# A New Backus-Gilbert Meshless Method for Initial-Boundary Value Partial Differential Equations

Christopher D. Blakely[1]

**Abstract:** The purpose of this paper is to introduce a Backus-Gilbert approximation method as a tool for numerically solving initial-boundary value problems. The formulation of the method with its connection to the standard moving least-squares formulation will be given along with some numerical examples including a numerical solution to the viscous nonlinear Burgers equation in two dimensions. In addition, we highlight some of the main advantages of the method over previous numerical methods based on meshless collocation approximation in order to validate its robust approximating power and easy handling of initial-boundary value problems.

*Keywords:* Meshless Approximation, Moving Least-Squares, Radial Basis Functions, Numerical PDEs

## 1 Introduction

As with other moving least-squares techniques for solving partial differential equations such as the ones found in Belytschko, Lu, and Gu [5], and Fasshauer [9], one of the main advantages comes from the fact that no computational mesh of the domain is needed. This leads to an easy implementation of the method on domains which have complex geometries where a mesh as constructed in finite-element type methods can be computationally tedious and non-trivial to generate.

While meshless methods have grown popular for dealing with multivariate approximation problems, extending these methods for their use in solving partial differential equations has led to a variety of different techniques typically based on either a Galerkin formulation where numerical integration is used, namely Element-free Galerkin (e.g. [2], [4]) or a collocation technique (e.g. [13], [9]). The Backus-Gilbert approximation method discussed in this paper is based on collocation and thus no numerical integration is required.

The first meshless collocation method applied to numerically solving partial differential equations was developed by Kansa in [13] where direct interpolation using translations and

[1]University of Maryland, Center for Scientific Computation and Mathematical Modeling, cblakely@cscamm.umd.edu.

dialations of radial basis functions, notably multiquadric functions, approximated the solution and its derivatives. Thereafter, successful applications of the method to elliptic and time-dependent problems such as the shallow-water equations began appearing. However, as studied in many papers such as [22], the condition numbers associated with the interpolation matrices are extremely large and grow as the amount of collocation nodes in the domain increased. Furthermore, when using globally supported radial basis functions such as the multiquadric function used in [14] which possess the best approximation results as opposed to compactly supported radial functions, the shape parameter $\beta$ controlling the dialation of the function has been demonstrated to be extremely sensitive to the multivariate scattered data interpolation in terms of convergence of the approximation. Finding an optimal or near optimal shape parameter for a certain problem can so far only be accomplished numerically, namely no *a priori* knowledge of an optimal $\beta$ which minimizes the error in any norm can be found analytically. This is a severe hinderance to the method since finding an optimal shape parameter for very large problems where node counts reach an order of $\mathcal{O}(10^5)$ can be computationally infeasable.

In [9], a moving least-squares approximation for time-dependant PDEs was proposed based on *approximate approximation*. In this method, a Backus-Gilbert MLS formulation constructs a reproducing kernel generated by a reproducing space taken as polynomials shifted on a set of collocation nodes by the evaluation points defined on the domain and boundary while symmetric weights are taken as radial functions. This approach to the Backus-Gilbert MLS formulation via translated polynomials offers fast construction of the reproducing kernel since the Lagrange multipliers that are needed can be found explicitly for a polynomial basis of low order. In effect, it is a fast way to achieve the robust approximation power of moving least-squares approximation, but at the cost of not being equipped with the flexibility of choosing an approximation space for constructing the reproducing kernel. The translated polynomial basis must be used rendering the method inadmissible for problems in which discontinuities are known to occur in the solution.

Deducing from the disadvantages of the past two meshless methods, it is of high interest to construct a meshless method equipped with the following properties:

- The method is flexible when it comes to choosing an approximation space.
- Collocation matrices of the basis functions from the approximation space are well-conditioned.
- The shape parameter is flexible for any problem in that it offers large intervals of near optimal choices.
- The method yields high-convergence rates for smooth problems.

In this paper, it is our goal to present a new meshless collocation method based on Backus-Gilbert moving least squares where we strive to obtain the desired properties described above. In order to do this, we will first describe the construction of the Backus-Glibert approximation method in section 2 followed by a brief discussion on how to adapt the method to numerically solve initial-boundary valued partial differential equations. Section 3 will describe a new numerical scheme based on Backus-Gilbert followed by some numerical examples involving a

2D boundary-value elliptic problem and a viscous 2-D nonlinear Burgers equation. Finally, to conclude the paper, a discussion on the future research and application interests of this meshless collocation method will be provided.

# 2 Formulation

## 2.1 Backus-Gilbert Formulation

Being based on moving least squares approximation, this approximation scheme begins with the Backus-Gilbert formulation. In depth descriptions of this formulation can be found in [6] and in references therein. This approach for MLS approximation considers a quasi-interpolant of the form

$$Pf(\mathbf{x}) = \sum_{i=1}^{N} f(\mathbf{x}_i)\Psi_i(\mathbf{x}). \tag{1}$$

where $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_N)]^T$ represents the given data on a set of $N$ distinct evaluation nodes $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \Omega \cup \partial\Omega \subset \mathbb{R}^n$. The quasi-interpolant $\Psi_i(\mathbf{x})$, or discrete reproducing kernel in some literature, is constructed to be minimized in a discrete quadratic expression subject to some approximation space reproduction constraints. This is done by choosing a family of functions which are linearly independent such as a polynomial or trigonometric basis. In this presentation we will use a basis of radial functions as the approximation space, namely $U = \text{span}\{\phi(||\cdot - \xi_j||),\ \xi_j \in \Xi\}$ where $\Xi = \{\xi_1, \ldots, \xi_N\} \subset \Omega \cup \partial\Omega$ is a set of $M \leq N$ nodes which can be taken as $\mathcal{X}$ although any set of $M$ distinct nodes will work. For more detail on the techniques of radial basis approximation, the reader is referred to [11], [19], [21] and the references therein. Denoting $\phi_j(\mathbf{x}) := \phi(||\mathbf{x} - \xi_j||),\ j = 1, \ldots, M$, the approximation space reproduction constraints are written as

$$\sum_{i=0}^{N} \phi_j(\mathbf{x}_i)\Psi_i(\mathbf{x}) = \phi_j(\mathbf{x}), \quad \text{for all } \phi_j \in U. \tag{2}$$

or in matrix form as

$$A\Psi(\mathbf{x}) = \phi(\mathbf{x}),$$

with $A_{ji} = \phi_j(\mathbf{x}_i), i = 1, \ldots, N, j = 1, \ldots, M$. The objective is to minimize

$$\frac{1}{2}\sum_{i=0}^{N} \Psi_i^2(\mathbf{x})W(\mathbf{x}, \mathbf{x}_i) \tag{3}$$

using the above reproducing approximation space constraints. $W(\mathbf{x}, \mathbf{x}_i)$ is a positive weight function for any pair $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{x}_i \in \mathcal{X}$ which *moves* according to the evaluation node $\mathbf{x}$. However, when the approximation space is taken as a set of radial basis functions, one usually takes $W(\mathbf{x}, \mathbf{x}_i) = 1$.

