Improved Multiquadric Method for Elliptic Partial Differential
Equations via PDE Collocation on the Boundary

A. I. Fedoseyev*, M. J. Friedman† and E. J. Kansa‡
*Center for Microgravity and Materials Research,
University of Alabama in Huntsville, Huntsville, AL 35899
†Department of Mathematical Sciences,
University of Alabama in Huntsville, Huntsville, AL 35899
‡Embry-Riddle Aeronautical University, East Bay Campus, Oakland, CA 94621

October 2, 2000

Abstract

The Multiquadric Radial Basis Function (MQ) Method is a recent meshless collocation method with
global basis functions. It was introduced for discretizing partial differential equations (PDEs) by Kansa
in early nineties. The MQ method was originally used for interpolation of scattered data, and it was
shown to have exponential convergence for interpolation problems.

In [12] we have extended the Kansa-MQ method to numerical solution and detection of bifurcations
in 1D and 2D parametrized nonlinear elliptic PDEs. We have found there that the modest size nonlinear
systems resulting from the MQ discretization can be efficiently continued by a standard continuation
software, such as AUTO. We have observed high accuracy with small number of unknowns, as compared
with most known results from the literature.

In this paper we formulate an improved Kansa-MQ method with \textit{PDE collocation on the boundary}
(MQ PDECB): we add an additional set of nodes (which can lie inside or outside of the domain) adjacent
to the boundary and, correspondingly, add an additional set of collocation equations obtained via
collocation of the PDE on the boundary. Numerical results are given that show a considerable improvement
in accuracy of the MQ PDECB method over the Kansa-MQ method, with both methods having
exponential convergence with essentially the same rates.

\textbf{Keywords:} Radial basis functions, multiquadric method, numerical solution, continuation, bifurca-
tions, nonlinear elliptic PDEs.

1 Introduction.

The Multiquadric Radial Basis Function (MQ RBF or, simply, MQ) method is a recent meshless collocation
method, with global basis functions, for discretizing PDEs. It was originally proposed in 1970 [20], [21] for
interpolation of scattered data and was shown [28], [29], [33] to have an exponential convergence for function
approximation. The MQ method was introduced for solving PDEs in Kansa [25], [26] in early nineties. Since
then it was successfully applied for solving a number of 2D and 3D PDEs, see e.g. [5], [19], [30], [18], [22], [8]
and references there, while some convergence results for solving PDEs, based directly on the interpolation
error estimates, appeared only recently [15, 16]. Application of the MQ method to PDEs leads to finite
dimensional problems with full matrices. We also mention a recent paper [27], where MQ basis functions
with compact support were constructed, yielding banded collocation matrices of arbitrary band width.

The Kansa-MQ method was shown to give high accuracy with a relatively small number of unknowns
(tens or hundreds for 2D problems). The corresponding linear systems can be efficiently solved by direct

\*E-mail: alex@uhht.mn.uah.edu
†E-mail: friedman@math.uah.edu
‡E-mail: kansa1@lnl.gov
methods. In [12] we have extended the Kansa-MQ method to numerical solution of parametrized nonlinear elliptic PDEs. We presented there results of our numerical experiments with continuation of solutions to and detection of bifurcations in 1D and 2D nonlinear elliptic PDEs. We found that the modest size nonlinear systems resulting from the MQ discretization can be efficiently continued by a standard continuation software, such as AUTO [6].

Our observations have shown that the residual error is typically largest near the boundary (by one to two orders) compared to the residual error in the domain far from the boundary.

In this paper we formulate an improved Kansa-MQ method with PDE collocation on the boundary (PDECB): we add an additional set of nodes (which can lie inside or outside of the domain) adjacent to the boundary and, correspondingly, add an additional set of collocation equations obtained via collocation of the PDE on the boundary. The motivation for this modification of the Kansa-MQ method comes from our observations that 1) the residual is typically the largest near the boundary (by one to two orders larger than in the domain far away from the boundary), and 2) the residual is dramatically reduced when we use the PDE collocation on the boundary. The MQ PDECB method leads not only to a higher accuracy, but, for nonlinear problems, also to a higher efficiency due to the reduction of the number of unknowns in the continuation process by using a preprocessing.

We apply our MQ PDECB method to several model 1D and 2D linear and nonlinear elliptic PDEs and present results of our numerical experiments. These results demonstrate considerable improvement in convergence of the MQ PDECB method over the Kansa-MQ method, with both methods having exponential convergence with essentially the same rates. To our knowledge, this is the first demonstration of the exponential convergence for the MQ method applied to PDEs.

A related idea was successfully used for high Re number fluid flows in the cases of the RNS model [9], [13] and the Alexeev hydrodynamics equations [11] (in the framework of the finite element method) for the solution of 3D thermo-vibrational flows [10].

A class of global numerical methods for 1D and 2D problems, the numerical algorithms without saturation, was proposed in early eighties in [1]. This class includes a highly accurate discretization method for PDEs based on Chebyshev polynomials. This method was further developed in [3], [4], where it was found to be more accurate and better conditioned than the spectral method.

In Section 2 we formulate the Kansa-MQ and the MQ PDECB methods for a linear elliptic PDE. In Section 3 we describe in detail the Kansa-MQ and the MQ PDECB methods for continuation of solutions to parametrized nonlinear elliptic PDEs. For clarity of presentation, Section 3 is written independently of Section 2. In Section 4 numerical examples are given that illustrate the accuracy of our method. In Section 5 we summarize our results.

