

Strong form collocation: input for optimization

E. J. Kansa

Convergent Solutions, Livermore, CA, USA

Abstract

Integrated volumetric methods such as finite elements and their “meshless” variations are typically smoother because of the application of Gauss’s theorem than the strong form finite difference and radial basis function collocation methods. Minimization methods can be either local (such as finite elements) or global; gradient based methods tend to go to the nearest location where the local Jacobian is zero; when the eigenvalues of the Hessian are positive, then that location is either a local or global minimum.

The starting point for the global volume-integrated minimization procedure of Galperin–Zheng, is the set of expansion coefficients obtained from the simple strong form radial basis function (RBF) discretization of the partial differential equations (PDE). The volume integrated RBFs do not require a mesh for integration because these integrated RBFs are evaluated at the global endpoints rather than locally about a test point. This aspect is completely distinct from finite elements or the so-called meshless methods finite element variations that require a tessellation for local integration. Using these globally integrated RBFs and the strong form expansion coefficients in the functional, the evaluation of the functional L_∞ errors are recorded for various choices of data centers, evaluation centers and shape parameter distributions. These L_∞ errors form a complex landscape of maxima and shallow and deep minima. Finding the deepest possible minimum is a complex process and not as simple as local gradient search methods even using information from the Hessian matrix.

Keywords: meshless radial basis functions, multiquadric, strong and weak formulation, partial differential equations, global minimization.



1 Introduction

The minimization of the errors of the discretized multi-variant partial differential, integral, or integro-differential equations is the objective of numerical schemes. Such problems can be cast in terms of seeking the minimum of a function of n variables, $\mathbf{F}(\vec{\mathbf{x}})$, see Harris and Stocker [1]. Define, \mathbf{J} , as the Jacobian to be a vector of the first order partial derivatives of \mathbf{F} , and, \mathbf{H} , the Hessian matrix to be a square matrix of second-order partial derivatives of \mathbf{F} . A Taylor series expansion of \mathbf{F} is given by:

$$\mathbf{F}(\vec{\mathbf{x}} + \Delta \vec{\mathbf{x}}) = \mathbf{F}(\vec{\mathbf{x}}) + \Delta \vec{\mathbf{x}} \mathbf{J} + 1/2 \Delta \vec{\mathbf{x}}^T \mathbf{H} \Delta \vec{\mathbf{x}} + \dots \quad (1)$$

where $\vec{\mathbf{x}}_k$ is a critical point. If the Jacobian, $\mathbf{J}(\vec{\mathbf{x}}_k) = 0$, $\mathbf{F}(\vec{\mathbf{x}}_k)$ may be at a local maximum, minimum, or inflection point. To determine what type of point $\vec{\mathbf{x}}_k$ is, one must examine the eigenvalues of the Hessian matrix at a critical point, $\vec{\mathbf{x}}_k$. The following test can be applied at any critical point for which the Hessian matrix is invertible:

1. If the Hessian possesses all positive eigenvalues at $\vec{\mathbf{x}}_k$, then $\mathbf{F}(\vec{\mathbf{x}}_k)$ is a local minimum.
2. If the Hessian possesses all negative eigenvalues at $\vec{\mathbf{x}}_k$, then $\mathbf{F}(\vec{\mathbf{x}}_k)$ is a local maximum.
3. If the Hessian has both positive and negative eigenvalues at $\vec{\mathbf{x}}_k$, then $\mathbf{F}(\vec{\mathbf{x}}_k)$ is a saddle point.

The finite element method (FEM) is based on minimizing the residual of the interior partial differential equations and boundary conditions in a least-squares sense. The method minimizes a least squares functional that consists of a weighted sum of the residuals; FEM has a nice theoretical foundation in which the least squares functional is minimized. FEM integrates local low order polynomials that requires mesh generation. Mesh generation over irregular domains in higher dimensional domains can present serious implementational difficulties. Another appealing aspect of FEM is the fact that integration increases the convergence rate and Gauss theorem:

$$\int_V (\nabla \cdot \vec{\mathbf{v}}) dV = \int_S (\vec{\mathbf{v}} \cdot \vec{\mathbf{n}}) dS \quad (2)$$

reduces orders of differentiation.