To solve this moving-least squares problem, Lagrangian multipliers $\lambda_j(\mathbf{x}), j = 1, \ldots, M$ are introduced which leads to a system where we solve for both the Langrangian multipliers and the reproducing kernel functions. This system can be written in block matrix form as

$$\begin{bmatrix} \mathbf{Q}(\mathbf{x}) & -\mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Psi(\mathbf{x}) \\ \lambda(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \phi(\mathbf{x}) \end{bmatrix} \tag{4}$$

where $Q(\mathbf{x}) = \text{diag}\Big(W(\mathbf{x}, \mathbf{x}_1), \ldots, W(\mathbf{x}, \mathbf{x}_1)\Big)$ which is positive definite for any $\mathbf{x}$. Since we are assuming that the sets $\Xi$ and $\mathcal{X}$ are comprised of distinct nodes in the domain of interest, $\mathbf{A}$ has full rank and we can apply Gaussian elimination to the block matrix and arrive at

$$\lambda(\mathbf{x}) = G(\mathbf{x})^{-1}\phi(\mathbf{x}) \tag{5}$$

where we define $G(\mathbf{x}) = AQ^{-1}(\mathbf{x})A^T$ as the Gram matrix. The explicit form of the reproducing kernels can then be given as

$$\Psi_i(\mathbf{x}) = \sum_{j=1}^{M} \lambda_j(\mathbf{x})\phi(\mathbf{x}_i, \xi_i)\frac{1}{W(\mathbf{x}, \mathbf{x}_i)}$$

So for every evaluation node $\mathbf{x} \in \mathcal{X}$, an $M \times M$ system of equations must be solved in order to calculate $\Psi_i(\mathbf{x})$. We note here that the closeness of the generating functions $\Psi_i(\mathbf{x})$ to the pointwise cardinal functions, (i.e., $\Psi_i(\mathbf{x}_j) = \delta_{i,j}, i, j = 1, ...N$) determines how well the summation in (1) approximates the function $f(\mathbf{x})$.

## 2.2 Approximating Solutions to PDEs using Backus-Gilbert

This approximation scheme can now be adapted to numerically solve time-dependent boundary-valued partial differential equations. One way to formulate such an adaption which we present in this paper is akin to the theory of Generalized Hermite interpolation (see e.g. [17] and references therein). The idea of this adaption is to construct a reproducing kernel which reproduces a certain differential operator acting on an approximation space defined on a set of collocation nodes. As we will discuss later, this routine leads to a numerical method for solving time-dependant problems which is dependent on the time-integration scheme used.

To begin, suppose $f \in L^\infty(\mathcal{H}, [0, \infty])$ where $\mathcal{H}$ is normed function space on $\Omega \in \mathbb{R}^n$ and suppose we wish to solve the linear hyperbolic partial differential equation

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + Lu(\mathbf{x}, t) = f(\mathbf{x}, t) \;\; \mathbf{x} \in \Omega, \;\; t > 0 \tag{6}$$

with initial conditions

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \;\; \text{for} \;\; \mathbf{x} \in \Omega$$

and boundary condition

$$u(\mathbf{x}, t) = g(\mathbf{x}, t) \;\; \text{for} \;\; \mathbf{x} \in \partial\Omega$$

where $L$ is a real linear differentiable operator of order $s$. To initiate the Backus-Gilbert numerical approximation of this problem, we first discretize in time using a standard implicit method disregarding the negligable $o(\Delta t)$ time error. Here, while using the backwards Euler scheme, we require the two sides of the equation to be equal at a set of $N - K$ distinct

collocation nodes $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N-K}\} \in \Omega$. This gives, using the notation for time indices $u(\mathbf{x}_i, t_n) := u^n(\mathbf{x}_i)$

$$u^n(\mathbf{x}_j) + \Delta t L u^n(\mathbf{x}_j) = u(\mathbf{x}_j)^{n-1} + \Delta t f^n(\mathbf{x}_j) \text{ for } j = 1, \ldots, N \tag{7}$$

for any time step $t_n > 0$. Furthermore, for the boundary, we create a set of $K$ collocation nodes $\mathcal{X}_\mathcal{B} = \{\mathbf{x}_{N-K+1}, \ldots, \mathbf{x}_N\} \in \partial\Omega$ and require

$$u^n(\mathbf{x}_j) = g^n(\mathbf{x}_j) \text{ for } \mathbf{x}_j \in \mathcal{X}_\mathcal{B}.$$

In [18], a similar problem involving transport equations on a periodic domain $\mathbb{T}^n$ was considered as a Hermite interpolation problem and approximated using periodic basis functions at each time step. That is, assuming we know $u^{n-1}(\mathbf{x}_j)$, interpolation is achieved by using information about derivatives which leads to restating the discretized transport equation distributional notation as follows. At each time step, find $v \in C^s(\mathbb{T}^n)$ such that

$$\langle \delta(\mathbf{x}_j) + \Delta t L \delta(\mathbf{x}_j), v \rangle = d_j, \text{ for } \mathbf{x}_j \in \mathbb{T}^n$$

where $d_j = u^{n-1}(\mathbf{x}_i) + \Delta t f^n(\mathbf{x}_i)$ is the given data at each time step. Periodic basis functions which were defined to be continuous functions $\psi : \mathbb{T}^n \to \mathbb{R}$ such that all the coefficients in its Fourier expansion, $\psi(\mathbf{x}) = \sum_n \hat{\psi(n)} \exp in \cdot x$ ($n$ multi-index) are strictly positive were then proposed as a tool for solving the above interpolation problem. In this formulation however, we consider solving 6 as a multivariate scattered-data generalized Hermite interpolation problem by constructing quasi-interpolants which discretely reproduce differential operators acting on an approximation space defined on a set of collocation nodes. Let $L$ be the real linear differential operator defined in (6) and let $\{\phi_j(\mathbf{x})\}_{j=1}^M$ be a family of smooth linear independent functions such that $\text{span}\{\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x})\} \subset C^s(\mathbb{R}^n)$, $M \leq N$. Using the two sets of nodes $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N-K}\}$ and $\mathcal{X}_\mathcal{B} = \{\mathbf{x}_{N-K+1}, \ldots, \mathbf{x}_N\}$, the generalized Hermite interpolation problem is to find $v \in C^s(\mathbb{R}^n)$ such that for any time step t,

$$\langle \delta(\mathbf{x}_j) + \Delta t L \delta(\mathbf{x}_j), v(\mathbf{x}) \rangle = d_j, \text{ for } \mathbf{x}_j \in \mathcal{X}$$

$$\langle \delta(\mathbf{x}_j), v(\mathbf{x}) \rangle = g^n(\mathbf{x}_j), \text{ for } \mathbf{x}_j \in \mathcal{X}_\mathcal{B}$$

