_{\mathbf{x}_1} & y|_{\mathbf{x}_2} & \cdots & y|_{\mathbf{x}_n} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_1^* \\ w_x^* \\ w_y^* \end{bmatrix} = \begin{bmatrix} L\phi(\|\mathbf{x}-\mathbf{x}_1\|)|_{\mathbf{x}_c} \\ L\phi(\|\mathbf{x}-\mathbf{x}_2\|)|_{\mathbf{x}_c} \\ \vdots \\ L\phi(\|\mathbf{x}-\mathbf{x}_n\|)|_{\mathbf{x}_c} \\ L(1)|_{\mathbf{x}_c} \\ L(x)|_{\mathbf{x}_c} \\ L(y)|_{\mathbf{x}_c} \end{bmatrix} \quad (5)$$

or, in abbreviated notation:

$$\begin{bmatrix} A & P^T \\ P & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{w}^* \end{bmatrix} = \begin{bmatrix} L\phi \\ LP \end{bmatrix} \quad (6)$$

Here, P is a (3,n) matrix of the Taylor monomials (1, x, y) at the stencil node points, and LP is the (3,1) vector of evaluations of the linear operator L applied to each of these three monomials and then evaluated at the location $\mathbf{x}_c$ (where we want our approximation to be accurate). The column vector $\mathbf{w}$ contains the weights of the resulting stencil, while the vector $\mathbf{w}^*$ should be discarded. Practical experience suggests the use of polynomials of such degrees that the number of rows in P become around half the number of rows in A (or slightly less). *Case* 2 stencils in the test problems below have used $n = 19$ nodes with polynomial support up to and including third degree (cubic) terms.

Case 3: RBF-FD stencils across interfaces

Although the RBF-FD approach easily allows nodes to be placed optimally around interfaces, it is still crucial to combine this with a further strategy that allows for accurate differentiation of the PDE's dependent variables across that interface. Traditional FD weights are usually determined under the assumption that the data to be differentiated is smooth. As noted above, many current FD seismic simulation routines then apply these weights also to data that (physically correctly) lack smoothness across an interface (Robertsson, et al., 2012), (Vishnevsky, et al., 2014).

When data is not smooth across an interface, we can use our knowledge of interface continuity conditions together with the governing PDE to create a customized set of piecewise polynomial basis functions to accurately represent also this data. These piecewise polynomials (rather than regular ones) are then added to the set of RBF basis functions when determining weights within RBF-FD stencils that cross the interface. When using only linear polynomials, this concept was tested for the 2-D Maxwell's equations in (Yu, et al., 2011).

In the present 2-D isotropic elastic wave equation case, the governing equations are:

$$\begin{cases} \rho u_t = f_x + g_y \\ \rho v_t = g_x + h_y \\ f_t = (\lambda + 2\mu) u_x + \lambda v_y \\ g_t = \mu(u_x + v_y) \\ h_t = (\lambda + 2\mu) v_y + \lambda u_x \end{cases} . \tag{7}$$

Here, u, v are local medium velocities in the x and y directions, respectively, f, g, h are the elements of the 2-D (symmetric) stress tensor, ρ denotes density, and the Lamé parameters λ, μ describe the material properties with regard to pressure and shear. At the interface, velocity and traction (stress in the direction of the interface normal) must be continuous.

Determining Case 3 polynomial basis functions: 1-D simplification

For the purpose of illustration, we consider first the 1-D case, for which these equations reduce to the well-known 2-way wave equation:

$$\begin{bmatrix} u_t \\ f_t \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{1}{\rho}\right)\frac{\partial}{\partial x} \\ \rho c_p^2 \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} u \\ f \end{bmatrix} \triangleq D \begin{bmatrix} u \\ f \end{bmatrix} \tag{8}$$

In (8), c_p^2 is the speed of pressure (primary) waves within a medium. Higher-order time derivatives and their equivalences in terms of spatial derivatives can be obtained by applying the differential operator D multiple times:

$$\begin{bmatrix} u_{(k)t} \\ f_{(k)t} \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{1}{\rho}\right)\frac{\partial}{\partial x} \\ \rho c_p^2 \frac{\partial}{\partial x} & 0 \end{bmatrix}^k \begin{bmatrix} u \\ f \end{bmatrix} \triangleq D^k \begin{bmatrix} u \\ f \end{bmatrix} \tag{9}$$

Continuity of velocity and traction are simple in 1-D: both u and f must be continuous across an interface. Therefore, $u_{(k)t}$, $f_{(k)t}$ are also continuous there for $k = 1, 2, 3, \ldots$. Suppose for simplicity that the interface in the present 1-D example is located at $x = 0$. If we refer to the left and right sides of the interface as sides L and R, respectively, then for $k = 0, 1, 2, \ldots$:

$$D_L^k \begin{bmatrix} u_L \\ f_L \end{bmatrix} - D_R^k \begin{bmatrix} u_R \\ f_R \end{bmatrix} = \begin{bmatrix} D_L^k, -D_R^k \end{bmatrix} \begin{bmatrix} u_L \\ f_L \\ u_R \\ f_R \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{10}$$

Here, the subscripted operators and data fields are those entities evaluated as we approach the interface from the left and from the right, respectively. 1-D FD stencils that uphold these truths can be constructed as described below.