2 A linear elliptic PDE.

We consider a well-posed elliptic boundary value problem: for given \( g(x) \), \( f(x) \) find \( u(x) \) from

\[
Lu(x) = f(x), \quad \text{in } \Omega \subset \mathbb{R}^d, \\
Bu(x)|_{\partial \Omega} = g(x),
\]

where \( \Omega \) is a bounded domain with the boundary \( \partial \Omega \), \( L \) is a linear elliptic partial differential operator, and \( B \) is a boundary operator.

2.1 The Kansa-MQ method.

Introduce a set \( \Theta_k \) of nodes (Fig. 1)

\[
\Theta_k = \left\{ \{x_i\}_{i=1}^N \subset \Omega, \quad \{x_i\}_{i=N+1}^{N+N_b} \subset \partial \Omega \right\}
\]

and the MQ basis functions

\[
g_j(x) \equiv g_j(c_j, x) = \sqrt{||x-x_j||_2 + c_j^2}, \quad j = 1, \ldots, N + N_b, \quad g_{N+N_b+1}(x) = 1,
\]
where \( x, x_j \in \mathbb{R}^d \) and \( \| x - x_j \|_{\mathbb{R}^d} \) is the Euclidean norm in \( \mathbb{R}^d \), \( c_j \geq 0 \) are called shape parameters [26]. We look for the approximate solution \( u_h \) to (1) in the form

\[
u_h(x) = \sum_{j=1}^{N+N_b+1} a_j g_j(x), \tag{4}\]

Substituting \( u_h(x) \) into (1) and using collocation at the nodes \( \Theta_h \), we obtain the finite dimensional problem

\[
\begin{align*}
L \left( \sum_{j=1}^{N+N_b+1} a_j g_j(x_i) \right) &= \sum_{j=1}^{N+N_b+1} a_j L g_j(x_i) = f(x_i), \quad i = 1, \ldots, N, \\
B \left( \sum_{j=1}^{N+N_b+1} a_j g_j(x_i) \right) &= \sum_{j=1}^{N+N_b+1} a_j B g_j(x_i) = g(x_i), \quad i = N + 1, \ldots, N + N_b, \\
\sum_{j=1}^{N+N_b} a_j &= 0.
\end{align*} \tag{5}
\]

Introducing the notation: \( a = (a_1, \ldots, a_{N+N_b+1})^T \), \( b = (f(x_1), \ldots, f(x_N), g(x_{N+1}), \ldots, g(x_{N+N_b}), 0)^T \in \mathbb{R}^{N+N_b+1} \)

\[
L_g = \begin{bmatrix} L g_1(x_1) & \cdots & L g_{N+N_b+1}(x_1) \\ \vdots & \ddots & \vdots \\ L g_1(x_N) & \cdots & L g_{N+N_b+1}(x_N) \end{bmatrix}, \\
B_g = \begin{bmatrix} B g_1(x_{N+1}) & \cdots & B g_{N+N_b}(x_{N+1}) & B g_{N+N_b+1}(x_{N+1}) \\ \vdots & \ddots & \vdots & \vdots \\ B g_1(x_{N+N_b}) & \cdots & B g_{N+N_b}(x_{N+N_b}) & B g_{N+N_b+1}(x_{N+N_b}) \end{bmatrix}, \quad W = \begin{bmatrix} L_g \\ B_g \end{bmatrix}, \tag{6}
\]

we can rewrite the system (5) in the matrix form as

\[
W a = b, \tag{7}
\]

whose solution is

\[
\alpha = W^{-1} b. \tag{8}
\]
2.2 The MQ PDECB method.

Introduce a set $\Theta_h^1$ of nodes (see Fig. 1)

$$\Theta_h^1 = \left\{ \{x_i\}_{i=1}^{N} \subset \Omega, \quad \{x_i\}_{i=N+1}^{N+N_\ell} \subset \partial \Omega, \quad \{x_i\}_{i=N+N_\ell+1}^{N+2N_\ell} \subset \mathbb{R}^d \setminus \partial \Omega \right\},$$

(9)

where the nodes $\{x_i\}_{i=N+1}^{N+N_\ell+1}$, which can be inside $\Omega$ or outside $\overline{\Omega}$, are adjacent to the boundary $\partial \Omega$, and the MQ basis functions

$$g_j(x) = g_j(c_j, x) = \sqrt{\|x - x_j\|_2^2 + e_j^2}, \quad j = 1, ..., N + N_\ell, \quad g_{N+2N_\ell+1}(x) = 1.$$

(10)

We look for the approximate solution $u_h$ to (1) in the form

$$u_h(x) = \sum_{j=1}^{N+2N_\ell+1} a_j g_j(x),$$

(11)

Substituting $u_h(x)$ into (1) and using collocation at the nodes $\Theta_h^1$, we obtain the finite dimensional problem

$$L \left( \sum_{j=1}^{N+2N_\ell+1} a_j g_j(x_i) \right) = \sum_{j=1}^{N+2N_\ell+1} a_j L g_j(x_i) = f(x_i), \quad i = 1, ..., N + N_\ell,$$

$$B \left( \sum_{j=1}^{N+2N_\ell+1} a_j g_j(x_i) \right) = \sum_{j=1}^{N+2N_\ell+1} a_j B g_j(x_i) = g(x_i), \quad i = N + 1, ..., N + N_\ell,$$

$$\sum_{j=1}^{N+2N_\ell} a_j = 0.$$

(12)