where $d_j$ is defined above. Using this information, we wish to construct a discrete reproducing kernel $\Psi_i(\mathbf{x}) \in C^s(\mathbb{R}^n)$ generated from the approximation space $\text{span}\{\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x})\}$ such that $\mathcal{P}v(\mathbf{x}) = \sum_{i=1}^N \Psi_i(\mathbf{x}) v(\mathbf{x}_i)$. After reproducing the initial condition known on $\mathcal{X} \cup \mathcal{X}_\mathcal{B}$ as $\mathcal{P}u(\mathbf{x}, 0) = \sum_{i=1}^N \Psi_i(\mathbf{x}) u(\mathbf{x}_i, 0)$, we can approximate the solution $u(t_n, \mathbf{x})$ at any time step $t_n > 0$ by solving

$$\langle \delta(\mathbf{x}_j) + \Delta t L \delta(\mathbf{x}_j), \sum_{i=0}^N \Psi_i(\mathbf{x}) v(\mathbf{x}_i) \rangle =$$

$$\sum_{i=1}^N \Big( \Psi_i(\mathbf{x}_j) + \Delta t L \Psi_i(\mathbf{x}_j) \Big) v(\mathbf{x}_i) = u^{n-1}(\mathbf{x}_j) + \Delta t f^n(\mathbf{x}_j), \text{ for } \mathbf{x}_j \in \mathcal{X}$$

and

$$\sum_{i=1}^N \Psi_i(\mathbf{x}_j) v(\mathbf{x}_i) = g^n(\mathbf{x}_j), \text{ for } \mathbf{x}_j \in \mathcal{X}_\mathcal{B}$$

We calculate $L\Psi_i(\mathbf{x})$ by using the construction of $\Psi_i(\mathbf{x})$. This gives

$$L\Psi_i(\mathbf{x}) = \sum_{j=1}^{M} L\Big(\lambda_j(\mathbf{x})\frac{1}{W(\mathbf{x},\mathbf{x}_i)}\Big)\phi_j(\mathbf{x}_i)$$

However, using an approximation space $\mathcal{U} = \text{span}\{\phi_j(\mathbf{x})\}$ of radial basis functions $\phi_j(\mathbf{x}) \equiv \phi(||\mathbf{x}-\xi_j||,\beta)$ where $\beta$ is the appropriate shape parameter, one usually takes $W(\mathbf{x},\mathbf{x}_i) \equiv 1$.

After solving this system of equations at each time step for $v(\mathbf{x}_i)$, we set $u^n(\mathbf{x}_i) := v(\mathbf{x}_i)$ From this system of equations, it is apparent that the quasi-interpolants $\Psi_i(\mathbf{x}_j)$ must be constructed to belong to $C^s(\mathbb{R}^n)$ when assuming $u(\mathbf{x},t)$ is a smooth solution for $t > 0$ on $\Omega$. Indeed, this property is dependent on the smoothness of the approximation space $\mathcal{U} = \text{span}\{\phi_1(\mathbf{x}),\ldots,\phi_N(\mathbf{x})\}$ from which we construct the quasi-interpolants.

# 3 A New Backus-Gilbert Approximation Technique

## 3.1 Construction

In the Backus-Gilbert formulation described in the previous section when constructing the discrete reproducing kernel, a linear system of size $M \times M$ needed to be solved at each evaluation point in the domain. This can be a computational burden for approximation spaces of large dimensions. In this section however, we construct a Backus-Gilbert approximation technique that will avoid calculating solutions to linear systems at each evalutation node.

To begin, we consider the approximation space $\mathcal{U} = \text{span}\{\phi_j(\cdot),\ j = 1,\ldots,M\}$ which was used to provide the reproduction constraints in the Backus-Gilbert formulation. Furthermore, we set the weight function $W(\cdot,\mathbf{x}_i) \equiv 1$. Suppose we have $N$ evaluation nodes in the domain of interest $\Omega \cup \partial\Omega$ which can either be uniformly or randomly distributed. We can thus write the $N \times M$ interpolation matrix based on these evaluation nodes as

$$\mathbf{A} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & & & \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{pmatrix} \tag{8}$$

Using this matrix, consider the resulting $Q$ matrix from a $QR$ decompositon algorithm, namely $A = QR$. A new basis $\{\tilde{\phi}_j(\cdot)\}_{j=1}^{M}$ from the rows of $Q$ is provided which is orthonormal with respect to the evalutation nodes in the sense that

$$\langle\tilde{\phi}_j,\tilde{\phi}_k\rangle = \sum_{i=1}^{N}\tilde{\phi}_j(\mathbf{x}_i)\tilde{\phi}_k(\mathbf{x}_i) = \delta_{j,k}$$

Going back to the Backus-Gilbert formulation using this new basis, we have the reproduction constraints written as

$$\begin{pmatrix} \langle\tilde{\phi}_1,\tilde{\phi}_1\rangle & \langle\tilde{\phi}_1,\tilde{\phi}_2\rangle & \cdots & \langle\tilde{\phi}_1,\tilde{\phi}_M\rangle \\ \langle\tilde{\phi}_2,\tilde{\phi}_1\rangle & \langle\tilde{\phi}_2,\tilde{\phi}_2\rangle & \cdots & \langle\tilde{\phi}_2,\tilde{\phi}_M\rangle \\ \vdots & & & \\ \langle\tilde{\phi}_M,\tilde{\phi}_1\rangle & \langle\tilde{\phi}_M,\tilde{\phi}_2\rangle & \cdots & \langle\tilde{\phi}_M,\tilde{\phi}_M\rangle \end{pmatrix} \begin{pmatrix} \lambda_1(\mathbf{x}) \\ \lambda_2(\mathbf{x}) \\ \vdots \\ \lambda_M(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \tilde{\phi}_1(\mathbf{x}) \\ \tilde{\phi}_2(\mathbf{x}) \\ \vdots \\ \tilde{\phi}_M(\mathbf{x}) \end{pmatrix}. \tag{9}$$

But since $\langle \tilde{\phi}_j, \tilde{\phi}_k \rangle = \delta_{j,k}$ this means that $\lambda_j(\mathbf{x}) = \tilde{\phi}_j(\mathbf{x})$ for $j = 1, \ldots, M$ and so the Backus-Gilbert formulation yields a discrete reproducing kernel

$$\Psi_i(\mathbf{x}_k) = \sum_{j=1}^{M} \lambda_j(\mathbf{x}_k)\tilde{\phi}_j(\mathbf{x}_i) = \sum_{j=1}^{M} \tilde{\phi}(||\mathbf{x}_k - \xi_j||)\tilde{\phi}(||\mathbf{x}_i - \xi_j||).$$

Unlike with the original Backus-Glibert MLS approximation, no linear systems need to be solved at each evaluation node in order to produce the approximation. Instead, for a given radial basis approximation space of dimension $M$ in $\mathbb{R}^n$, $\mathcal{U} = \text{span}\{\phi(||\mathbf{x} - \xi_j||);\ \xi_j \in \Xi, j = 1, \ldots, N\}$, and a given set of $N$ evaluation nodes $\mathcal{X} \subset \Omega \cup \partial\Omega$, an orthonormal basis is constructed with respect to $\mathcal{X}$ via QR decomposition of the matrix defined in 8.