Meshfree methods have been used to solve partial differential, integral, and integro-differential equations; no mesh generation becomes a significant advantage whenever the number of spatial dimensions is greater than or equal to three. Traditional low order compactly supported methods such as finite difference, element, volume methods do require meshes to define the connectivity relations. Low mesh connectivity has the definite computational advantage of sparse systems of equations that are well conditioned or are readily preconditioned. The disadvantages are that these methods possess polynomial convergence and the sparse systems of equations grow increasingly denser along with increasingly poorer conditioning as the spatial dimensionality increases.



Mesh free radial basis functions (RBFs) are particularly attractive because, regardless of the dimensionality of the problem, they are univariate functions of distance (usually Euclidean). RBFs may be globally supported that are either finitely or infinitely differentiable or RBFs can be compactly supported possessing finite differentiability. Only the C^∞ RBFs such as multiquadrics, Gaussian, Mathern splines, etc. possess exponential convergence. The objective is to find efficient methods for the numerical solution of partial differential equations (PDEs), integral equations (IEs), and integro-partial differential equations (IPDEs). Assume that a dependent variable, $U(\vec{x},t)$ is an unknown piece-wise continuous function on $\Omega \in \mathbb{R}^n$. Over the interior, let \mathcal{L} , be a linear or nonlinear hyperbolic, elliptic, or elliptic differential operator with the associated forcing function, $f(\vec{x},t)$. Let \wp be well-posed boundary operators, and let $g(\vec{x},t)$ be the associated forcing boundary condition. The set of well-posed interior and boundary conditions are:

$$\mathcal{L}U=f, \text{ over } \Omega \setminus \partial\Omega, \tag{3}$$

$$\wp U=g, \text{ on } \partial\Omega. \tag{4}$$

Any dependent variable, $U(\vec{x},t)$, see [2–4], can be expanded in terms of the N radial basis functions, ϕ_j , in terms of N expansion coefficients, α_j .

$$U(\vec{x},t) = \sum_{j=1}^N \phi_j(\vec{x})\alpha_j(t). \tag{5}$$

The interpolation matrix is given as:

$$\mathbf{A}_{\text{intp}} \vec{\alpha} = \begin{bmatrix} \Phi_{i,i} & \Phi_{i,b} \\ \Phi_{b,i} & \Phi_{b,b} \end{bmatrix} \begin{bmatrix} \vec{\alpha}_i \\ \vec{\alpha}_b \end{bmatrix} = \begin{bmatrix} \mathbf{U}_i \\ \mathbf{U}_b \end{bmatrix} = \mathbf{U} \tag{6}$$

where i refers to the set of interior points and b refers to boundary points. The interpolation matrix can easily be expanded to account for various degrees of polynomial precision, see [4]. The approximate function, $U(\vec{x},t)$, is expanded in terms of C^∞ RBFs as

$$\phi_j = (1 + (r_j/c_j)^2)^\gamma, \text{ (generalized MQ), } r_j = \|\vec{x} - \vec{y}_j\|, \tag{7}$$

where c_j is the local shape parameter, $\gamma \geq -1/2$, $\{\vec{y}\}$ is the set of data centers, and $\{\vec{x}\}$ is the set of evaluation points; $\{\vec{x}\}$ and $\{\vec{y}\}$ can be distinct or the same. Note that c_j^2 plays the roles of a wavelet dilation parameter. Madych and Nelson [5] and Madych [6] has provided theoretical guidance for optimizing the interpolation procedure. He showed that C^∞ radial basis functions, such as MQ, converge exponentially as:

$$\text{convergence rate} \sim O(\lambda^{\xi-m}), \text{ where } \xi=(c/h), \text{ and } 0 < \lambda < 1, \tag{8}$$

h is the fill distance, and m is the order of differentiation where m is positive for differentiation and negative for integration. The dependent variables may be



continuous, or multi-valued; if multi-valued, then special basis functions can be appended to the set of basis functions, see [4].

