ABSTRACT

When thermal diffusivity does not vary smoothly within a computational domain, standard numerical methods for solving heat equilibrium problems often converge to an inaccurate solution. In the present paper, we discuss a mesh-free, radial basis function-generated finite difference (RBF-FD)-based method for designing stencil weights that can be applied directly to data that crosses an interface. The approach produces a very accurate solution when thermal diffusivity varies smoothly on either side of an interface. It continues to produce high-quality results when a region between two interfaces is much smaller that the distance between adjacent discrete data nodes in the domain (as becomes the case for thin, nearly insulating layers). We give several test cases that demonstrate the method solving heat equilibrium problems to 4^{th}-order accuracy in the presence of smoothly-curved interfaces.

1. INTRODUCTION

Since the 1970s and 1980s, significant effort has gone into numerically solving parabolic and elliptic equations that model heat- or otherwise diffusivity-related transport processes in domains with interfaces or challenging boundary conditions. Some of the earliest works in this area include Babuska's finite element approach to elliptic problems [1], Peskin's immersed boundary method used for modeling blood flow in the heart [2], and Mayo's work with integral equations to solve Poisson problems and the biharmonic equations in irregular regions [3].

As with the hyperbolic PDEs that model wave transport, many investigators continue to propose new and improved numerical solutions to diffusive interface problems. A significant amount of this work has involved FEMs or other weak-form methods [4],[5],[6].

Other efforts have focused on correcting the error in using traditional finite difference stencils to differentiate non-smooth data across an interface, such as the immersed interface method introduced by LeVeque and Li in [7]. Around the same time, Li and Mayo [8] published work on an alternating direction implicit (ADI) scheme for solving heat equations with interfaces to 2nd-order accuracy. Li and Shen [9] followed up on this method with an ADI approach that allowed smoothly-variable model parameters on either side of an interface. Wiegmann and Bube [10] and Linnick and Fasel [11] presented methods that account for the jump in directional derivatives along Cartesian grid lines when an interface is crossed. Fornberg and Meyer-Spasche [12] and Fornberg [13] described a simple interface correction method for a class of free boundary problems. Some recent interest has been devoted to the study of compact stencils and their correction near interfaces, as in Mittal et al. [14]. Still other approaches use smooth data extensions or "ghost" methods [15],[16], the use of embedded domain [17], and coordinate transformation of a curvilinear interface problem into a rectilinear one [18].

A number of very recent studies on the numerical solution of PDEs have focused on the application of a radial basis function-generated finite difference (RBF-FD) approach that allows determination of finite difference-like collocation weights on a mesh-free cloud of data nodes. Fornberg and Flyer [19] published a comprehensive primer on the subject last year, and papers exploring the use of such methods to handle challenging boundary conditions in elliptic problems (e.g. [20],[21]) are increasingly seen in computational literature.

The present method is an adaptation of our work with time-dependent wave equations in [22] and [23] that uses RBF-FD with an included support space of piecewise polynomials to solve PDEs to a high degree of accuracy in the presence of interfaces. It offers the following combination of features:

- It can be used on a mesh-free cloud of data nodes whose structure accounts for interfaces, as in [24], or conforms to irregular boundaries, as in [20].
- Stencils that include data nodes on both sides of an interface (or possibly multiple interfaces) include all important mathematical information about that interface (curvature, etc.) in the stencil weights. No consideration of the interface is needed after the weights are formed.
- The method requires no formation of fictitious data extensions across interfaces.
- RBF-FD avoids the necessary meshing of FEM techniques.
- Support functions and collocation weights for stencils are determined via relatively small and simple matrix problems.
- The method can achieve any degree of spatial accuracy demanded by the user (though there may be a practical upper limit for stable solutions depending on interface properties in a given problem).

Section 2 of the present paper offers a brief introduction to the isotropic heat equation. Section 2.1 gives a short overview of RBF-FD stencil formation in regions away from interfaces. Section

2.2 explains a simple way to explicitly determine the piecewise polynomial structure of temperature data as that data crosses a curved interface, and how one may use that information within RBF-FD stencils that cross interfaces. When doing so, interfacial thermal diffusion problems may be solved to a high order of accuracy. In section 3, we present numerical examples which show that the method solves such problems to 4^{th}-order accuracy. Section 3.1 verifies the method presented here vs. an analytical solution in the presence of linear interfaces. Section 3.2 focuses on solution of a simple boundary value problem in a domain with two mildly-curved interfaces. Finally, section 3.3 presents results from a domain with two circular interfaces that enclose a very thin, strongly insulating region.

2. RBF-FD METHODOLOGY

The main goal of the present paper is to solve the isotropic 2-D heat equation in domains where its parameters are discontinuous or non-smooth:

$$u_t = \frac{\partial}{\partial x}\left[\alpha(x,y)\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[\alpha(x,y)\frac{\partial u}{\partial y}\right] = (\alpha u_x)_x + (\alpha u_y)_y \qquad (1)$$

In (1), u represents temperature and $\alpha(x,y)$ thermal diffusivity. At an interface, both u and $\mathbf{n}\cdot(\alpha\nabla u)$ must be continuous, where $\mathbf{n}$ is the interface's normal vector. The latter constraint enforces continuity of heat flux across the interface. This paper focuses on solutions to the equilibrium problem where (1) is equal to zero and there is no change in a solution over time. Even so, information from the time-dependent equation will be used to ensure that the equilibrium solutions are correct to a high order of accuracy.

Figure 1 shows an example 2-D domain containing a single curved interface. The parameter α may change instantly in value between the upper and lower sides of that interface. In this domain, RBF-FD nodes are distributed with a quasi-uniform structure that accounts for the interface shape.

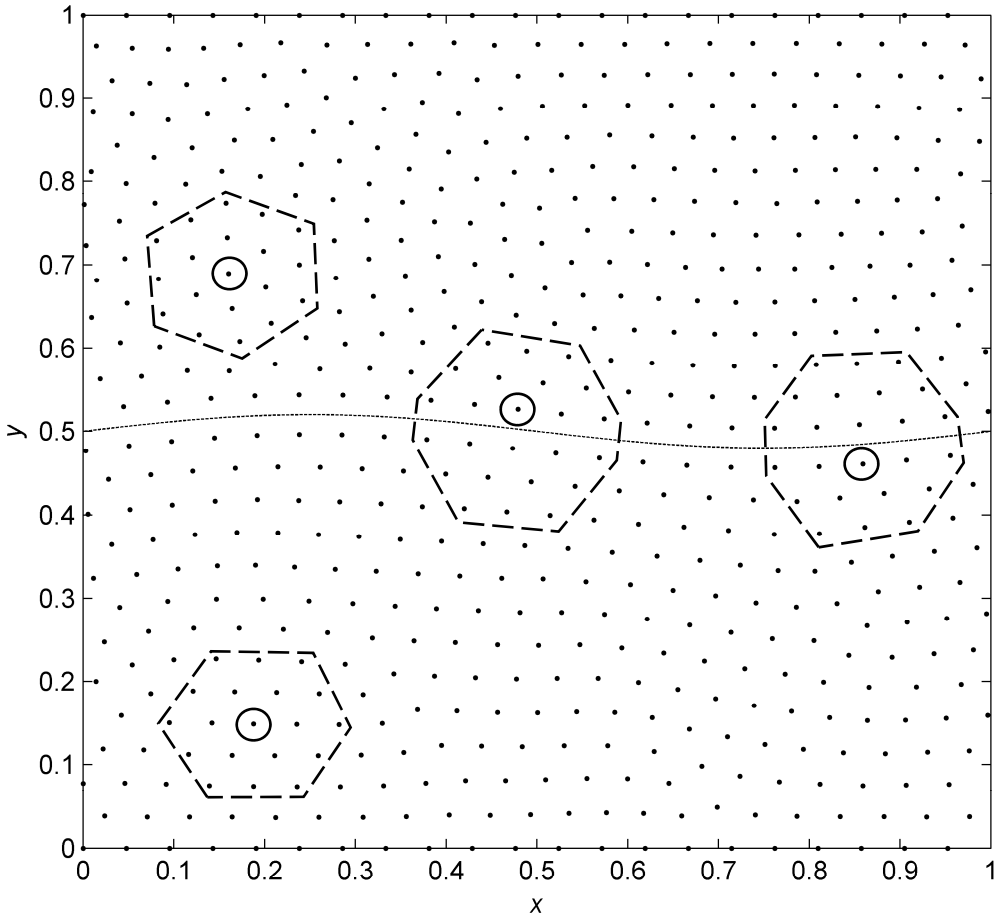

Figure 1. RBF-FD nodes near a curved interface. The polygonal regions show various stencils, some which cross the interface and some which do not. Stencils are designed to approximate PDE operator values at the circled nodes.

For stencils that do not cross an interface, weights are obtained through standard RBF-FD methodology described briefly in section 2.1 and in more detail in [19],[25]. In stencils that do cross the interface, continuity of temperature and heat flux help to create a special set of RBF-FD weights that ensure high-order accuracy when applied simultaneously to data on both sides of the interface.

2.1 *Determination of RBF-FD stencils away from an interface*

Traditional finite difference weights for approximating a 2-D operator L (such as $L = \partial / \partial x$ or $L = \partial^2 / \partial x^2 + \partial^2 / \partial y^2$) cannot be determined in a useful way within an arbitrary stencil of data nodes. Attempting to use the standard polynomial approach from 1-D in a 2-D setting results in unstable and sometimes even singular linear systems that must be solved. However, one can pair a set of 2-D polynomials with RBFs $\phi(\| \underline{x} - \underline{x}_k \|_2)$, with one RBF placed at each stencil point $\underline{x}_k = (x_k, y_k)$. A linear system similar to the one in (2) is then solved to determine stencil weights. Although only constant and linear polynomial terms are seen in (2), higher-order terms may be added to achieve more accurate stencils. Entries in the 1, x, and y columns are evaluations of those polynomial terms at a particular node in the stencil (for example, x_k would represent the evaluation of the function $f(x, y) = x$ at node k).

$$\begin{bmatrix} & & & | & 1 & x_1 & y_1 \\ & A & & | & \vdots & \vdots & \vdots \\ & & & | & 1 & x_n & y_n \\ - & - & - & + & - & - & - \\ 1 & \cdots & 1 & | & & & \\ x_1 & \cdots & x_n & | & & 0 & \\ y_1 & \cdots & y_n & | & & & \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ - \\ w_{n+1} \\ w_{n+2} \\ w_{n+3} \end{bmatrix} = \begin{bmatrix} L\phi(\| \underline{x} - \underline{x}_1 \|)|_{\underline{x}=\underline{x}_c} \\ \vdots \\ L\phi(\| \underline{x} - \underline{x}_n \|)|_{\underline{x}=\underline{x}_c} \\ - \\ L1|_{\underline{x}=\underline{x}_c} \\ Lx|_{\underline{x}=\underline{x}_c} \\ Ly|_{\underline{x}=\underline{x}_c} \end{bmatrix} \tag{2}$$

Entries in the (symmetric) matrix A are evaluations of the RBFs placed at each data node (rows) and evaluated at each node (columns): $a_{i,j} = \phi(\| \underline{x}_i - \underline{x}_j \|)$. The RHS terms are evaluated at the evaluation node of the stencil, $\underline{x}_c$. If the stencil node structure is reasonable and if the RBFs are any of the commonly used types, including multiquadrics (MQ): $\phi(r) = \sqrt{1 + (\varepsilon r)^2}$, Gaussians (GA): $\phi(r) = e^{-(\varepsilon r)^2}$, or polyharmonic splines (PHS) $\phi(r) = \begin{cases} r^m & , m \text{ odd} \\ r^m \log r & , m \text{ even} \end{cases}$, the resulting linear system will generally be nonsingular. In the solution vector, $w_1, \ldots, w_n$ are the weights applied to data at nodes $\underline{x}_k, k = 1, 2, \ldots, n$. The rest of the *w*-entries are discarded. For more information about RBF-FD stencil determination, see [19] (its section 5.1.4 provides a derivation of (2)).

The approach above is used in this paper for stencils that do not cross interfaces. Although the paper focuses on modification of the supplemental support polynomials to enforce interface continuity conditions, we also describe and implement a method for modifying the RBFs to help achieve the same goal. In the case of a stencil that crosses an interface, techniques from section 2.2 are used to determine exactly how each polynomial term (up to the desired degree of accuracy)

changes as the interface is crossed. Evaluation of polynomial terms seen in (2) is then dependent on which side of that interface each node resides.

2.2 *Determination of RBF-FD stencils that cross interfaces*

While RBF-FD can be used to place nodes strategically around interfaces, this approach must still be coupled with another that ensures accurate data treatment near and across those interfaces. In [22] and [23], interface continuity conditions together with a PDE informed the creation of special stencil weights that approximated derivatives of cross-interface data. Special piecewise polynomial functions were added to the RBFs that allow determination of weights within RBF-FD stencils. In [26] it was reported that the polynomial functions from section 2.1 dominate convergence behavior of RBF-FD solutions to PDEs during node refinement. The approach described here will focus on adapting the polynomial functions to account for continuity of temperature and heat flux that must hold at an interface.