Consider the polynomials in a fourth-order expansion of both data fields about the interface:

$$\mathbf{u} \approx \begin{cases} \mathbf{u}_L = u_{L,1}1 + u_{L,x}x + u_{L,x^2}x^2 + u_{L,x^3}x^3 + u_{L,x^4}x^4 & if\ x \le 0 \\ \mathbf{u}_R = u_{R,1}1 + u_{R,x}x + u_{R,x^2}x^2 + u_{R,x^3}x^3 + u_{R,x^4}x^4 & if\ x > 0 \end{cases} \tag{11}$$

$$\mathbf{f} \approx \begin{cases} \mathbf{f}_L = f_{L,1}1 + f_{L,x}x + f_{L,x^2}x^2 + f_{L,x^3}x^3 + f_{L,x^4}x^4 & if\ x \le 0 \\ \mathbf{f}_R = f_{R,1}1 + f_{R,x}x + f_{R,x^2}x^2 + f_{R,x^3}x^3 + f_{R,x^4}x^4 & if\ x > 0 \end{cases} \tag{12}$$

In (11) and (12), we have switched to vector notation to indicate that we are now considering vectors of polynomial coefficients in $\mathbb{P}_4(\mathbb{R})$. For example, $\mathbf{u}_L$ is composed of the scalar $u_{L,1}, u_{L,x}, u_{L,x^2}$, etc.:

$$\mathbf{u}_L = \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} \tag{13}$$

In 1-D, we are only considering Taylor monomials in our FD basis, so we can define the PDE differential operators D_L and D_R entirely by their action on these monomial basis functions:

$$\begin{bmatrix} \mathbf{u}_{L,(k)t} \\ \mathbf{f}_{L,(k)t} \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{1}{\rho_L}\right)\frac{\partial}{\partial x} \\ \rho_L c_{p,L}^2 \frac{\partial}{\partial x} & 0 \end{bmatrix}^k \begin{bmatrix} \mathbf{u}_L \\ \mathbf{f}_L \end{bmatrix} \triangleq D_L{}^k \begin{bmatrix} \mathbf{u}_L \\ \mathbf{f}_L \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} \mathbf{u}_{R,(k)t} \\ \mathbf{f}_{R,(k)t} \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{1}{\rho_R}\right)\frac{\partial}{\partial x} \\ \rho_R c_{p,R}^2 \frac{\partial}{\partial x} & 0 \end{bmatrix}^k \begin{bmatrix} \mathbf{u}_R \\ \mathbf{f}_R \end{bmatrix} \triangleq D_R{}^k \begin{bmatrix} \mathbf{u}_R \\ \mathbf{f}_R \end{bmatrix} \tag{15}$$

Here, the discrete differential block operator $\partial / \partial x$ is also defined by its action on the standard monomial basis set in $\mathbb{P}_4(\mathbb{R})$:

$$\frac{\partial}{\partial x}\begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} \tag{16}$$

We can now return to (10) and use powers of the differential operator D to turn continuity of time derivatives at the interface into relationships between data expansion coefficients on either side of the interface:

$$
\begin{bmatrix} \\ \begin{bmatrix} D_L^0, & -D_R^0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_L \\ \mathbf{f}_L \\ \mathbf{u}_R \\ \mathbf{f}_R \end{bmatrix}_{x=0} \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \rightarrow \quad \begin{cases} u_{1,1} = u_{R,1} \\ f_{1,1} = f_{R,1} \end{cases}
$$

$$
\begin{bmatrix} \\ \begin{bmatrix} D_L^1, & -D_R^1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_L \\ \mathbf{f}_L \\ \mathbf{u}_R \\ \mathbf{f}_R \end{bmatrix}_{x=0} \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \rightarrow \quad \begin{cases} \left(\frac{1}{\rho_L}\right) f_{L,x} = \left(\frac{1}{\rho_R}\right) f_{R,x} \\ \rho_L c_L^2 u_{L,x} = \rho_R c_R^2 u_{R,x} \end{cases} \tag{17}
$$

$$
\begin{bmatrix} \\ \begin{bmatrix} D_L^2, & -D_R^2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_L \\ \mathbf{f}_L \\ \mathbf{u}_R \\ \mathbf{f}_R \end{bmatrix}_{x=0} \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \rightarrow \quad \begin{cases} c_L^2 u_{L,x^2} = c_R^2 u_{R,x^2} \\ c_L^2 f_{L,x^2} = c_R^2 f_{R,x^2} \end{cases}
$$

$$
\vdots
$$

In (17), the equivalences between expansion coefficients on the right side of each equation were obtained by examining the rows of each left-hand-side matrix that reference constant expansion terms. Constants are the only monomials that evaluate to a nonzero value at the interface location ($x = 0$ in our chosen coordinate system). Therefore, any combination of data expansion coefficients that end up in a row for constant monomials after a given power of D is applied in (17) must all add up to zero. In 1-D, this process of generating non-smooth basis functions is simple and decoupled; we can determine the relationships between expansion coefficients one by one, just by looking at individual constant-term columns of the left-hand-sides above. In 2-D or 3-D, we will need to collect "null rows" together into a big continuity matrix C. After this is done, we can find a basis set of coupled monomial coefficients that satisfy all the truths at once: we just need to find a basis for $(col(C))^{\perp}$, or $null(C^T)$. When applied in a 1-D case, this more general procedure reduces to (17).

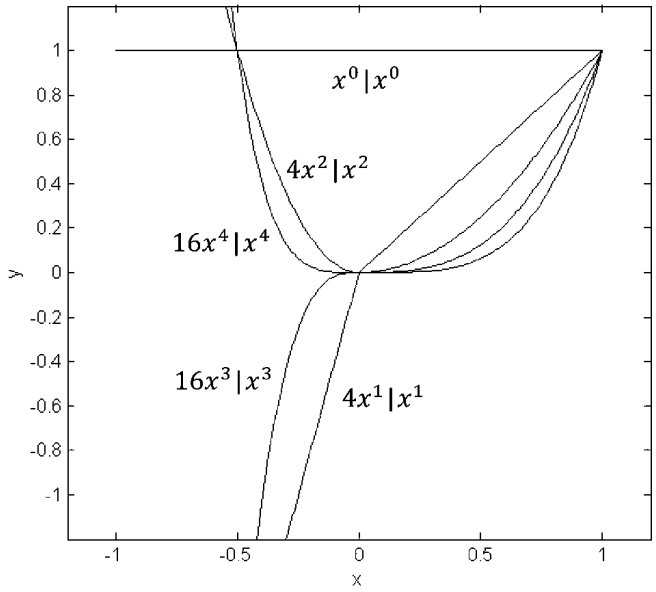


Figure 2. Illustration of the non-smooth basis underlying differential stencils for data field *u* used to solve the present 1-D test problem. For the *f*-variable, the couplings become slightly different: $\{x^0|x^0, x^1|x^1, 4x^2|x^2, 4x^3|x^3, 16x^4|x^4\}$.