Madych also showed differentiation reduces the rate of convergence; however, if the parameter, ξ , is sufficiently large, the derivatives can be very good approximates. Since integration is anti-differentiation, the convergence rate should increase, as observed by Mai-Duy and Tran-Cong [7]. The strong collocation is similar in concept to the finite difference formulation. To find the set of expansion coefficients, assume Eqs (1–2) are well-posed, and let \mathcal{L} and \wp operate upon the interior and boundary respectively, yielding the following system of N equations in N unknowns:

$$\mathbf{H} \vec{\alpha}_{\text{PDE}} = \begin{bmatrix} \mathcal{L}\Phi_{i,i} & \mathcal{L}\Phi_{i,b} \\ \wp\Phi_{b,i} & \wp\Phi_{b,b} \end{bmatrix} \begin{bmatrix} \vec{\alpha}_i \\ \vec{\alpha}_b \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} = \theta. \quad (9)$$

If H is invertible, then

$$\vec{\alpha}_{\text{PDE}} = \mathbf{H}^{-1}\theta. \quad (10)$$

Then with expansion coefficients, $\{\vec{\alpha}_{\text{PDE}}\}$, the numerical approximation to \mathbf{U}_{PDE} is given by

$$\mathbf{U}_{\text{PDE}} = \mathbf{A}_{\text{intp}} \vec{\alpha}_{\text{PDE}}. \quad (11)$$

The L_∞ error used in this document is:

$$L_\infty = \max_{\mathbf{k}} | \mathbf{U}(\vec{\mathbf{x}}_{\mathbf{k}})_{\text{exact}} - \mathbf{U}(\vec{\mathbf{x}}_{\mathbf{k}})_{\text{PDE}} |. \quad (12)$$

2 Dependency of errors on the general parameter set

The usual procedure presented in the meshless RBF literature to solve PDEs, IEs, and IDEs is to choose either a common shape parameter or a distribution, $\{c_j^2\}$, a set of data centers, $\{\vec{\mathbf{y}}\}$ and a set of evaluation points, $\{\vec{\mathbf{x}}\}$, then find a set of expansion coefficients by solving a set of $N \times N$ coupled linear or nonlinear equations, then compare the error norms. The set of adjustable parameters, $\mathbf{Q} = \{\vec{\mathbf{x}}, \vec{\mathbf{y}}, c_j^2, \vec{\alpha}\}$ determines how well the MQ expansion approximates the true solution; hence, different sets of \mathbf{Q} will produce either very deep minimum norms, shallow norms, saddlepoints, or even maximum norms of varying heights. The objective to find the deepest minimum error norm possible that is acceptable.

The process of finding the expansion coefficients, $\{\vec{\alpha}\}$, by solving systems of linear equations is unfortunately not always stable. The expansion coefficients, $\{\vec{\alpha}\}$, depend upon the parameter subset, $\{\vec{\mathbf{x}}, \vec{\mathbf{y}}, c_j^2\}$.

$$\vec{\alpha} = \vec{\alpha}(\vec{\mathbf{x}}, \vec{\mathbf{y}}, c_j^2). \quad (13)$$

Unless one is exceedingly fortunate, a single guess of $\mathbf{Q}_1 = \{\vec{\mathbf{x}}, \vec{\mathbf{y}}, c_j^2\}$ will not yield the minimum value of the functional, \mathcal{F} . Unless one is exceedingly fortunate, an initial guess of \mathbf{Q}_1 will not yield the minimum value of the functional. Near the candidate minimum, $\vec{\mathbf{x}}_k$, one can switch to the Newton–Raphson method; if

\vec{x}_k is within the ball of convergence, the Newton–Raphson method will converge quadratically to the local root, according to the Kantorovich theorem.

The global minimization procedure would be greatly simplified if the C^∞ RBFs were ortho-normal, because one optimized basis function at a time could be added to the optimized basis set. Although the modified Gram-Schmidt algorithm is superior to the standard Gram-Schmidt algorithm, it unfortunately can lose orthogonality due to finite precision round-off errors, see Paige *et al.* [8]. In a future study, various studies of the number of digits will be investigated in the modified Gram-Schmidt algorithm. However in the present study, the partial set, $Q_1 = \{\vec{x}, \vec{y}, c_j^2\}$ is specified, and the set of $\{\vec{\alpha}_{PDE}\}$ is found. Although the optimization tools are not optimal, some advantages can be gained from experience such as:

- Using more spatial refinement in high gradient regions as indicated by the forcing interior and boundary conditions,
- Extending the interior domain slightly outside of the computational boundaries, see Fedoseyev *et al.* [14].
- Following Wertz *et al.* [15], allow $\{c_j^2\}_{\partial\Omega} > \{c_j^2\}_{\Omega \setminus \partial\Omega}$.
- Some insight of the location of fine length scales can be gained from the effect of the source functions, $f(\vec{x}, t)$ over $\Omega \setminus \partial\Omega$ and $g(\vec{x}, t)$ on $\partial\Omega$.
- In addition, Heryudonol and Driscoll [23] and Libre *et al.* [16] describe techniques to refine certain regions efficiently.

Using either a very fine discretization everywhere in the domain may have a firm theoretical foundation, but it leads to ill-conditioned system of equations on finite precision computers. On real world computers, compromises must be made to obtain the objective of the most accurate results in the most efficient manner possible, such as the use of extended precision. It is not efficient, on a CPU count, to use uniform very fine discretization over the entire domain, Ω , unless the length scale is small everywhere.

The search for the optimal method to solve PDEs, IEs, and IDEs is evolving. Huang *et al.* [9] and Cheng [10] demonstrated that when using extended precision arithmetic, a very coarse distribution of data and evaluation centers coupled with large shape parameters yields extremely accurate numerical solutions and is very computationally efficient. Intuitively, this seems wrong because the CPU time for multiple precision is much longer than for double precision. The reason why intuition is incorrect is that although the CPU time per data and evaluation center is larger with extended precision, the very fact that the total number of points for specific target accuracy is orders of magnitude smaller, hence the total cost is reduced many orders of magnitude.

3 Forming the global minimization method

A general unifying approach is presented by Galperin and Zheng [11, 12] to solve PDEs, IEs, and IDEs as a global minimization problem in which the interior operator, acting upon the C^∞ RBFs is volume integrated over $\Omega \setminus \partial\Omega$ and the boundary operator, \wp , acting upon the C^∞ RBFs is integrated over each $\partial\Omega_k$.



The functional, \mathcal{F} , to be minimized, over the parameter set Q is:

$$\min_{\mathbf{q} \in Q} \mathcal{F} = \varpi \left| \int_{\Omega \setminus \partial\Omega} (\mathcal{L}\mathbf{U} - \mathbf{f}) d\vec{\mathbf{x}} \right| + (1 - \varpi) \left| \int_{\partial\Omega} (\wp\mathbf{U} - \mathbf{g}) d\vec{\mathbf{x}} \right| \leq \eta \quad (14)$$

where Q is the set of free parameters, $Q = \{ \vec{\mathbf{x}}, \vec{\mathbf{y}}, c_j^2, \alpha \}$, $\varpi < 1$ and η is a prescribed error criterion. Note that unlike the strong collocation method, there is a prescribed amount of “fuzziness” in which the solution trajectory is allowed to possess.

The integrations can be performed globally without the need for tessellation as required by finite elements or the finite-element meshless methods. The operators, \mathcal{L} and \wp , act upon the RBFs first, then the results are integrated. Fortunately, in 2D, the integrals of $\mathcal{L}\phi(\vec{\mathbf{x}} - \vec{\mathbf{y}}_j) d\vec{\mathbf{x}}$ and $\wp\phi(\vec{\mathbf{x}} - \vec{\mathbf{y}}_j) d\vec{\mathbf{x}}$ are integrated analytically and stored as vectors.

The integrated results are stored as vectors:

$$\psi(\vec{\mathbf{y}}_j) = \int_{\Omega \setminus \partial\Omega} \mathcal{L}\phi(\vec{\mathbf{x}} - \vec{\mathbf{y}}_j) d\vec{\mathbf{x}}, \quad (15)$$

$$\chi(\vec{\mathbf{y}}_j) = \int_{\partial\Omega} \wp\phi(\vec{\mathbf{x}} - \vec{\mathbf{y}}_j) d\vec{\mathbf{x}}. \quad (16)$$