2.2.1 Determining polynomial basis functions across an interface: 1-D simplification

Before presenting the full, 2-D RBF-FD method, we address the problem of interfaces with a similar, simpler, completely polynomial approach in 1-D. In 1-D, (1) reduces to:

$$u_t = \frac{\partial}{\partial x}\left[\alpha(x)\frac{\partial u}{\partial x}\right] \equiv Du \tag{3}$$

Time derivatives of any order can be found by applying the differential operator D multiple times:

$$u_{(k)t} = \left(\frac{\partial}{\partial x}\left[\alpha(x)\frac{\partial u}{\partial x}\right]\right)^k \equiv D^k u \tag{4}$$

The subscript k indicates the k^{th} time derivative of u. Since u is continuous at an interface, its time derivatives must be continuous there. Information about the time derivatives of heat flux across the interface is also required:

$$(\mathbf{n}\cdot\alpha\nabla u)_{(k)t} = \alpha\frac{\partial}{\partial x}\left[u_{(k)t}\right] \equiv \alpha\frac{\partial}{\partial x}\left[D^k u\right] \tag{5}$$

Continuity of temperature and heat flux require both (4) and (5) to be continuous across an interface for all values of k. If the left and right sides of the interface are referenced as sides L and R, respectively, then for $k = 0,1,2,\ldots$, the following two conditions must be true at the interface:

$$D_L^k u_L = D_R^k u_R \tag{6}$$

$$\alpha_L\frac{\partial}{\partial x}\left[D_L^k u_L\right] = \alpha_R\frac{\partial}{\partial x}\left[D_R^k u_R\right] \tag{7}$$

If there is an interface at $x = 0$, and if a 4th-degree Taylor expansion of temperature data is desired about that location (for determining a 4th-order stencil), the following 10 terms (coefficients) should be considered:

$$u \approx \begin{cases} \mathbf{u}_L = u_{L,1}1 + u_{L,x}x + u_{L,x^2}x^2 + u_{L,x^3}x^3 + u_{L,x^4}x^4 & if\ x \le 0 \\ \mathbf{u}_R = u_{R,1}1 + u_{R,x}x + u_{R,x^2}x^2 + u_{R,x^3}x^3 + u_{R,x^4}x^4 & if\ x \ge 0 \end{cases} \tag{8}$$

In (8), the boldface notation indicates vectors of polynomial coefficients for u in $\mathbb{P}_4(\mathbb{R})$. We will henceforth store them and operate on them as column vectors of coefficients in that space:

$$\mathbf{u}_L = \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix},\ \mathbf{u}_R = \begin{bmatrix} u_{R,1} \\ u_{R,x} \\ u_{R,x^2} \\ u_{R,x^3} \\ u_{R,x^4} \end{bmatrix} \tag{9}$$

The operators D_L and D_R are defined to apply the PDE to these 4th-order expansions:

$$\mathbf{u}_{L,(k)t} = \left[D_x \alpha_L D_x\right]^k \mathbf{u}_L \equiv {D_L}^k \mathbf{u}_L \tag{10}$$

$$\mathbf{u}_{R,(k)t} = \left[D_x \alpha_R D_x\right]^k \mathbf{u}_R \equiv {D_R}^k \mathbf{u}_R \tag{11}$$

Here, the matrix operator D_x is built to work on vectors of polynomial coefficients. For example, the expansion of u on the left side of the interface changes under differentiation:

$$\begin{aligned} &\frac{\partial}{\partial x}\left[(1)u_{L,1} + (x)u_{L,x} + (x^2)u_{L,x^2} + (x^3)u_{L,x^3} + (x^4)u_{L,x^4}\right] = \\ &\left[(1)u_{L,x} + (2x)u_{L,x^2} + (3x^2)u_{L,x^3} + (4x^3)u_{L,x^4}\right] \end{aligned} \tag{12}$$

We define D_x to express this action as a matrix operator on the vector of polynomial coefficients in $\mathbb{P}_4(\mathbb{R})$. For example, its action on the expansion of u on the left side of the interface is as follows:

$$D_x \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} \tag{13}$$

We also allow α to be smoothly variable near the interface:

$$\boldsymbol{\alpha} \approx \begin{cases} \boldsymbol{\alpha}_L = \alpha_{L,1}1 + \alpha_{L,x}x + \alpha_{L,x^2}x^2 + \alpha_{L,x^3}x^3 + \alpha_{L,x^4}x^4 & \text{if } x < 0 \\ \boldsymbol{\alpha}_R = \alpha_{R,1}1 + \alpha_{R,x}x + \alpha_{R,x^2}x^2 + \alpha_{R,x^3}x^3 + \alpha_{R,x^4}x^4 & \text{if } x \geq 0 \end{cases} \tag{14}$$

The thermal diffusivity parameter in (10) and (11) can be replaced with a matrix representation as was done with the differentiation operator. The matrix form of the parameter describes the 4th-order result from multiplying α by u on the left side of the interface:

$$\alpha_L \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} = \begin{bmatrix} \alpha_{L,1} & 0 & 0 & 0 & 0 \\ \alpha_{L,x} & \alpha_{L,1} & 0 & 0 & 0 \\ \alpha_{L,x^2} & \alpha_{L,x} & \alpha_{L,1} & 0 & 0 \\ \alpha_{L,x^3} & \alpha_{L,x^2} & \alpha_{L,x} & \alpha_{L,1} & 0 \\ \alpha_{L,x^4} & \alpha_{L,x^3} & \alpha_{L,x^2} & \alpha_{L,x} & \alpha_{L,1} \end{bmatrix} \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} \tag{15}$$

An analogous form is used on the right side of the interface. If thermal diffusivity is constant on either side of the interface, these matrices simply reduce to scalar forms. Next, it is necessary to constrain the coefficients in (8) so that (6) and (7) hold to the degree of accuracy desired for a stencil.

2.2.2 Determining Type 3 polynomial basis functions for a specific 1-D problem

Suppose a heat equation problem must be solved on $[-1,1]$, with a thermal diffusivity parameter defined as in Figure 2.

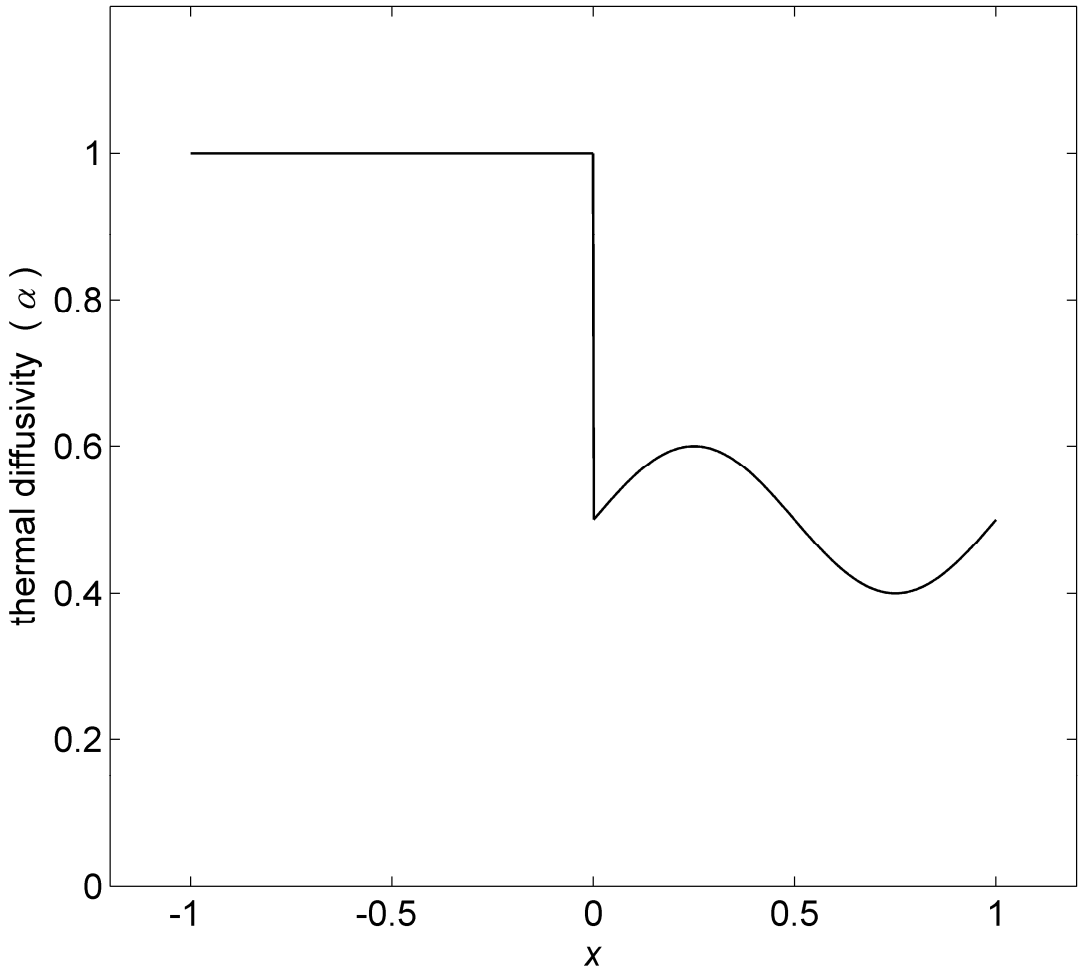

Figure 2. Thermal diffusivity profile for 1-D illustration

Thermal diffusivity is defined here as:

$$\alpha = \begin{cases} 1, & x \in [-1, 0) \\ 0.5 + 0.1\sin(2\pi x), & x \in [0, 1] \end{cases} \tag{16}$$

The following matrices represent multiplication of polynomials by α near the interface at $x = 0$ (as described abstractly in (15) from section 2.2.1):

$$\alpha_L \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} \tag{17}$$

$$\alpha_R \begin{bmatrix} u_{R,1} \\ u_{R,x} \\ u_{R,x^2} \\ u_{R,x^3} \\ u_{R,x^4} \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0.63 & 0.5 & 0 & 0 & 0 \\ 0 & 0.63 & 0.5 & 0 & 0 \\ -4.13 & 0 & 0.63 & 0.5 & 0 \\ 0 & -4.13 & 0 & 0.63 & 0.5 \end{bmatrix} \begin{bmatrix} u_{R,1} \\ u_{R,x} \\ u_{R,x^2} \\ u_{R,x^3} \\ u_{R,x^4} \end{bmatrix} \tag{18}$$

If k is set to zero and (6) evaluated with the previously detailed discrete operators, the interface relation below must hold to ensure continuity of temperature:

$$\mathbf{u}_{L,(0)t}\Big|_{x=0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (1)u_{L,1} \\ (x)u_{L,x} \\ (x^2)u_{L,x^2} \\ (x^3)u_{L,x^3} \\ (x^4)u_{L,x^4} \end{bmatrix}_{x=0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (1)u_{R,1} \\ (x)u_{R,x} \\ (x^2)u_{R,x^2} \\ (x^3)u_{R,x^3} \\ (x^4)u_{R,x^4} \end{bmatrix}_{x=0} = \mathbf{u}_{R,(0)t}\Big|_{x=0}$$

(19)

Since the interface is at $x = 0$, (19) implies that $u_{L,1}$ and $u_{R,1}$ (constant terms) are equal:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{R,1} \\ u_{R,x} \\ u_{R,x^2} \\ u_{R,x^3} \\ u_{R,x^4} \end{bmatrix} \tag{20}$$

We next evaluate (7), still with $k = 0$. This time, continuity of heat flux is enforced across the interface.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} (1)u_{L,1} \\ (x)u_{L,x} \\ (x^2)u_{L,x^2} \\ (x^3)u_{L,x^3} \\ (x^4)u_{L,x^4} \end{bmatrix}_{x=0} = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0.6 & 1.0 & 0 & 0 \\ 0 & 0 & 1.3 & 1.5 & 0 \\ 0 & -4.1 & 0 & 1.9 & 2.0 \\ 0 & 0 & -8.3 & 0 & 2.5 \end{bmatrix} \begin{bmatrix} (1)u_{R,1} \\ (x)u_{R,x} \\ (x^2)u_{R,x^2} \\ (x^3)u_{R,x^3} \\ (x^4)u_{R,x^4} \end{bmatrix}_{x=0} \tag{21}$$

Since heat flux is continuous at $x = 0$, the constant expansion term of heat flux (and all time derivatives of that term) must be equal between different sides of the interface. The first row of linear relationships between Taylor coefficients of u seen in (21) – representing continuity in the constant coefficient of heat flux – must be true. The first row of (21) is added to the equivalence of constant temperature expansion terms from (20):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{R,1} \\ u_{R,x} \\ u_{R,x^2} \\ u_{R,x^3} \\ u_{R,x^4} \end{bmatrix} \tag{22}$$

Additional relationships between expansion coefficients of u are obtained through evaluation of (6) and (7) for larger values of k. For every k value, one linear relationship between expansion coefficients of u is gathered to enforce continuity of the k^{th} time derivative of temperature, and another relationship is gathered to enforce continuity of the k^{th} time derivative of heat flux. After evaluation through $k = 2$ with (6) and through $k = 1$ with (7), the following relationships have been established:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 24 \end{bmatrix} \begin{bmatrix} u_{L,1} \\ u_{L,x} \\ u_{L,x^2} \\ u_{L,x^3} \\ u_{L,x^4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0.6 & 1.0 & 0 & 0 \\ 0 & 0 & 1.3 & 1.5 & 0 \\ 0 & -12 & 1.6 & 7.5 & 6.0 \end{bmatrix} \begin{bmatrix} u_{R,1} \\ u_{R,x} \\ u_{R,x^2} \\ u_{R,x^3} \\ u_{R,x^4} \end{bmatrix} \tag{23}$$

Throughout this paper, the matrices above are denoted "continuity matrices", and represented by the letter C. Using this convention, (23) may be rewritten as:

$$C_L \mathbf{u}_L = C_R \mathbf{u}_R \tag{24}$$

This relation can be used to determine data expansion coefficients on one side of the interface if their counterparts on the other side of the interface are known. For example:

$$\mathbf{u}_L = C_L^{-1} C_R \mathbf{u}_R \tag{25}$$

If we assume the standard basis set of polynomials for $\mathbf{u}_R$, for instance, we can determine how those right-side polynomial functions must change as they pass through the interface. The result of this process for the example problem is shown in Figure 3.