The approach described above was used on a particular test problem in [-1,1] to generate the basis for data field *u* shown in Figure 2. This basis can be used for stencils that cross the interface and apply to data field *u*. In this test problem, $c_p = 1$ for $x < 0$ and $c_p = 2$ for $x \geq 0$, and $\rho = 1$ throughout the domain.

Determining Case 3 non-smooth polynomial basis functions: 2-D example

An appropriate set of non-smooth, piecewise polynomial basis functions can easily be incorporated into FD schemes in 1-D. The situation in 2-D is somewhat more involved. For example, if we then want to enforce interface conditions to 2nd order accuracy, an analogous process to that above (examining how powers of the differential operator relate spatial derivatives of motion and traction) produces an explicit 12-dimensional space of 2-D coupled basis polynomials for *u* and *v*,

and a 24-dimensional space of basis polynomials for *f*, *g* and *h*. Adding this type of support to a 2-D Cartesian stencil would not only be difficult to achieve, but likely also be highly unstable. However, the method through which explicit polynomial support is added to RBF-FD stencils (as shown in equations (5) and (6)) can readily be generalized to handle it.

In 2-D, we begin by evaluating weights for a stencil that crosses the interface the same way as we did in 1-D. For illustration, assume that we wish to add polynomial support up to and including second degree terms to an RBF-FD stencil crossing an interface. We pick a point on the interface that lies near the point where the stencil approximates a linear operator. As in the 1-D example, we designate the nearby point on the interface as the origin for purposes of evaluating support polynomials, and we initially assume that all expansion coefficients of all data fields may vary from one side of the interface to another.

We have the following data expansions for which we want to determine coefficients:

$$\begin{cases} \mathbf{u} \approx \begin{cases} \mathbf{u}_L = u_{L,1}1 + u_{L,x}x + u_{L,y}y + u_{L,x^2}x^2 + u_{L,xy}xy + u_{L,y^2}y^2 & \text{if } y < c(x) \\ \mathbf{u}_R = u_{R,1}1 + u_{R,x}x + u_{R,y}y + u_{R,x^2}x^2 + u_{R,xy}xy + u_{R,y^2}y^2 & \text{otherwise} \end{cases} \\ \mathbf{v} \approx \begin{cases} \mathbf{v}_L = v_{L,1}1 + v_{L,x}x + v_{L,y}y + v_{L,x^2}x^2 + v_{L,xy}xy + v_{L,y^2}y^2 & \text{if } y < c(x) \\ \mathbf{v}_R = v_{R,1}1 + v_{R,x}x + v_{R,y}y + v_{R,x^2}x^2 + v_{R,xy}xy + v_{R,y^2}y^2 & \text{otherwise} \end{cases} \\ \vdots \\ \mathbf{h} \approx \begin{cases} \mathbf{h}_L = h_{L,1}1 + h_{L,x}x + h_{L,y}y + h_{L,x^2}x^2 + h_{L,xy}xy + h_{L,y^2}y^2 & \text{if } y < c(x) \\ \mathbf{h}_R = h_{R,1}1 + h_{R,x}x + h_{R,y}y + h_{R,x^2}x^2 + h_{R,xy}xy + h_{R,y^2}y^2 & \text{otherwise} \end{cases} \end{cases} \tag{18}$$

Although left (*L*) and right (*R*) don't have absolute meaning in 2-D, we've kept the notation from our 1-D illustration for consistency: *L* and *R* subscripts indicate data values or expansion coefficients as we approach an interface from one side or the other. The differential operator *D* is a bit more complicated in 2-D:

$$\begin{bmatrix} u_{(k)t} \\ v_{(k)t} \\ f_{(k)t} \\ g_{(k)t} \\ h_{(k)t} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \left(\frac{1}{\rho}\right)\frac{\partial}{\partial x} & \left(\frac{1}{\rho}\right)\frac{\partial}{\partial y} & 0 \\ 0 & 0 & 0 & \left(\frac{1}{\rho}\right)\frac{\partial}{\partial x} & \left(\frac{1}{\rho}\right)\frac{\partial}{\partial y} \\ (\lambda+2\mu)\frac{\partial}{\partial x} & \lambda\frac{\partial}{\partial y} & 0 & 0 & 0 \\ \mu\frac{\partial}{\partial y} & \mu\frac{\partial}{\partial x} & 0 & 0 & 0 \\ \lambda\frac{\partial}{\partial x} & (\lambda+2\mu)\frac{\partial}{\partial y} & 0 & 0 & 0 \end{bmatrix}^k \begin{bmatrix} u \\ v \\ f \\ g \\ h \end{bmatrix} \triangleq D^k \begin{bmatrix} u \\ v \\ f \\ g \\ h \end{bmatrix} \tag{19}$$

Consider an RBF-FD stencil near a curved interface, as shown in Figure 3. At this location, the following must be true for continuity of traction and motion to hold (note that there is no continuity relation for *f*):

$$\begin{cases} \left[u'_L - u'_R\right]\Big|_{(x=0,y=0)} = 0 \\ \left[v'_L - v'_R\right]\Big|_{(x=0,y=0)} = 0 \\ \left[g'_L - g'_R\right]\Big|_{(x=0,y=0)} = 0 \\ \left[h'_L - h'_R\right]\Big|_{(x=0,y=0)} = 0 \end{cases} \tag{20}$$

In (20), the primed data values are the same physical quantities we have been discussing already (horizontal particle velocity, vertical particle velocity, etc.), but expressed in the rotated coordinate system shown in Figure 3.