Introducing the notation: $a = (a_1, ..., a_{N+2N_\ell+1})^T$, $b = (f(x_1), ..., f(x_{N+N_\ell}), g(x_{N+N_\ell+1}), ..., g(x_{N+2N_\ell}), 0)^T \in \mathbb{R}^{N+2N_\ell+1}$,

$$L_g = \begin{bmatrix}
Lg_1(x_1) & \ldots & Lg_{N+2N_\ell+1}(x_1) \\
\vdots & \ddots & \vdots \\
Lg_1(x_{N+N_\ell}) & \ldots & Lg_{N+2N_\ell+1}(x_{N+N_\ell})
\end{bmatrix},$$

$$B_g = \begin{bmatrix}
Bg_1(x_{N+1}) & \ldots & Bg_{N+2N_\ell+1}(x_{N+1}) \\
\vdots & \ddots & \vdots \\
Bg_1(x_{N+N_\ell}) & \ldots & Bg_{N+2N_\ell+1}(x_{N+N_\ell}) \\
1 & \ldots & 1
\end{bmatrix}, \quad W = \begin{bmatrix}
L_g \\
B_g
\end{bmatrix},$$

(13)

we can rewrite the system (12) in the matrix form as

$$Wa = b.$$

(14)

3 Continuation for nonlinear elliptic PDEs.

Consider a boundary value problem for a second order system of $n$ parametrized nonlinear elliptic PDEs:

$$F(u(x), \lambda) \equiv D(\lambda) \Delta u - f(\nabla u, x, \lambda) = 0, \quad \text{in} \quad \Omega \subset \mathbb{R}^d, \lambda \in \mathbb{R}, u(\cdot) \in \mathbb{R}^n,$$

$$Bu(x)|_{\partial \Omega} = 0,$$

(15)

where $\Omega$ is a bounded domain, $D(\lambda)$ is a positive diagonal $n \times n$ matrix, $f$ is smooth, and $B$ is a boundary operator which we assume, for simplicity, to be linear. For the bifurcation analysis in the process of continuation we also need to consider the eigenvalue problem for the linearization $D_1 F(u, \lambda)$ of $F$ about the solution $u$ of (15)

$$D_1 F(u, \lambda)v(x) = \mu v(x), \quad \text{in} \quad \Omega,$$

$$Bu(x)|_{\partial \Omega} = 0.$$
3.1 The Kansa-MQ method.

To formulate the approximate problem, we first introduce the set $\Theta_h$ of nodes

$$\Theta_h = \left\{ \{x_i\}_{i=1}^N \subset \Omega, \quad \{x_j\}_{j=N+1}^{N+N_h} \subset \partial \Omega \right\}$$

(17)

and the MQ basis functions,

$$g_j(x) \equiv g_j(c_j, x) = \sqrt{|x - x_j|^2 + c_j^2}, \quad j = 1, \ldots, N + N_h, \quad g_{N+N_h+1}(x) = 1.$$

(18)

The same points $x_i, \ i = 1, \ldots, N + N_h$ will be used as the collocation points. We next define an MQ finite dimensional subspace

$$S_h := \left\{ \chi = \sum_{j=1}^{N+N_h+1} a_j g_j(x) : \sum_{j=1}^{N+N_h} a_j = 0, \quad B \chi(x_i) = 0, \quad i = N + 1, \ldots, N + N_h \right\}. \quad (19)$$

The problems (15) and (16), respectively, are approximated by the collocation equations

$$F(u_h(x), \lambda) = 0, \quad u_h \in S_h, \quad i = 1, \ldots, N, \quad (20)$$

$$Lv_h(x_i) \equiv D_1 F(u_h, \lambda)v_h(x_i) = \mu v_h(x_i), \quad v_h \in S_h, \quad i = 1, \ldots, N. \quad (21)$$

Substituting

$$u_h(x) = \sum_{j=1}^{N+N_h+1} a_j g_j(x), \quad (22)$$

$$v_h(x) = \sum_{j=1}^{N+N_h+1} b_j g_j(x) \quad (23)$$

into (20) and (21), respectively, and using the definition (19) of $S_h$, we obtain the following finite dimensional problems:

$$(G(a, \lambda))_i \equiv F \left( \sum_{j=1}^{N+N_h+1} a_j g_j(x_i), \lambda \right) = 0, \quad i = 1, \ldots, N,$$

$$\sum_{j=1}^{N+N_h+1} a_j Bg_j(x_i) = 0, \quad i = N + 1, \ldots, N + N_h, \quad (24)$$

$$\sum_{j=1}^{N+N_h} a_j = 0,$$

$$\sum_{j=1}^{N+N_h+1} b_j Lg_j(x_i) = \mu \sum_{j=1}^{N+N_h+1} b_j g_j(x_i), \quad i = 1, \ldots, N, \quad (25)$$

$$\sum_{j=1}^{N+N_h+1} b_j Bg_j(x_i) = 0, \quad i = N + 1, \ldots, N + N_h,$$

$$\sum_{j=1}^{N+N_h} b_j = 0.$$
Introducing the notation: \( a = (a_1, \ldots, a_{N+N_i+1})^T, \ b = (b_1, \ldots, b_{N+N_i+1})^T \in \mathbb{R}^{n \times (N+N_i+1)}, \)

\[
B_g = \begin{bmatrix}
Bg_1(x_{N+1}) & \ldots & Bg_{N+N_i}(x_{N+1}) & Bg_{N+N_i+1}(x_{N+1}) \\
\vdots & & \vdots & \vdots \\
Bg_1(x_{N+N_i}) & \ldots & Bg_{N+N_i}(x_{N+N_i}) & Bg_{N+N_i+1}(x_{N+N_i}) \\
1 & \ldots & 1 & 0
\end{bmatrix},
\]

\[
L_g = \begin{bmatrix}
Lg_1(x_1) & \ldots & Lg_{N+N_i+1}(x_1) \\
\vdots & \vdots & \vdots \\
Lg_1(x_N) & \ldots & Lg_{N+N_i+1}(x_N)
\end{bmatrix}, \quad \Gamma = \begin{bmatrix}
g_1(x_1) & \ldots & g_{N+N_i+1}(x_1) \\
\vdots & \vdots & \vdots \\
g_1(x_N) & \ldots & g_{N+N_i+1}(x_N)
\end{bmatrix}
\]

we can rewrite the problems (24) and (25) in the matrix form as

\[
G(a, \lambda) = 0, \quad B_ga = 0, \quad (27)
\]

\[
L_gb = \mu \Gamma b, \quad B_gb = 0. \quad (28)
\]