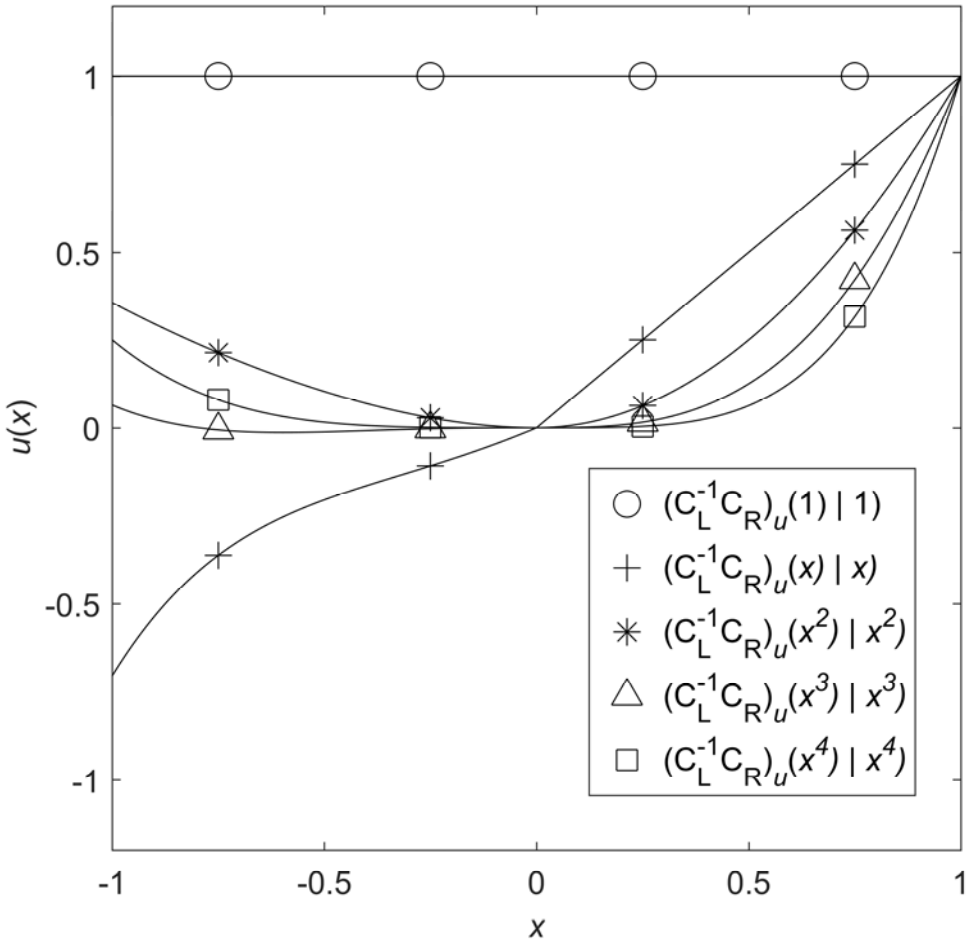

Figure 3. Standard basis functions for $\mathbf{u}_R$ and their counterparts in $\mathbf{u}_L$, determined explicitly to enforce continuity of temperature and heat flux.

Before time stepping can begin, finite difference weights must be created to accurately approximate the differential operator D. In the current 4th-order example, weights must reproduce operator values of the piecewise polynomials plotted in Figure 3 instead of the standard basis

$\{1, x, \ldots, x^4\}$. Figure 4 shows how stencil weights for D differ from traditional FD weights near the interface (in the 1-D example problem).

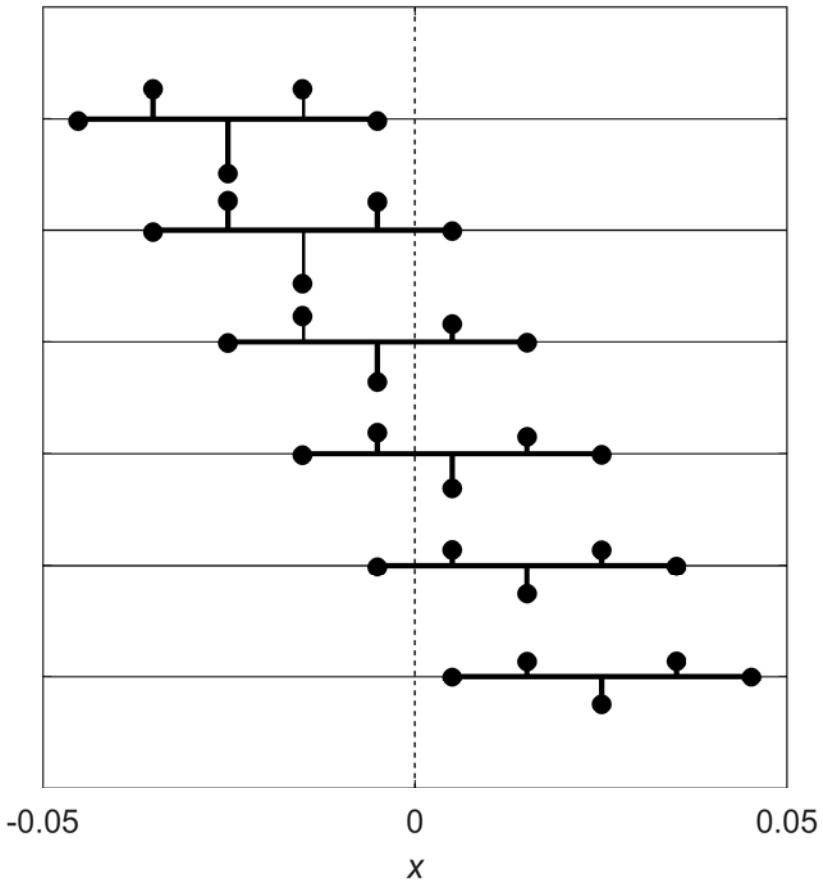

Figure 4. Plot of weights within $(\alpha u_x)_x$ stencils near the interface in the 1-D test problem, with h = 0.01. Height of dots above or below each vertical line indicates magnitude and sign (above = positive; below = negative) of a stencil weight.

2.2.3 Determining polynomial basis functions for stencils that cross an interface: 2-D example

In 1-D, a complete basis of piecewise Taylor expansion functions can be created to uphold continuity of temperature and heat flux to a high degree of accuracy, and stencil weights can be

determined with a basis of such functions using a traditional, "polynomial-only" approach. As mentioned in section 2.1, this is not practical in 2-D. Fortunately, RBF-FD methodology allows the use of piecewise 2-D polynomials that are analogous to the 1-D piecewise polynomials of section 2.2.2, and these functions can help create high-order stencils for 2-D data that crosses an interface (or multiple interfaces, as explored in section 3.3).

For illustration, assume that polynomial support through second degree terms must be added to an RBF-FD stencil that contains points on either side of an interface. Near the data node where the stencil approximates a linear operator, a point on the interface (described by the equation $y = c(x)$) is chosen. The nearby point on the interface is designated as the origin for evaluating functions that support the stencil (both RBFs and polynomials).

The following data expansions require coefficient determination:

$$\mathbf{u} \approx \begin{cases} \mathbf{u}_L = u_{L,1}1 + u_{L,x}x + u_{L,y}y + u_{L,x^2}x^2 + u_{L,xy}xy + u_{L,y^2}y^2 & \text{if } y < c(x) \\ \mathbf{u}_R = u_{R,1}1 + u_{R,x}x + u_{R,y}y + u_{R,x^2}x^2 + u_{R,xy}xy + u_{R,y^2}y^2 & \text{otherwise} \end{cases} \tag{26}$$

Even though left (L) and right (R) don't necessarily mean anything in 2-D, the notation is retained from the 1-D examples in sections 2.2.1 and 2.2.2. L and R subscripts still denote quantities on different sides of an interface. For 2-D work, we return to the definition of the differential operator D in (1).

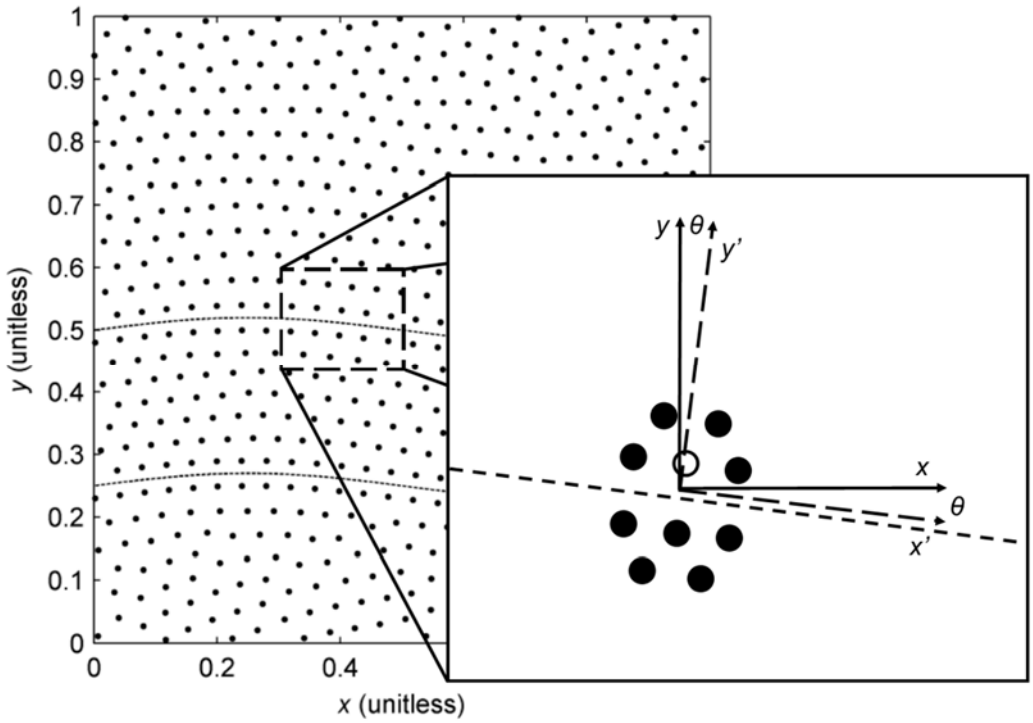

Figure 5. RBF nodes surround interfaces shown by curved lines. (Zoomed-in view) A very small stencil straddles the interface. Weights are applied to all nodes in the zoomed-in view, with stencil evaluation at the unfilled node above the interface.

In Figure 5, two rows of data nodes orthogonally straddle an interface. We began using this strategy after consulting [24], thinking that such a node layout might yield more accurate solutions. This type of node structure has also been helpful in maintaining stability of solutions, particularly when model parameter jump discontinuities are severe (as in section 3.2). While several rows of the nodes are placed deliberately about the interface, most nodes in the domain are distributed in the observed quasi-uniform arrangement through a simulated static repulsion routine.

In 1-D, support monomials were characterized across an interface by enforcing discrete forms of (6) and (7) for constant expansion coefficients (and constant term time derivatives) of temperature and heat flux. In 2-D, continuity of temperature is similarly upheld by enforcement of

linear relationships between data expansion terms on both sides of an interface. If a mildly-curved interface is well-described (at least locally) by a linear approximation ($y' \approx 0$), these relationships may be obtained by observing time derivatives of the first ($p - 2k$) monomial terms in the coordinate x' (i.e., 1, x', $(x')^2$, etc.) of u after k applications of the discrete form of (1). Here, p is the maximum degree of polynomial support to be added across the interface. As in 1-D, the discrete PDE operator turns these time derivatives into linear relationships (rows of coefficients) between the (2-D) expansion terms of u on both sides of the interface. Rows of coefficients corresponding to time derivatives of 1, x', $(x')^2$, etc. on each side of the interface are collected into continuity matrices C_L and C_R.

Continuity of heat flux is also relatively simple to enforce if a locally linear approximation to interface shape is assumed. In that case, the interface's normal direction is just the direction of y'. Relationships of expansion coefficients across the interface can be further constrained by observing time derivatives of the first ($p - 2k - 1$) monomial terms in the coordinate x' after k applications of the discrete form of (27). The corresponding rows of coefficients in the result are gathered into C_L and C_R.

$$(\mathbf{n} \cdot \alpha \nabla u)_{(k)t} = \alpha \frac{\partial}{\partial y'} \left[u_{(k)t} \right] \equiv \alpha \frac{\partial}{\partial y'} \left[D^k u \right] \tag{27}$$

Enforcing linear relationships between the expansion coefficients of u as described in the last two paragraphs ensures that continuity of temperature and heat flux are upheld to $O(h^p)$ accuracy at $y' = 0$ for a mildly-curved interface that is locally very well-represented by a linear approximation. When enough unique linear relationships between the expansion coefficients of u

on either side of the interface have been collected, the following expression relates square, well-conditioned matrices C_L and C_R:

$$C_L \mathbf{u}_L = C_R \mathbf{u}_R \tag{28}$$

As in 1-D, given an expansion for u on one side of the interface, the shape of that data on the other side of the interface may be determined by inverting either of the matrices in (28). Assumption of a standard polynomial basis on one side of an interface, along with explicit computation of the necessary changes to that basis that occur across the interface (through continuity matrix inversion), determines the piecewise polynomial functions that must be present in an expansion of temperature data near the interface. These piecewise polynomial functions support RBFs in determining RBF-FD stencil weights across an interface (replacing the traditional Taylor monomials found in section 2.1).

As performed above, the method for determining RBF-FD stencil weights generally works well if the interface is locally very well approximated by a linear function. If desired, one can also account more rigorously for curvature of a smooth interface. To enforce continuity of temperature along a curved interface, after application of a given power *k* of the discrete form of the differential operator in (1), a local expansion for the interface shape itself (again, in terms of x') can be inserted into entries for y'. After this is done, each row of coefficients that corresponds to a time derivative of a single 2-D Taylor term is transformed into a set of contributions it must make to time derivatives of u's expansion restricted to the interface itself:

$$\mathbf{u}_{\text{int}} \approx \begin{cases} \mathbf{u}_{L,\text{int}} = u_{L,1} 1 + u_{L,x'} x' + u_{L,(x')^2} (x')^2 & \text{if } y' < c(x') \\ \mathbf{u}_{R,\text{int}} = u_{R,1} 1 + u_{R,x'} x' + u_{R,(x')^2} (x')^2 & \text{otherwise} \end{cases} \tag{29}$$

For example, consider the $x'y'$ term on side L in u's 2-D expansion around the interface. If the interface is located at $y' = 0.2 + 0.01x'$, then, along the interface, $x'y' = x'(0.2 + 0.01x') = 0.2x' + 0.01(x')^2$. In determining the coefficient $u_{L,x'}$ from (29), then, 0.2 times u's $x'y'$ expansion term must be added on side L of the interface. Likewise, in determining $u_{L,(x')^2}$, 0.01 times u's $x'y'$ term must be added on side L. All of the 2-D expansion terms for u contribute in this manner to determine the (1-D) expansion terms of $\mathbf{u}_{\text{int}}$. In other words, each expansion term of $\mathbf{u}_{\text{int}}$ is a linear combination of the 2-D expansion terms of u, determined by the definition of y' in terms of x'.