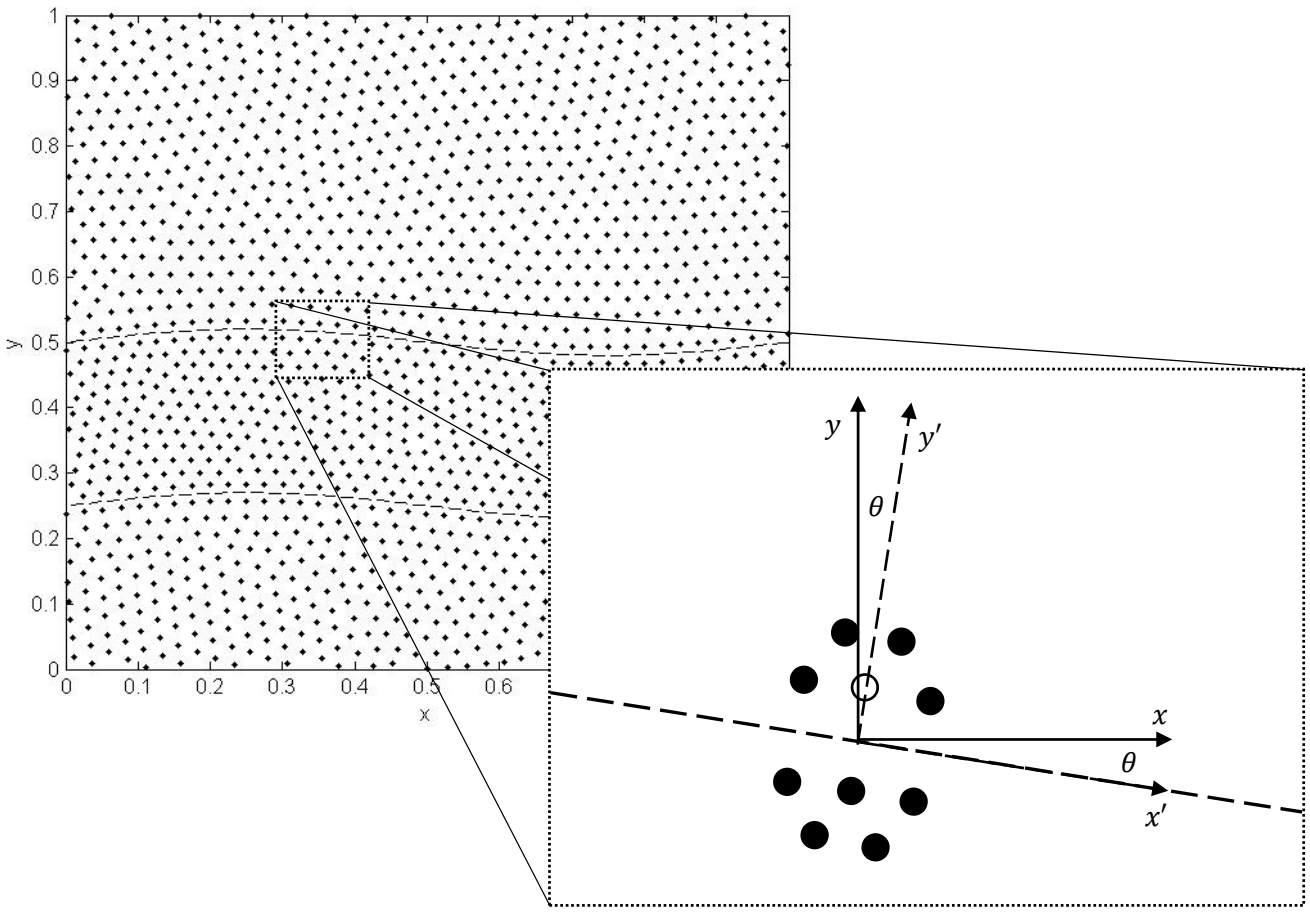


Figure 3. (ABOVE) RBF nodes surround curved interfaces, shown by dashed lines. (BELOW) A very small stencil for evaluating the spatial derivatives at the RBF node indicated by the empty circle. Weights are applied to data at all RBF node (circle) locations. The stencil is created using RBF-FD for a horizontal interface in the x', y' coordinate system.

We can express each of the 5 data fields in the rotated coordinate system:

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \triangleq R_{uv} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u' \\ v' \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} \cos^2(\theta) & 2\sin(\theta)\cos(\theta) & \sin^2(\theta) \\ -\sin(\theta)\cos(\theta) & \cos^2(\theta)-\sin^2(\theta) & \sin(\theta)\cos(\theta) \\ \sin^2(\theta) & -2\sin(\theta)\cos(\theta) & \cos^2(\theta) \end{bmatrix} \begin{bmatrix} f \\ g \\ h \end{bmatrix} \triangleq R_{fgh} \begin{bmatrix} f \\ g \\ h \end{bmatrix} = \begin{bmatrix} f' \\ g' \\ h' \end{bmatrix} \tag{22}$$

We can also express time derivatives in the rotated coordinate system:

$$\begin{bmatrix} u'_{(k)t} \\ v'_{(k)t} \\ f'_{(k)t} \\ g'_{(k)t} \\ h'_{(k)t} \end{bmatrix} = \begin{bmatrix} R_{uv} & 0 \\ 0 & R_{fgh} \end{bmatrix} \left(D^k \begin{bmatrix} u \\ v \\ f \\ g \\ h \end{bmatrix} \right). \tag{23}$$