Then, the vectors are used to form a specific value of $\mathcal{F}(\mathbf{q})$:

$$\begin{aligned} \min_{\mathbf{q} \in Q} \mathcal{F}(\mathbf{q}) = & \varpi \left| \sum_{j=1}^N \{ \vartheta(\vec{\mathbf{y}}_j) \vec{\alpha}_{\text{PDE}} - \int_{\Omega \setminus \partial\Omega} \mathbf{f} d\vec{\mathbf{x}} \right| + (1 - \varpi) \left| \right. \\ & \left. \times \sum_{j=1}^N \chi(\vec{\mathbf{y}}_j) - \int_{\partial\Omega} \mathbf{g} d\vec{\mathbf{x}} \right| \leq \eta. \end{aligned} \quad (17)$$

If one wishes to have a finite element analog, then local integration over the tessellation can yield N equations in N unknowns.

Nonlinear optimization problems may have multiple local and global maxima, minima, and saddle regions. Gradient based methods such as steepest descent, Newton–Raphson or combinations thereof may only lead to the nearest local or global minimum. Searching the complicated multi-dimensional landscape for global minima can be exceedingly time-consuming since such minima may be clustered closely to widely dispersed in parameter space. Although theoretically, the Galperin and Zheng procedure eliminates the need to solve sets of linear or nonlinear equations, global minimization requires a huge number of evaluations of the functional, \mathcal{F} . Fast reliable global minimization algorithms capable of dealing with at least $4N$ parameters are not yet perfected. One way to minimize the number of functional evaluations is reduce the total number of free parameters.

4 Implementation of the Galperin–Zheng method

Experience has shown that the set of data and evaluation centers, optimal shape parameters and expansion coefficients are inter-related. Changing either the location and fill distance of the data centers and evaluation centers changes the magnitude of the parameter, $\gamma = c/h$, that, in turn, changes the condition number of the system involved in determining the expansion coefficients. While a uniform mesh is nice for analysis, it is wasteful in higher dimensional problems ($d > 3$), see Kansa and Geiser [17]. The PDE forcing functions over $\Omega \setminus \partial\Omega$ and boundary conditions on $\partial\Omega$ usually correlate with the loci of maxima, minima, and inflection points. By having a minimal number of data centers in such regions, a dense covering is not necessary.

Aside from the general recipe of Madych and Nelson [5] and Madych [6] showing that the ratio of c/h should be as large as possible, practical recipes for choosing either a uniform or variable shape distribution still require more development. Luh [18–22] has made significant progress for interpolation problems for some of the more popular RBFs. Kansa [2, 3] found using the MQ RBF that, if the solutions are either monotonic increasing or decreasing, then the following power law works well:

$$c_j^2 = c_{\min}^2 * (c_{\max}^2 / c_{\min}^2)^{(j-1)/(N-1)}, \quad j=1,2,\dots,N, \quad (18)$$

where c_{\max}^2 and c_{\min}^2 are input estimates of the squares of the largest and smallest length scales. Wertz *et al.* [15] demonstrated that the $\{c_j^2\}_{\partial\Omega}$ associated with boundaries should be larger than those associated with the interior.

The usual procedure is to choose a set of N $\{\vec{x}\}$ and $\{\vec{y}\}$ and either a single shape parameter or a distribution of shape parameters, then solve an $N \times N$ set of equations to find the expansion coefficients. To overcome the limitations of double precision, the MATLAB compatible multi-precision package was obtained from www.advanpix.com. To improve the convergence rate, ξ should become very large. This can be accomplished in two ways: (1) The h-scheme (spatial refinement) increases the number, N_h , of data centers, but a relatively small $\langle c_j^2 \rangle$. (2) The c-scheme increases c , but requires a significantly smaller number of data centers, N_c , where $N_c \ll N_h$; the c-scheme requires $O(N_c^3) \ll O(N_h^3)$ operations. For purposes of efficiency, the c-scheme is superior and preferable over the h-scheme. The disadvantage of either the h-scheme, or the c-scheme is that as $\xi \rightarrow \infty$, the system of linear equations becomes very ill-conditioned and subject to extreme round-off errors on computers with limited arithmetic precision.

