


A Numerical Method for Solving Ill-Conditioned Equation Systems Arising from Radial Basis Functions

Edward J. Kansa 

Convergent Solutions, Livermore, CA, USA

Email: edwardjkansa@gmail.com

How to cite this paper: Kansa, E.J. (2023) A Numerical Method for Solving Ill-Conditioned Equation Systems Arising from Radial Basis Functions. *American Journal of Computational Mathematics*, 13, 356-370. <https://doi.org/10.4236/ajcm.2023.132019>

Received: March 15, 2023

Accepted: June 24, 2023

Published: June 27, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Continuously differentiable radial basis functions (C^∞ -RBFs), while being theoretically exponentially convergent are considered impractical computationally because the coefficient matrices are full and can become very ill-conditioned. Similarly, the Hilbert and Vandermonde have full matrices and become ill-conditioned. The difference between a coefficient matrix generated by C^∞ -RBFs for partial differential or integral equations and Hilbert and Vandermonde systems is that C^∞ -RBFs are very sensitive to small changes in the adjustable parameters. These parameters affect the condition number and solution accuracy. The error terrain has many local and global maxima and minima. To find stable and accurate numerical solutions for full linear equation systems, this study proposes a hybrid combination of block Gaussian elimination (BGE) combined with arbitrary precision arithmetic (APA) to minimize the accumulation of rounding errors. In the future, this algorithm can execute faster using preconditioners and implemented on massively parallel computers.

Keywords

Continuously Differentiable Radial Basis Functions, Global Maxima and Minima, Solutions of Ill-Conditioned Linear Equations, Block Gaussian Elimination, Arbitrary Precision Arithmetic

1. Introduction

Hardy invented an interpolation function for scattered data arrangements based upon geodesics between pairs of points, see [1]. Theoretically, the continuously differentiable radial basis functions (C^∞ -RBFs) have very impressive properties. They converge exponentially with refined discretization, and also converge faster

as the spatial dimensionality increases, see [2]. Two very commonly considered C^∞ -RBF are:

$$\phi_j(x) = \exp\left(-(\mathbf{x} - \mathbf{y}_j)^2 / c_j^2\right) \quad (\text{Gaussian}) \quad (1)$$

$$\phi_j(\mathbf{x}) = 1 + (\mathbf{x} - \mathbf{y}_j)^2 / c_j^2 \quad (\text{Multiquadric}) \quad (2)$$

where \mathbf{x} and \mathbf{y} are the evaluation centers and data centers, respectively, in d -dimensional space, \mathfrak{R}^d . The set of variables, $\{c_j^2\}$, is a set of shape parameters that either narrows or broadens the resulting distribution. For the multiquadric (MQ) RBF, m is the exponent, $\mu \geq -1/2$ and excludes all integers. The exponent, μ , either steepens or flattens the distribution near the data center, \mathbf{y}_j . Many applications of C^∞ -RBFs such as: interpolation, approximation, integral, integro-differential, and partial differential equations use the concept of interpolation as the starting point for various applications, see [3] [4].

Assume there are n_i data and n_i evaluation centers defined over the interior, and n_b data and n_b evaluation centers such that the sum equals the total number of centers, N . That is:

$$N = n_i + n_b. \quad (3)$$

Assume that the set of n_i interior points are denoted by the subscript, \mathbf{i} , and the set of n_b boundary points are denoted by the subscript, \mathbf{b} . These points may be either the interior data centers or evaluation centers and, similarly, the boundary data or evaluation centers.

The interpolation problem can be expressed in matrix form as:

$$\begin{pmatrix} \Phi_{i,i} & \Phi_{i,b} \\ \Phi_{b,i} & \Phi_{b,b} \end{pmatrix} \begin{pmatrix} \alpha_i \\ \alpha_b \end{pmatrix} = \begin{pmatrix} U_i \\ U_b \end{pmatrix} \quad (4)$$

Denote the general boundary operators that can be either Dirichlet, Neumann or Robin operators as \mathcal{B} , over the n_b boundary points, and let \mathcal{L} be an integral, integro-differential operator, or an elliptic, hyperbolic, parabolic operator over n_i interior points. Define the appropriate forcing functions, $g(x, t)$ on the boundary, and $f(x, t)$ over the interior. The resulting set of N equations over N unknown expansion coefficients for PDEs, IEs, or I-PDEs are written as:

$$\begin{pmatrix} \mathcal{L}\Phi_{i,i} & \mathcal{L}\Phi_{i,b} \\ \mathcal{B}\Phi_{i,b} & \mathcal{B}\Phi_{b,b} \end{pmatrix} \begin{pmatrix} \alpha_i \\ \alpha_b \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \quad (5)$$

To achieve a compact notation that illustrates the solution method, it is convenient to define the following matrices:

$$\Phi = \begin{pmatrix} \Phi_{i,i} & \Phi_{i,b} \\ \Phi_{b,i} & \Phi_{b,b} \end{pmatrix}, \quad (6)$$

$$\alpha = \begin{bmatrix} \alpha_i \\ \alpha_b \end{bmatrix}, \quad (7)$$

$$U = \begin{bmatrix} U_i \\ U_b \end{bmatrix}, \quad (8)$$

$$\mathbf{h} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad (9)$$

and

$$\Psi = \begin{pmatrix} \mathcal{L}\Phi_{i,i} & \mathcal{L}\Phi_{i,b} \\ \mathcal{B}\Phi_{i,b} & \mathcal{B}\Phi_{b,b} \end{pmatrix}. \quad (10)$$