As in 1-D, we can uphold continuity of motion and traction by enforcing

$$\left[D_L^k \begin{bmatrix} u'_L \\ v'_L \\ g'_L \\ h'_L \end{bmatrix} - D_R^k \begin{bmatrix} u'_R \\ v'_R \\ g'_R \\ h'_R \end{bmatrix} \right]_{y'=0} = \left[\begin{bmatrix} D_L^k, -D_R^k \end{bmatrix} \begin{bmatrix} u'_L \\ v'_L \\ g'_L \\ h'_L \\ u'_R \\ v'_R \\ g'_R \\ h'_R \end{bmatrix} \right]_{y'=0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{24}$$

In 1-D, we determined each support monomial for an FD stencil that crosses an interface by enforcing that every row associated with constant expansion terms after *k* applications of the differential operator (equation (17)) sum to zero. In 2-D, we similarly uphold continuity of motion and traction by enforcing that all expansion coefficients associated with the first (*p*-*k*) monomial terms of the rotated coordinate x' (that is, 1, x', $(x')^2$, etc.) after *k* applications of the operators in (24) sum to zero, where *p* is the maximum degree of polynomial support we wish to add across the interface. This ensures that continuity of motion and traction are upheld to $O(h^p)$ accuracy at $y' = 0$ for a mildly-curved interface that is well-represented locally by a linear approximation. The second of our two test cases uses this method.

If desired, one can also account more rigorously for curvature of a smooth interface. The trigonometric functions involved in equations (21) and (22) may be replaced by local expansions of those functions in terms of the local horizontal coordinate x'. After application of a given power *k* of the differential operator in (24), a local expansion for the interface shape itself (again, in terms of

x') can be inserted into entries for y'. After gathering like powers of x', enforcing that expansion coefficients associated with the first (*p*-*k*) monomial terms of x' sum to zero then upholds continuity of motion and traction to $O(h^p)$ accuracy not only for a locally flat interface, but also for any smooth interface. This method is used in the first of our two test cases.

In the numerical solutions examined so far, we have not observed the latter, rigorous treatment of interface curvature to improve solution error much compared to using a locally linear approximation for the interface. However, future study may show that more extreme interface shapes and higher-order RBF-FD stencils justify the more complex procedure.

RBF-FD TEST CASES

Example 1. Simple p-wave test problem; two interfaces

Our first test problem involves two mildly curved interfaces within a doubly periodic unit square (Figure 4). In this domain, A horizontal p-wave front begins at $y = 0.75$ and travels in the negative y-direction for 0.3 time units, encountering the mild sinusoidal interfaces near $y = 0.5$ and $y = 0.25$. Figure 5(a) shows the data field v at $t = 0.3$. Figures 5(b) and 5(c) show errors for N = 160,000 node pseudospectral (PS, on a Cartesian grid) and RBF-FD computed data fields v at t = 0.3, respectively, using an N = 640,000-node RBF-FD solution as a reference. The chaotic error in the PS solution has resulted from its naïve differentiation across the interfaces. Second degree polynomial support was here added to RBF-FD stencils that cross interfaces, and other RBF-FD stencils featured third degree polynomial support. Both the PS and RBF-FD methods used 4^{th}-order Runge-Kutta time integration with a time step of 2.5E-04.

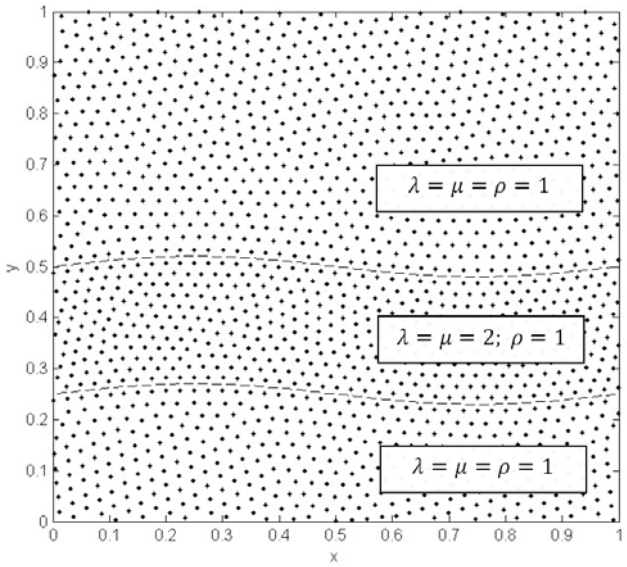
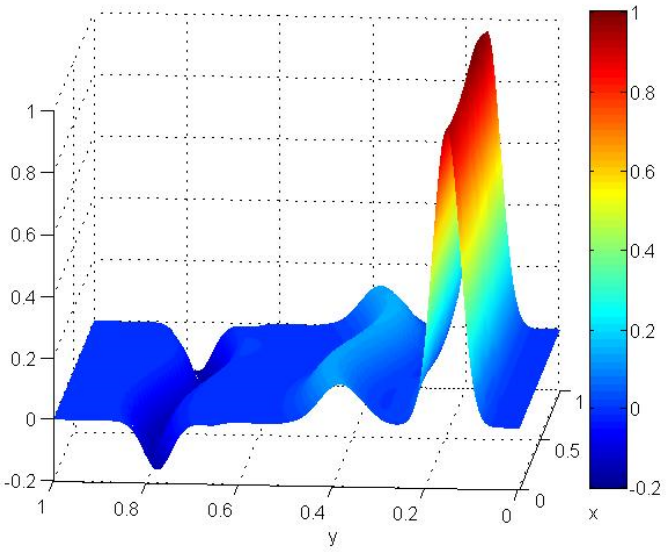
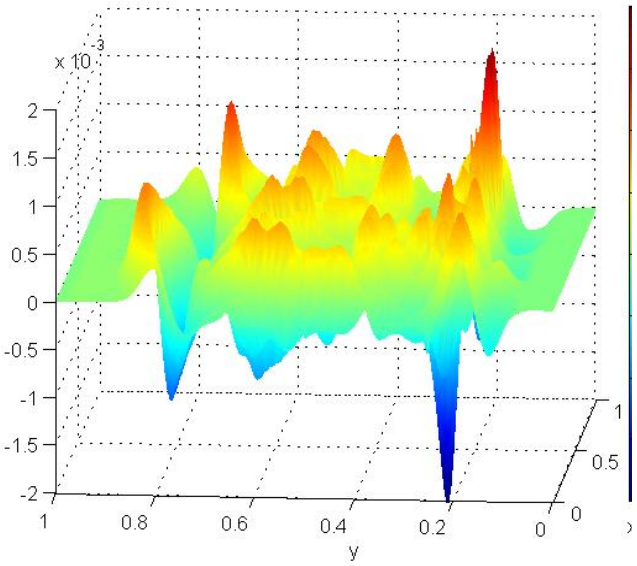


