DEPARTMENT OF MATHEMATICS AND STATISTICS

MASTER DISSERTATION

Kansa radial basis function method with fictitious centres for solving nonlinear boundary value problems

Andreas Katsiamis

Advisor: Andreas Karageorghis

13 May 2022

Abstract

A Kansa–radial basis function (RBF) collocation method is applied to two–dimensional second and fourth order nonlinear boundary value problems. The solution is approximated by a linear combination of RBFs, each of which is associated with a centre and a different shape parameter. As well as the RBF coefficients in the approximation, these shape parameter values are taken to be among the unknowns. In addition, the centres are distributed within a larger domain containing the physical domain of the problem. The size of this larger domain is controlled by a dilation parameter which is also included in the unknowns. In fourth order problems where two boundary conditions are imposed, two sets of (different) boundary centres are selected. The Kansa–RBF discretization yields a system of nonlinear equations which is solved by standard software. The proposed technique is applied to four problems and the numerical results are analyzed and discussed.

Acknowledgements

Words cannot express my deepest gratitude to my professor Karageorghis Andreas for his guidance, patience and support. I have benefited greatly from your wealth of knowledge and meticulous editing. His unassuming approach to research and science is a source of inspiration. This approach is reflected by his simple but clear writing style, which is something I hope to carry forward throughout my career. Lastly, I would be remiss in not mentioning my family. Their belief in me has kept my spirits and motivation high during this process. To my family, I give everything, including this.

Contents

1. Introduction

Different variations of the Kansa-radial basis function (RBF) method have been implemented for second and fourth-order nonlinear boundary value problems (BVPs) in two dimensional spaces, in recent studies [11–13]. The shape parameter(s), as well as the RBF expansion coefficients, was (were) included as a part of the problem's unknowns. These were the main characteristics utilized in these investigations. Particularly, in [12,13] the same shape parameter was incorporated into the nonlinear RBF discretization system unknowns and its value was correlated with each RBF (and centre) utilized. On the contrary, in [11] each RBF (and centre) utilized was related to a different shape parameter (multi–shape parameter), resulting in the same number of shape parameters as centres, and their values were incorporated in the nonlinear system's unknowns. Using the idea of an equilibrated matrix, [17] proposed a distinct methodology to find out such multi–shape parameters. In contrast, a single shape parameter was optimally calculated in [18] by the minimization of an energy gap functional. Standard MATLAB© nonlinear solvers were utilized for solving the nonlinear systems in both of these methodologies. The centres related to each RBF are spread in a domain encompassing the problem's domain [4], rather than inside the closure of the problem's domain, as is customary. More specifically, a physical magnification includes these *fictitious centres*. In addition to the RBF expansion coefficients and the shape parameters' values, the magnification parameter governing this distribution has been included in the problem unknowns. The concept of having one collection of boundary centres on a magnification of the physical boundary while setting the other collection on a magnification of it is proposed for fourth-order problems, where two boundary conditions ought to be forced. This additional magnification parameter is likewise considered to be one of the discretized problem's unknowns. In spite of the addition of some unknowns to the nonlinear solvers utilized, this approach lends itself nicely to the solution of nonlinear problems, i.e. by including these parameters as unknowns.

The suggested Kansa-RBF methodology for solving second and fourth-order BVPs is explained in Section 2 of the thesis. In Section 3, some implementation issues are discussed, whereas in Section 4, specific numerical examples are studied to illustrate the methodology. Lastly, some conclusions and ideas for potential future studies are discussed in Section 5.

2. Kansa RBF method

2.1. Second order problems.

2.1.1. *The problem.* Firstly, we examine a BVP in $\mathbb{R}^2$ described by the second order partial differential equation

$$\mathcal{N}u = f \quad \text{in the domain } \Omega, \text{ where } \mathcal{N} \text{ represents a nonlinear elliptic operator of second order [6]}, \tag{2.1a}$$

subject to the condition described by the operator $\mathcal{B}$

$$\mathcal{B}u = g \quad \text{on the boundary } \partial\Omega. \tag{2.1b}$$

2.1.2. *The methodology.* The approximation of the solution u for BVP (2.1) is based on Kansa's method [14], as follows:

$$u_{\mathsf{N}}(x,y) = \sum_{\mathsf{n}=1}^{\mathsf{N}} a_{\mathsf{n}}\, \Phi(c_{\mathsf{n}}, r_{\mathsf{n}}), \qquad (x,y) \in \overline{\Omega}, \tag{2.2}$$

where each RBF $\Phi(c_{\mathsf{n}}, r_{\mathsf{n}})$ is related to a different *centre point* (or *centre*) $(\mathrm{x}_{\mathsf{n}}, \mathrm{y}_{\mathsf{n}})$ via the formula $r_{\mathsf{n}}^2 = (x - \mathrm{x}_{\mathsf{n}})^2 + (y - \mathrm{y}_{\mathsf{n}})^2$ and a different *shape parameter* c_{n}. In most approaches the shape parameters is the same, namely $c_1 = c_2 = \ldots = c_{\mathsf{N}} = c$, where c is preadjusted. Besides, as the RBF literature documents, the most demanding problem is to achieve the optimal shape value(s). To overcome this difficulty, in [12, 13] c was taken to be part of the unknowns with the RBF coefficients $a_{\mathsf{n}}, \mathsf{n} = 1, \ldots, \mathsf{N}$, while in [11] every different $c_{\mathsf{n}}, \mathsf{n} = 1, \ldots, \mathsf{N}$, was included in the unknowns. Furthermore, $u_{\mathsf{N}}(x,y)$ does not include polynomial basis functions in contrast to [11, 13].

The *collocation points* are set in the suggested discretization as $\{(x_{\mathsf{m}}, y_{\mathsf{m}})\}_{\mathsf{m}=1}^{\mathsf{M}} \in \overline{\Omega}$. Particularly, we take $\mathsf{M}_{\mathrm{int}}$ *interior* $\{(x_{\mathsf{m}}, y_{\mathsf{m}})\}_{\mathsf{m}=1}^{\mathsf{M}_{\mathrm{int}}}$ and $\mathsf{M}_{\mathrm{bry}}$ *boundary* points $\{(x_{\mathsf{m}}, y_{\mathsf{m}})\}_{\mathsf{m}=\mathsf{M}_{\mathrm{int}}+1}^{\mathsf{M}_{\mathrm{int}}+\mathsf{M}_{\mathrm{bry}}}$ and set $\mathsf{M} = \mathsf{M}_{\mathrm{int}} + \mathsf{M}_{\mathrm{bry}}$. Additionally, the points $\{(\tilde{\mathrm{x}}_{\mathsf{n}}, \tilde{\mathrm{y}}_{\mathsf{n}})\}_{\mathsf{n}=1}^{\mathsf{N}}$ (interior and boundary) are selected, where $\mathsf{N} = \mathsf{N}_{\mathrm{int}} + \mathsf{N}_{\mathrm{bry}}$, in $\overline{\Omega}$. The corresponding *fictitious centres* $\{(\mathrm{x}_{\mathsf{n}}, \mathrm{y}_{\mathsf{n}})\}_{\mathsf{n}=1}^{\mathsf{N}}$ will be placed in $\overline{D}$ where D is a domain comprising Ω. Specifically, we take

$$(\mathrm{x}_{\mathsf{n}}, \mathrm{y}_{\mathsf{n}}) = \eta\, (\tilde{\mathrm{x}}_{\mathsf{n}}, \tilde{\mathrm{y}}_{\mathsf{n}}), \quad n = 1, \ldots, \mathsf{N}, \quad \text{where } \eta \text{ is a constant.} \tag{2.3}$$

This scheme has been utilized in [4]. In the current study, the magnification parameter η restraining the size of D will be considered an unknown value calculated as a portion of the solution. Clearly, the number of collocation points exceeds (or equals) the number of centres, i.e. $\mathsf{M} \geq \mathsf{N}$.