In compact notation, the trial solution is given by:

$$\mathbf{U}^{\text{trial}} = \Phi \{ \Psi^{-1} \mathbf{h} \} = \Phi \alpha. \quad (11)$$

Assume that the exact and the trial calculated solutions are given. In comparing the trial solution, $\mathbf{U}^{\text{trial}}$, to the exact solution, $\mathbf{U}^{\text{exact}}$, the error, ζ , is denoted by norm of the difference between the exact solution and the exact solution, where the exact solution is defined by the user. Thus,

If

$$\zeta = \|\mathbf{U}^{\text{trial}} - \mathbf{U}^{\text{exact}}\| > \eta, \text{ reject}, \quad (12.a)$$

else

$$\zeta = \|\mathbf{U}^{\text{trial}} - \mathbf{U}^{\text{exact}}\| \leq \eta, \text{ accept}. \quad (12.b)$$

In this study, the error denoted by, ζ will be reported as the well-known root mean square error, (RMSE), if $U^{\text{exact}}(\mathbf{x}) \leq 1$, and if $U^{\text{exact}}(\mathbf{x}) > 1$, the normalized root mean square error is used. The error, ζ , depends upon set or linear or linearized expansion coefficients, $\{\alpha_j\}$, but also upon the nonlinear n parameter set, \mathbf{Q} , where:

$$\mathbf{Q} = \{ \mathbf{x}, \mathbf{y}, \mathbf{c}, \mu \}. \quad (13)$$

On a finite precision computer, rounding errors can severely degrade the accuracy of the numerical results. The machine epsilon is proportional to the inverse of the absolute condition number, κ , that indicates the number of digits of reliability of the numerical solution, see [4]. The condition numbers of both the interpolation matrix, Φ and the PDE matrix, Ψ , are implicit functions of the parameter set, \mathbf{Q} .

The procedure used is to vary the N data centers, the N evaluation centers, the N shape parameters, and the MQ exponent to find the optimal set, \mathbf{Q} , that minimizes the errors. These $3N + 1$ free parameters are then used to construct a full $N \times NPDE$, IE, or IPDE coefficient matrix, Ψ .

The matrix from the set of equations used to solve for the unknown expansion coefficients is asymmetric, full, and potentially ill-conditioned. As the shape parameters become larger, and the separation distances become smaller, and the equation system size increases, so does the potential for severe ill-conditioning occur.

2. A Very Brief Survey Method to Solve Full Ill-Conditioned Linear Equations

Consider an asymmetric, full, and very ill-conditioned matrix, \mathbf{A} , such that:

$$\mathbf{Ax} = \mathbf{b} . \quad (14)$$

The condition number of a matrix is the ratio of the largest singular value of that matrix to the smallest singular value, denoted by $\kappa(A)$. Computer arithmetic and mathematics are not identical to idealized Platonic arithmetic and mathematics due to the finite precision approximations of numbers and functions on electronic computers. The larger number of computer bits representing a number or a function, the closer is the computer representations to the ideal Platonic world, even though the goal is unattainable. In addition, in the computer universe, the creation of computer beings and their interactions requires a finite amount of time (cost) unlike that of the beings in the Platonic universe. Numbers, functions, and operations require a finite time (cost) to perform.

In some engineering applications, very fast iterative based approximations are needed. Where human life is at stake, extremely accurate direct methods are required, and computational speed is a secondary concern. In either case, computational speed can be addressed by utilizing highly optimized massively parallel computers for both scenarios.

Some examples of direct methods are: Gaussian elimination, LU decomposition, SVD, or QR decomposition. If there were no rounding errors, then direct methods are theoretically exact. There are multitudes of iterative schemes such as overlapping or non-overlapping domain decomposition and blending, iterative refinement, and reconditioning.

Truncated singular value decomposition (SVD) is a direct method to treat very ill-conditioned systems by discarding singular values smaller than cut-off values near the magnitude of the machine epsilon. For problems requiring high accuracy, information is discarded. Tikhonov regularization modifies ill-conditioned matrices by multiplying the identity matrix by a small parameter and adding it to the matrix, \mathbf{A} ,

$$(\mathbf{A} + \delta \mathbf{I})x = b , \quad (15)$$

where δ is a small positive constant. The disadvantage of this approach is that the exact original problem is not solved.

Pinpointing, see [4], modifies SVD decomposition by splitting the singular values that are greater than the double precision epsilon and those less than or equal to the double precision epsilon and treated these singular values with quadruple precision. This method was successful for treating ill-conditioned solid mechanics problems using MQ discretization, see [5]. However, in some applications, the SVD singular values can be even smaller than the quadruple precision epsilon. Iterative refinement is another popular method to improve accuracy. This approach is successful if the approximate solution, \mathbf{x}' , and the approximate inverse, (A^{-1}) are close to correct values.