Time derivatives of all of the $\mathbf{u}_{\text{int}}$ coefficients in terms of 2-D expansion coefficients currently present in u on either side of the interface can be similarly defined. The first $(p - 2k)$ of the equivalent relationships between data expansion coefficients are collected and placed as rows in continuity matrices C_L and C_R.

Note that in the presence of a flat interface, this curved interface method is identical to the much simpler approach described above. The only difference is that the method requires far less arithmetic when y' is equal to zero. We will examine the tradeoff between simplicity and accuracy of the local linear interface approximation in section 3.1.

Upholding continuity of heat flux along a smoothly-curved interface is only slightly more complicated. As the interface curves, the definition of the interface normal direction changes:

$$(\mathbf{n} \cdot \alpha \nabla u)_{(k)t} = \alpha \left(\cos(\theta') \frac{\partial}{\partial y'} - \sin(\theta') \frac{\partial}{\partial x'} \right) \left[D^k u \right] \tag{30}$$

In (30), θ' is the angle of the interface, and for a smoothly-curving interface the angle is a well-behaved function of x'. The functions $\cos(\theta')$ and $\sin(\theta')$ are replaced with matrices that express the result of multiplying the 2-D expansion of **u** by the expansion of each trigonometric function. This is analogous to how α was replaced by a matrix in (15). The same treatment is given to α in (30).

As when enforcing continuity of temperature, an expansion of the interface itself ($y' = f(x')$) is used to restrict (30) to a 1-D expansion along the interface, and to enforce that time derivatives of the first ($p - 2k - 1$) restricted (1-D) terms are equal across the interface.

Although it may be difficult to achieve analytical expansions for all the trigonometric functions and interface expansions required here, we have found a numerical approach (using finite difference stencils to approximate derivatives of interface representation and associated trigonometric functions) to work quite well in all solutions tested to date.

While the procedure for translating polynomial information from one side of the interface to the other may seem involved, once this is done, stencil weights are determined almost identically to the procedure discussed in section 2.1. One must simply determine on which side of an interface each stencil node resides. For all nodes present on the same side of the interface as the evaluation node, one may simply evaluate each standard 2-D Taylor basis monomial up through the desired degree of accuracy, placing the result in the appropriate column or row of the matrix in (2) (corresponding to that monomial and data node). For nodes that reside on the other side of the interface, one uses the procedures detailed here to transform each basis monomial into its necessary counterpart on the other side of the interface (so that high-order continuity conditions are upheld), and evaluations of this *transformed* combination of Taylor terms gets placed in the appropriate

column or row (corresponding to the correct basis monomial on the initial, evaluation node side of the interface).

2.2.4 Modification of RBFs across interfaces

So far in this paper, continuity conditions have been upheld through specialized treatment of the polynomials involved in determining weights for RBF-FD stencils that cross interfaces. Some work can also be done to make sure that the RBFs that aid in stencil weight determination uphold continuity of temperature and heat flux, at least to first order.

Consider the case in Figure 6, where an RBF is centered in the x, y-plane just below a flat interface at $y = 0.2x$. In this case, $\alpha = 1$ below the interface, and $\alpha = 1/2$ above the interface. Above the interface, the RBF has been spatially "warped" by a coordinate stretching that enforces continuity of heat flux. Aside from a single set of data represented in Figure 10 of section 3.1, all numerical experiments presented in section 3 use this altered RBF structure.

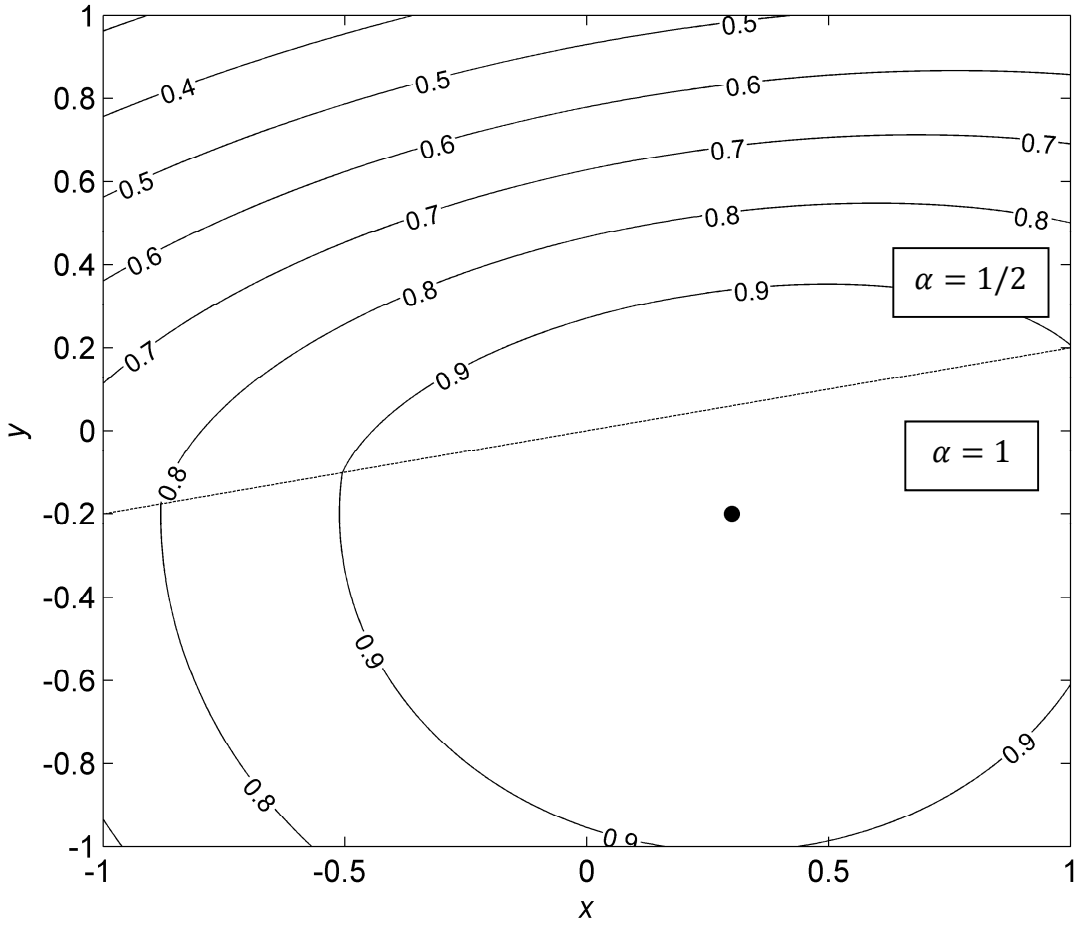

Figure 6. Contours of a warped Gaussian RBF. The standard RBF (with shape parameter $\varepsilon = 0.4$) present below the interface gives way to a warped counterpart in the region above the interface in order to uphold continuity of heat flux.

3. RBF-FD TEST CASES

The equilibrium equation $u_t = 0$ is solved in all test problems, and all problems are carried out in the x-periodic unit square with Dirichlet boundary conditions at $y = 0$ and $y = 1$. All RBF-FD stencils in use Gaussian RBFs with a shape parameter (ε) calculated as follows:

$$\varepsilon = \frac{0.4}{d} \tag{31}$$

Here, d is the distance from the evaluation node of a stencil to its nearest neighbor. Normalizing the shape parameter in this manner can help maintain a balance between stencil accuracy and conditioning of the linear systems solved to obtain stencil weights. The general form $\varepsilon = O(1/d)$ has been adopted in several previous publications, including [27],[28]. Although a value of 0.4 in the numerator of (31) has proven reliable in solving the test cases presented here, additional accuracy could possibly be gained by decreasing the numerator to a smaller constant (as long as the resulting RBF-FD weights produce an acceptably stable solution).

Except for the iterative method solution comparison of test case 3, all RBF-FD stencils have the following properties: RBF-FD stencils that do not cross an interface and are far away from the domain boundary contain the evaluation node and its 41 nearest neighbors for a total of 42 nodes. These stencils are supported by all polynomials up through 5th degree ($x^5, x^4y, \ldots, y^5$). RBF-FD stencils that cross an interface or are close to the domain boundary contain 30 nodes and are supported by all polynomials up through 4th degree. RBF-FD stencils that cross an interface incorporate curvature information into calculation of stencil weights unless a flat interface assumption is explicitly mentioned (as from particular test case data shown in Figures 10 and 14).

3.1 *Test case 1: Moderate jump discontinuities in α across two flat interfaces*

The first 2-D test case was carried out in the x-periodic unit square with simple, linear interfaces present in order to verify the RBF-FD treatment of interfaces in a case where an analytical solution is available. Parameters and Dirichlet boundary conditions are defined as follows:

$$\alpha = \begin{cases} 0.2 & y \in [0.6, 0.8] \\ 1 & otherwise \end{cases} \tag{32}$$

$$u(\Omega) = \begin{cases} \sin(2\pi x) & y = 1 \\ 0 & y = 0 \end{cases} \tag{33}$$

Since temperature and heat flux are continuous at the two interfaces, what follows must be the (unique) analytical solution to the problem:

$$u(x,y) = \begin{cases} \sin(2\pi x)\left[c_1 \exp(2\pi y) + c_2 \exp(-2\pi y)\right], & y \in [0.0, 0.6) \\ \sin(2\pi x)\left[c_3 \exp(2\pi y) + c_4 \exp(-2\pi y)\right], & y \in [0.6, 0.8) \\ \sin(2\pi x)\left[c_5 \exp(2\pi y) + c_6 \exp(-2\pi y)\right], & y \in [0.8, 1.0] \end{cases} \tag{34}$$

The following are enforced to determine constants $c_1, c_2, \ldots c_6$:

- Continuity of u at $y = 0.6$ (the lower interface)
- Continuity of $\mathbf{n} \cdot (\alpha \nabla u) = \alpha u_y$ at $y = 0.6$
- Continuity of u at $y = 0.8$ (the upper interface)
- Continuity of $\mathbf{n} \cdot (\alpha \nabla u) = \alpha u_y$ at $y = 0.8$
- The condition that $u = 0$ along $y = 0$
- The condition that at $t = 0$, $u = \sin(2\pi x)$ along $y = 1$

After determining these constants, we have an analytical solution against which we can judge the performance of our RBF-FD solutions. In Figure 7, errors in RBF-FD solutions are plotted vs. number of data nodes in the domain. We observe the expected 4th order convergence here.

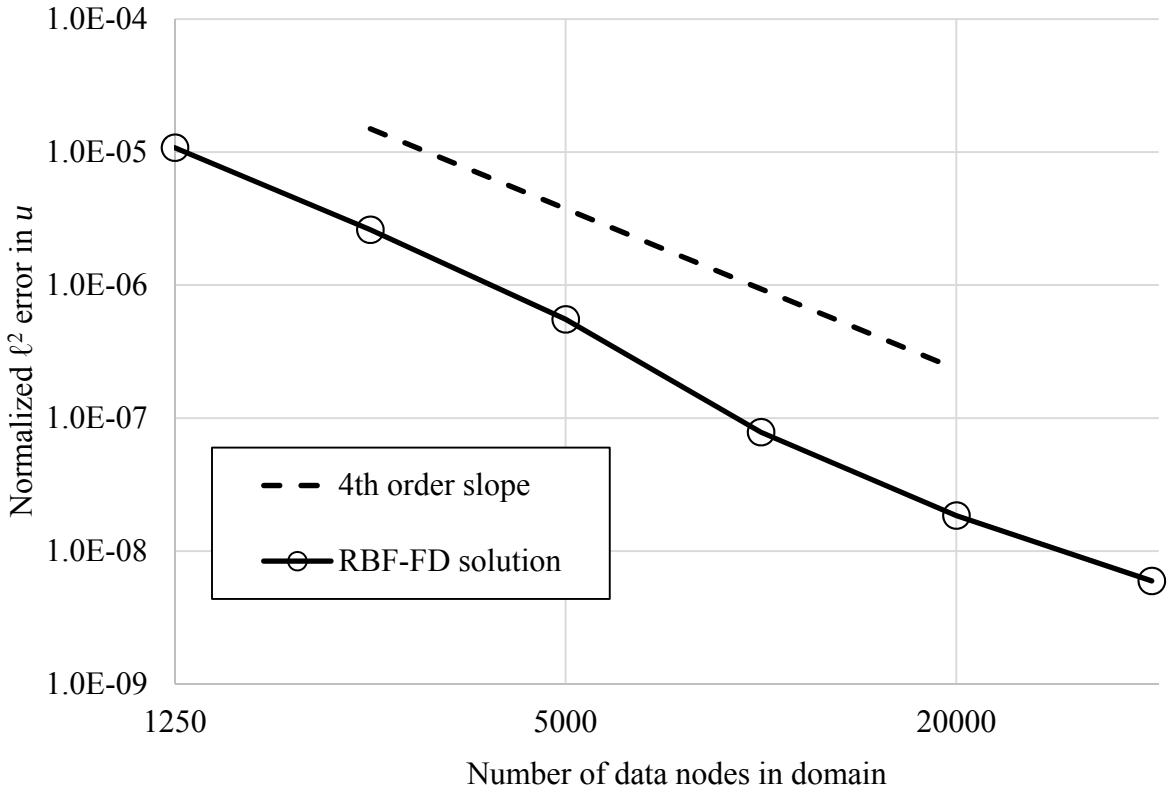

Figure 7. Convergence of RBF-FD solutions for the case 1 problem.

3.2 *Test case 2: Moderate jump discontinuities in α across two curved interfaces*

The next 2-D test problem involves two mildly curved interfaces within a unit square that is periodic in the x-direction and closed at $y = 0$ and $y = 1$ (Figure 8). Between the two interfaces, α is defined by the following equation:

$$\alpha = 0.2 + 0.1\sin(2\pi x)\sin(2\pi y) \tag{35}$$

Elsewhere in the domain, α is equal to 1.