The unknown shape parameter values $\{c_{\mathsf{n}}\}_{\mathsf{n}=1}^{\mathsf{N}}$, the unknown coefficients $\{a_{\mathsf{n}}\}_{\mathsf{n}=1}^{\mathsf{N}}$, along with the unknown quantity of the magnification parameter η restraining the size of D in equation (2.3), are calculated from the M collocation equations

$$\mathcal{N} u_{\mathsf{N}}(x_{\mathsf{m}}, y_{\mathsf{m}}) = f(x_{\mathsf{m}}, y_{\mathsf{m}}), \quad \mathsf{m} = 1, \ldots, \mathsf{M}_{\mathrm{int}}, \tag{2.4a}$$
$$\mathcal{B} u_{\mathsf{N}}(x_{\mathsf{m}}, y_{\mathsf{m}}) = g(x_{\mathsf{m}}, y_{\mathsf{m}}), \quad \mathsf{m} = \mathsf{M}_{\mathrm{int}} + 1, \ldots, \mathsf{M}_{\mathrm{int}} + \mathsf{M}_{\mathrm{bry}}. \tag{2.4b}$$

A sum of M equations in $2\mathsf{N}+1$ unknowns results from this technique, specifically, the coefficients $\boldsymbol{a} = [a_1, a_2, \ldots, a_{\mathsf{N}}]^T$, the shape parameters $\boldsymbol{c} = [c_1, c_2, \ldots, c_{\mathsf{N}}]^T$ and the parameter η and, thus we must choose $\mathsf{M} \geq 2\mathsf{N} + 1$ in order to have a sufficient number of equations.

A nonlinear system of equations is obtained by the equations (2.4) as

$$\boldsymbol{F}(\boldsymbol{a},\boldsymbol{c},\eta) := \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{\mathsf{M}} \end{bmatrix} = \begin{bmatrix} \mathcal{N}u_{\mathsf{N}}(x_1,y_1) - f(x_1,y_1) \\ \vdots \\ \mathcal{N}u_{\mathsf{N}}(x_{\mathsf{M}_{\text{int}}},y_{\mathsf{M}_{\text{int}}}) - f(x_{\mathsf{M}_{\text{int}}},y_{\mathsf{M}_{\text{int}}}) \\ \mathcal{B}u_{\mathsf{N}}(x_{\mathsf{M}_{\text{int}}+1},y_{\mathsf{M}_{\text{int}}+1}) - g(x_{\mathsf{M}_{\text{int}}+1},y_{\mathsf{M}_{\text{int}}+1}) \\ \vdots \\ \mathcal{B}u_{\mathsf{N}}(x_{\mathsf{M}},y_{\mathsf{M}}) - g(x_{\mathsf{M}},y_{\mathsf{M}}) \end{bmatrix} = \mathbf{0}\,. \tag{2.5}$$

The MATLAB© [19] optimization toolbox routines `fsolve` or `lsqnonlin` are utilized, details in order to solve the system (2.5) of which can be found, e.g. in [13]. We mostly utilised `fsolve` because it was more computationally efficient than `lsqnonlin` for the numerical examples analysed in this study.
The variables $\boldsymbol{a}_0, \boldsymbol{c}_0$ and η_0 designate the initial values of the unknowns $\boldsymbol{a}, \boldsymbol{c}$ and η, respectively. Therefore, to carry out the above routines we have to provide these initial values.

2.2. **Fourth order problems.**

2.2.1. *The problem.* We examine the BVP in $\mathbb{R}^2$ described by the fourth order partial differential equation

$$\mathcal{N}u = f \quad \text{in the domain } \Omega, \text{ where } \mathcal{N} \text{ is a nonlinear elliptic operator of fourth–order [2]}, \tag{2.6a}$$

subject to the conditions

$$\mathcal{B}_1\, u = g_1 \quad \text{and} \quad \mathcal{B}_2\, u = g_2 \quad \text{on the boundary } \partial\Omega. \tag{2.6b}$$

2.2.2. *The methodology.* The approximation of the solution u for the BVP (2.6) is taken as given by (2.2). We select the collocation points $\{(x_{\mathsf{m}}, y_{\mathsf{m}})\}_{\mathsf{m}=1}^{\mathsf{M}} \in \overline{\Omega}$ consisting of M_{int} interior and M_{bry} boundary collocation points in the Kansa–RBF discretization, as in Section 2.1.2. Furthermore, we set the interior points $\{(\tilde{x}_{\mathsf{n}}, \tilde{y}_{\mathsf{n}})\}_{\mathsf{n}=1}^{\mathsf{N}_{\text{int}}}$ and *two different collections* of boundary points $\{(\tilde{x}_{\mathsf{n}}, \tilde{y}_{\mathsf{n}})\}_{\mathsf{n}=\mathsf{N}_{\text{int}}+1}^{\mathsf{N}_{\text{int}}+\mathsf{N}_{\text{bry}}}$, and $\{(\tilde{x}_{\mathsf{n}}, \tilde{y}_{\mathsf{n}})\}_{\mathsf{n}=\mathsf{N}_{\text{int}}+\mathsf{N}_{\text{bry}}+1}^{\mathsf{N}_{\text{int}}+2\mathsf{N}_{\text{bry}}}$. Now, however, we need to take $\mathsf{N} = \mathsf{N}_{\text{int}} + 2\mathsf{N}_{\text{bry}}$. The centres are derived by (2.3), as in the second order problem.