**Implementation 1.** Let

\[
a^1 = (a_1, \ldots, a_N)^T \in \mathbb{R}^{n \times N}, \quad a^2 = (a_{N+1}, \ldots, a_{N+N_i+1})^T \in \mathbb{R}^{n \times (N+N_i+1)};
\]

\[
L_g^1 = \begin{bmatrix}
Lg_1(x_1) & \ldots & Lg_N(x_1) \\
\vdots & \vdots & \vdots \\
Lg_1(x_N) & \ldots & Lg_N(x_N)
\end{bmatrix}, \quad L_g^2 = \begin{bmatrix}
Lg_{N+1}(x_1) & \ldots & Lg_{N+N_i+1}(x_1) \\
\vdots & \vdots & \vdots \\
Lg_{N+1}(x_N) & \ldots & Lg_{N+N_i+1}(x_N)
\end{bmatrix},
\]

\[
\Gamma^1 = \begin{bmatrix}
g_1(x_1) & \ldots & g_N(x_1) \\
\vdots & \vdots & \vdots \\
g_1(x_N) & \ldots & g_N(x_N)
\end{bmatrix}, \quad \Gamma^2 = \begin{bmatrix}
g_{N+1}(x_1) & \ldots & g_{N+N_i+1}(x_1) \\
\vdots & \vdots & \vdots \\
g_{N+1}(x_N) & \ldots & g_{N+N_i+1}(x_N)
\end{bmatrix}
\]

\[
B_g^1 = \begin{bmatrix}
Bg_1(x_{N+1}) & \ldots & Bg_N(x_{N+1}) \\
\vdots & \vdots & \vdots \\
Bg_1(x_{N+N_i}) & \ldots & Bg_N(x_{N+N_i}) \\
1 & \ldots & 1
\end{bmatrix},
\]

\[
B_g^2 = \begin{bmatrix}
Bg_{N+1}(x_{N+1}) & \ldots & Bg_{N+N_i}(x_{N+1}) & Bg_{N+N_i+1}(x_{N+1}) \\
\vdots & \vdots & \vdots & \vdots \\
Bg_{N+1}(x_{N+N_i}) & \ldots & Bg_{N+N_i}(x_{N+N_i}) & Bg_{N+N_i+1}(x_{N+N_i}) \\
1 & \ldots & 1 & 0
\end{bmatrix}.
\]

Substituting this into (27), we rewrite it as:

\[
G(a^1, \lambda) \equiv G(a^1, a^2, \lambda) = 0, \quad (29)
\]

where \( a^2 \) solves

\[
B_g^2 a^2 = -B_g^1 a^1. \quad (30)
\]

Similarly, we rewrite (28) as

\[
L_g^1 b^1 + L_g^2 b^2 = \mu (\Gamma^1 b^1 + \Gamma^2 b^2),
\]

\[
B_g^1 b^1 + B_g^2 b^2 = 0.
\]
or, eliminating $b^2$, as

$$L_g b^1 - L_g \left( B_g^2 \right)^{-1} B_g b^1 = \mu \left( \Gamma_1 b^1 + \Gamma_2 \left( B_g^2 \right)^{-1} B_g b^1 \right).$$  \hspace{1cm} (31)$$

We are interested in the continuation of solutions to (29). Therefore, in addition to $a^1$, we also treat $\lambda$ as unknown, and add an algebraic constraint

$$G_c(a^1, \lambda) = 0,$$  \hspace{1cm} (32)$$

which defines a parametrization of the solution curve.

**Algorithm 1** (Continuation algorithm for the system (29), (32)). Given current approximations to $a^1 \in \mathbb{R}^{n \times N}$ and $\lambda \in \mathbb{R}$, a complete Newton iteration consists of the following steps:

1. Compute the matrices $B_g^1$, $B_g^2$, $\Gamma_1$, $\Gamma_2$.
2. Solve the system (30) to find $a^2$.
3. Use the expressions (29), (32) to compute the residuals $-G(a^1, \lambda)$, $-G_c(a^1, \lambda)$ and then compute the matrices $D_1 G \equiv D_1 G(a^1, \lambda)$, $D_2 G \equiv D_2 G(a^1, \lambda)$, $D_1 G_c \equiv D_1 G_c(a^1, \lambda)$, and $D_2 G_c \equiv D_2 G(a^1, \lambda)$ by differencing.
4. Solve the system

$$D_1 G \delta a^1 + D_2 G \delta \lambda = -G(a^1, \lambda),$$
$$D_1 G_c \delta a^1 + D_2 G_c \delta \lambda = -G_c(a^1, \lambda),$$  \hspace{1cm} (33)$$

where we omitted iteration indices for $\delta a^1$ and $\delta \lambda$ in (33).
5. Update $a^1 \rightarrow a^1 + \delta a^1$ and $\lambda \rightarrow \lambda + \delta \lambda$.
6. Solve the generalized eigenvalue problem

$$D_1 G b^1 = \mu \left( \Gamma_1 + \Gamma_2 \left( B_g^2 \right)^{-1} B_g \right) b^1,$$  \hspace{1cm} (34)$$

and

$$D_1 G b^1 = L_g b^1 - L_g \left( B_g^2 \right)^{-1} B_g b^1,$$  \hspace{1cm} (35)$$

where $D_1 G b^1 = L_g b^1 - L_g \left( B_g^2 \right)^{-1} B_g b^1$, see (31).