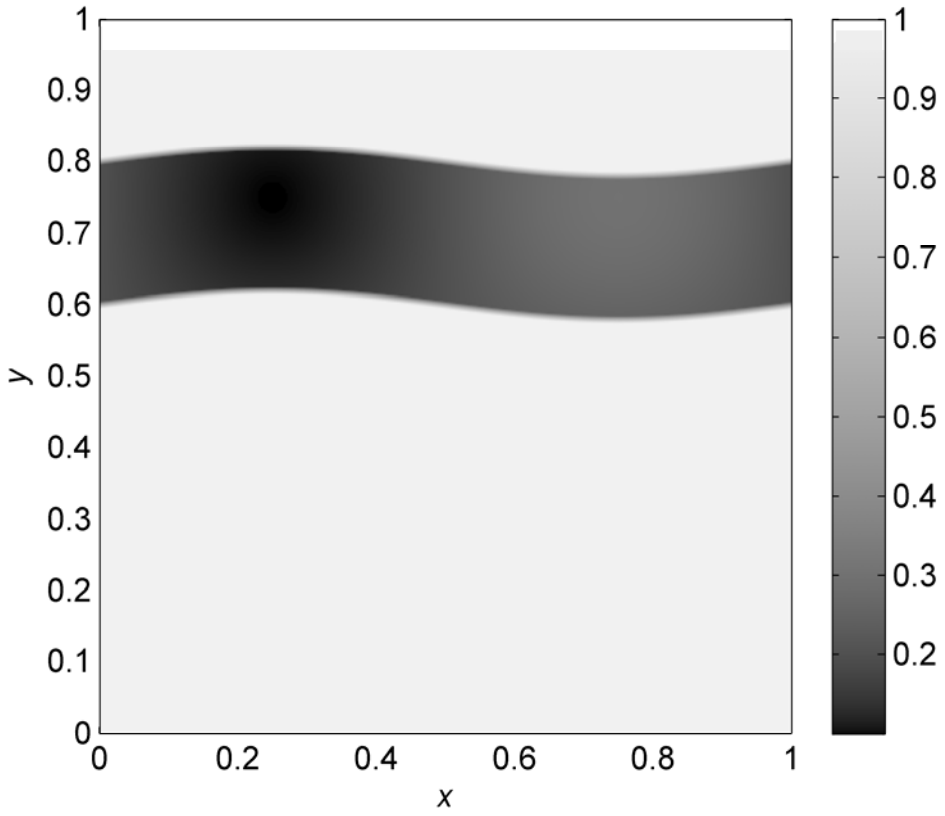

Figure 8. Greyscale map of thermal diffusivity for test case 2.

The following conditions are enforced on the Dirichlet boundaries of the domain:

$$u(\Omega) = \begin{cases} \sin(2\pi x), & y = 1 \\ 0, & y = 0 \end{cases} \tag{36}$$

Figure 9 shows a mesh plot of u for a 40,000-node RBF-FD solution to the equilibrium problem $u_t = 0$ in this domain.

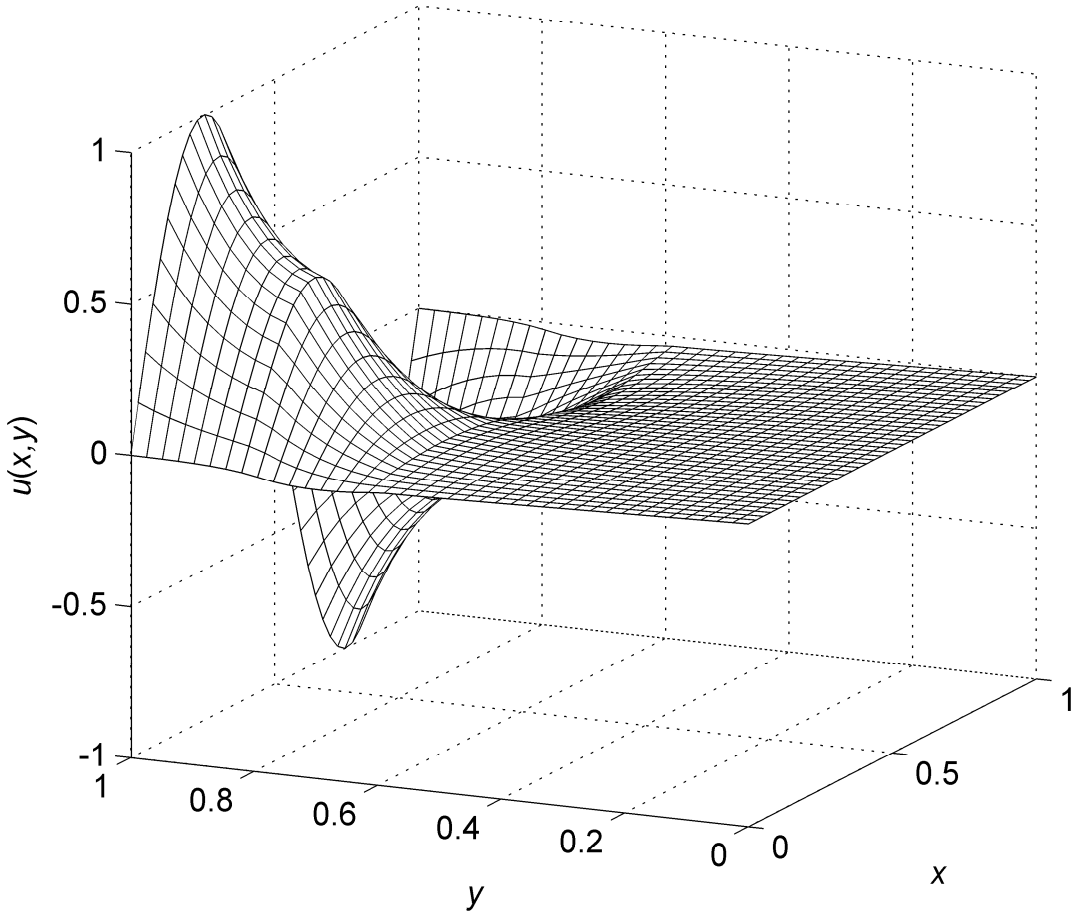

Figure 9. Solution $u(x, y)$ from the second test case, solved with the RBF-FD method on 40,000 data nodes.

Figure 10 shows numerical solution errors for different methods used to solve the case 2 problem carried out at different resolutions. Errors are calculated using a 160,000-node RBF-FD solution as a reference. The three methods represented use naïve 4th-order finite differences (FD4), RBF-FD stencils that assume a local linear approximation to the interface ("flat int."), and RBF-FD stencils that incorporate interface curvature information into the stencil weight calculation ("curv. incl.") for spatial differentiation of temperature data. The problem was solved with MATLAB's backslash operator (sparse direct solver). Note how much more accurate solutions are when interface curvature is incorporated.

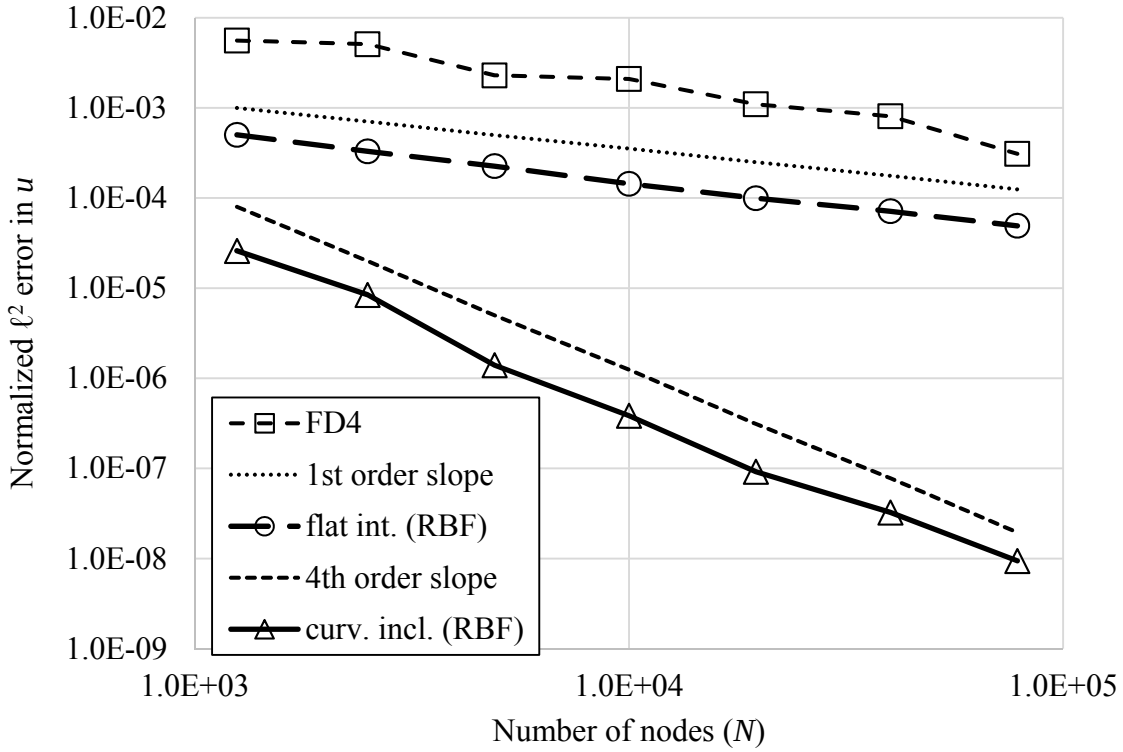

Figure 10. Convergence plot for different methods used to solve test case 2. Extrapolation of the top line indicates that, to reach an error of 10^{-8}, both the FD4 and flat int. (RBF) methods would require around 10^{11} data nodes.

Figure 11 shows another error comparison of solutions using two different methods used to solve the case 2 problem. Both of them are RBF-FD-based methods that incorporate curvature information into the stencil weights. The methods vary in their node placement strategy near the interface and in the shape of the RBFs that contribute to the calculation of stencil weights. The error plot for "no warp; no straddling" solutions represent trials completed on sets of data nodes that were distributed in a manner that didn't take into account the location of the interface, and whose stencils were determined with traditional RBFs (rather than the modified RBFs, as presented in 2.2.4). "With warp and straddling" solutions featured an interface-straddling strategy as pictured in Figure 5 and also modified RBFs that uphold continuity conditions to 1st order at interfaces.

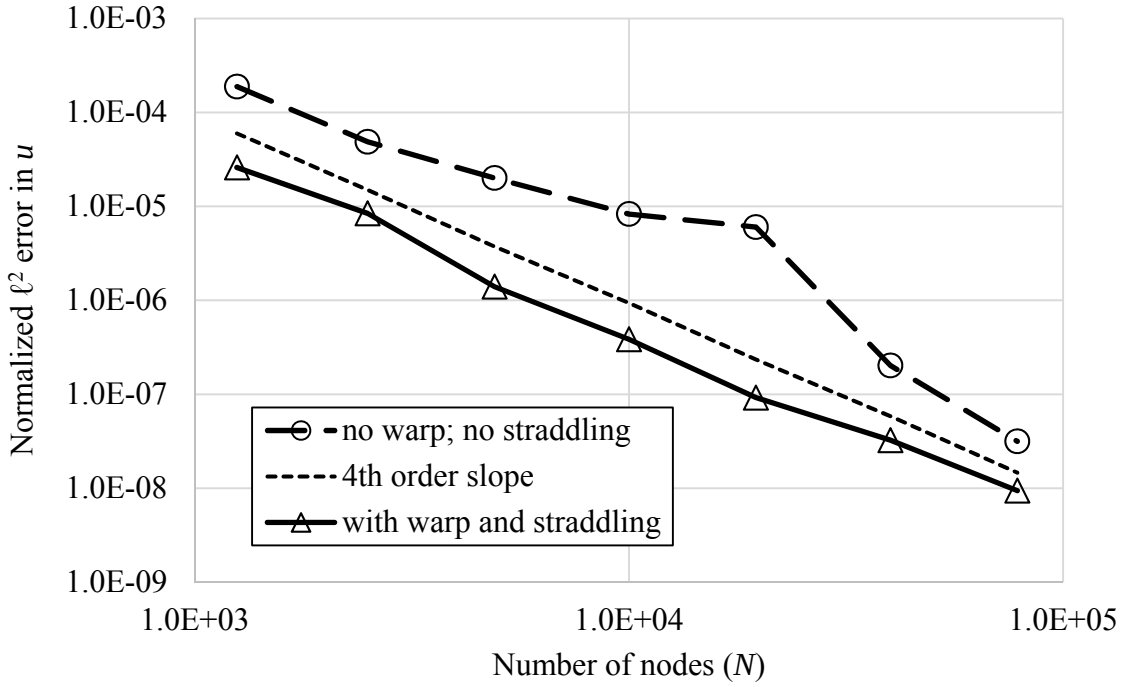

Figure 11. Errors of two different RBF-FD techniques for solving test case 2. It generally helps solution quality and stability to both distribute nodes carefully about the interface and using RBFs that are modified to uphold continuity of temperature and heat flux to 1st order. The lowest curve here ("with warp and straddling") is the same as in Figure 10.

3.3 *Test case 3: A very thin, highly insulating layer surrounds a central cooling region*

The final test case examines RBF-FD performance when applied to a much more extreme scenario of interface contrasts and domain feature scale in the x-periodic unit square. Figure 12 shows both a view of the entire domain and a zoomed-in perspective near the interface zone. In this test case, a very thin (0.001 units wide) layer of insulating material is present in the domain. The radius r from the center point (0.5,0.5) and heat diffusivity are defined as follows:

$$r \equiv \left[(x-0.5)^2 + (y-0.5)^2 \right]^{0.5} \tag{37}$$

$$\alpha = \begin{cases} 1/1500 + (1/3000)\sin(2\pi x)\sin(2\pi y) & \text{if } 0.349 \le r \le 0.35 \\ 1 & \text{otherwise} \end{cases} \tag{38}$$

There is also a third, circular Dirichlet boundary at $r = 0.05$ to simulate a cooling unit inside the insulating ring. The boundary conditions are:

$$u(\Omega) = \begin{cases} \sin(6\pi x), & y = 1 \\ \sin(6\pi x), & y = 0 \\ 0 & r = 0.05 \end{cases} \tag{39}$$

To generate polynomial basis functions for stencils that cross both interfaces defining the very thin insulating ring, one can choose a particular point (and region) within the stencil as the center of the standard basis set of 2-D polynomials up through the desired degree. The polynomials can then be "translated" into the appropriate shapes (other polynomials) through inversion of the appropriate continuity matrices each time an interface is crossed (for collocation at other points in the stencil that lie across one or both interfaces).