5 Numerical results

There are several steps involved in calculating the Galperin–Zheng global minimization, see Eq (17). Firstly, the data centers are tensor product meshes; in each coordinate direction, the well-known geometric progression of increasing finer data centers were generated as x_1 or x_2 approaches unity. The boundary itself



is defined by the corners: (0,0), (1,0), (1,1), and (0,1). Following Fedoseyev *et al.* [14], extra data centers are placed slightly beyond the boundary locus by adding 1D randomly generated numbers multiplied by $2.03e-3$ in the x_1 -direction and $2.51e-3$ in the x_2 -direction. There are N_{x_1} points along the x_1 -axis and N_{x_2} points along the x_2 -axis.

The test problem is the Poisson equation over a unit 2D square:

$$\nabla^2 U(x_1, x_2) = (a^2 + b^2) \exp(ax_1 + bx_2) \text{ over } \Omega, \quad (19)$$

(a and b are parameters that will be varied) with the following Dirichlet conditions on

$$\begin{aligned} U(x_1 = 0, x_2) &= \exp(ax_1) \text{ on } \partial\Omega(x_2 = 0) \\ \text{and } U(x_1, x_2) &= \exp(bx_2) \text{ on } \partial\Omega(x_1 = 0), \end{aligned} \quad (20)$$

and Neumann conditions on the opposite boundaries:

$$\begin{aligned} \partial U / \partial x_1 &= a \cdot \exp(ax_1 + b) \text{ on } \partial\Omega(x_2 = 1) \\ \text{and } \partial U / \partial x_2 &= b \cdot \exp(a + bx_2) \text{ on } \partial\Omega(x_1 = 1). \end{aligned} \quad (21)$$

The exact solution is:

$$U(x_1, x_2) = \exp(ax_1 + bx_2). \quad (22)$$

The test domain was chosen over a unit square, and various exponential parameters for the forcing function, f , were chosen: The parameters, a and b were specified as $a = 2$ and $b = 3$. The maximum values of U at $x_1 = x_2 = 1$, is, $U = \exp(5) = 148.4132$.

Since most scientists and engineers are familiar with finite difference, element, or volume methods, the customary procedure is to use extremely fine meshing. This approach is workable for 2 and 3-dimensional problems, but impractical for 6D Boltzmann equations due to the curse of dimensionality. To minimize the number of data centers, a geometric progression of increasingly finer discretization was used, and the origin was appended in the construction of the tensor product mesh. In the paper of Fedoseyev *et al.* [14], the computational domain, Ω was extended slightly beyond the boundaries, $\partial\Omega_k$ given by the line segments of the unit square: (0,1) (0,0), (0,1) (1,1) (0,1), The domain was extended to 1.022 in the x_1 direction and 1.02412 in the x_2 direction.

The highlights of the parameter tests are summarized below:

- The shape parameters corresponding to either the two Dirichlet and two Neumann boundary data centers were multiplied by adjustable input factors, see Wertz *et al.* [15].
- While the power law recipe seems to work for monotonic functions, and yields the desired wavelet relations, see [4]; this power-law recipe is too simplistic and is used only as a temporary measure.



- Since the maximum value of U occurs at $x_1 = x_2 = 1$, the geometric progression algorithm produced successively finer discretization in the x_1 and x_2 directions. The first point was fixed at 0.16 for x_1 and 0.18 for x_2 .
- Afterwards, the minimum values of x_1 and x_2 was -0.02 and the maximum values of x_1 and x_2 was 1.025 were appended to the tensor product 2D mesh. The boundary locus line segments : (0,0), (0,1), (1,1), (1,0) were also added.. and
- The set of expansion coefficients, $\{\vec{\alpha}_{PDE}\}$, was obtained by solving the point collocation for the PDE and boundary conditions.
- Using the set of expansion coefficients, $\{\vec{\alpha}_{PDE}\}$, and the integrated basis functions over $\Omega \setminus \partial\Omega$ and $\partial\Omega$, the global L_∞ error was calculated for various choices of evaluation, data, $\{c_j^2\}$ tuneable parameters. and number of digits of precision.