Because of this construction, this kernel function forms a *partition of unity* in the sense that $\sum_{i=1}^{N} \Psi_i(\mathbf{x}_k) = 1$ for $\mathbf{x}_k \in \mathcal{X}$. To see this, we use the definition of the kernel function and write for some $\mathbf{x}_k \in \mathcal{X}$

$$\begin{aligned}
\sum_{i=1}^{N} \Psi_i(\mathbf{x}_k) &= \sum_{i=0}^{N}\sum_{j=0}^{M} \lambda_j(\mathbf{x}_k)\tilde{\phi}_j(\mathbf{x}_i) && (10)\\
&= \sum_{i=1}^{N}\sum_{j=1}^{M} \tilde{\phi}_j(\mathbf{x}_k)\tilde{\phi}_j(\mathbf{x}_i) && (11)\\
&= \sum_{i=1}^{N}\Big(\sum_{j=1}^{M} \tilde{\phi}(||\mathbf{x}_k - \xi_j||)\tilde{\phi}(||\mathbf{x}_i - \xi_j||)\Big) && (12)\\
&= \sum_{i=1}^{N} \delta_{i,k} = 1 && (13)
\end{aligned}$$

We used the fact that the QR decomposition of the $N \times N$ matrix $A$ produces a $Q$ which has orthonormal rows and columns.

Restating the Backus-Gilbert formulation with respect to the constructed orthonormal basis $\{\tilde{\phi}_j(\cdot)\}_{j=1}^{M}$, for given data $\mathbf{f} = \{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)\}$, $\mathbf{x}_i \in \mathcal{X}$, the approximation using the kernel functions $\Psi_i(\mathbf{x}_k)$ is given as

$$\mathcal{P}f(\mathbf{x}_k) = \sum_{i=1}^{N} f(\mathbf{x}_i)\Psi_i(\mathbf{x}_k)$$

where

$$\Psi_i(\mathbf{x}_k) = \sum_{j=1}^{M} \tilde{\phi}(||\mathbf{x}_k - \xi_j||)\tilde{\phi}(||\mathbf{x}_i - \xi_j||)$$

## 3.2 Approximating Linear Differential Operators

We now discuss the problem of approximating a linear differential operator $L$ acting on the given data $\mathbf{f} = \{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)\}$ via MLS with the constructed orthonormal basis. In the standard Backus-Gilbert formulation using the radial basis approximation space $\mathcal{U} =$

$\text{span}\{\phi(||\mathbf{x}-\xi_j||);\ \xi_j \in \Xi, j=1,\ldots,M\}$, the approximation of the linear operator $L$ acting on the given data $\mathbf{f}$ is made simply by calculating for $\mathbf{x} \in \mathcal{X}$

$$\begin{aligned} \mathcal{P}Lf(\mathbf{x}) &= \sum_{i=1}^{N} f(\mathbf{x}_i) L\Psi_i(\mathbf{x}) && (14) \\ &= \sum_{i=1}^{N} f(\mathbf{x}_i) \sum_{j=1}^{M} L\lambda_j(\mathbf{x})\phi_j(\mathbf{x}_i) && (15) \end{aligned}$$

Denoting $\lambda(\mathbf{x}) = [\lambda_1(\mathbf{x}),\ldots,\lambda_N(\mathbf{x})]^T$, the $L\lambda_j(\mathbf{x})$'s are calculated as

$$L\lambda(\mathbf{x}) = G^{-1}L\phi(\mathbf{x})$$

where $L\phi(\mathbf{x}) = [L\phi_1(\mathbf{x}),\ldots,L\phi_M(\mathbf{x})]^T$ and $G = AA^T$ is the Gram matrix.

The problem we face with the constructed orthonormal basis approach to Backus-Gilbert MLS approximation is that the basis $\{\tilde{\phi}_j(\cdot)\}_{j=1}^{M}$ is not explicitly known, so we do not know how the operator $L$ acts on the new basis. In effect, they must be projected onto the original approximation space $\mathcal{U} = \text{span}\{\phi_j(\mathbf{x});\ j=1,\ldots,M\}$ with which we used to compute an orthonormal basis via $QR$ decomposition. This will then allow approximation of the linear differential operator $L$. To this end, for each $j=1,\ldots,M$, we project $\tilde{\phi}_j(\mathbf{x})$ onto $\text{span}\{\phi_j(\mathbf{x});\ j=1,\ldots,M\}$ by calculating the vector $\mathbf{c}^j \in \mathbb{R}^M$ defined as

$$\begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_M(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_M(\mathbf{x}_2) \\ \vdots & & & \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_M(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} c_1^j \\ c_2^j \\ \vdots \\ c_M^j \end{pmatrix} = \begin{pmatrix} \tilde{\phi}_j(\mathbf{x}_1) \\ \tilde{\phi}_j(\mathbf{x}_2) \\ \vdots \\ \tilde{\phi}_j(\mathbf{x}_N) \end{pmatrix}, \quad \text{for } \mathbf{x}_i \in \mathcal{X} \qquad (16)$$

We can then approximate $L\tilde{\phi}_j(\mathbf{x}_k)$ for some $\mathbf{x}_k \in \mathcal{X}$ as

$$L\tilde{\phi}_j(\mathbf{x}_k) = \sum_{i=0}^{M} c_i^j L\phi_i(\mathbf{x}_k)$$

These projections can be efficiently implemented by storing the $LU$ decomposition of the matrix above and then applying direct substition for each vector $\mathbf{c}^j \in \mathbb{R}^M$, $j=1,\ldots,M$.

With the goal being to construct a discrete kernel $\Upsilon_i(\mathbf{x})$ such that

$$Lf(\mathbf{x}) = \sum_{i=1}^{N} f(\mathbf{x}_i)\Upsilon_i(\mathbf{x}) \quad \mathbf{x} \in \mathcal{X}$$

via Backus-Gilbert MLS approximation using the orthonormal basis, we can apply the reproducing constraints to the discrete kernel as

$$L\tilde{\phi}(\mathbf{x}) = \sum_{i=1}^{N} \tilde{\phi}(\mathbf{x}_i)\Upsilon_i(\mathbf{x}) \quad \mathbf{x} \in \mathcal{X}$$

while minimizing $\frac{1}{2}\sum_{i=1} \Upsilon_i^2(\mathbf{x})$. To solve for the discrete kernel $\Upsilon$, we use the process described in section (2.1) and introduce Lagrangian multipliers $\tilde{\lambda}_j,\ j = 1\ldots, M$ to this constrained optimization problem which yields the system

$$\begin{pmatrix} \langle\tilde{\phi}_1,\tilde{\phi}_1\rangle & \langle\tilde{\phi}_1,\tilde{\phi}_2\rangle & \cdots & \langle\tilde{\phi}_1,\tilde{\phi}_M\rangle \\ \langle\tilde{\phi}_2,\tilde{\phi}_1\rangle & \langle\tilde{\phi}_2,\tilde{\phi}_2\rangle & \cdots & \langle\tilde{\phi}_2,\tilde{\phi}_M\rangle \\ \vdots & & & \\ \langle\tilde{\phi}_M,\tilde{\phi}_1\rangle & \langle\tilde{\phi}_M,\tilde{\phi}_2\rangle & \cdots & \langle\tilde{\phi}_M,\tilde{\phi}_M\rangle \end{pmatrix} \begin{pmatrix} \tilde{\lambda}_1(\mathbf{x}) \\ \tilde{\lambda}_2(\mathbf{x}) \\ \vdots \\ \tilde{\lambda}_M(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} L\tilde{\phi}_1(\mathbf{x}) \\ L\tilde{\phi}_2(\mathbf{x}) \\ \vdots \\ L\tilde{\phi}_M(\mathbf{x}) \end{pmatrix}. \tag{17}$$