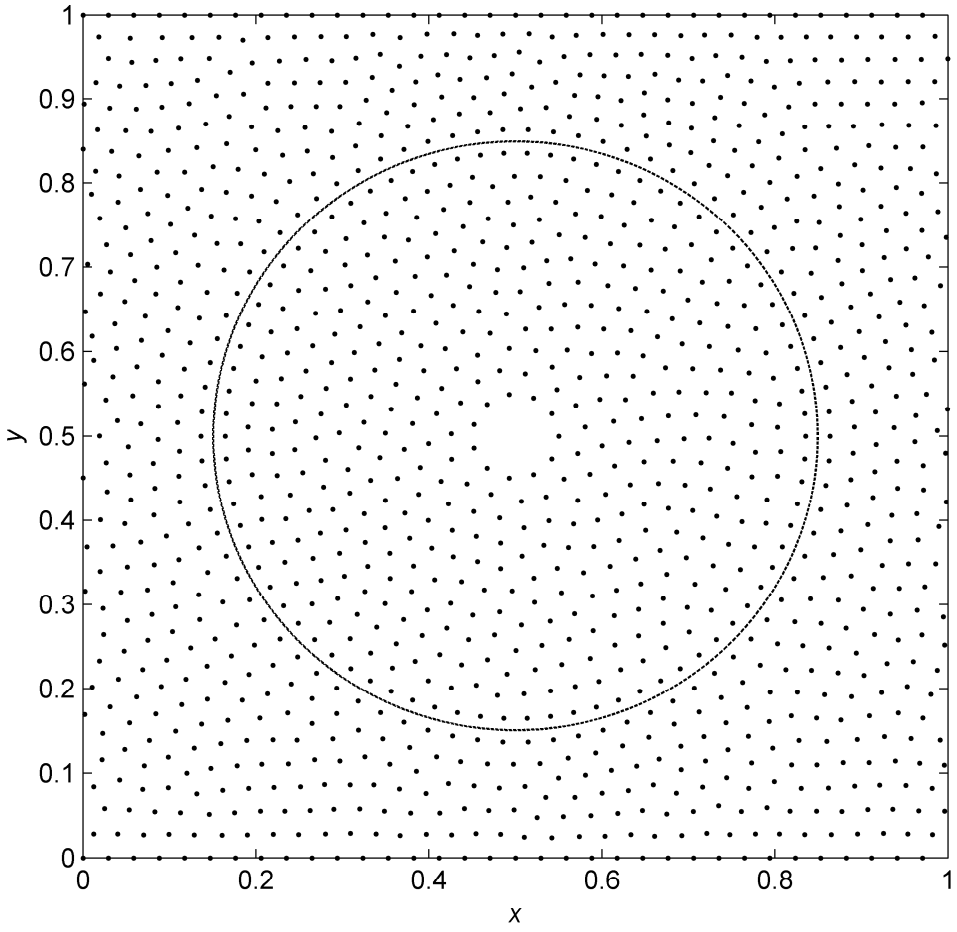

Figure 12a

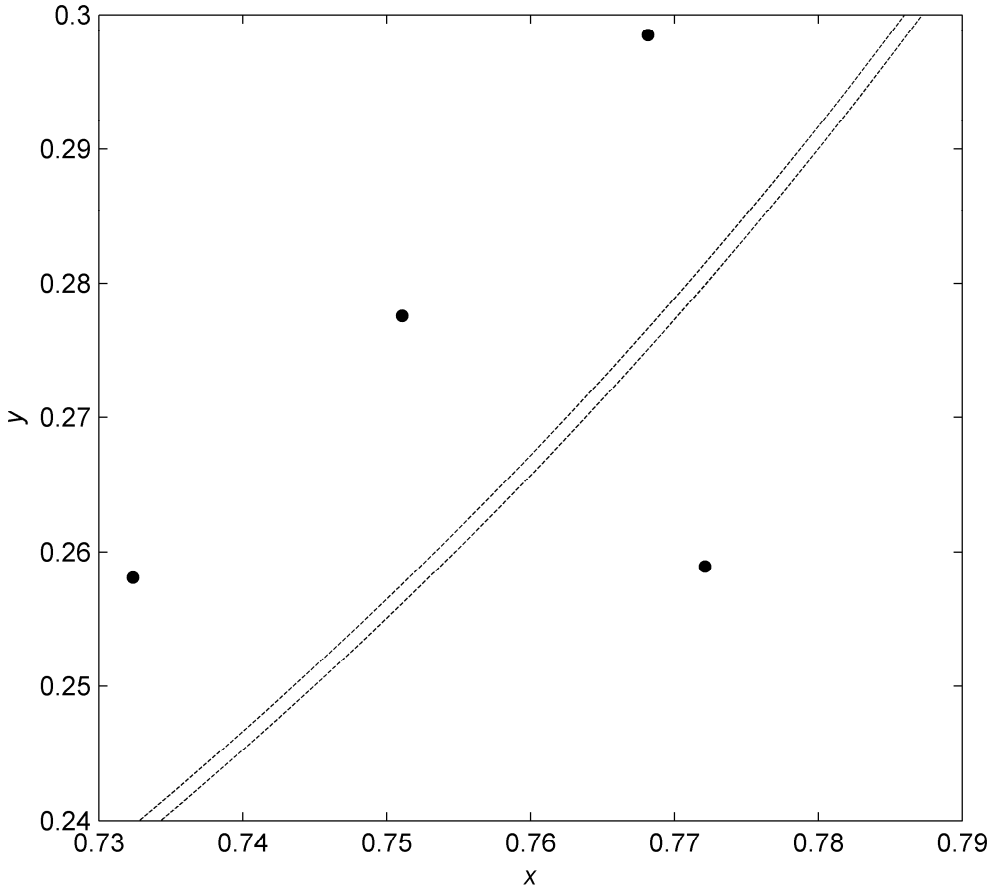

Figure 12b

Figure 12. Test domain for the third 2-D test problem. (12a) View of the entire domain, covered with a coarse set of RBF-FD nodes. (12b) Zoomed-in view of the insulating layer (much thinner than the node spacing).

3.3.1 Test case 3 results obtained using MATLAB's backslash operator

Figure 13 shows a mesh plot of a 40,000-node RBF-FD solution to the equilibrium problem $u_t = 0$ in the case 3 domain. As in solutions to test case 1 and 2, MATLAB's backslash operator was used to solve the problem.

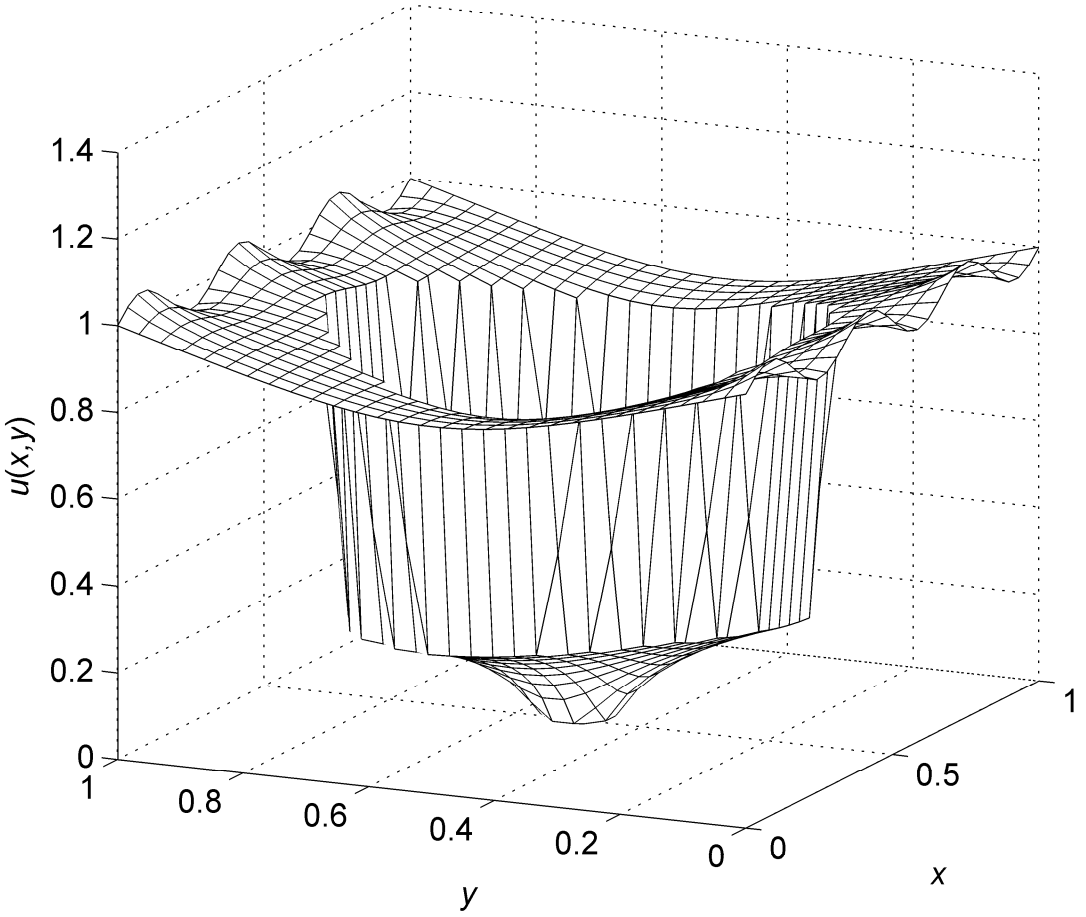

Figure 13. Mesh plot of a 40,000-node RBF-FD solution to the test case 3 problem.

Corresponding to Figure 10 for test case 2, Figure 14 compares results using a locally flat interface assumption vs. results incorporating curvature information into stencils, along with results from a traditional FD approach. Once again, inclusion of curvature information is seen to be critically important, especially as the node set becomes more and more refined. Also note how the extremely small scale of the insulating feature prevents the traditional FD4 method from even starting to converge until the node set is very refined (~1600 nodes in each Cartesian direction).

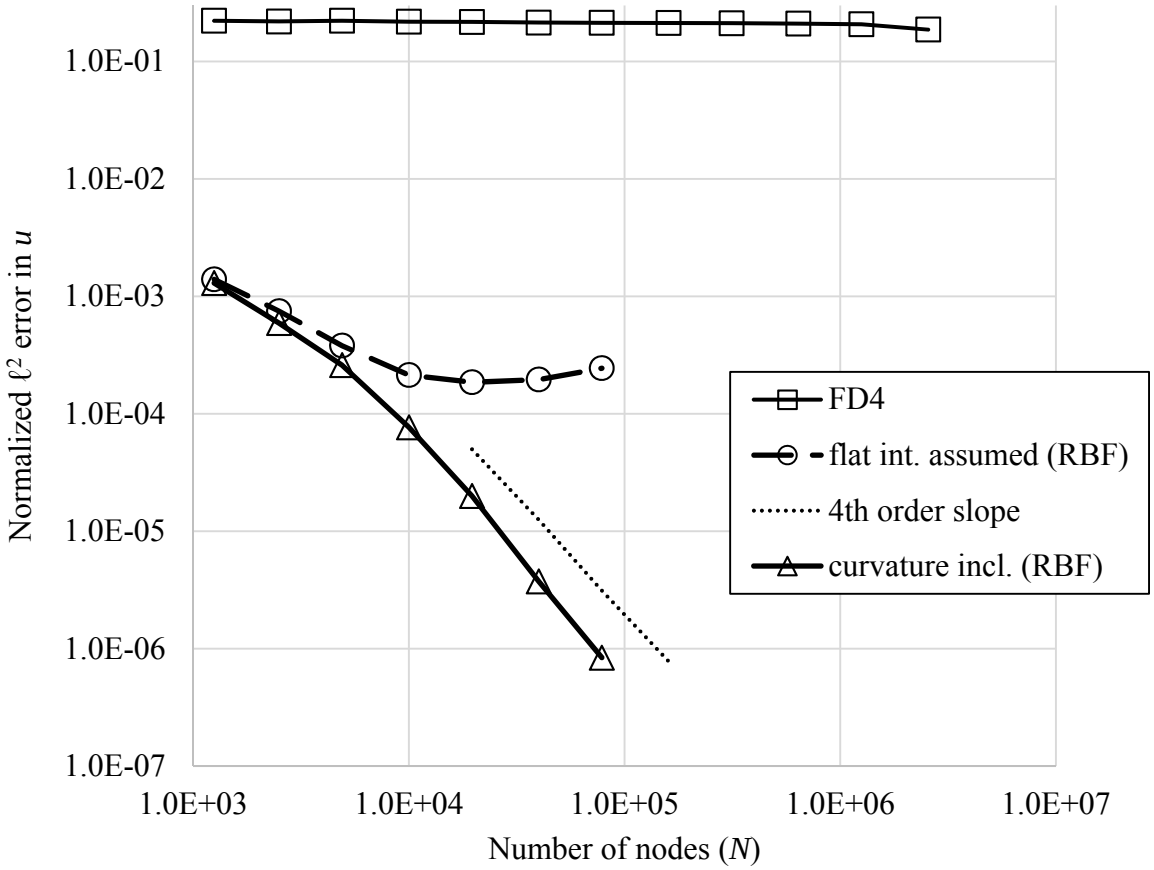

Figure 14. Errors of FD4 and two different RBF-FD techniques for solving test case 3.

3.3.2 Results from solving the test case 3 problem iteratively

Although MATLAB's backslash operator is convenient for prototyping solution methods for heat equilibrium problems in 2-D when the number of data nodes is small, it is typically not practical for use on 3-D or even large 2-D problems. It would furthermore be impossible to rely on the operator when coding in lower-level languages for solution execution on highly-parallel computing hardware.

In this section, solutions to the third test problem were obtained with MATLAB's built-in gmres and bicgstab iterative solvers in conjunction with RBF-FD spatial differentiation. Here, RBF-FD stencils contain only 19 nodes and polynomial support up through and including 3rd-degree terms. Nodes always orthogonally straddle the circular interfaces as depicted in Figure 12. Curvature information is included in all stencils. RBFs that cross either or both interfaces are modified so that they uphold continuity of heat flux to first order.