Preconditioning is very popular for direct or iterative methods to solve ill-conditioned linear problems by reducing the condition number of the original matrix, \mathbf{A} . Assume the preconditioner is represented by a matrix, \mathbf{P} .

Then new linear problem becomes:

$$(PA)x = Pb, \quad (16)$$

As a result,

$$\kappa(PA) \ll \kappa(A). \quad (17)$$

The construction of a successful preconditioner is not always easy and depends upon the structure of the matrix, A . Preconditioners for sparse, symmetric matrices are relatively easy. A more difficult task that was successfully achieved for asymmetric matrices, see [6].

The difficult task of constructing preconditioners for the full, asymmetric matrices arising from C^∞ RBF applications was performed in the following papers, see [7]. These preconditioners were only able to lower the condition number by only 6 orders of magnitude; in most cases with large shape parameters and larger system sizes, the condition numbers could easily become enormous.

The focus of this manuscript is not on engineering problems for which crude approximations are acceptable, but on problems that required direct methods with the highest possible accuracy. The next sections will focus on highly accurate results.

3. The Hybrid Block Gaussian Elimination-Arbitrary Precision Arithmetic Algorithm

Computer mathematics is not synonymous with ideal Platonic mathematics because firstly, all numbers only possess finite precision and secondly, all library functions are only finitely precise. The Institute of Electrical and Electronic Engineers (IEEE) defines a single precision word to consist of 32 bits and a double precision word to consist of 64 bits.

The condition number, κ , of a matrix is the ratio of the maximum to minimum matrix singular values and is a measure of the confidence of linear equation solutions in single or double precision. The cutoff κ for double precision is $O(1e+16)$. If κ exceeds the inverse of the machine epsilon, there are zero digits of precision and numerical results can be unreliable.

Arbitrary Precision Arithmetic (APA) is a very powerful tool to avoid the consequences of accumulating rounding errors. Assuming a computer has sufficient storage, any number can be stored as a string of smaller base numbers. There is no fixed limit on number of the number of base types used, only the available amount of computer memory. APA are software codes that rearrange memory to store the arbitrary precision variables and perform operations on these numbers with the desired Note that there are two valid notations for the number zero: an empty vector and a vector with a single zero digit. Some programming languages such as Lisp, Python, Perl, Haskell, etc. allow an arbitrary number of digits of precision. Numbers are stored as a vector in which each element is a single digit of that number, and digits are stored from least to most significant. There is no fixed limit on the number of base types used to represent numbers, just whatever the computer memory can hold. All operations are im-

plemented so the result does not have any leading zeros. All operations that might result in a number with leading zeros should be followed by a code that removes them. Lastly, APA can be used to avoid overflow that is not possible with fixed precision arithmetic. Since obtaining more precision via computer chips is unlikely, software methods such as the *MPLAPACK package*, see [8], that is based upon *BLAS*, and the *Advanpix* package, [9], are required. *Advanpix* is very fast compared to other APA packages because of proprietary implementations. The objective of solving the linearized set of equations is to find the expansion coefficients, $\{a\}$, so that a trial solution can be constructed to determine whether the solution error, ζ .

In linear algebra applications, a matrix is a rectangular or square table arranged in columns and rows consisting of numbers, symbols, or expressions. In addition, a matrix may be composed of a table of elements that are matrices that are arranged in columns and rows. The analog of dividing an element is to multiply that matrix element by the inverse of another matrix element. The equivalent of a matrix element being unity is the identity matrix of size p ; the equivalent of a matrix being zero is the null matrix of size p .

As the size of a matrix and its bandwidth increases, so does its condition number, κ . The full matrices, such as those associated with C^∞ -RBF applications can rapidly become so ill-conditioned on single and precision computers that the numerical results can become unreliable. Rather than solving the entire full system of equations at one time, the approach taken here is to consider solving *many smaller, but better conditioned systems*.

The numerical solution of large matrices with large bandwidths has used the Block Gaussian Elimination (BGE) scheme used for many years, see [10] [11] [12]. The concept for the Block Gaussian Elimination method is simple. The large $N \times N$ matrix with N rows and N columns is partitioned into $K \times K$ blocks along the K rows and K columns. In turn, within each block, there are $p \times p$ elements. Assume the $N \times N$ matrix is subdivided into $K \times K$ uniformly sized blocks, each of which consists of $p \times p$ elements, such that:

$$N = K \cdot p . \quad (18)$$

To illustrate the process, assume that the matrix A is subdivided into $K \times K$ blocks and each block contains $p \times p$ elements: (**Figure 1**)

The procedure used is block Gaussian elimination (BGE), see [6] [7] [8], without pivoting since pivoting induces stalling, see [2] [3]. The superscript, m , on the block matrices, including the column vector or augmented matrix, indicates the m -th diagonal operation. Having calculated the math diagonal inverse, the remaining block matrices are multiplied by that inverse. This procedure can be performed in parallel. The remaining blocks of the j -th row are sent to separate processors to form the Schur product matrices and subtract the Schur matrices to construct new block matrices. Since the inverse of the $p \times p$ diagonal block matrix is considerably smaller, the inverse is mildly contaminated by rounding errors (**Figure 2**).