6 Summary of results

The L_∞ errors obtained from solving the set PDE expansion coefficients, $\{\alpha_{PDE}\}$, calculating U_{PDE} with these expansion coefficients relative to the exact solution ranged from 0.186 to 2e-28 with the total number of points ranging from 12 to 110. Intuitively, systems of equations with good condition numbers should yield very accurate answers. However, consider the following counter examples:

$N = 56, n_{digit} = 16, K_{pde} = 8.939e+13, \text{ and } L_\infty = 0.469924,$

$N = 56, n_{digit} = 128, K_{pde} = 2.199e+13, \text{ and } L_\infty = 0.0.186132.$

To analyze numerically the question whether the weak formulation is viable, several steps are required.

1. Choose the distribution of $\{\vec{x}\}$ and $\{\vec{y}\}$ and the extent that the interior points extend beyond the boundary.
2. Choose a distribution of shape parameters and the multiplicative factors for the Neumann and Dirichlet shape parameters.
3. Construct the MQ basis functions and the integrated 2D RBFs for the interior and the 1D integrated Neumann and Dirichlet RBFs, and choose the weighting factor, ϖ , for the interior and boundary problem.
4. Construct the strong form system of equation and find $\{\alpha_{PDE}\}$
5. Substitute $\{\alpha_{PDE}\}$ into the strong form and reject if the L_∞ error exceeds.
6. Continue varying the input parameters until the deepest L_∞ error is achieved.

$$\min_{\mathbf{q} \in \mathcal{Q}} \mathcal{F}(\mathbf{q}) = \varpi \left| \int_{\Omega \setminus \partial\Omega} (\mathcal{L}\mathbf{U}-\mathbf{f})d\vec{\mathbf{x}} \right| + (1 - \varpi) \left| \int_{\partial\Omega} (\wp\mathbf{U}-\mathbf{g})d\vec{\mathbf{x}} \right| \leq \eta. \quad (23)$$

The following table shows the constants a and b for the forcing term, ax_1 and ax_2 that are the first terms, respectively for the geometric progression of the x_1 and x_2 coordinates, c_{max}^2 and c_{min}^2 are the maximum and minimum parameters for the power law shape parameter distribution, χ_{Dir} and χ_{Neu} are the multiplicative parameters to modify the shape parameters associated with the Dirichlet and Neumann boundary coordinates, respectively, K_{PDE} is the condition number



Table 1: RMS errors of the Galperin–Zheng formulation: first term of geometric progression: $ax_1 = 0.27$, $ax_2 = 0.277$ and exponential factors: $a = 2$, $b = 3$ held constant.

	n_{digits}	c_{max}^2	c_{min}^2	χ_{Dir}	χ_{Neu}	K_{PDE}	$error_{strong}$	$\mathcal{F}(\mathbf{q})$
N	32	1.021	0.9149	0.5	0.85	3.2210^{13}	1.8910^{-25}	0.0033
56	128	1.021	0.9149	0.5	0.85	3.6410^{13}	1.8910^{-25}	0.0022
56	512	1.0e5	0.0019	0.0007	79.31	3.6410^{13}	2.4410^{-25}	0.0022
56	1024	1.0e5	0.0019	0.0007	90.0	3.2210^{13}	2.9110^{-25}	0.0011
63	512	1.021	0.9149	1	1	3.6410^8	3.4910^{-30}	0.0003
90	32	1.021	0.9149	0.5	0.85	1.4610^{18}	1.2710^{-22}	0.0060
90	128	1.021	0.9149	0.5	0.85	2.4810^{18}	1.3610^{-22}	0.0002

associated with the strong form of the PDE matrix, $error_{strong}$ and $error_{weak}$ are the strong and weak form L_∞ norm errors.

Table 1 shows the results of increasing the number data centers for the Galperin–Zheng weak formulation. During previous testing, and using an increasing number of digits of precision, the values of c_{min}^2 and c_{max}^2 were pushed, fixing the set, $\{c_j^2\}$, using the set $\{\alpha\}$ from the solution of the point collocation scheme, and fixing the starting points of the geometric progression for the tensor product meshes.

Note that increasing the number of data centers, h-refinement, while fixing the set, $\{c_j^2\}$ does show successively smaller RMS errors, but it is questionable whether the extra effort is warranted. The point of this exercise is to provide a good starting point for the global optimization process in which all free parameters are varied to find very deep global minima instead of a local minimum.