Using the approximation of $L\tilde{\phi}_j(\mathbf{x})$ from the original basis, the expression for the dicrete kernel then becomes

$$\Upsilon_i(\mathbf{x}) = \sum_{j=1}^{M} \tilde{\lambda}_j(\mathbf{x})\tilde{\phi}_j(\mathbf{x}_i) = \sum_{j=1}^{M} L\tilde{\phi}_j(\mathbf{x})\tilde{\phi}_j(\mathbf{x}_i) \quad \text{for } \mathbf{x} \in \mathcal{X}$$

To summarize, efficiently computing the reproducing kernel associated with the linear differential operator $L$ requires two sets of nodes in the domain, $\mathcal{X}$ on which the evaluations of the basis and the reproducing of the operator are done, and $\Xi$ of $M$ nodes which creates the approximation basis. Moreover, one $QR$ decomposition of the interpolation matrix $A$ is required along with one $LU$ decomposition of the same matrix in order to produce the approximation of operator $L$ acting on the computed orthonormal basis.

## 3.3 Solving Initial-Boundary Valued Problems

Extending this Backus-Gilbert approach to numerically solve initial-value problems should now be straight forward. Consider the following problem

$$\frac{\partial u}{\partial t} = Lu(\mathbf{x},t) \quad \mathbf{x} \in \Omega \ \ t > 0 \tag{18}$$

$$u(\mathbf{x},0) = f(\mathbf{x}) \quad \mathbf{x} \in \Omega$$

where $L$ is a linear spatial differential operator which we will assume forms a well-posed linear initial-value PDE. This new moving-least squares technique begins by calculating the orthonormal basis with respect to the evaluation nodes along with constructing the discrete reproducing kernel. To this end, let $\{\Psi_i\}_{i=1}^N$ be the kernel associated with the orthonormal basis on the set of evaluation nodes $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \Omega \cup \partial\Omega$ and let $\{\Upsilon_i\}_{i=1}^N$ be the kernel associated with the differential operator $L$. Discretizing 18 first in time using a high-order explicit scheme while applying discrete reproducing kernel to the initial condition, $\mathcal{P}u(\mathbf{x}_k,0) = \sum_{i=0}^{N} u(\mathbf{x}_i,0)\Psi_i(\mathbf{x}_k),\ \forall \mathbf{x}_k \in \Omega$, each new timestep $n+1$ can then be solved for simply by applying $\Upsilon_i(\cdot)$ to the solution at the previous time step $n$. For example, suppose a 2nd order Adams-Bashford method is used for the timestepping. Namely,

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \frac{3L\mathbf{u}^n - L\mathbf{u}^{n-1}}{2}. \tag{19}$$

Here $\mathbf{u}^n = [u^n(\mathbf{x}_1), \ldots, u^n(\mathbf{x}_N)]$ is the solution from the previous timestep at each node $\mathbf{x}_k \in \Omega$. The approximation of $Lu^n(\mathbf{x}_k)$ and $Lu^{n-1}(\mathbf{x}_k)$ at a node $\mathbf{x}_k$ is calculated from

$$Lu^n(\mathbf{x}_k) = \sum_{i=0}^{N} u^n(\mathbf{x}_i)\Upsilon_i(\mathbf{x}_k)$$

We remark that since an explicit time integration scheme is being employed, no linear system needs to be solved.

An implicit time scheme can be used by considering the discrete reproducing kernel $\Upsilon_i(\cdot)$ which satisfies reproducing constraints of the form

$$\sum_{i=1}^{N} \tilde{\phi}(\mathbf{x}_i)\Upsilon_i(\mathbf{x}_k) = (I - C\Delta t L)\tilde{\phi}(\mathbf{x}_k) \quad \forall\, \mathbf{x}_k \in \mathcal{X}.$$

where $C$ is some constant depending on the implicit method and $I$ is the identity operator. After discretizing in time using the implicit method and calculating the initial condition, we can advance the timestepping at step $n$ by solving for $\mathbf{u}^{n+1}$ from the linear system

$$\begin{pmatrix} \Upsilon_1(\mathbf{x}_1) & \Upsilon_2(\mathbf{x}_1) & \cdots & \Upsilon_N(\mathbf{x}_1) \\ \Upsilon_1(\mathbf{x}_2) & \Upsilon_2(\mathbf{x}_2) & \cdots & \Upsilon_N(\mathbf{x}_2) \\ \vdots & & & \\ \Upsilon_1(\mathbf{x}_N) & \Upsilon_2(\mathbf{x}_N) & \cdots & \Upsilon_N(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} u^{n+1}(\mathbf{x}_1) \\ u^{n+1}(\mathbf{x}_2) \\ \vdots \\ u^{n+1}(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} u^n(\mathbf{x}_1) \\ u^n(\mathbf{x}_2) \\ \vdots \\ u^n(\mathbf{x}_N) \end{pmatrix}. \tag{20}$$

In the next section on numerical examples, we start by showing the approximation results for a linear elliptic boundary-valued problem which is tied to using an implicit scheme for solving a time dependant problem since at every time step $n$, a system like the one above must be solved in order to produce an approximation for the next time step $n+1$.

### Dirichlet Boundary Conditions

Adding Dirichlet boundary conditions

$$u(\mathbf{x}, t) = g(\mathbf{x}), \quad \mathbf{x} \text{ on } \partial\Omega \;\; t > 0$$

to the initial-value problem adds additional constraints to linear system in 7. To this end, we split the set of $N$ nodes $\mathcal{X}$ into two sets, namely $\mathcal{X}_\Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_1}\}$ of interior nodes and $\mathcal{X}_{\partial\Omega} = \{\mathbf{x}_{N_1+1}, \ldots, \mathbf{x}_N\}$ composed uniquely of nodes on the boundary $\partial\Omega$. Thus we require that for $\mathbf{x}_k \in \mathcal{X}_{\partial\Omega}$, $u(\mathbf{x}_k, t) = g(\mathbf{x}_k)$ and for $\mathbf{x}_k \in \mathcal{X}_\Omega$, $\partial u(\mathbf{x}_k, t)/\partial t = \mathcal{L}u(\mathbf{x}_k, t)$ for any $t > 0$. An explicit time-discretization scheme gives a system for any time step $n$

$$\begin{pmatrix} u^{n+1}(\mathbf{x}_1) \\ \vdots \\ u^{n+1}(\mathbf{x}_{N_1-1}) \\ u^{n+1}(\mathbf{x}_{N_1}) \\ \vdots \\ u^{n+1}(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} \mathcal{F}^n(u^n(\mathbf{x}_1)) \\ \vdots \\ \mathcal{F}^n(u^n(\mathbf{x}_{N_1-1})) \\ g(\mathbf{x}_{N_1}) \\ \vdots \\ g(\mathbf{x}_N) \end{pmatrix}. \tag{21}$$

where

$$\mathcal{F}^n(\mathbf{u}^n(\cdot)) = \mathbf{u}^n + \Delta t \frac{3\mathcal{L}\mathbf{u}^n - \mathcal{L}\mathbf{u}^{n-1}}{2}$$