To obtain the magnification parameter η, the coefficients $\{a_{\mathsf{n}}\}_{\mathsf{n}=1}^{\mathsf{N}}$ and the shape parameters $\{c_{\mathsf{n}}\}_{\mathsf{n}=1}^{\mathsf{N}}$ in (2.2), the differential equation (2.6a) and the boundary conditions (2.6b) are arranged in the following way:

$$\mathcal{N}u_{\mathsf{N}}(x_{\mathsf{m}}, y_{\mathsf{m}}) = f(x_{\mathsf{m}}, y_{\mathsf{m}}), \quad \mathsf{m} = 1, \ldots, \mathsf{M}_{\text{int}}, \tag{2.7a}$$

$$\mathcal{B}_1 u_{\mathsf{N}}(x_{\mathsf{m}}, y_{\mathsf{m}}) = g_1(x_{\mathsf{m}}, y_{\mathsf{m}}), \quad \mathsf{m} = \mathsf{M}_{\text{int}} + 1, \ldots, \mathsf{M}_{\text{int}} + \mathsf{M}_{\text{bry}}, \tag{2.7b}$$

$$\mathcal{B}_2 u_{\mathsf{N}}(x_{\mathsf{m}}, y_{\mathsf{m}}) = g_2(x_{\mathsf{m}}, y_{\mathsf{m}}), \quad \mathsf{m} = \mathsf{M}_{\text{int}} + 1, \ldots, \mathsf{M}_{\text{int}} + \mathsf{M}_{\text{bry}}. \tag{2.7c}$$

Needless to say, we have $\mathsf{M} = \mathsf{M}_{\text{int}} + 2\mathsf{M}_{\text{bry}}$ equations in $2\,(\mathsf{N}_{\text{int}} + 2\mathsf{N}_{\text{bry}}) + 1$ unknowns, which are the coefficients $\boldsymbol{a} = [a_1, a_2, \ldots, a_{\mathsf{N}}]^T$, the shape parameters $\boldsymbol{c} = [c_1, c_2, \ldots, c_{\mathsf{N}}]^T$ (where in this case $\mathsf{N} = \mathsf{N}_{\rm int} + 2\,\mathsf{N}_{\rm bry}$) and the parameter η. As a result we must choose $\mathsf{M}_{\rm int} + 2\mathsf{M}_{\rm bry} \geq 2\,(\mathsf{N}_{\rm int} + 2\mathsf{N}_{\rm bry}) + 1$.

A nonlinear system of equations, as in (2.5), obtained by the equations (2.7) as

$$\boldsymbol{F}(\boldsymbol{a}, \boldsymbol{c}, \eta) := \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{\mathsf{M}_{\rm int}+2\mathsf{M}_{\rm bry}} \end{bmatrix}$$

$$= \begin{bmatrix} \mathcal{N} u_{\mathsf{N}}(x_1, y_1) - f(x_1, y_1) \\ \vdots \\ \mathcal{N} u_{\mathsf{N}}(x_{\mathsf{M}_{\rm int}}, y_{\mathsf{M}_{\rm int}}) - f(x_{\mathsf{M}_{\rm int}}, y_{\mathsf{M}_{\rm int}}) \\ \mathcal{B}_1 u_{\mathsf{N}}(x_{\mathsf{M}_{\rm int}+1}, y_{\mathsf{M}_{\rm int}+1}) - g_1(x_{\mathsf{M}_{\rm int}+1}, y_{\mathsf{M}_{\rm int}+1}) \\ \vdots \\ \mathcal{B}_1 u_{\mathsf{N}}(x_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}, y_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}) - g_1(x_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}, y_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}) \\ \mathcal{B}_2 u_{\mathsf{N}}(x_{\mathsf{M}_{\rm int}+1}, y_{\mathsf{M}_{\rm int}+1}) - g_2(x_{\mathsf{M}_{\rm int}+1}, y_{\mathsf{M}_{\rm int}+1}) \\ \vdots \\ \mathcal{B}_2 u_{\mathsf{N}}(x_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}, y_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}) - g_2(x_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}, y_{\mathsf{M}_{\rm int}+\mathsf{M}_{\rm bry}}) \end{bmatrix} = \boldsymbol{0}, \tag{2.8}$$