$$\begin{matrix}
 A^0_{1,1} & A^0_{1,2} & A^0_{1,3} & \dots & A^0_{1,K} & \beta^0_1 \\
 A^0_{2,1} & A^0_{2,2} & A^0_{2,3} & \dots & A^0_{2,K} & \beta^0_2 \\
 A^0_{K,1} & A^0_{K,2} & A^0_{K,3} & \dots & A^0_{K,K} & \beta^0_K
 \end{matrix}$$

Figure 1. The full matrix A is partitioned to $K \times K$ blocks where each block contains $p \times p$ elements.

$$\begin{matrix}
 I & 0 & A^2_{1,3} \dots & A^2_{1,K} & \beta^2_1 \\
 0 & I & A^2_{2,3} \dots & A^2_{2,K} & \beta^2_2 \\
 0 & 0 & A^2_{3,3} \dots & A^2_{3,K} & \beta^2_3 \\
 \vdots & & & & \\
 0 & 0 & A^2_{K,3} \dots & A^2_{K,K} & \beta^2_K
 \end{matrix}$$

Figure 2. The partial decomposition of the matrix, A , at the third diagonal.

Continuing in a similar manner, the last diagonal is reached, and the last matrix block is inverted. The last column matrices are eliminated. The augmented matrix contains the expansion coefficients, see **Figure 3**.

Since $N = K \cdot p$, it is a simple matter to transform the β column into the α expansion coefficient column of size N .

$$\begin{matrix}
 I & 0 & 0 \dots & 0 & \beta^k_1 \\
 0 & I & 0 \dots & 0 & \beta^k_2 \\
 0 & 0 & I \dots & 0 & \beta^k_3 \\
 \vdots & & & & \\
 0 & 0 & 0 \dots & I & \beta^k_K
 \end{matrix}$$

Figure 3. The final decomposition of the matrix A at the last diagonal.

In the next section, it will be shown that the size and condition number of the block diagonal matrices are considerably smaller than that of the initial matrix, A . To further assist in obtaining high accuracy, arbitrary precision arithmetic (APA) will be used to minimize the accumulation of rounding errors in the formation of Schur matrix components and their subtraction in the block elimination procedure.

4. Elimination-Arbitrary Precision Arithmetic Algorithm

Computer mathematics is not synonymous with ideal Platonic mathematics because firstly, all numbers only possess finite precision and secondly, all library functions are only finitely precise. The Institute of Electrical and Electronic Engineers (IEEE) defines a single precision word to consist of 32 bits and a double

precision word to consist of 64 bits.

The condition number, κ , of a matrix is the ratio of the maximum to minimum matrix singular values and is a measure of the confidence of linear equation solutions in single or double precision. The cutoff κ for double precision is $O(1e+16)$. If κ exceeds the inverse of the machine epsilon, there are zero digits of precision and numerical results can be unreliable.

Arbitrary Precision Arithmetic (APA) is a very powerful tool to avoid the consequences of accumulating rounding errors. Assuming a computer has sufficient storage, any number can be stored as a string of smaller base numbers. There is no fixed limit on number of the number of base types used, only the available amount of computer memory. APA are software codes that rearrange memory to store the arbitrary precision variables and perform operations on these numbers with the desired Note that there are two valid notations for the number zero: an empty vector and a vector with a single zero digit. Some programming languages such as Lisp, Python, Perl, Haskell, etc. allow an arbitrary number of digits of precision. Numbers are stored as a vector in which each element is a single digit of that number, and digits are stored from least to most significant. There is no fixed limit on the number of base types used to represent numbers, just whatever the computer memory can hold. All operations are implemented so the result does not have any leading zeros. All operations that might result in a number with leading zeros should be followed by a code that removes them. Lastly, APA can be used to avoid overflow that is not possible with fixed precision arithmetic. Since obtaining more precision via computer chips is unlikely, software methods such as the *MPLAPACK package*, see [8], that is based upon *BLAS*, and the *Advanpix* package, [9], are required. *Advanpix* is very fast compared to other APA packages because of proprietary implementations. The objective of solving the linearized set of equations is to find the expansion coefficients, $\{a\}$, so that a trial solution can be constructed to determine whether the solution error, ζ .

5. Calculations with Ill-Conditioned Systems

The first set of numerical experiments examines the root mean squared errors (RMSEs) is the square of the difference between the exact solution at $U^{\text{exact}}(x_j)$ and the numerical solution at x_j divided by the number of samples, N :

$$\text{RMSE} = \sum_j (U_j^{\text{exact}} - U_j^{\text{trial}})^2 / N \quad (19)$$

If the magnitude of U^{exact} is greater than one, the normalized root mean squared.

Error (NRMSE) is:

$$\text{NRMSE} = \left[\sum_j (U_j^{\text{exact}} - U_j^{\text{trial}})^2 / N (U^{\text{exact}})^2 \right]^{1/2} \quad (20)$$

Double precision has about 16 digits of precision. The upper limit for the absolute condition number is $O(1e+16)$ in double precision; while it is possible

to obtain accurate numerical results in double precision if the matrix condition number exceeds $O(1e+16)$, this not the universal case. In the first study, a set of equations arising from a (16×16) Hilbert matrix, see [13], is studied in double and quadruple precision. This study also investigates the effect upon the RMSEs, the maximum block condition number, and the execution time, the maximum block condition number, and the execution time (**Table 1**).