**Implementation 2.** Let $U = (U_1, ..., U_N)^T$ be the vector of nodal values of the solution $u_h$ (22) of the collocation problem (20), and let \{ $\phi_i$ \}$_{i=1}^N$ be the Lagrange basis in $S_h$:

$$\{ \phi_j \in S_h : \phi_j(x_i) = \delta_{ij}, \ i, j = 1, ..., N \}. $$  \hspace{1cm} (35)$$

Then $u_h$ can be written as

$$u_h(x) = \sum_{j=1}^N U_j \phi_j(x),$$  \hspace{1cm} (36)$$

Combining this with the definitions (22) of $u_h$ and (26) of $B_g$ and $\Gamma$, we have

$$\Gamma a = U,$$
$$B_g a = 0,$$  \hspace{1cm} (37)$$

that defines the 1-to-1 correspondence between $U \in \mathbb{R}^{n \times N}$ and $a \in \mathbb{R}^{n \times (N+N_i+1)}$. The problems (20) and (21), respectively, are written as

$$G(U, \lambda) \equiv G(a, \lambda) = 0,$$  \hspace{1cm} (38)$$

7
where $a$ solves (37), and

$$D_1 G(U, \lambda)V = \mu V, \quad V \in \mathbb{R}^{n \times N}.$$  \hfill (39)

As before, to define a parametrization of the solution curve, we add an algebraic constraint

$$G_c(U, \lambda) = 0.$$  \hfill (40)

**Algorithm 2** (Continuation algorithm for the system (38), (40)). Given current approximations to $U \in \mathbb{R}^{n \times N}$ and $\lambda \in \mathbb{R}$, a complete Newton iteration consists of the following steps:

1. Compute the matrices $B_g$, $\Gamma$.
2. Solve the system (37) to find $a$.
3. Use the expressions (38), (40) to compute the residuals $-G(U, \lambda)$, $-G_c(U, \lambda)$ and then compute the matrices $D_1 G = D_1 G(U, \lambda)$, $D_2 G = D_2 G(U, \lambda)$, $D_1 G_c = D_1 G_c(U, \lambda)$, and $D_2 G = D_2 G_c(U, \lambda)$ by differencing.
4. Solve the system

$$D_1 G \delta U + D_2 G \delta \lambda = -G(U, \lambda),$$

$$D_1 G_c \delta U + D_2 G_c \delta \lambda = -G_c(U, \lambda),$$  \hfill (41)

where we omitted iteration indices in (41) for $\delta U$ and $\delta \lambda$.
5. Update $U \rightarrow U + \delta U$ and $\lambda \rightarrow \lambda + \delta \lambda$.
6. Solve the eigenvalue problem (39) (to detect bifurcations).

**Remark 1** For our numerical experiments, we implemented in AUTO [6] Algorithm 2 for the Kansa-MQ method and Algorithm 2a, below, for the MQ PDECB method. The principle reason for choosing Algorithm 2 rather than Algorithm 1 is that the eigenvalue problem (39) (and (56)) is a standard eigenvalue problem whose solution is supported by AUTO. On the other hand, the eigenvalue problem (34) is a complicated generalized eigenvalue problem whose solution is not supported by AUTO.

### 3.2 The MQ PDECB method.

To formulate the approximate problem, we first introduce the set $\Theta_h$ of nodes

$$\Theta_h^1 = \left\{ \{x_i\}_{i=1}^N \subset \Omega, \quad \{x_i\}_{i=N+1}^{N+N_1} \subset \partial \Omega, \quad \{x_i\}_{i=N+N_1+1}^{N+N_1+N_2} \subset \mathbb{R}^d \setminus \partial \Omega \right\},$$  \hfill (42)

where the nodes $\{x_i\}_{i=N+N_1+1}^{N+N_1+N_2}$, which can be inside $\Omega$ or outside $\overline{\Omega}$, are adjacent to the boundary $\partial \Omega$, and the MQ basis functions,

$$g_j(x) \equiv g_j(c_j, x) = \sqrt{||x - x_j||^2 + c_j^2, \quad j = 1, \ldots, N + N_b, \quad g_{N+N_1+N_2+1}(x) = 1.}$$  \hfill (43)

We remark here that only the points $x_i, \ i = 1, \ldots, N + N_b$ that lie in $\Omega$ and on $\partial \Omega$ will be used as the collocation points. In particular, the points $x_i, \ i = N + 1, \ldots, N + N_b$ on $\partial \Omega$ will be used as the collocation points for both the PDE and the boundary condition.