the solution of which is achieved using the MATLAB© optimization toolbox routines `fsolve` or `lsqnonlin`. The variables $\boldsymbol{a}_0, \boldsymbol{c}_0$ and η_0 once more designate the initial values of the unknowns $\boldsymbol{a}, \boldsymbol{c}$ and η, accordingly.

3. Implementational details

The approximate solution u_{N} was computed on a collection of L test points in $\overline{\Omega}$. The maximum relative error E, where the exact solution u is known, given by

$$\mathrm{E} = \frac{||u - u_{\mathsf{N}}||_{\infty,\overline{\Omega}}}{||u||_{\infty,\overline{\Omega}}} \tag{3.1}$$

and the root mean square error (RMSE) $\mathcal{E}$ defined by

$$\mathcal{E} = \sqrt{\frac{1}{L}\sum_{\ell=1}^{L} \left[u(x_\ell, y_\ell) - u_{\mathsf{N}}(x_\ell, y_\ell)\right]^2}\,, \tag{3.2}$$

were calculated. Also, the maximum absolute error

$$\mathsf{e} = ||u - u_{\mathsf{N}}||_{\infty,\overline{\Omega}}, \tag{3.3}$$

was computed in some cases. In addition, except when specified, we utilized the normalized multiquadric (MQ) RBFs

$$\Phi(c_{\mathsf{n}}, r_{\mathsf{n}}) = \sqrt{(c_{\mathsf{n}}\, r_{\mathsf{n}})^2 + 1}, \tag{3.4}$$

and the Appendix contains some useful derivatives of these.

The shape parameters' initial values, as in [11], were distributed in two ways:

Approach 1: Here we took $\boldsymbol{c}_0 = c_0\,[1, 1, \dots, 1]$ where c_0 designates the (preassigned) initial value for every shape parameter.

Approach 2: We formed uniformly a distribution of the initial shape parameter (see [20])

$$\boldsymbol{c}_0(\ell) = d_{\min} + (d_{\max} - d_{\min})\frac{(\ell - 1)}{(\mathsf{N} - 1)}, \quad \ell = 1, \dots, \mathsf{N},$$

where $d_{\min}$ and $d_{\max}$ are preassigned.

We also determined the minimum ($c_{\min}$) and maximum ($c_{\max}$) values of the final vector's ($\boldsymbol{c}$) entries.

If boundary $\partial\Omega$ is star shaped, i.e. defined parametrically by

$$(x, y) = r(\vartheta)\,(\cos\vartheta, \sin\vartheta)\,, \quad 0 \le \vartheta \le 2\pi, \tag{3.5}$$

a uniform boundary collocation point distribution can be computed based on [16]. In this uniform distribution, the length of each segment is $S/\mathsf{M}_{\text{bry}}$, where S is the boundary curve's length.
To find the angles ϑ_k that yield a uniform distribution, we start with $\vartheta_1 = 0$ and solve, for $k = 1, \dots, M_\vartheta - 1$, the nonlinear equations

$$F(t) = \sqrt{\left(r(t)\cos t - r(\vartheta_k)\cos\vartheta_k\right)^2 + \left(r(t)\sin t - r(\vartheta_k)\sin\vartheta_k\right)^2} - \frac{S}{\mathsf{M}_{\text{bry}}} = 0, \tag{3.6}$$

to get $t = \vartheta_k, k = 2, \dots, \mathsf{M}_{\text{bry}}$, respectively. From the angles $\vartheta_k, k = 1, \dots, \mathsf{M}_{\text{bry}}$, we can create a uniform boundary collocation point distribution by taking

$$(x_{\mathsf{M}_{\text{int}}+i}, y_{\mathsf{M}_{\text{int}}+i}) = r(\vartheta_i)\,(\cos\vartheta_i, \sin\vartheta_i)\,, \quad i = 1, \dots, \mathsf{M}_{\text{bry}}.$$

We consider the interior collocation points $(x_i, y_i)\,, i = 1, \dots, \mathsf{M}_{\text{int}}$ as the *Halton* points [9, Appendix A.1] in most scenarios. We compute the (boundary and interior) points $\{(\tilde{\mathrm{x}}_{\mathsf{n}}, \tilde{\mathrm{y}}_{\mathsf{n}})\}_{\mathsf{n}=1}^{\mathsf{N}}$ in the same way, and the centres are determined using (2.3).