7 Discussion

The papers of Galperin and Zheng [11, 12] offer a general framework for the numerical solution of PDEs, IEs, and IDEs. The test solutions are expanded in terms of C^∞ RBFs, then operated upon by the domain and boundary operators that are volume and surface integrated, respectively. A functional $\mathcal{F}(\mathbf{q})$ is constructed from these test expansions that is constrained to be less than or equal to an error bound. The functional is minimized by global optimization methods that are unfortunately not very robust given a large number of parameters. Presently, a decoupling process is required in which the set of data and evaluation centers are input parameters as well as either a uniform or variable shape parameter distribution, then the expansion coefficients are solved as a set of possible ill-conditioned equation systems requiring extended arithmetic precision.

Unless some breakthrough in the global minimization algorithm is developed, parameter splitting methods will need to be continued to be used. Optimizing the minimization procedure will require further research and development.

References

- [1] Harris, J.W., Stocker, H., *Handbook of Mathematics and Computational Science*, Springer Verlag, New York, 1998.
- [2] Kansa, E.J., *Comput Math Appl.*, **19**(8/9): pp.127–147, 1990.
- [3] Kansa, E.J., *Comput. Math. Appl.*, **19**(6–8): pp.147–161, 1990.
- [4] Sarra, S.A., Kansa, E.J., Multiquadric Radial Basis Function Approximation Methods for the Numerical Solution of Partial Differential Equations, *Advances in Computational Mechanics*, Vol. 2, 2009, ISSN:1940–5820.
- [5] Madych, W.R., Nelson S.A., *Math. Comp.* 54: pp.211–230, 1990.
- [6] Madych, W.R., *Comput. Math. Applic.* 24(12): pp.121–138, 1992.
- [7] Mai-Duy, N., Tran-Cong, T., *Int. J. Numer. Methods in Fluids* 37(1): pp.65–86, 2001.
- [8] Paige, C.C., Rosloznic, M., Strakos. Z., *SIAM J. Matrix Anal. Applic.* v28, pp.264–284, 2006.
- [9] Huang, C.-S., Lee, C.-F., Cheng, A.H.D., *Eng. Anal. Bound. Elem.* 31: pp.615–623, 2007.
- [10] Cheng, A.H.D., *Eng. Anal. Bound Elem.* 36: pp.220–239, 2012.
- [11] Galperin, E.A., Zheng, Q., *Comput. Math. Applic.* 25(10/11): pp.103–118, 1993.
- [12] Galperin, E.A., Zheng, Q., *Comput. Math. Applic.* 25(10/11): pp.119–124, 1993.
- [13] Galperin, E.A., Kansa, E.J., Makroglou. A., Nelson, S.A., *J. Comput. Appl. Math.* 115: pp.193–211, 2000.
- [14] Fedoseyev, A.I., Friedman, M.J., Kansa, E.J., *Comput. Math. Appl.* 43(3–5): pp.491–500, 2002.
- [15] Wertz, J., Kansa, E.J., Ling, L., *Comput. Math. Applic.* 51(8): pp.1335–1348, 2006.
- [16] Libre, N.A., Emdadi, A., Kansa, E.J., Shekarchi, M., Rahimian, M., *Eng. Anal. Bound. Elem.* 33: pp.901–914, 2009.
- [17] Kansa, E.J. and Geiser, J., *Eng. Anal. Bound. Elem.* 37: pp.637–645, 2013.
- [18] Luh, L.-T., *Inter. J. Numer. Methods Appl.* 1(2): pp.101–120, 2009.
- [19] Luh, L.-T., arXiv:pp.1002.2082, 2010.
- [20] Luh, L.-T., *Eng. Anal. Boundary Elem.* 37(5): pp.988–993, 2013.
- [21] Luh, L.-T., arXiv:pp.1004.0759, 2010.
- [22] Luh, L.-T., *Inter. J. Numer. Methods Appl.* 1(2): pp.155–174, 2009.
- [23] Heryudono, A.R.H., Driscoll, T.A. Radial Basis Function Interpolation on Irregular Domain through Conformal Transplantation, *Journal of Scientific Computing*, 44, pp. 289–300, 2010.