We next define an MQ finite dimensional set (which is not a subspace, in general)

$$S_h^1 := \left\{ \chi = \sum_{j=1}^{N+N_2+N_1+1} a_j g_j(\cdot) : \sum_{j=1}^{N+N_1} a_j = 0, \quad B \chi(x_i) = 0, \quad F(\chi(x_i), \lambda) = 0, \ i = N + 1, \ldots, N + N_b \right\}.$$  \hfill (44)
The problems (15) and (16), respectively, are approximated by the collocation equations

\[
F(u_h(x_i), \lambda) = 0, \quad u_h \in S_h^1, \quad i = 1, \ldots, N,
\]

\[
Lv_h(x_i) \equiv D_1 F(u_h(x_i), \lambda) v_h(x_i) = \mu v_h(x_i), \quad v_h \in S_h^1, \quad i = 1, \ldots, N.
\]

Substituting

\[
u_h(x) = \sum_{j=1}^{N+2N_1+1} b_j g_j(x),
\]

\[
u_h(x) = \sum_{j=1}^{N+2N_1+1} b_j g_j(x),
\]

into (45) and (46), respectively, and using the definition (44) of \(S_h^1\), we obtain the following finite dimensional problems:

\[
(G(a, \lambda))_i \equiv F \left( \sum_{j=1}^{N+2N_1+1} a_j g_j(x_i), \lambda \right) = 0, \quad i = 1, \ldots, N,
\]

\[
(G(a, \lambda))_i \equiv F \left( \sum_{j=1}^{N+2N_1+1} a_j g_j(x_i), \lambda \right) = 0, \quad i = N + 1, \ldots, N + N_b,
\]

\[
\sum_{j=1}^{N+2N_1+1} a_j g_j(x_i) = 0, \quad i = N + 1, \ldots, N + N_b,
\]

\[
\sum_{j=1}^{N+2N_1+1} b_j g_j(x_i) = \mu \sum_{j=1}^{N+2N_1+1} b_j g_j(x_i), \quad i = 1, \ldots, N + N_b,
\]

\[
\sum_{j=1}^{N+2N_1+1} b_j g_j(x_i) = 0, \quad i = N + 1, \ldots, N + N_b,
\]

\[
\sum_{j=1}^{N+2N_1+1} b_j = 0.
\]

Introducing the notation: \(a = (a_1, \ldots, a_{N+2N_1+1})^T\), \(b = (b_1, \ldots, b_{N+2N_1+1})^T \in \mathbb{R}^{n \times (N+2N_1+1)}\),

\[
B_g = \begin{bmatrix}
B_{g1}(x_{N+1}) & \cdots & B_{gN+2N_1}(x_{N+1}) & B_{gN+2N_1+1}(x_{N+1}) \\
\vdots & \ddots & \vdots & \vdots \\
B_{g1}(x_{N+N_1}) & \cdots & B_{gN+2N_1}(x_{N+N_1}) & B_{gN+2N_1+1}(x_{N+N_1}) \\
1 & \cdots & 1 & 0
\end{bmatrix},
\]

\[
L_g = \begin{bmatrix}
L_{g1}(x_1) & \cdots & L_{gN+2N_1+1}(x_1) \\
\vdots & \ddots & \vdots \\
L_{g1}(x_{N+N_1}) & \cdots & L_{gN+2N_1+1}(x_{N+N_1})
\end{bmatrix}, \quad \Gamma = \begin{bmatrix}
g1(x_1) & \cdots & g_{N+2N_1+1}(x_1) \\
\vdots & \ddots & \vdots \\
g1(x_{N+N_1}) & \cdots & g_{N+2N_1+1}(x_{N+N_1})
\end{bmatrix},
\]

we can rewrite the problems (49) and (50) in the matrix form as

\[
(G(a, \lambda))_i = 0, \quad i = 1, \ldots, N,
\]

\[
(G(a, \lambda))_i = 0, \quad i = N + 1, \ldots, N + N_b,
\]

\[
B_g a = 0,
\]

\[
L_g b = \mu \Gamma b,
\]

\[
B_g b = 0.
\]
Implementation 2a. Let $U = (u_h(x_1), \ldots, u_h(x_N))^T$ be the vector of nodal values of the approximate solution $u_h$. Then by the definitions (47) of $u_h$ and (51) of $B_g$ and $\Gamma$, we have

\begin{align}
(G(a, \lambda))_i &= 0, \quad i = N + 1, \ldots, N + N_h, \\
\Gamma a &= U, \\
B_g a &= 0,
\end{align}

(54)

that defines the 1-to-1 correspondence between $U \in \mathbb{R}^{n \times N}$ and $a \in \mathbb{R}^{n \times (N + 2N_h + 1)}$.

The problems (45) and (46), respectively, are written as

\begin{equation}
(G(U, \lambda))_i \equiv (G(a, \lambda))_i = 0, \quad i = 1, \ldots, N,
\end{equation}

(55)

where $a$ solves (54), and

\begin{equation}
D_1 G(U, \lambda)V = \mu V, \quad V \in \mathbb{R}^{n \times N}.
\end{equation}

(56)

As before, to define a parametrization of the solution curve, we add an algebraic constraint

\begin{equation}
G_c(U, \lambda) = 0.
\end{equation}

(57)

Algorithm 2a (Continuation algorithm for the system (55), (57)). Given current approximations to $U \in \mathbb{R}^{n \times N}$ and $\lambda \in \mathbb{R}$, a complete Newton iteration consists of the following steps:

1. Compute the matrices $B_g$, $\Gamma$.
2. Solve the system (54) to find $a$.
3. Use the expressions (55), (57) to compute the residuals $-G(U, \lambda)$, $-G_c(U, \lambda)$ and then compute the matrices $D_1 G \equiv D_1 G(U, \lambda)$, $D_2 G \equiv D_2 G(U, \lambda)$, $D_1 G_c \equiv D_1 G_c(U, \lambda)$, and $D_2 G \equiv D_2 G_c(U, \lambda)$ by differencing.
4. Solve the system

\begin{align}
D_1 G\delta U + D_2 G\delta \lambda &= -G(U, \lambda), \\
D_1 G_c\delta U + D_2 G_c\delta \lambda &= -G_c(U, \lambda),
\end{align}

(58)

where we omitted iteration indices in (58) for $\delta U$ and $\delta \lambda$.
5. Update $U \rightarrow U + \delta U$ and $\lambda \rightarrow \lambda + \delta \lambda$.
6. Solve the eigenvalue problem (56) (to detect bifurcations).