To get a baseline idea for iterative method performance on the problem, a control scenario was first designed in the same domain. The control problem has all the same boundary conditions and model parameter values as test case 3, without the thin, circular insulating layer (no interfaces are actually present, and thermal diffusivity has value 1 throughout the domain). MATLAB's backslash operator, gmres, and bicgstab functions were all used to solve this control problem. A plot of the control problem solution is shown in Figure 15, and a performance plot (error vs. computational time to solution) for the problem is presented in Figure 16.

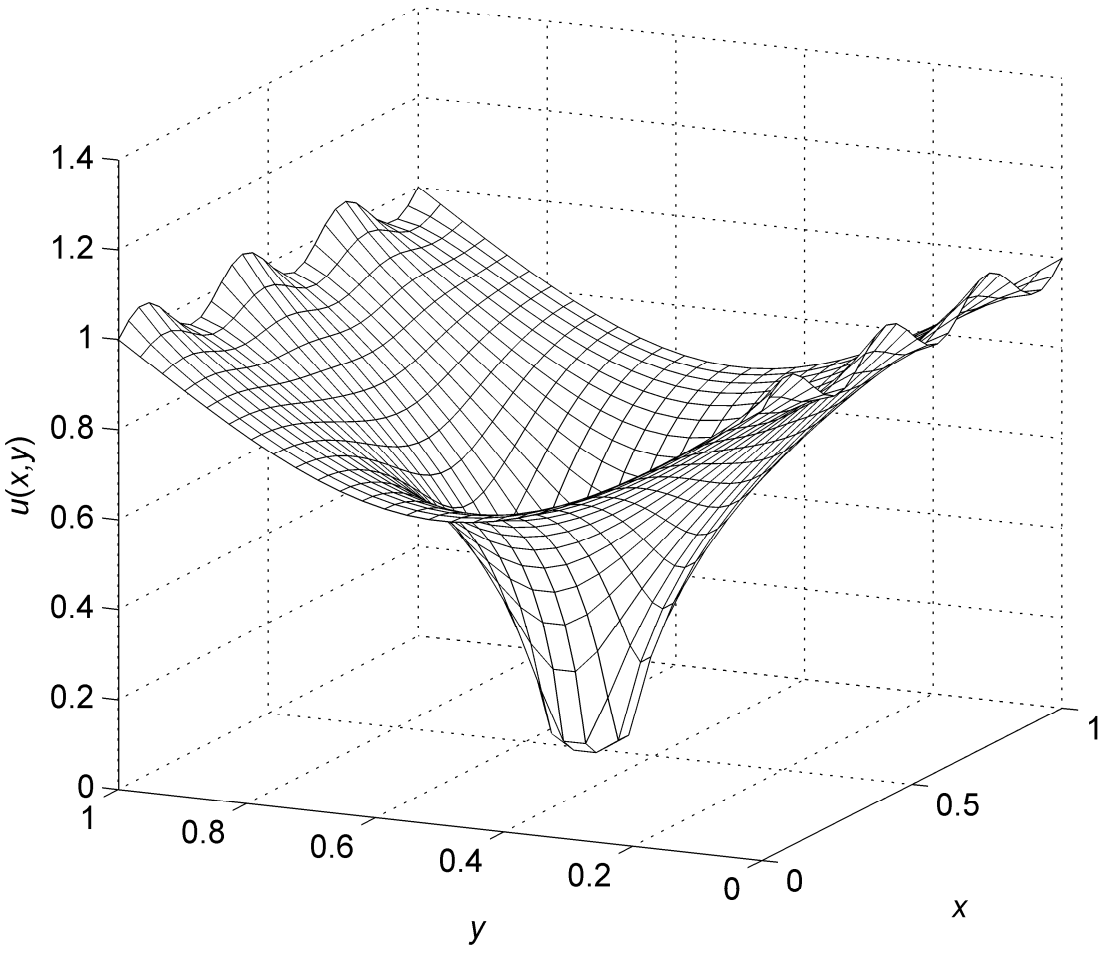

Figure 15. Test case 2 control problem solution. A 40,000-node RBF-FD solution to the problem is shown (no interfaces or insulating layers are present here).

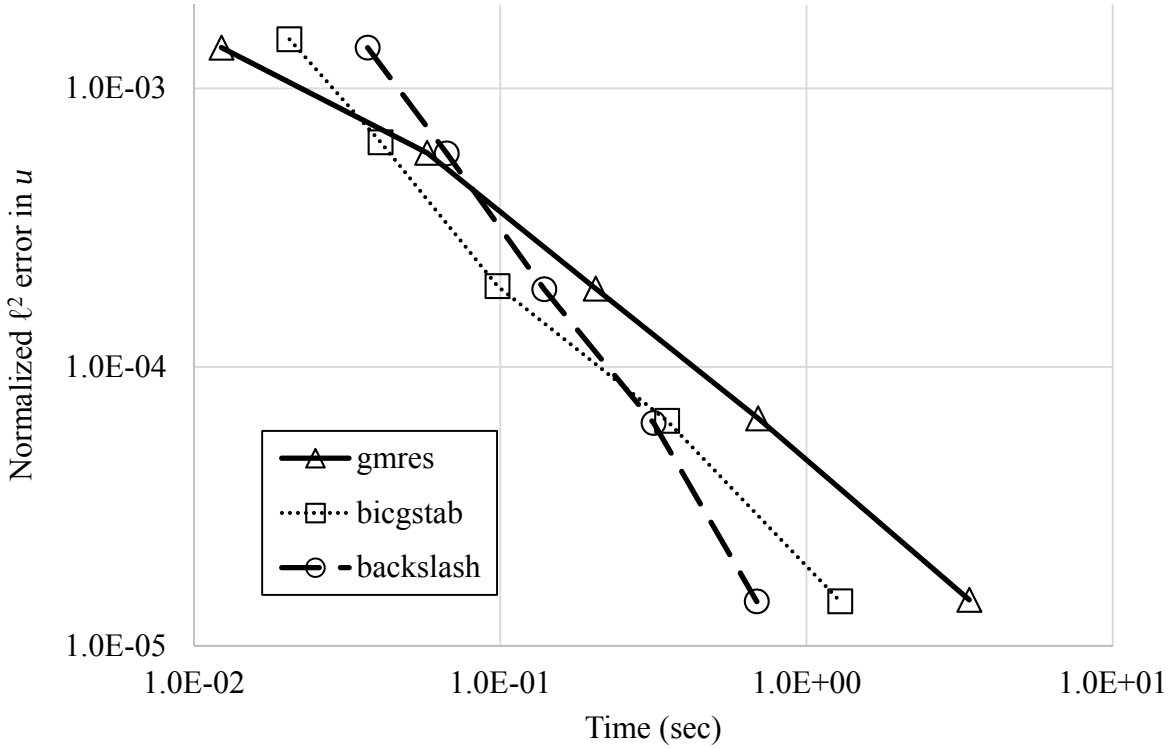

Figure 16. Performance plots for solving the control problem. MATLAB's backslash operator, gmres, and bicgstab functions were timed when applied to the test case 2 control problem (no interfaces are actually present in the domain).

From Figure 16, it is apparent that when no interfaces are present in the domain, the iterative solvers perform quite well relative to the backslash operator.

Next, these iterative methods were applied to the actual test case 3 problem (with interfaces). Figure 17 shows errors in gmres and backslash operator solutions to the second test problem plotted vs. the computational time taken to arrive at a solution, with the RBF-FD-based differential operator created as described above. Bicgstab data are not plotted here, because the solver completely failed to work on the problem. And although the gmres method did indeed converge to a solution, it did so at a rate far slower than it accomplished in the control problem.

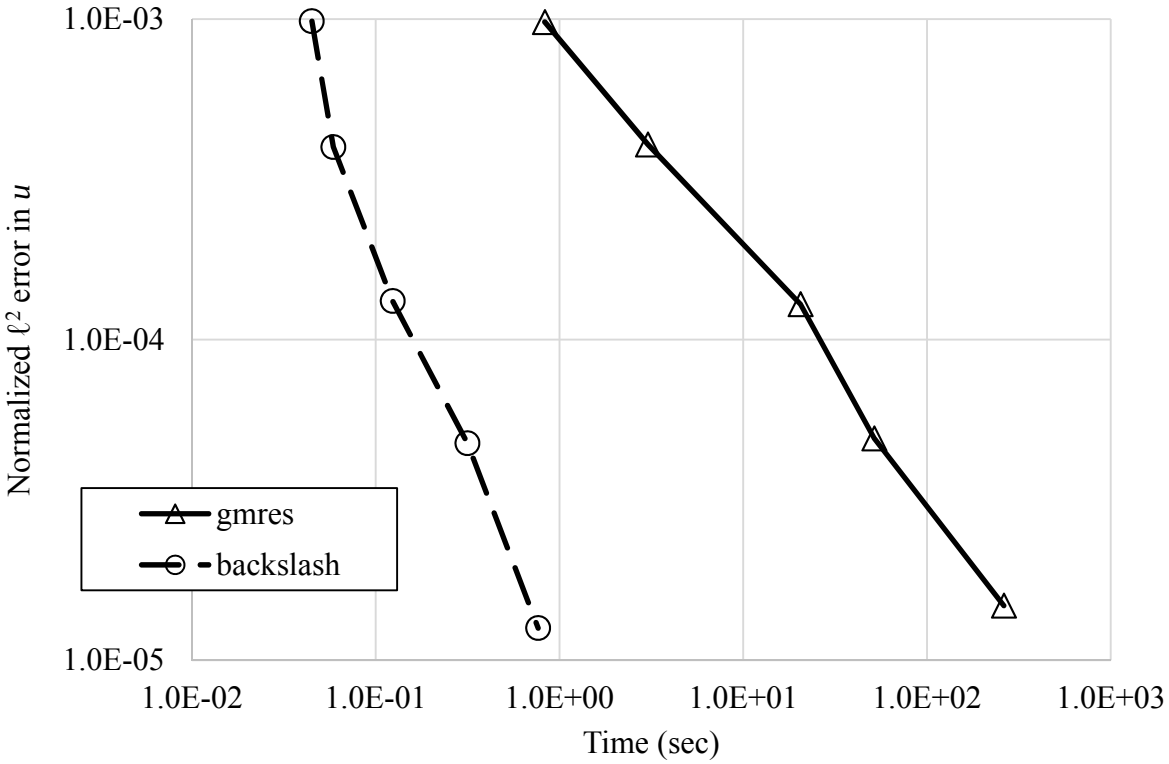

Figure 17. Performance plot of iterative methods. The plots displayed are errors vs. elapsed computational time of MATLAB's backslash operator and gmres function when applied to the test case 2 problem.

While we do not understand exactly why both iterative methods to converge efficiently to a solution in the presence of a strong interface, the source of the problem seems to be algebraic in nature (with the altered collocation weights disrupting matrix properties such as diagonal dominance that usually lead to an efficient iterative solution of elliptic problems). However, if even a crude preconditioner is applied to the problem (details of the preconditioning method can be found in [29]), almost all of the performance of gmres and bicgstab may be restored (as

compared to their solutions to a problem with no interface). This is shown by the performance plot in Figure 18.

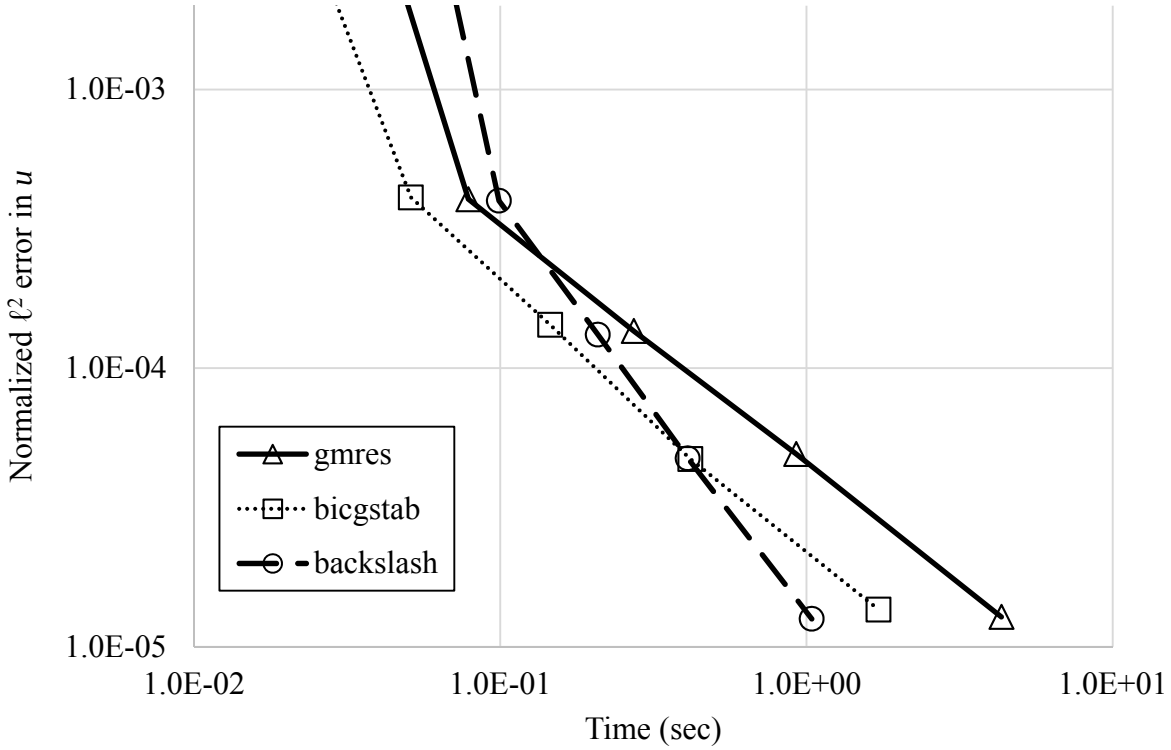

Figure 18. Performance plots, after preconditioning. Performance plot of MATLAB's backslash operator, gmres, and bicgstab functions when applied to the test case 3 problem after preconditioning has been completed.