There are two boundary conditions when solving fourth-order BVPs. Thus, to avoid a singular system matrix if we have a square system, the collection of bounndary centres subject to the boundary conditions must be distinct, see e.g. [22]. The first collection of boundary centres $\{(\mathrm{x}_{\mathsf{n}}, \mathrm{y}_{\mathsf{n}})\}_{\mathsf{n}=\mathsf{N}_{\text{int}}+1}^{\mathsf{N}_{\text{int}}+\mathsf{N}_{\text{bry}}}$ is created as described above.
The second collection of centres $\{(\mathrm{x}_{\mathsf{n}}, \mathrm{y}_{\mathsf{n}})\}_{\mathsf{n}=\mathsf{N}_{\text{int}}+\mathsf{N}_{\text{bry}}}^{\mathsf{N}_{\text{int}}+2\mathsf{N}_{\text{bry}}}$ were distributed in three ways:

Method 1: By taking $\vartheta_1 = \pi/\mathsf{N}_{\text{bry}}$ in (3.6), i.e. by shifting the original collection of boundary centres, we choose these boundary centres at different points.

Method 2: Based on the assumption that the domain's boundary is star-shaped and centred at the origin, we use a magnification of the first collection of centres to position the second collection of centres on a curve that lies outside the domain, see [5]. More specifically, we choose the second collection to be

$$\left(\mathrm{x}_{\mathsf{N}_{\text{int}}+\mathsf{N}_{\text{bry}}+i}, \mathrm{y}_{\mathsf{N}_{\text{int}}+\mathsf{N}_{\text{bry}}+i}\right) = \xi\left(\mathrm{x}_{\mathsf{N}_{\text{int}}+i}, \mathrm{y}_{\mathsf{N}_{\text{int}}+i}\right), \quad i = 1, \dots, \mathsf{N}_{\text{bry}}, \tag{3.7}$$

where ξ is a predetermined (fixed) magnification parameter.

Method 3: Same as Method 2, however we allow the magnification parameter ξ to be free and consider it as one of the unknowns. Note that $\boldsymbol{F}(\boldsymbol{a}, \boldsymbol{c}, \eta)$ in (2.8) is substituted by $\boldsymbol{F}(\boldsymbol{a}, \boldsymbol{c}, \eta, \xi)$ in this method and there is a total of $2\,(\mathsf{N}_{\text{int}} + 2\mathsf{N}_{\text{bry}}) + 2$ unknowns.

4. Numerical examples

4.1. **Example 1.** First we study the second order BVP [13, 23, 24] in a peanut–shaped domain (see Fig.2)

$$\mathcal{N}u = \Delta u - 4u^3 = 0 \quad \text{in} \quad \Omega, \tag{4.1}$$

subject to Dirichlet boundary conditions that lead to the exact solution

$$u(x, y) = \frac{1}{4 + x + y}\,, \tag{4.2}$$

shown in Figure 1. The boundary of the domain $\partial\Omega$ can be parametrized as in (3.5) with