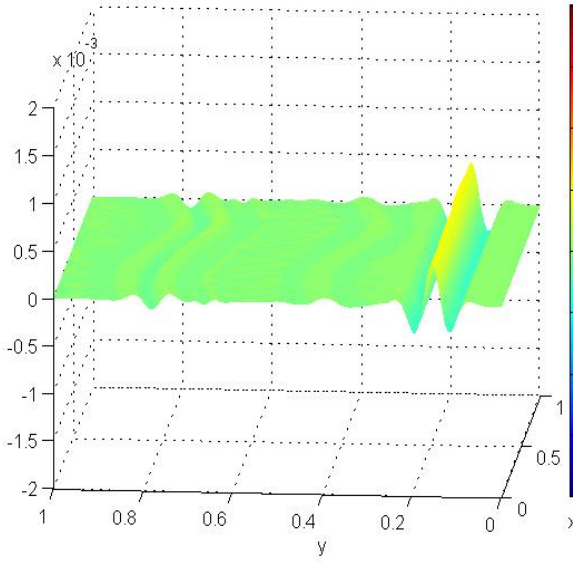
Figure 4. An example of a coarse (N = 1,600) node set for Example 1. Note that nodes adjacent to the interface orthogonally straddle it. Here, only *Case* 2 and *Case* 3 stencils are used; Cartesian structure could have been used for the parts of the domain away from interfaces (*Case* 1).



(a)



(b)



(c)

Figure 5. (a) Data field v at t = 0.3 for the p-wave test problem in Example 1, (b) Error in an N = 160,000-node PS solution at t = 0.3. (c) Error in an N = 160,000-node RBF-FD solution at t = 0.3. The vertical scale is the same in plots (b) and (c). In the shown perspective, the initial wave front travels from left to right. In (c), we note that the error is purely dispersive; a high-order FD method could have been used away from the interface (Case 1) to significantly reduce these errors.

Tables 1 and 2 compare normalized ℓ^2 errors for the PS and RBF-FD methods in the variable *v* at *t* = 0.3. Note how RBF-FD preserves global 3rd-order accuracy.

Table 1. PS solution errors: Example 1

Number of nodes	Normalized ℓ^2 error	Error ratio (2*h*:*h*)
10,000	3.5 E-03	
40,000	1.1 E-03	3.2
160,000	3.6 E-04	3.1

Table 2. RBF-FD solution errors: Example 1

Number of nodes	Normalized ℓ^2 error	Error ratio (2*h*:*h*)
10,000	6.3 E-03	
40,000	8.8 E-04	7.2
160,000	9.7 E-05	9.1

Example 2. Point source in a "mini-Marmousi" domain

Figure 6 shows a "mini-Marmousi" model, and Figure 7 results from a point source in the region.

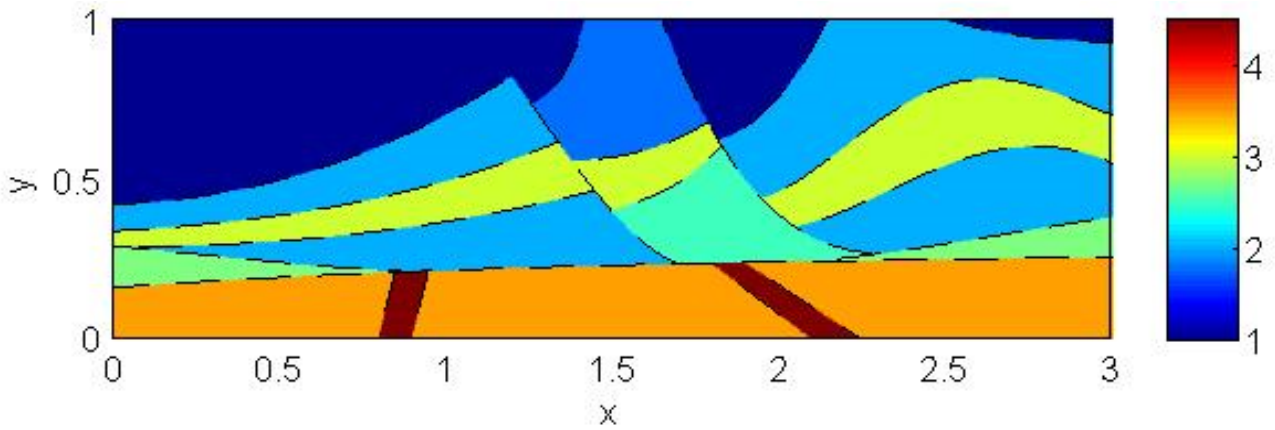


Figure 6. Relative p-wave velocity structure used to test new numerical methods. The structure is inspired by the Marmousi model: http://www.caam.rice.edu/~benamou/testproblem.html