Similarly, enforcing boundary conditions while using an implicit scheme can be accomplished by splitting the set of nodes $\mathcal{X}$ into the two sets $\mathcal{X}_\Omega$ and $\mathcal{X}_{\partial\Omega}$ and use the reproducing operator kernel $\Upsilon$ applied to $u^{n+1}(\mathbf{x}_k)$ for $\mathbf{x}_k \in \mathcal{X}_\Omega$ and the reproducing kernel $\Psi$ for $\mathbf{x}_k \in \mathcal{X}_{\partial\Omega}$. For example, using the backward Euler implicit method yields the system

$$\begin{pmatrix} \Upsilon_1(\mathbf{x}_1) & \Upsilon_2(\mathbf{x}_1) & \cdots & \Upsilon_N(\mathbf{x}_1) \\ \Upsilon_1(\mathbf{x}_2) & \Upsilon_2(\mathbf{x}_2) & \cdots & \Upsilon_N(\mathbf{x}_2) \\ \vdots & & & \\ \Upsilon_1(\mathbf{x}_{N_1-1}) & \Upsilon_2(\mathbf{x}_{N_1-1}) & \cdots & \Upsilon_N(\mathbf{x}_{N_1-1}) \\ \Psi_1(\mathbf{x}_{N_1}) & \Psi_2(\mathbf{x}_{N_1}) & \cdots & \Psi_N(\mathbf{x}_{N_1}) \\ \vdots & & & \\ \Psi_1(\mathbf{x}_N) & \Psi_2(\mathbf{x}_N) & \cdots & \Psi_N(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} u^{n+1}(\mathbf{x}_1) \\ \vdots \\ u^{n+1}(\mathbf{x}_{N_1-1}) \\ u^{n+1}(\mathbf{x}_{N_1}) \\ \vdots \\ u^{n+1}(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} u^n(\mathbf{x}_1) \\ \vdots \\ u^n(\mathbf{x}_{N_1-1}) \\ g(\mathbf{x}_{N_1}) \\ \vdots \\ g(\mathbf{x}_N) \end{pmatrix}. \quad (22)$$

It is easy to see from the construction of the kernel $\Upsilon$ that the solution $\mathbf{u}^{n+1}$ will satisfy the PDE for any node $\mathbf{x}_k \in \mathcal{X}_\Omega$ and $\sum_{i=1}^{N} u^n(\mathbf{x}_i)\Psi_i(\mathbf{x}_k) = g(\mathbf{x}_k)$ for $\mathbf{x}_k \in \mathcal{X}_{\partial\Omega}$.

## 3.4 Numerical Examples

However, before engaging in numerical examples with time dependent problems, we wish to study the approximating power of this Backus-Gilbert method in reproducing linear differential operators and its dependence on the shape parameter $\beta$. It has been known since the work of Kansa [13] that the shape parameter in multivariate approximation using radial basis functions is of critial importance in dealing with the convergence of the method. Many studies in recent years in employing radial basis functions demonstrate the difficulty in obtaining an approximate optimal shape parameter with respect to minimal $L_1$ error.

Due to their multivariate scattered data approximation ability studied by Franke in [11] we use throughout these numerical examples a radial basis of translated multiquadric functions. In these examples we begin by constructing a radial basis using multiquadric functions on $\Omega = [0, \pi] \times [0, \pi]$. We chose $N$ distinct nodes which were randomly distributed in $\Omega \cup \partial\Omega$ on which the multiquadric functions were translated. The set of evaluation nodes $\mathcal{X}$ were chosen to be equally distributed throughout the domain. Using this basis on the evaluation set $\mathcal{X}$, the interpolation matrix $A$ as in 8 is constructed, followed by the $QR$ decomposition of $A$. Using the orthogonal matrix $Q$, a new basis is contructed with respect to the evaluation nodes. This new basis, $\mathcal{Q} = \text{span}\{\tilde{\phi}_j(\cdot);\ j = 1, \ldots, N\}$ will now be used as the reproducing approximation space for the Backus-Gilbert formulation.

To show the ability of this Backus-Gilbert approach on approximating linear differential operators, we consider the problem on $\Omega = [0, \pi] \times [0, \pi]$

$$\Delta u = -2\sin(x)\sin(y) \quad x, y \in \Omega$$

$$u = 0 \quad x, y \in \partial\Omega$$

which has as solution $f(x,y) = \sin(x)\sin(y)$. Using the orthonormal basis $\mathcal{Q} = \text{span}\{\tilde{\phi}_j(\cdot)\ j = 1,\ldots,N\}$, we construct the reproducing kernel $\Psi_i(\mathbf{x})$ and create the linear system associated with this Poisson boundary value problem which the solution $u$ satisfies.

$$\begin{pmatrix} \Delta\Psi_1(\mathbf{x}_1) & \Delta\Psi_2(\mathbf{x}_1) & \cdots & \Delta\Psi_N(\mathbf{x}_1) \\ \Delta\Psi_1(\mathbf{x}_2) & \Delta\Psi_2(\mathbf{x}_2) & \cdots & \Delta\Psi_N(\mathbf{x}_2) \\ \vdots & & & \\ \Delta\Psi_1(\mathbf{x}_{N_1-1}) & \Delta\Psi_2(\mathbf{x}_{N_1-1}) & \cdots & \Delta\Psi_N(\mathbf{x}_{N_1-1}) \\ \Psi_1(\mathbf{x}_{N_1}) & \Psi_2(\mathbf{x}_{N_1}) & \cdots & \Psi_N(\mathbf{x}_{N_1}) \\ \vdots & & & \\ \Psi_1(\mathbf{x}_N) & \Psi_2(\mathbf{x}_N) & \cdots & \Psi_N(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_{N_1-1}) \\ u(\mathbf{x}_{N_1}) \\ \vdots \\ u(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_{N_1-1}) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (23)$$

The $L_1$ errors associated with different grid sizes on various $\beta$ values are shown in the two tables below. The $N$ evaulation nodes $\mathcal{X}$ are taken to be uniformly distributed and the $M$ collocation nodes $\Xi$ are randomly distributed. In these examples, we take $N = M$.