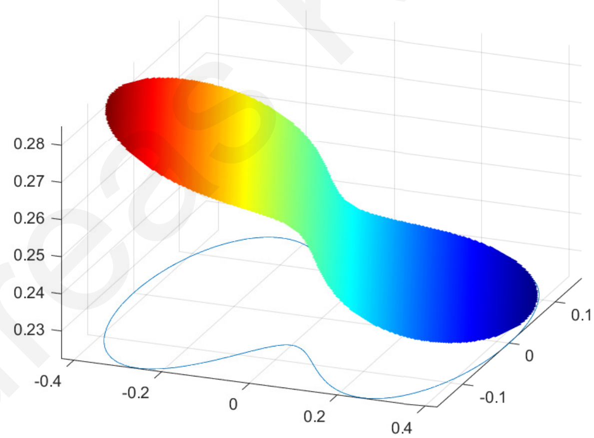
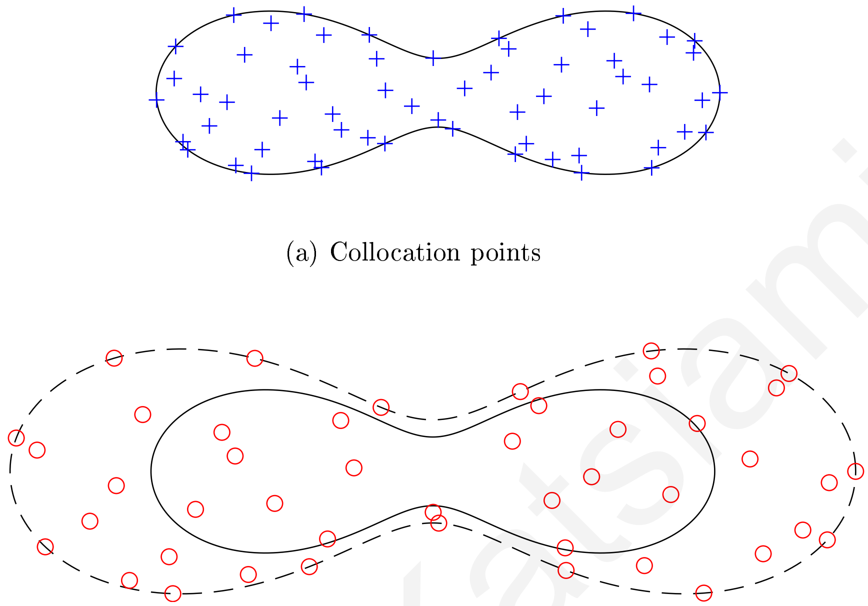


Figure 1. Examples 1 and 3: Exact solution

$$r(\vartheta) = 0.3\sqrt{\cos 2\vartheta + \sqrt{1.1 - \sin^2 2\vartheta}}. \tag{4.3}$$

In this example, standard distributions of collocation and centre points are presented in Figure 2 and the test points are a collection of $L = 300$ interior Halton points.



(a) Collocation points

(b) Centres

FIGURE 2. Example 1: Standard distributions of centres (o) and collocation points (+).

For Approaches 1 and 2, respectively, we present standard results for various initial values η_0 of the magnification parameter (with final value η) and different iteration numbers `niter` in Tables 1 and 2, with M_{int}=200, $\mathsf{M} = 300$, $\mathsf{N}_{\text{int}} = 50$, $\mathsf{N} = 70$. In contrast to the findings presented in [11] (which were already an improvement on those in [13]), these results demonstrate a considerable improvement in precision. Furthermore, the precision achieved with Approaches 1 and 2 is almost identical.

4.2. **Example 2.** Based on [8,10] and [11], the governing equation for the second order BVP is

$$\mathcal{N}u = -\varepsilon^2 \Delta u - u + u^3 \quad \text{in the unit square} \quad \Omega, \tag{4.4a}$$

with a homogeneous Dirichlet boundary condition

$$\mathcal{B}u = u = 0 \quad \text{on} \quad \partial\Omega, \tag{4.4b}$$

where $\varepsilon > 0$ is a given constant. The BVP (4.4) has the analytical solution

$$u(x,y) = \left(1 + \mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-x/\varepsilon} - \mathrm{e}^{(x-1)/\varepsilon}\right)\left(1 + \mathrm{e}^{-1/\varepsilon} - \mathrm{e}^{-y/\varepsilon} - \mathrm{e}^{(y-1)/\varepsilon}\right). \tag{4.5}$$

The M_{bry} boundary collocation points are uniformly distributed on $\partial\Omega$, while M_{int} Halton points are selected in Ω for the interior collocation points. Consequently, N_{int} interior points $\{(\tilde{\mathsf{x}}_{\mathsf{n}}, \tilde{\mathsf{y}}_{\mathsf{n}})\}_{\mathsf{n}=1}^{\mathsf{N}}$ and N_{bry} boundary points are chosen in $\overline{\Omega}$ and the centres are calculated using (2.3). We choose