A pressure source located at (0.5, 0.8) has created a radial disturbance traveling outward from that location. Figures 7(a) and 7(b) show the data field v at $t = 0.15$ and $t = 0.3$, respectively. Figure 8 shows errors of N = 38,400-node and 153,600-node pseudospectral (PS) and RBF-FD solutions for the variable v at $t = 0.3$, using an N = 614,400-node RBF-FD solution as a reference (using an extremely highly resolved PS calculation as reference give equivalent results). For the PS solution, the naïve differentiation across the interfaces leads to highly oscillatory errors which only decrease relatively slowly under node refinement. In contrast, the RBF-FD solution improves by a factor of about 10 (both near to and away from interfaces) when the distances between the nodes is halved. For a third order accurate method, a factor of eight would have been expected. Second degree polynomial support was here added to RBF-FD stencils that cross the interfaces (*Case 3*), and all other RBF-FD stencils (*Case 2*) featured third degree polynomial support. Weights for *Case 2* stencils that cross an interface near an interface cusp or intersection between two interfaces were determined without the special interface treatment described here. Both RBF-FD and PS methods used standard 4th-order Runge-Kutta time integration with a time step of 1.25E-04.

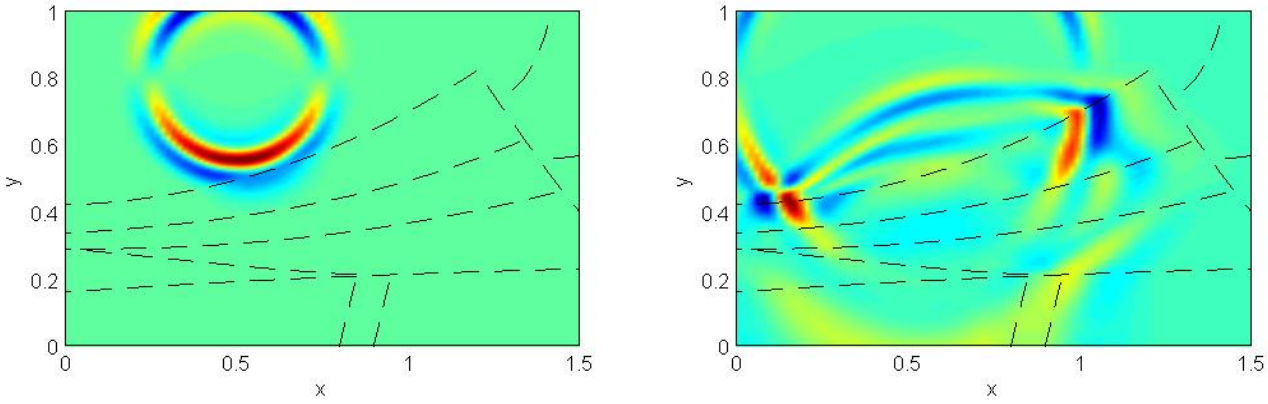


Figure 7. Data field v at $t = 0.15$ (a) and $t = 0.3$ (b) from the point source test problem in the mini-Marmousi domain from Figure 6.