Table 1: $L_1$ error for $10 \times 10$ and $12 \times 12$ uniform grid nodes for various $\beta$ values.

| $10 \times 10$ | | $12 \times 12$ | |
|---|---|---|---|
| $L_1$ error | $\beta$ | $L_1$ error | $\beta$ |
| 6.06394e-05 | 14 | 6.4211e-06 | 10 |
| 9.37156e-06 | 34 | 8.27118e-06 | 18 |
| 3.57931e-06 | 48 | 4.53544e-06 | 28 |
| 2.58906e-05 | 54 | 1.26301e-05 | 42 |
| 5.23094e-05 | 100 | 3.2699e-05 | 48 |
| 9.22903e-05 | 102 | 4.33146e-05 | 54 |

Table 2: $L_1$ error for $15 \times 15$ and $20 \times 20$ uniform grid nodes for various $\beta$ values.

| $15 \times 15$ | | $20 \times 20$ | |
|---|---|---|---|
| $L_1$ error | $\beta$ | $L_1$ error | $\beta$ |
| 4.57714e-05 | 6 | 8.1885e-07 | 6 |
| 1.01192e-06 | 10 | 7.41084e-06 | 10 |
| 4.72437e-06 | 18 | 3.71961e-05 | 20 |
| 9.65463e-06 | 32 | 5.83601e-05 | 26 |
| 2.54876e-05 | 42 | 2.74848e-05 | 30 |
| 4.52623e-05 | 62 | 7.28861e-05 | 40 |

*Nonlinear viscous Burgers Equation*

In order to validate this Backus-Gilbert construction for solving initial and boundary valued problems, we apply the method to a 2D nonlinear viscous Burgers equation which features

both a nonlinear advection term and a linear diffusion term to smooth the solution in time. Extending this moving-least squares approach to numerically solve a time-dependant nonlinear problem such the viscous Burgers Equation

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{2} u \frac{\partial u}{\partial x_i} = \epsilon \Delta u$$

$$u(0, x, y) = u_0 \quad x, y \in (0, \pi)^2 = \Omega \;\; u(t, x, y) = 0 \;\; x, y \in \partial\Omega \;\; t > 0$$

is handled in a similar manner as above. Discretizing the equation first in time using any time stepping scheme such as a 4th order explicit multistep scheme, a reproducing kernel must be created for both the Laplacian and the advection operators. Figure 1 depicts plots of the solution for various times using a grid composed of $30 \times 30$ uniformly distributed evalutation nodes, $30 \times 30$ random collocation nodes, and setting $\beta = 6$ for the multiquadric basis as used in the elliptic problem above. Furthermore, we set the viscosity $\epsilon = .5$ in Figure 1 and $\epsilon = .1$ in Figure 2. The time-stepping uses an explicit 4th-order Adams-Bashforth scheme.

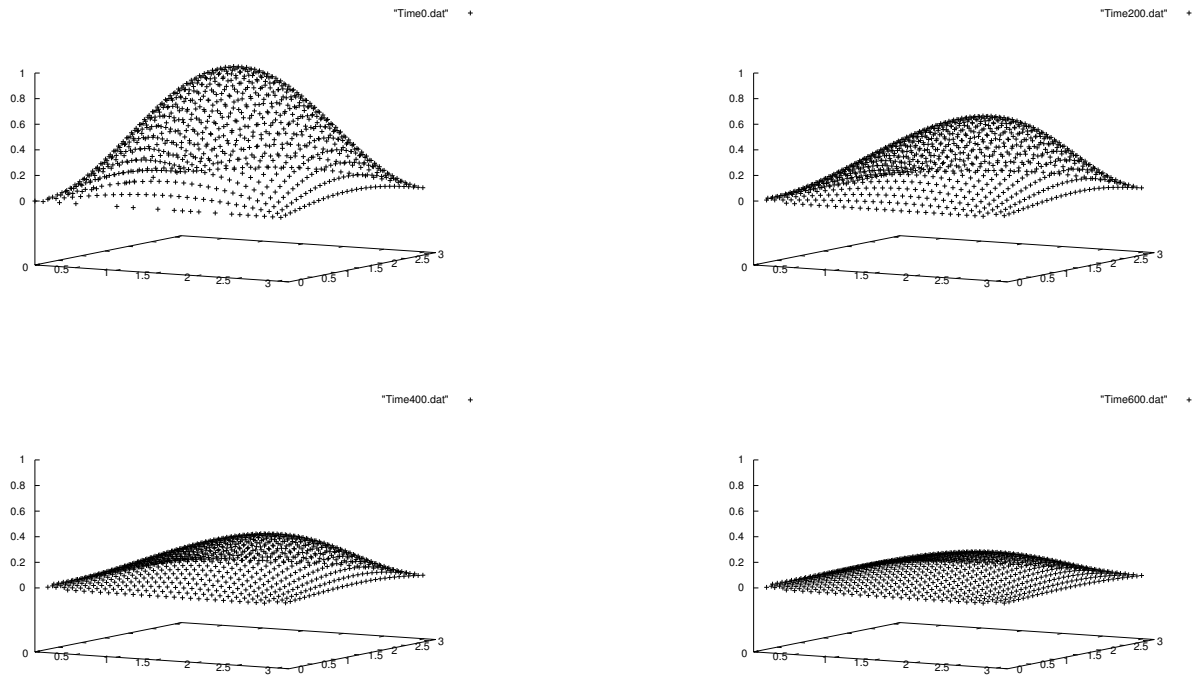


Figure 1: Plot of solution to Burgers Equation for Time = 0,.2,.4,.6, $\epsilon = .5$

## 3.5 Concluding Remarks

In this presentation of a new Backus-Gilbert approximation method for solving initial-boundary valued problems, we attempted to demonstrate that its fine approximation results could make it an attractive substitute for traditional radial basis collocation methods. Although a $QR$ and