The original 16×16 Hilbert matrix has the advantage of reducing the maximum block condition number for a maximum 2×2 block condition number of $2.88e+3$. The RMSE for this partitioning is only reduced from 14.05 to 8.82. The source of this error is most likely from the formation of the Schur matrices involved forming product matrices and subtracting the resultant during the block elimination process. The same problem is solved again, but instead of using 16 digits of precision, 32 digits of precision we used, with a much-improved outcome.

Table 2 shows more dramatic reduction of η and RMSE with the combination of block size reduction and extra digits of precision. Note that the RMSEs are dramatically reduced, especially for the (2×2) blocks,

Using 32 digits of precision alters the runs. Notice the run times on a serial computer are longer, and the maximum block condition numbers have increased due to more accurate calculations of the singular values. The most noticeable improvement in the reduction of the RMSEs. The next exercise is the calculation of a 300×300 Hilbert system of equations (**Table 3**).

Table 1. (16×16) Hilbert system with different block sizes, maximum block condition number, and RMSEs with 16 digits of precision.

16 digits	Max lock cond. No.	RMSE
4 (8×8)	1.33e+11	14.0537
16 (4×4)	4.99+08	9.102
64 (2×2)	2.82e+03	3.832

Table 2. (16×16) Hilbert system with different block sizes, execution time, RMS errors with 32 digits of precision.

32 digits of precision	Max lock Cond. No.	RMSE
4 (8×8)	1.33e+12	3.8e-09
16 (4×4)	1.55e+09	1.63e-12
64 (2×2)	1.54e+04	1.80e-16

Table 3. RMSEs for block decomposition of a (300×300) Hilbert matrix with 48 and 200 digits of precision.

Blocks Size 48 digits	RMSE	Blocks Size 200 digits	RMSE
10 (30×30)	7.32e-01	10 (30×30)	1.1e-52
15 (20×20)	2.72e-15	15 (20×20)	4.3e-527
30 (10×10)	1.9e-20	30 (10×10)	4.9e-537

Notice that using 48 digits id precision, the RMS errors dropped to dramatically to a $1e-24$ for the fine 5×5 blocks. The drop in RMS errors is even more dramatic with 200 digits of precision dropping to $4e-537$. Using 200 digits of precision reduced the accumulation of rounding errors.

The next examples are the Vandermonde matrices, see [14], defined as a matrix of vectors raised to a power, commonly given by:

$$A_v(i, j) = v(j)^{(N-j)}, \quad (21)$$

where $v(j) = [v_0, v_0 + dv, v_0 + 2dv, \dots, v_N]$. Specialized fast Vandermonde solvers exist, see [15], exist that are more efficient than block Gaussian elimination, but the optimal choice of algorithms for Vandermonde systems, Hilbert, Wilson, etc. systems is beyond the scope of this study.

The study of the Vandermonde system of equations will be made into two parts. The first part of the study will be performed on a limited memory serial computer whose maximum size with APA is 144. The second part of the study will be performed on a parallel machine consisting of 16 (Graphical Processing Units) GPUs cores that is capable of handling many more digits of precision. For the size 100×100 matrix, $dv = 0.5$, and $v = 50.5$, and for the size 144 matrix, $dv = 0.5$, and $v = 72.5$. The condition number of the size 100 matrix is estimated to be $8e+202$. With 360 digits of precision, the results for the 25 (4×4) blocks the RMSE = $4e-44$, and with 50 (2×2) blocks, respectively, the RMSE is $4e-95$. The condition number of the global 144×144 matrix is $4e+313$ (Table 4).

Table 4. Block size, maximum condition number, digits used, run time, and global MS error for the Vandermonde equations.

Linear system	Cond. No.	Digits Used	Rune Time	RMSE
100×100	$7.9e+202$	300	0.03	$7.9e-203$
200×200	$7.1e+463$	600	0.41	$5.7e-931$
300×300	$1.9e+746$	1200	3.35	$1.8e-2007$
400×400	$1.8e+1278$	1500	14.38	$3.3e-2597$
500×500	$1.3e+1671$	1600	30.16	$4.8e-3284$
600×600	$1.4e+1671$	1800	47.38	$1.4e-3284$
700×700	$1.3e+1906$	2100	58.40	$1.3e-3880$

P. Holoborodko has constructed parallel machines consisting of 32 or 64 Graphical Processing Units (GPUs).

To construct a trial solution, U^{trial} , the N unknown expansion coefficients need to be solved. The sets of equations to find the expansion coefficients can be ill-conditioned on electronic digital computers.

There are many potential causes of numerical ill-conditioning such as: the size of the equation system, the minimal distance between data and evaluation centers, the magnitude of the shape parameters, and the magnitude of the exponen-

tial, μ . The same BGM-APA hybrid algorithm that was used to solve very ill-conditioned Hilbert and Vandermonde equations will be used for C^∞ RBF applications.