3.3.3 Examining robustness in the case of extreme interfaces

As a test of the method's applicability under the extremely high diffusivity contrasts that may occur in real-world engineering applications, we conclude with a series of modifications to test case 3. These test cases are derived by adding a scaling parameter s to the definition of thermal diffusivity in the domain:

$$\alpha = \begin{cases} 1/(1.5s) + (1/(3s))\sin(2\pi x)\sin(2\pi y) & \text{if } (0.35 - 1/s) \le r \le 0.35 \\ 1 & \text{otherwise} \end{cases} \quad (40)$$

All other problem specifications were identical to test case 3. The derivative cases here were begun with an s value of 10^3, replicating exactly the conditions found in test case 3. From here, we multiplied s by several factors of 10, each time solving the resulting equilibrium problem at a variety of resolutions. This was carried all the way to an s value of 10^{11}, where ill-conditioning of the continuity relationships across interfaces caused a breakdown of solution convergence. Qualitatively, the solutions appear quite like those of the initial test case 3; here, decreasing thermal diffusivity in the insulating layer is balanced by a commensurate decrease in the thickness of that layer. For instance, solutions with an s value of 10^9 take place in a domain whose circular insulating layer is only a billionth of a unit thick.

Figure 19 shows a convergence plot of solutions featuring a wide range of s values. These solutions seem to perform almost as well, if not identically, to the original case 3 problem until s reaches a value of around 10^8-10^9, with more marked failure occurring around $s = 10^{11}$.

Average condition number for the continuity matrices associated with extreme values of thermal diffusivity in the domain (in the thin insulating layer) is plotted vs. s in Figure 20. Given the data here, it is unsurprising that solutions fail to perform well at extremely high diffusivity contrasts. In such scenarios, it may be valuable to find ways to eliminate or improve the conditioning difficulties encountered here, or to avoid the problem entirely by assuming $\alpha = 0$ (treating the insulating layer as a boundary rather than an interface). A number of factors may determine whether the latter approach is acceptable, including accuracy desired from solutions,

precise geometry, scale, and extremity of an insulator, magnitude of temperature gradients expected in the solution, etc.

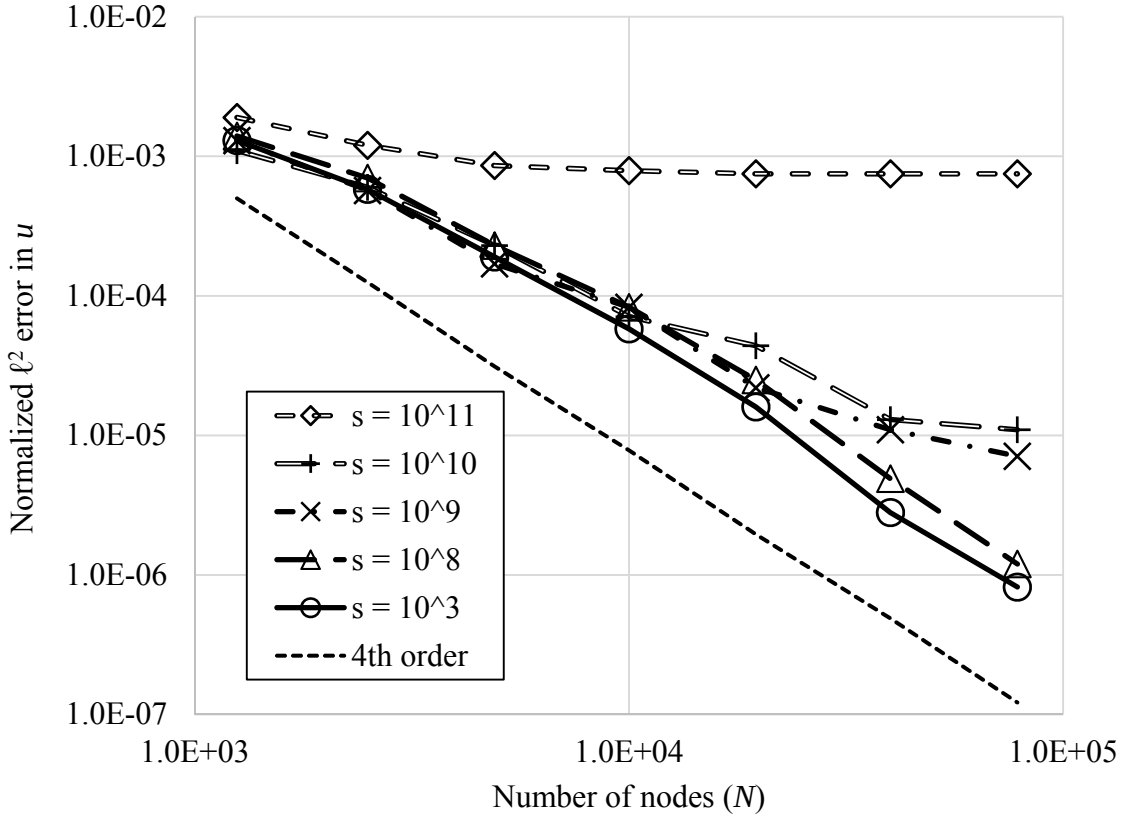

Figure 19. Errors of RBF-FD solutions to test case 3 at various values of extremizing parameter s.

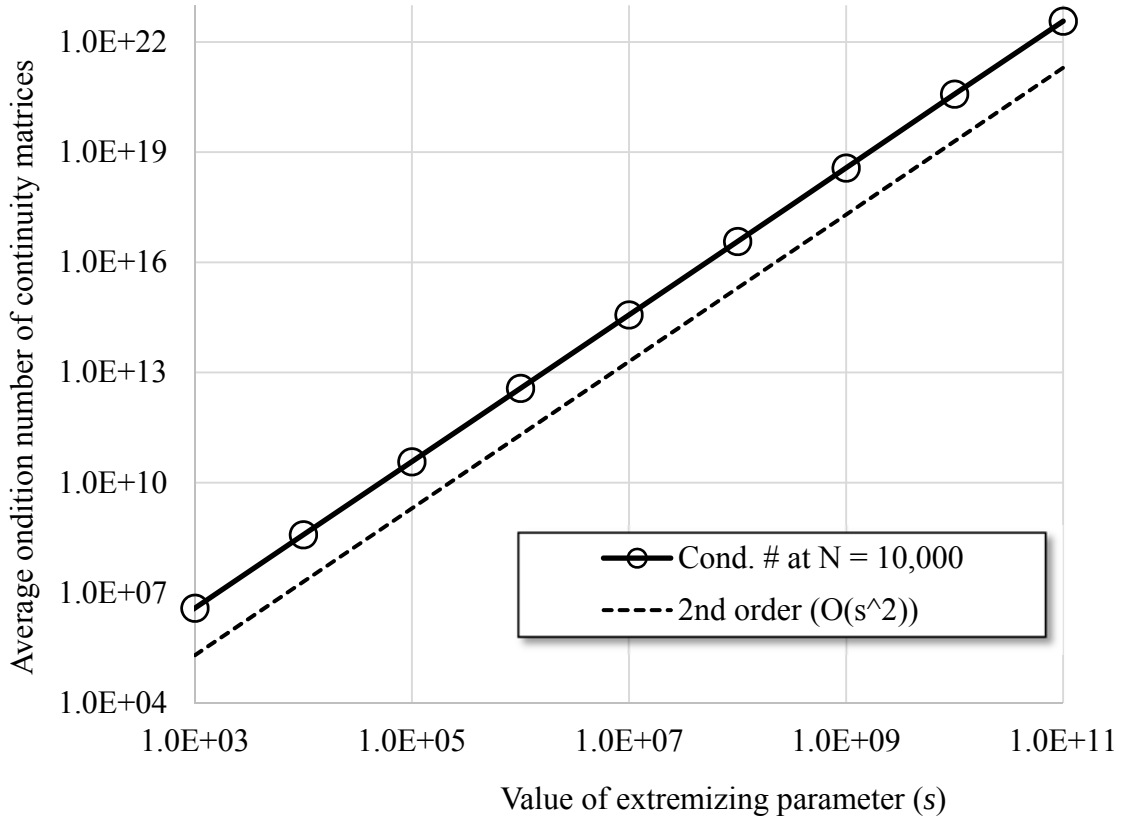

Figure 20. Condition number of continuity matrices vs. extremizing parameter s, with resolution fixed (N = 10,000 nodes in the domain).

4. CONCLUSIONS

We presented and tested an RBF-FD-based method for heat equilibrium modeling that can achieve fourth-order convergence even when stencils cross one or more curved interfaces. To the best of our knowledge, it is the only current method that makes use of a single piecewise

polynomial basis near one or more curved interface(s), where polynomial data is explicitly "translated" into other polynomial data as each curved interface is crossed. This translation is achieved through the inversion of small matrices, and ensures that the desired continuity conditions are upheld across the interface(s) to the desired order of accuracy.

Issues that require further study include:

- How to design a set of basis functions near corners and cusps (in a non-smooth interface) that can help guarantee high-order convergence to a solution
- Methods for and value of modifying RBFs further at interfaces (beyond a 1^{st}-order correction) to increase accuracy and stability of solutions
- Comparing the computational simplicity and efficiency with that of other methods, when applied to practical 3-D problems
- Determination of scenarios in which preconditioning will be necessary for efficient operation of iterative solvers in the presence of high-contrast interfaces
- Strategies for understanding and mitigating ill-conditioning that may result in the case of extreme parameter contrasts at interfaces

ACKNOWLEDGEMENTS

We thank the Graduate School of the University of Colorado at Boulder for financial support during final preparation of the manuscript. We also thank Natasha Flyer (National Center for

Atmospheric Research) and the rest of the RBF research group at CU-Boulder for frequent and very helpful feedback on the mathematical techniques in this paper.

REFERENCES

[1] I. Babuska, The finite element method for elliptic equations with discontinuous coefficients, Computing. 5 (1970) 207–213. doi:10.1007/bf02248021.

[2] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271. doi:10.1016/0021-9991(72)90065-4.

[3] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (1984) 285–299. doi:10.1137/0721021.

[4] J.H. Bramble, J.T. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math. 6 (1996) 109–138. doi:10.1007/bf02127700.

[5] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202. doi:10.1007/s002110050336.

[6] T. Lin, Q. Yang, X. Zhang, Partially penalized immersed finite element methods for parabolic interface problems, (2015).

[7] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044. doi:10.1137/0731054.

[8] Z. Li, A. Mayo, ADI method for heat equations with discontinuities along an arbitrary interface, in: Proc. Symp. Appl. Math., AMS, 1994: pp. 311–315.

[9] Z. Li, Y.-Q. Shen, A numerical method for solving heat equations involving interfaces, in: Electronic Journal of Differential Equations, 1999.

[10] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827–862. doi:10.1137/s0036142997328664.

[11] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192. doi:10.1016/j.jcp.2004.09.017.

[12] B. Fornberg, R. Meyer-Spasche, A finite difference procedure for a class of free boundary problems, J. Comput. Phys. 102 (1992) 72–77. doi:10.1016/s0021-9991(05)80006-3.

[13] B. Fornberg, A finite difference method for free boundary problems, J. Comput. Appl. Math. 233 (2010) 2831–2840.

[14] H.V.R. Mittal, J.C. Kalita, R.K. Ray, A class of finite difference schemes for interface problems with an HOC approach, Int. J. Numer. Methods Fluids. (2016). doi:10.1002/fld.4231.

[15] K. Xia, M. Zhan, G.-W. Wei, MIB Galerkin method for elliptic interface problems, J. Comput. Appl. Math. 272 (2014) 195–220. doi:10.1016/j.cam.2014.05.014.

[16] K. Ito, Z. Li, Y. Kyei, Higher-order, Cartesian grid based finite difference schemes for elliptic equations on irregular domains, SIAM J. Sci. Comput. 27 (2005) 346–367. doi:10.1137/03060120x.

[17] Y. Epshteyn, M. Medvinsky, On the solution of the elliptic interface problems by difference potentials method, in: Lect. Notes Comput. Sci. Eng., 2015: pp. 197–205. http://dx.doi.org/10.1007/978-3-319-19800-2_16 (accessed May 11, 2016).

[18] B.. Jovanovic, L.G. Vulkov, Finite difference approximation of an elliptic interface problem with variable coefficients, in: Lect. Notes Comput. Sci., 2005: pp. 46–55. http://dx.doi.org/10.1007/978-3-540-31852-1_5 (accessed May 11, 2016).

[19] B. Fornberg, N. Flyer, A Primer on Radial Basis Functions with Applications to the Geosciences, SIAM, Philadelphia, 2015.

[20] S.Y. Reutskiy, A meshless radial basis function method for 2D steady-state heat conduction problems in anisotroptic and inhomogeneous media, Eng. Anal. Bound. Elem. 66 (2016) 1–11. doi:10.1016/j.enganabound.2016.01.013.

[21] V. Bayona, N. Flyer, B. Fornberg, G.A. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, J. Comput. Phys. 332 (n.d.) 257–273. doi:10.1016/j.jcp.2016.12.008.

[22] B. Martin, B. Fornberg, A. St-Cyr, Seismic modeling with radial-basis-function-generated finite differences, Geophysics. 80 (2015) T137–T146. doi:10.1190/geo2014-0492.1.

[23] B. Martin, B. Fornberg, Seismic modeling with RBF-FD - a simplified treatment of interfaces, J. Comput. Phys. (2016).

[24] B. Fornberg, The pseudospectral method: Accurate representation of interfaces in elastic wave calculations, Geophysics. 53 (1988) 625–637. doi:10.1190/1.1442497.

[25] B. Fornberg, N. Flyer, Solving PDEs with radial basis functions, Acta Numer. 24 (2015) 215–258. doi:10.1017/S0962492914000130.

[26] N. Flyer, B. Fornberg, V. Bayona, G.A. Barnett, On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy, J. Comput. Phys. 321 (2016) 21–38. doi:10.1016/j.jcp.2016.05.026.

[27] R.E. Carlson, T.A. Foley, The parameter R^2 in multiquadric interpolation, Comput. Math. Appl. 21 (1991) 29–42. doi:10.1016/0898-1221(91)90123-l.

[28] B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, Comput. Math. Appl. 54 (2007) 379–398. doi:10.1016/j.camwa.2007.01.028.

[29] B. Martin, Application of RBF-FD to wave and heat transport problems in domains with interfaces, Ph.D. thesis, University of Colorado, Boulder, 2016. http://www.bradleypmartin.com (accessed July 24, 2016).