4 Numerical experiments for 1D and 2D elliptic PDEs.

We present examples of solution of linear 1D and 2D elliptic PDEs and continuation of solutions to nonlinear 1D and 2D Gelfand-Bratu equation. Each problem is discretized by the Kansa-MQ method, see Eq. (38), and the MQ PDEC method, see Eq. (55).

In the case of nonlinear problems, we perform continuation of solutions by Algorithm 2 for the Kansa-MQ method and by Algorithm 2a for the MQ PDEC method. We compare the accuracy of the detection of the limit point (or fold) by the two methods. We recall that a solution $(u_0, \lambda_0)$ of equation $f(u, \lambda) = 0$ is a (simple) limit point if the solution curve in $(u(s), \lambda(s))$, for some parametrization $s$, makes a turn at $(u_0, \lambda_0)$. This is expressed formally as $\dim \mathcal{N}(f_u(u_0, \lambda_0)) = 1$ and $f_\lambda(u_0, \lambda_0) \notin \mathcal{R}(f_u(u_0, \lambda_0))$.

We will use throughout the notation $h$ for the average distance between the nodes. Then $h = 1/(K - 1)$ for a 1D problem on $(0, 1)$ and for a 2D problem on $(0, 1) \times (0, 1)$, where $K$ is the number of nodes along each axis.

To improve the accuracy, we employ 2 simple adaptation strategies for the shape parameters $C = \{c_1, \ldots, c_{N + N_h}\}$ for the Kansa-MQ method, see Eq. (18), and $C^1 = \{c_1, \ldots, c_{N + 2N_h}\}$ for the MQ PDEC method, see Eq. (43); for the nodes $\Theta_h$ for the Kansa-MQ method, see Eq. (17), and $\Theta^1_h$ for the MQ PDEC
method, see Eq. (42). To be specific, assume that $\Omega = (0,1) \times (0,1)$ and consider the case of the Kansa-MQ method. Let $r(x,y, C, \Theta_h)$ be the residual. Our strategies are all based on the Nonlinear Least Squares Method which minimizes the $L_2$ norm $\varphi(C, \Theta_h) \equiv \|r\|_2$ of the residual. By the quasi-uniform distribution of nodes we will mean the distribution of nodes, where the nodes adjacent to the boundary $\partial \Omega$ are placed at the distance $\delta = \delta h_0, 0 < \delta \leq 1$, from $\partial \Omega$, while the remaining nodes are distributed uniformly with the distance $h_0$ between them.

**Strategy 1.** Uniform distribution of nodes $\Theta_h$; $c_1 = \ldots = c_{N+T_1} = c; \min_c \varphi(C, \Theta_h)$,

**Strategy 2.** Quasiuniform distribution of nodes $\Theta_h$; $c_1 = \ldots = c_{N+T_1} = c; \min_{c, \delta} \varphi(C, \Theta_h)$.

In all examples below we use the adaptation strategy 2.

**Example 2 A 1D model linear problem**

$$u_{xx} + (2\pi)^2 \sin(2\pi x) = 0, \quad \text{in } \Omega = (0,1),$$

$$u(0) = u(1) = 0.$$  \hspace{1cm} (59)

*The analytical solution is*

$$u_{exact} = \sin(2\pi x).$$

*Numerical results are presented in Fig. 2a.*

![1D PDE BY PDECB AND KANSA-MQ](image1.png) ![1D BRATU-GELFAND EQUATION BY PDECB](image2.png)

(a) 1D linear PDE solution  
(b) 1D continuation

**Figure 2:** Convergence properties of the Kansa-MQ method and the MQ PDECB method:

(a) 1D linear problem, Eq. (59); the $L_\infty$ norm of the solution error is plotted, in the logarithmic scale, versus $1/h$, where $h$ is the average distance between the nodes. The roundoff error starts to dominate at $1/h \approx 11$ for Kansa-MQ method and at $1/h \approx 18$ for the MQ PDECB method.

(b) The location $\lambda$ of the limit point for 1D Bratu-Gelfand problem, Eq. (61). Relative error in $\lambda$ is plotted in the logarithmic scale versus $1/h$.

**Example 3 A 2D model linear problem**

$$\Delta u - (2x^2 y^2 + 2x^2 y + 2xy^2 - 6xy)e^{(u+y)} = 0, \quad \text{in } \Omega = (0,1) \times (0,1),$$

$$u|_{\partial\Omega} = 0.$$  \hspace{1cm} (60)
Figure 3: Convergence properties of the Kansa-MQ and the MQ PDECB methods:
(a) 2D linear PDE, Eq. (60): the $L_\infty$ norm of the solution error is plotted, in the logarithmic scale, versus $1/h$, where $h$ is the average distance between the nodes. The roundoff error starts to dominate at $1/h \sim 9$ for the Kansa-MQ method and at $1/h \sim 11$ for the MQ PDECB method. We also provide, for comparison, the error in the MQ interpolation of the exact solution $u_{exact}$.
(b) The location $\lambda$ of the limit point for 2D Bratu-Gelfand problem, Eq. 62. Relative error in $\lambda$ is plotted, in the logarithmic scale, versus $1/h$.

The analytical solution is

$$u_{exact} = x(x - 1)y(y - 1)e^{(x + y)}.$$  

Numerical results are presented in Fig. 3a. We do not have an explanation of why the MQ PDECB solution is more accurate than the interpolation.