6. Solutions of a 2D Eigenvalue Poisson Equation

The incentive to develop the Block Gaussian elimination arbitrary precision arithmetic (BGE-APA) method is to find accurate numerical solutions of PDEs, IEs, and IPDEs using C^∞ -RBFs, see [16]. The global minimization procedure is taken from [17] [18] for small memory computers; the presented procedure can be modified for more powerful computers. Note the solutions from C^∞ -RBFs are more complex than either the Hilbert or Vandermonde systems. As the members of the parameter set, Q , are varied, different outcome RMSEs are produced, and a search of the RMSE landscape is required to find the global minima. Because the RMSEs can vary so much, many calculations of the RMSEs are needed to determine which Q produces the minimum RMSE.

Sometimes collocation will be used in which both the evaluation centers coincide with the data centers, but at other times these centers may be distinct. The evaluation centers are allowed to extend slightly beyond the domain, see [19]. In addition, the shape parameters can be either uniform or there may be distinct shape parameters, see [20] [21]. Also, the shape parameters on the boundary may be larger than those on the interior and the MQ exponent may be larger than $\frac{1}{2}$, see [21].

This produces a system of N linear (linearized) equations in N unknown expansion coefficients. Given the trial expansion coefficients, U_t^{rial} is constructed by interpolation and compared to U^{exact} . The goal of this manuscript is to solve IEs, PDEs, and I-PDEs with a minimum number of data and evaluation centers as possible to avoid the curse of dimensionality for more complex problems occurring in higher dimensions.

The test problem is the solution of the 2D eigenvalue Poisson PDE:

$$\nabla^2 U(x_1, x_2) = 50 \exp(5x_1 + 5x_2) \text{ over } r\Omega \setminus \partial\Omega, \quad (22)$$

$$U(x_1, 0) = \exp(5x_1, 0) \text{ along } \partial\Omega_1 \quad (23)$$

$$U(0, x_2) = \exp(0, 5x_2) \text{ along } \partial\Omega_2 \quad (24)$$

$$U(x_1, 1) = \exp(5x_1, 0) \text{ along } \partial\Omega_3 \quad (25)$$

$$U(1, x_2) = \exp(0, 5x_2) \text{ along } \partial\Omega_4 \quad (26)$$

This test problem will be solved numerically on a unit square using scattered data configurations.

The largest value of U occurs at $x_1 = 1, x_2 = 1$ where $U = 2.2026e+04$. At both $x_1 = 0, x_2 = 1$ and $x_1, x_2 = 0$, $U(x_1, x_2) = 148.41$, and at $\mathbf{x}_1 = \mathbf{0}, \mathbf{x}_2 = \mathbf{0}$, $U = 1$.

Case 1. The above problem, Equations (20)-(24), is discretized with 100 (10 \times 10) data and evaluation centers. There are 64 randomly generated interior 2D points for each of the data and evaluation centers; there were 36 randomly generated 1D boundary points.

There were 100 distinct shape parameters that varied from 4 to 199. Because the forcing functions for both the boundary and interior vary very rapidly approaching near $x_1 = 1$, $x_2 = 1$, the data and evaluation centers are clustered near $x_1 = 1$, $x_2 = 1$. There are a total of 36 points on the boundary. The interior evaluation centers extended up to 0.015 beyond the boundary centers.

The boundary shape parameters were 3.8 larger than the interior ones. The MQ exponential, $\mu = 1.43$. There were 200 digits of precision, and NRMSE is $17.4e-27$.

Unlike the simpler Hilbert and Vandermonde problems, the MQ-PDE problems are much more complicated because the RBFs are nonlinear in nature and the NRMSE landscape is filled with many peaks and valleys and the locations of the global minima valleys are not known a priori. There were hundreds of realizations of the 2D Poisson PDE. Only a small sample of the most significant realizations will be presented and summarized. The global condition number = $1.9e50$, and the NRMSE = $8.9e-18$. While this NRMS error may be improved, a variation of the Newton-Raphson method may lower the NRMSE further.

Case 2. In this test case, there were a total of 100×10 (10,000) randomly $\mu = 1.43$. Generated data and evaluation centers and the evaluation centers extend 0.001 beyond the boundary. There are 3600 randomly generated 1D boundary points, and 6400 randomly generated 2D interior points. In addition, there are $1e4$ randomly generated 1D shape parameters that are scaled from $5e0$ to $1.8e4$. Those shape parameters assigned to the boundary were multiplied by 3.2, and $\mu = 1.43$. The condition number of the coefficient matrix was $3.4e159$, and the NRMSE was $6.2e-124$.

Case 3. In this test, there were $4e4$ randomly generated 1D shape parameters, and linearly scaled from $1.2e1$ to $3.0e4$, and $\mu = 1.43$. The boundary shape parameters were 8 times greater than the interior parameters. There are $1.6e4$ randomly generated 1D boundary randomly generated data and evaluation centers and $2.4e4$ randomly generated 2D interior data and evaluation centers. The number of digits was set to 500 digits of precision. The condition number was $2e524$ and the NRMSE was $9.8e-228$.

Case 4. In this test, there are $4e4$ 1D boundary data and evaluation centers and $5e4$ 2D interior data and evaluation centers. Both sets of data and evaluation centers were randomly generated. The shape parameters were also generated as a 1D column of $9e4$ points that were linearly scaled from $2e1$ to $9e4$, $e4$, and $\mu = 1.43$. The boundary shape parameters were multiplied by 4.5. The condition number was $6.8e+713$ and the NRMSE was $9.8e-372$, and there were 800 digits of precision.