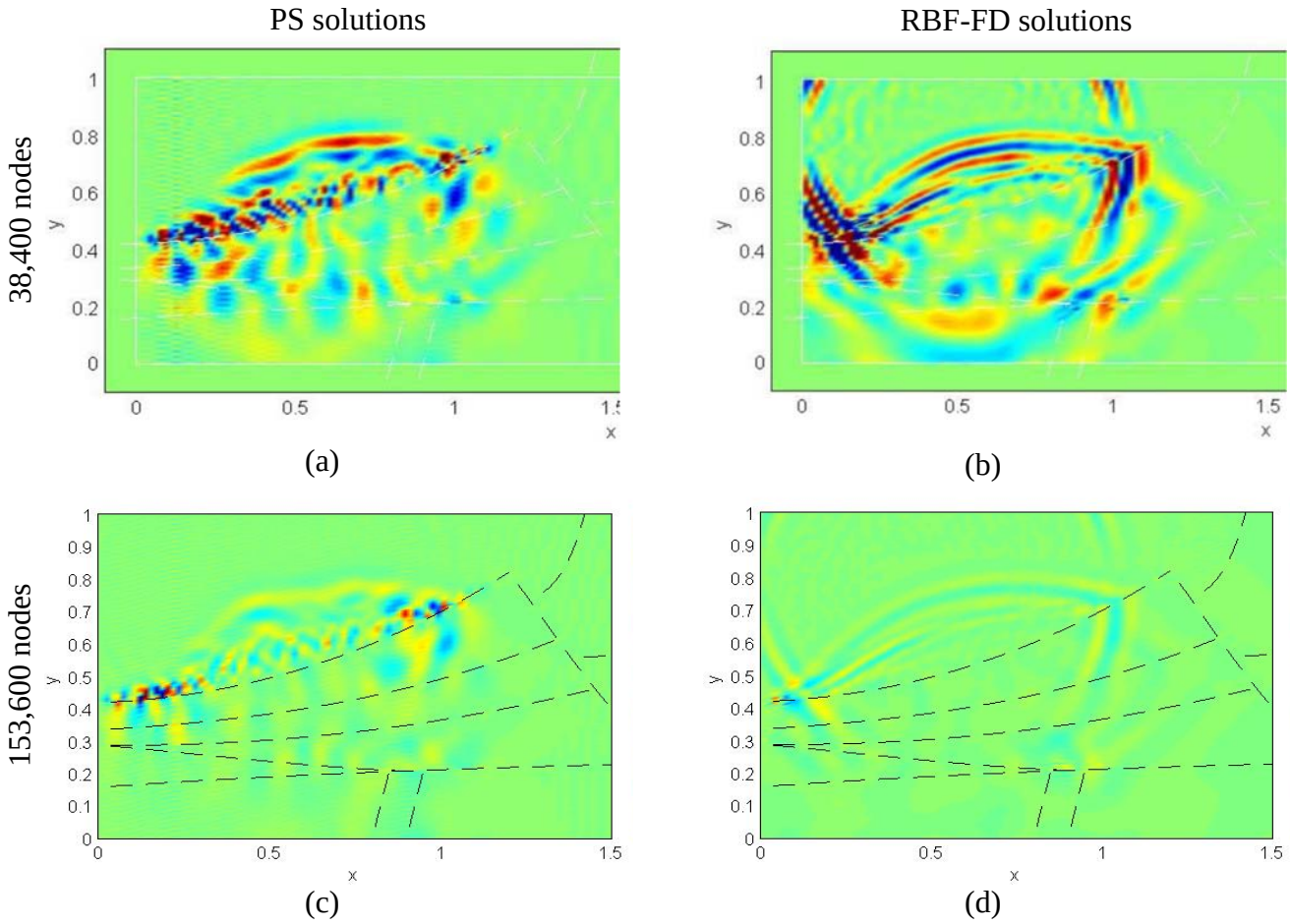


Figure 8. Errors in data field v at time $t = 0.3$ for N =38,400-node (top row) and N = 153,600-node (bottom row) solutions to the mini-Marmousi test problem when using PS (first column) and RBF-FD (second column) discretizations. The color map that represents error value is scaled exactly the same for each of the four plots above.

CONCLUSIONS

As implemented here, the present RBF-FD method achieves roughly third order convergence both with regard to dispersive errors and for reflection/transmission at interfaces. Because of its local nature, the method can be implemented successfully also for high-resolution 3-D simulations in a

cost competitive manner on large-scale distributed memory computer systems. Future studies will address CPU and GPU performances and give more specific large-scale feasibility assessments.

In the mini-Marmousi test problem, RBF-FD stencils near interface cusps and intersections were determined without the interface treatment discussed in this paper. Future work will also include an analysis of the error introduced by naïve treatment of these isolated point-like zones in the domain, and possibly provide suggestions for how also these errors can be minimized.

Although the present method has proved stable enough to produce the promising preliminary results shown here, future studies will address the stability of this approach more rigorously and discuss what (if any) additional measures should be taken.

REFERENCES

Driscoll, T. A., and B. Fornberg, 2002, Interpolation in the limit of increasingly flat radial basis functions: Computers & Mathematics with Applications, **43**, 413-422.

Fasshauer, G. E., 2007, Meshfree approximation methods with Matlab: World Scientific.

Flyer, N., E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr, 2011, A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere: Journal of Computational Physics, **231**, 4078-4095.

Flyer, N., G. B. Wright, and B. Fornberg, 2014, Radial basis function-generated finite differences: a mesh-free method for computational geosciences, *in* W. Freeden, M. Z. Nashed, and T. Sonar, eds., Handbook of geomathematics: Springer Reference.

Fornberg, B., 1987, The pseudospectral method: comparisons with finite differences for the elastic wave equation: Geophysics, **52**, 483-501.

Fornberg, B., 1988, The pseudospectral method: accurate representation of interfaces in elastic wave calculations: Geophysics, **53**, 625-637.

Fornberg, B., and E. Lehto, 2010, Stabilization of RBF-generated finite difference methods for convective PDEs: Journal of Computational Physics, **230**, 2270-2285.

Larsson, E., E. Lehto, A. Heryodono, and B. Fornberg, 2013, Stable calculation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions: SIAM Journal on Scientific Computing, **35**, A2096-A2119.

Li, Z., 1998, The immersed interface method using a finite element formulation: Applied Numerical Mathematics, **27**, 253-267.

Lombard, B., and J. Piraux, 2004, Numerical treatment of two-dimensional interfaces for acoustic and elastic waves: Journal of Computational Physics, **195**, 90-116.

Martin, B., N. Flyer, B. Fornberg, and A. St-Cyr, 2013, Development of mesh-less computational algorithms for seismic exploration: Presented at the 83rd Annual Meeting, SEG.

Robertsson, J. O. A., J. O. Blanch, K. Nihei, and J. Tromp, 2012, Numerical modeling of seismic wave propagation: two-way wave equation methods: SEG geophysics reprint series no. 28: SEG.

Schoenberg, I. J., 1938, Metric spaces and completely monotone functions: Annals of Mathematics, **39**, 811-841.

Shan, Y.Y., C. Shu, and Z. L. Lu, 2008, Application of local MQ-DQ method to solve 3D incompressible viscous flows with curved boundary: Computer Modeling in Engineering and Sciences, **25**, 99-113.

Symes, W. W., and T. Vdovina, 2009, Interface error analysis for numerical wave propagation: Computational Geosciences, **13**, 363-370.

Vishnevsky, D., V. Lisitsa, V. Tcheverda, and G. Reshetova, 2014, Numerical study of the interface errors of finite difference-type simulations of seismic waves: Geophysics, **79**, T219-T232.

Wright, G. B., and B. Fornberg, 2006, Scattered node compact finite difference-type formulas generated from radial basis functions: Journal of Computational Physics, **212**, 99-123.

Yu, Y., and Z. Chen, 2011, Implementation of material interface conditions in the radial point interpolation meshless method: IEEE Transactions on Antennas and Propagation, **59**, 2916-2923.

Zhang, C., and R. J. LeVeque, 1997, The immersed interface method for acoustic wave equations with discontinuous coefficients: Wave Motion, **25**, 237-263.

Zhebel, E., S. Minisini, A. Kononov, and W. A. Mulder, 2014, A comparison of continuous mass-lumped finite elements with finite differences for 3-D wave propagation: Geophysical Prospecting, **62**, 1111-1125.