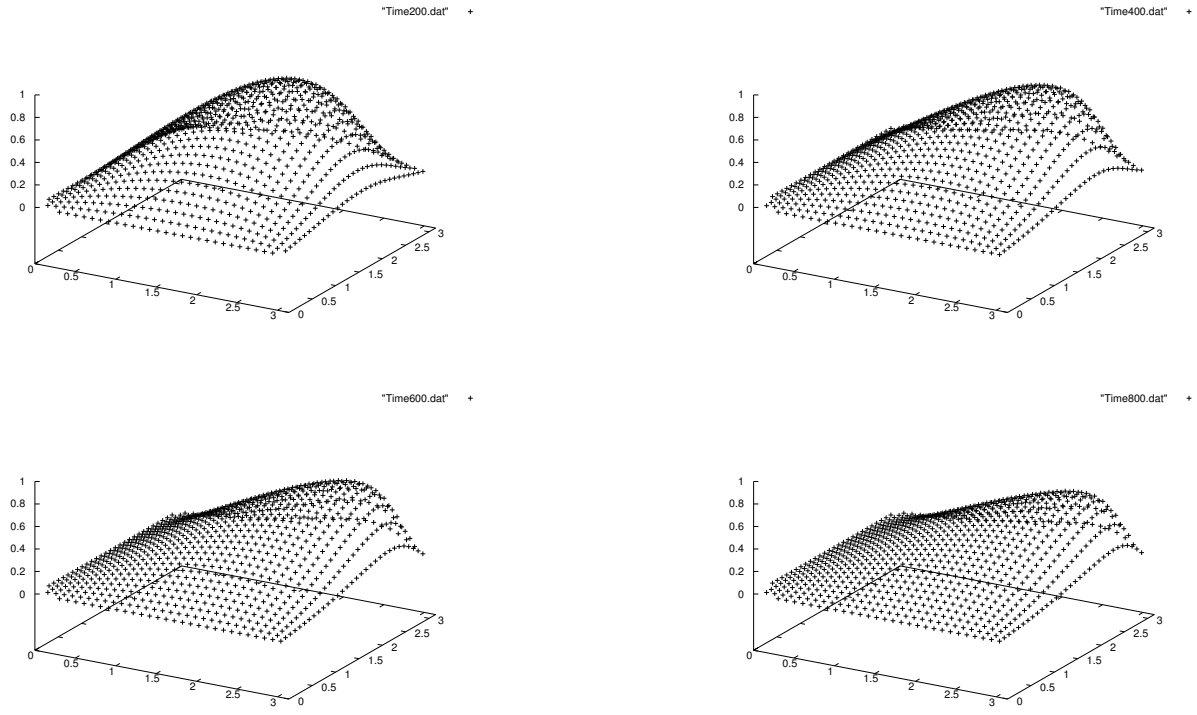


Figure 2: Plot of solution to Burgers Equation for Time = .2,.4,.6,.8, $\epsilon = .1$

$LU$ decomposition is needed in constructing the reproducing kernels for the linear operators, the resulting linear systems when discretizing PDEs have been shown to be slightly better conditioned than its radial basis collocation counterpart.

In future work on this Backus-Gilbert formulation, we plan to test constructions of reproducing kernels which are not generated by radial basis functions. This will include the most general sense of the original Backus-Glibert formulation, namely moving least squares as discussed in section (2.1).

In regards to the multiquadric case in generating the reproducing kernel as shown in this paper, for a fixed $\beta$ and an increasing amount of nodes to generate the radial basis, the moving-least squares method will not converge due to the dependence on the node density. However, unlike straight radial basis collocation, the Backus-Glibert approach offers greater flexibility in choosing a near optimal $\beta$ parameter since larger neighborhoods of near optimal values are available.

The results of the computations of these meshless schemes for solving partial differential equations presented in this paper show that accurate solutions can be obtained, but much work is still needed in adapting these schemes to larger scale problems such as 3D flow in large domains. Our current interest in these methods is to couple them with domain decomposition methods which have been used in finite-difference and finite-element methods with much success. Futhermore, this coupling has been shown to enjoy a computational structure easily adapted for the implementation on massively parallel processors.

# References

[1] Aronszajin, N. (1950). *Theory of Reproducing kernels* Transaction of the American Mathematical Society 68, 337-404.

[2] Babuska, I., Banerjee, U., Osborn, J., *Survey of Meshless and Generalized Finite Element Methods: A Unified Approach*, in preperation.

[3] R.K. Beatson, J.B. Cherrie and C.T. Mouat, *Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration*, Adv. Comput. Math. 11 (1999) 253-270.

[4] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl, *Meshless Methods: an overview and recent developments*, Comp. Meth. Appl. Mech. Eng. (139) 1996, 3-47.

[5] Belytschko T, Lu Y Y, Gu L *Element-free Galerkin methods.* Int. J. Numer. Methods Eng. 1994a 37: 229-256.

[6] L. P. Bos and K. Salkauskas, *Moving least-squares are Backus-Gilbert optimal*, J. Approx. Theory 59 (1989), 267-275.

[7] G.E. Fasshauer, *Matrix-free multilevel moving least-squares methods*, in Approximation Theory, X St. Louis, MO, 2002 271-278.

[8] G.E. Fasshauer, *Approximate moving least-squares approximation: a fast and accurate multivariate approximation method*, in Curve and Surface Fitting: Saint-Malo 2002 Nashboro Press, 2003 138 - 148.

[9] G.E. Fasshauer, *Appproximate moving least-squares approximation for Time-Dependant PDEs*, in *WCCM V* July 7-12, 2002 Vienna, Austria

[10] G. E. Fasshauer, *On smoothing for multilevel approximation with radial basis functions*, in Approximation Theory IX, Vol.II: Computational Aspects, Charles K. Chui, and L. L. Schumaker (eds.), Vanderbilt University Press, 1999, 55-62.

[11] Franke R. *Scattered Data Interpolation: Test of Some Methods* Journal Mathematics of Computations, 1982,33(157):181.

[12] Y. C. Hon and R. Schaback. *On unsymmetric collocation by radial basis functions.* Appl. Math. Comput., 119(2-3): 177-186, 2001.

[13] E. J. Kansa. *Multiquadrics: a scattered data approximation scheme with applications to computational Fluid-dynamics. I. Surface approximations and partial derivative estimates.* Comput. Math. Appl., 19(8-9): 127-145, 1990.

[14] Y.C. Hon, K.F. Cheung, X.Z. Mao, E. Kansa, *Multiquadric Solution for Shallow Water Equations* Journal of Hydraulic Engineering, Vol. 125, No. 5, May 1999, pp. 524-333

[15] L. Ling and E. J. Kansa. *A least-squares preconditioner for radial basis functions collocation methods.* Adv. Comput. Math., PIPS No: 5271809, 2004.

[16] F. J. Narcowich, R. Schaback and J.D. Ward. *Multilevel interpolation and approximation*, Appl. Comput. Harmon. Anal. 7 (1999), 243-261.

[17] F. J. Narcowich, J.D. Ward *Generalized Hermite Interpolation Via Matrix-Valued Conditionally Positive Definite Functions*, Math. Computation, 63 (1994) 208, pg. 661-687

[18] R.A. Lorentz, F.J. Narcowich, J.D. Ward *Collocation discretizations of the transport equation with radial basis functions* Appl. Math. Computation, 145 (2003) 97-116

[19] R. Schaback, *Optimal recovery in translation-invariant spaces of functions*, Annals of Numerical Mathematics, 4 (1997), 547-556.

[20] R. Schaback, *Native Hilbert spaces for radial basis functions, I*, in New Developments in Approximation Theory (Dortmund, 1998) Ser. Numer. Math., 132. Birkhuser, Basel, 1999, 255-282.

[21] R. Schaback, *On the efficiency of interpolation by radial basis functions*, in Surface Fitting and Multiresolution Methods, A. LeMehaute, C. Rabut, and L.L. Schumaker, Venderbilt University Press, Nashville TN, 1997, 309-318.

[22] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. in Comput. Math., 3 (1995), 251-264.

[23] H. Wendland, *Scattered Data Modelling by Radial and Related Functions*, Habilitation thesis, Univ. Gottingen, 2002.

[24] A. S. M. Wong, Y. C. Hon, T. S. Li, S. L. Chung, and E. J. Kansa. *Multizone decomposition for simulation of time-dependent problems using the multiquadric scheme.* Comput. Math. Appl., 37(8):23-43, 1999.

[25] Z. Wu, Hermite-Birkhoff interpolation of scattered data by radial functions *Approximation Theory and its Applications*, 8 (1992), 1-10.

[26] X. Zhou, Y. C. Hon, and Jichun Li. *Overlapping domain decomposition method by radial basis functions.* Appl. Numer. Math., 44(1-2):241-255, 2003.

03