The search for the smallest NRMSE is a global minimization problem that depends *nonlinearly* upon the number and location of the data centers, the evaluation centers, the boundary and interior shape parameters, and the exponents, μ of the MQ RBFs. This problem was discussed previously in [7] [8].

7. Conclusions

The application of C^{ν} -RBFs to PDEs, IEs, and I-PDEs involves nonlinear basis functions. The usual approach is to estimate appropriate data and evaluation centers, the shape parameters, and the MQ exponential. This study examines the hybrid combination of Block Gaussian Elimination (BGE) and Arbitrary Precision Arithmetic (APA) to form a new hybrid algorithm. The algorithm improves both accuracy and execution time since the BGE portion is inherently parallelizable and there is no pivoting that stalls the calculations.

By using many small diagonal block inversions with fewer digits of precision, shorter run times are required. Both BGE and APA complement each other. The Schur component matrices formed in Gaussian elimination can accumulate rounding errors if there is no increase in precision. The optimal balance of error control and run time is beyond the limited scope of this more study.

By using many small diagonal block inversions with fewer digits of precision, shorter run times are required. Both BGE and APA complement each other. The Schur component matrices formed in Gaussian elimination can accumulate rounding errors if there is no increase in precision.

The problem of ill-conditioning and solution stability is discussed in [22]. The more common cause of ill-conditioning is the choice of the shape parameters. Luh [23] [24] showed that if the problem to be solved resides in Sobolev space, then his theory very accurately predicts the optimal shape parameters for both Gaussian and multiquadric C^{ν} -RBFs.

A similar approach was used in [25] showing that decomposing a large full matrix has the main advantage of lowering the component condition numbers and increasing solution accuracy as measured from an exact solution. In addition, it was shown that the counter-diagonal elements whose distance from the main diagonal is greater than 1.42 could be set to zero with no significant loss of accuracy. In other words, the full matrices arising from C^{ν} -RBFs could have a sparse matrix structure with a larger bandwidth and still be quite accurate. The main advantage in this present work is the hybrid combination of arbitrary precision arithmetic that controlled the potential instabilities from Schur matrices.

The Hilbert and Vandermonde ill-conditioned linear systems are proxies for the numerical solutions to the C^{ν} -RBF based discretization of PDEs, IEs, and I-PDEs in \mathfrak{R}^d space. Because the C^{ν} -RBFs are nonlinear with respect to the data and evaluation centers, the shape parameters, and exponent, μ , the equation systems are far more varied than either the Hilbert or Vandermonde systems and can be quite ill-conditioned. The C^{∞} -RBF based PDE coefficient matrix is more complicated because it is composed of disjoint inhomogeneous matrix structures belonging to the interior and the boundary problem. The solution space of C^{∞} -RBF based problems can have many local and global minima and minima as well as saddle points. With such a complex solutions landscape of global and local maxima and minima, it is not surprising that in some unusual circumstances a singular solution is possible, see [26].

The algorithm presented here is distinct from domain decomposition. In the BGE-APA scheme all matrix and vector blocks are fully coupled, whereas in classic domain decomposition, one obtains independent solutions on separate subdomains, and iteration is used to blend the separate solutions. If the physical processes being modeled are sufficiently dissipative, then the iterative process can yield reliable results.

It would be an interesting study to optimize the execution time and the number of digits of precision because the desired accuracy of the numerical simulation. Certain engineering applications do not need such high accuracy whereas modeling a health issue for a human would need very high accuracy.

Conflicts of Interest

The author is unaware of any conflicts of interest.

References

- [1] Hardy, R.L. (1971) Multiquadric Equations of Topography and other Irregular Surfaces. *Journal of Geophysical Research*, **76**, 1905-1915. <https://doi.org/10.1029/JB076i008p01905>
- [2] Buhmann, M.D. (2003) Radial Basis Functions. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511543241>
- [3] Kansa, E.J. (1990) Multiquadrics—A Scattered Data Approximation Scheme with Applications to Computational Fluid-Dynamics. II. Solutions to Parabolic, Hyperbolic and Elliptic Partial Differential Equations. *Computers & Mathematics with Applications*, **19**, 147-161. [https://doi.org/10.1016/0898-1221\(90\)90271-K](https://doi.org/10.1016/0898-1221(90)90271-K)
- [4] Macon, N. and Spitzbart, A. (February 1958) Inverses of Vandermonde Matrices. *The American Mathematical Monthly*, **65**, 95-100. <https://doi.org/10.1080/00029890.1958.11989147>
- [5] Emdadi, A., Kansa, E.J., Ali Libre, N., Rahimian, M. and Shekarchi, M. (2008) Stable PDE Solution Methods for Large Multiquadric Shape Parameters. *Computer Modeling in Engineering & Sciences*, **25**, 23-42.
- [6] Benzi, M., Haws, N.J. and Umas, M. (2000) Pre-Conditioning Highly Indefinite and Nonsymmetric Matrices. *SIAM Journal on Scientific Computing*, **22**, 133-153. <https://doi.org/10.1137/S1064827599361308>
- [7] Ling, L. and Kansa, E.J. (2005) A Least Squares Preconditioner for Radial Basis Functions Collocation Methods. *Advances in Computational Mathematics*, **23**, 31-54.
- [8] MPlapack. <https://github.com/nakatamaho/mplapack>
- [9] Holoborodko. <https://www.advanpix.com>
- [10] Dammal, J.W., Highman, N.J. and Schreiber, R. (1992) Block LU Factorization, NASA-BR-97949, *Journal of Numerical Linear Algebra and Applications*.
- [11] Eldersveld, S.K. and Saunders, M.A. (1992) A Block-LU Update for Large-Scale Linear Programming. *SIAM Journal on Scientific Computing*, **13**, 191-201. <https://doi.org/10.1137/0613016>
- [12] Song, X., Jian, L. and Yang, S. (2017) A Chunk Updating LS-SVMs Based on Block Gaussian Elimination Method. *Applied Soft Computing*, **51**, 96-104. <https://doi.org/10.1016/j.asoc.2016.12.004>
- [13] Hilbert, D. (1894) Ein Beitrag zur Theorie des Legendre'schen Polynoms. *Acta Ma-*