**Example 4** 1D Gelfand-Bratu problem. This is a scalar problem

$$u'' + \lambda e^u = 0, \quad \text{in } \Omega = (0, 1),$$

$$u(0) = u(1) = 0. \quad (61)$$

that appears in combustion theory and is used as the demo example $\text{exp}$ in Auto97 [6] (fifth order adaptive orthogonal spline collocation method). There is a limit (fold) point on the solution curve. We take the value of $\lambda$ at the limit point found from demo $\text{exp}$ $(K \geq 50)$ as exact. The relative error in location of the limit point is shown in Fig. 2b. See also [12] for additional numerical results and references.

**Example 5** 2D Gelfand-Bratu problem

$$\Delta u + \lambda e^u = 0, \quad \text{in } \Omega = (0, 1) \times (0, 1),$$

$$u \Big|_{\partial \Omega} = 0. \quad (62)$$

This problem was studied by a number of authors. In [7] the problem was discretized with a high order orthogonal spline collocation method with sparse Jacobian. There is a limit (fold) point on the solution curve. The exact location of the limit point is assumed to be at the value of $\lambda$ obtained in [7] on a $16 \times 16$ mesh with $4 \times 4$ collocation points. The relative error in location of the limit point is shown in Fig. 3b. Note that the curve for the Kansa-MQ method was obtained [12] using quadruple precision which considerably slowed down computations, while we use only double precision with the MQ PDECB method here. See also [12] for additional numerical results, references and a discussion of the operation count.
Figure 4: (a) 1D linear problem with a boundary layer, Eq. (63) with $\epsilon = 10^{-3}$. The MQ PDECB solution with 21 nodes and the analytical solution $u_{exact}$ are plotted versus $x$ in $(0, 0.1)$. (b) The residuals for the solutions of 1D Gelfand-Bratu problem (61) by the Kansa-MQ and the MQ PDECB methods are plotted versus $x$ with $1/h = 9$, $\lambda = 2.5$. The $L_{\infty}$ residual norms are $4.2 \times 10^{-3}$ and $3.3 \times 10^{-6}$, respectively.

Example 6 A 1D model linear singular perturbation problem:

$$\epsilon u_{xx} + u_x = 0, \quad \text{in } \Omega = (0, 1),$$

$$u(0) = 0, \quad u(1) = 1. \quad (63)$$

The analytical solution is

$$u_{exact} = \frac{(1 - e^{-x/\epsilon})}{(1 - e^{-1/\epsilon})}.$$ 

This problem was studied by Hon in [24], who found that a standard Kansa-MQ method was too crude in the case $\epsilon \ll 1$ and formulated an efficient adaptive technique to treat such problems.

Here we use the MQ PDECB to solve this problem for $\epsilon \ll 1$ with relatively small number of nodes. For $\epsilon = 10^{-3}$, the MQ PDECB solution with 21 nodes and the exact solution are plotted versus $x$ in Fig. 4a. The $L_{\infty}$ norm of the solution error is 0.001 for the MQ PDECB method, while it is 0.22 for the Kansa-MQ method with 101 nodes (not shown). Note that for $\epsilon = 10^{-4}$ one can attain the same error 0.001 in the MQ PDECB solution with 41 nodes (not shown).

Example 7 Fig. 4b shows the residual distribution for the solutions of 1D Gelfand-Bratu problem (61) by the Kansa-MQ and the MQ PDECB methods with $1/h = 9$, $\lambda = 2.5$. The $L_{\infty}$ residual norms are $4.2 \times 10^{-3}$ and $3.3 \times 10^{-6}$, and the $L_2$ residual norms are $7.8 \times 10^{-4}$ and $1.1 \times 10^{-6}$ for the Kansa-MQ and the MQ PDECB methods, respectively.

5 Discussion.

We have formulated an improved Kansa-MQ method with the PDE collocation on the boundary (MQ PDECB). The idea of the method is to add an additional set of nodes adjacent to the boundary and, correspondingly, an additional set of collocation equations obtained via collocation of the PDE on the boundary. We have applied the MQ PDECB method to several model 1D and 2D linear and nonlinear elliptic PDEs and have presented results of our numerical experiments.
Numerical results demonstrate considerable improvement in convergence of the MQ PDECB method over the Kansa-MQ method, with both methods having exponential convergence with essentially the same rates.

We observe that the MQ PDECB method has considerably better conditioning of the corresponding MQ systems than the Kansa-MQ method, and hence the roundoff error starts to dominate for larger number of nodes in the case of the MQ PDECB method. However, ill-conditioning of the MQ systems, which is associated with ill-conditioning of the MQ interpolation matrix, still limits the accuracy that we can currently attain. In our experiments we notice the well known phenomena, studied theoretically in [31], that improving the accuracy (by increasing the shape parameter) increases ill-conditioning. In particular, increasing the number of nodes in our in our experiments would not lead to any significant improvement in accuracy. Notice, however, that even with a modest number of nodes that we have used, a very good accuracy has been achieved. In [27] domain decomposition was used to circumvent the ill-conditioning of the MQ matrices. It was shown there that the decomposition of the domain into successively finer subdivisions resulted in smaller block matrices with vastly improved condition numbers. In [2] a cardinal preconditioner was used for the full MQs, TPS, and other RBFs, and fast convergence was achieved for matrices of size 10,000-by-10,000, where the traditional methods would not work.

Typically, nonlinear problems are solved by Newton type methods. Using numerical continuation generally improves the convergence properties of the Newton method.

Acknowledgments

This work was supported in part by National Aeronautics and Space Administration through grant NAG8-1229. The first author is grateful to V.N. Belykh for fruitful discussions, that led us to the idea of the PDECB.

References