- thematica*, **1**, 155-159. <https://doi.org/10.1007/BF02418278>
- [14] Marcus, M. (1992) Vandermonde Matrix, §2.6.2. In: *A Survey of Matrix Theory and Matrix Inequalities*, Dover, New York, 15-16.
- [15] Highman, N.J. (1999) Fast Solution of Vandermonde-Like Systems Involving Orthogonal Polynomials. *IMA Journal of Numerical Analysis*, **8**, 473-486. <https://doi.org/10.1093/imanum/8.4.473>
- [16] Kansa, E.J. and Holoborodko, P. (2017) On the Ill-Conditioned Nature of C^∞ RBF Strong Collocation. *Engineering Analysis with Boundary Elements*, **78**, 26-36. <https://doi.org/10.1016/j.enganabound.2017.02.006>
- [17] Galperin, E.A., Kansa, E.J., Makroglou, A. and Nelson S.A. (1997) Mathematical Programming Methods in the Numerical Solution of Volterra Integral and Integro-Differential Equations with Weakly-Singular Kernel. *Nonlinear Analysis: Theory, Methods & Applications*, **30**, 1505-1513. [https://doi.org/10.1016/S0362-546X\(96\)00340-9](https://doi.org/10.1016/S0362-546X(96)00340-9)
- [18] Galperin, E.A. and Kansa, E.J. (2002) Application of Global Optimization and Radial Basis Functions to the Numerical Solution of Weakly Singular Volterra Integral Equations. *Computers & Mathematics with Applications*, **43**, 439-456. [https://doi.org/10.1016/S0898-1221\(01\)00300-5](https://doi.org/10.1016/S0898-1221(01)00300-5)
- [19] Fedoseyev, A.I., Friedman, M.J. and Kansa, E.J. (2002) Improved Multiquadric Method for all Partial Differential Equations via PDE Collocation on the Boundary. *Computers & Mathematics with Applications*, **43**, 439-455. [https://doi.org/10.1016/S0898-1221\(01\)00297-8](https://doi.org/10.1016/S0898-1221(01)00297-8)
- [20] Kansa E.J. and Carlson, R.A. (1992) Improved Accuracy of Multiquadric Interpolation Using Variable Shape Parameters. *Computers & Mathematics with Applications*, **24**, 99-120. [https://doi.org/10.1016/0898-1221\(92\)90174-G](https://doi.org/10.1016/0898-1221(92)90174-G)
- [21] Wertz, J., Kansa, E.J. and Ling, L. (2006) The Role of the Multiquadric Shape Parameters in Solving Elliptic Partial Differential Equations. *Computers & Mathematics with Applications*, **51**, 1335-1348. <https://doi.org/10.1016/j.camwa.2006.04.009>
- [22] Cervenka, M. and Skala, V. (2022) Conditionality Analysis of the Radial Basis Function Matrix. *Computational Science and Its Applications—ICCSA 2020*, Cagliari, 1-4 July 2020, 30-43. https://doi.org/10.1007/978-3-030-58802-1_3
- [23] Luh, L.T. (2016) The Mystery of the Shape Parameter IV. *Engineering Analysis with Boundary Elements*, **80**, 103-109.
- [24] Luh, L.-T. (2019) The Choice of the Shape Parameter-A Friendly Approach. *Engineering Analysis with Boundary Elements*, **98**, 103-109. <https://doi.org/10.1016/j.enganabound.2018.10.011>
- [25] Kansa, E.J. and Hon, Y.C. (2000) Circumventing the Ill-Conditioning Problem with Multiquadric Radial Basis Functions: Applications to Elliptic Partial Differential Equations. *Computers & Mathematics with Applications*, **39**, 123-137. [https://doi.org/10.1016/S0898-1221\(00\)00071-7](https://doi.org/10.1016/S0898-1221(00)00071-7)
- [26] Hon, Y.C. and Schaback, R. (2002) On Unsymmetric Collocation by Radial Basis Functions. *Applied Mathematics and Computation*, **119**, 177-188. [https://doi.org/10.1016/S0096-3003\(99\)00255-6](https://doi.org/10.1016/S0096-3003(99)00255-6)