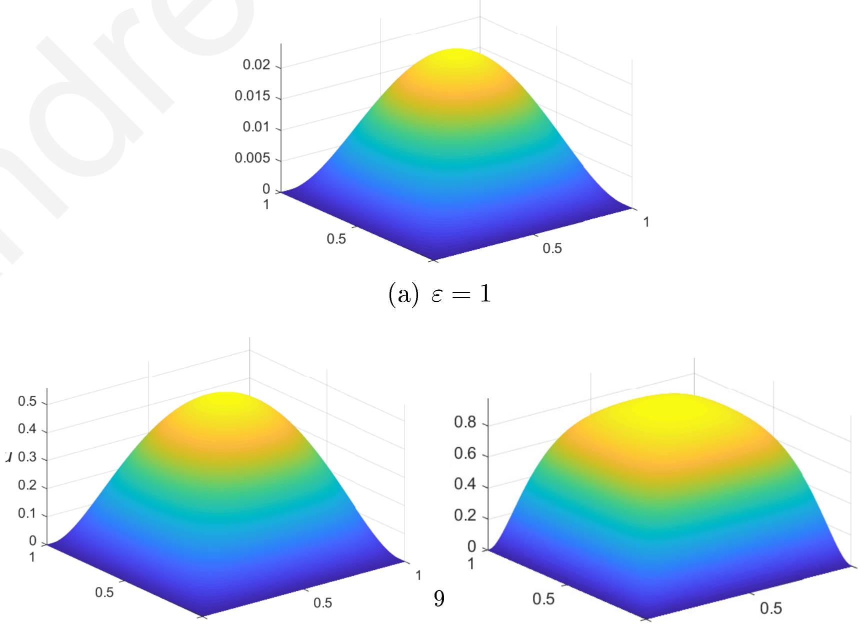
η_0	η	c_0	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	e	E	$\mathcal{E}$
1.500	1.454	1.000	0.476	1.272	101	24.87	9.869e-04	3.481e-03	3.793e-04
1.500	1.842	1.000	0.397	1.404	501	119.55	1.873e-04	6.605e-04	5.824e-05
1.500	2.376	1.000	0.241	1.568	1001	328.58	1.327e-05	4.682e-05	4.200e-06
1.5 00	2.665	1.000	0.215	1.598	2001	480.76	3.225e-06	1.138e-05	9.124e-07
3.000	3.299	1.000	0.842	1.078	101	31.97	1.025e-04	3.616e-04	3.375e-05
3.000	3.541	1.000	0.730	1.119	501	158.92	1.347e-05	4.751e-05	4.762e-06
3.000	3.646	1.000	0.667	1.152	1001	247.28	4.088e-06	1.442e-05	1.138e-06
3.000	3.720	1.000	0.632	1.188	2001	478.43	1.799e-06	6.346e-06	5.223e-07
5.000	5.087	1.000	0.960	1.032	101	33.12	2.903e-05	1.024e-04	8.934e-06
5.000	5.153	1.000	0.878	1.134	501	158.01	2.528e-06	8.917e-06	8.272e-07
5.000	5.166	1.000	0.869	1.137	1001	249.32	1.580e-06	5.574e-06	4.632e-07
5.000	5.254	1.000	0.848	1.089	2001	586.14	1.177e-06	4.150e-06	3.808e-07
6.500	6.533	1.000	0.986	1.027	101	27.12	1.996e-05	7.042e-05	6.502e-06
6.500	6.580	1.000	0.941	1.057	501	134.49	3.167e-06	1.117e-05	9.110e-07
6.500	6.609	1.000	0.900	1.063	1001	262.77	1.746e-06	6.157e-06	5.042e-07
6.500	6.637	1.000	0.859	1.088	2001	614.07	8.785e-07	3.099e-06	2.773e-07
8.000	8.017	1.000	0.995	1.019	101	31.77	1.510e-05	5.327e-05	5.277e-06
8.000	8.044	1.000	0.967	1.064	501	160.99	2.510e-06	8.852e-06	7.002e-07
8.000	8.060	1.000	0.943	1.098	1001	257.82	1.491e-06	5.259e-06	4.360e-07
8.000	8.190	1.000	0.763	1.223	2001	577.22	5.900e-07	2.081e-06	1.917e-07
9.500	9.521	1.000	0.984	1.056	101	26.67	6.509e-06	2.296e-05	2.270e-06
9.500	9.550	1.000	0.907	1.132	501	128.62	9.828e-07	3.467e-06	3.261e-07
9.500	9.576	1.000	0.877	1.216	1001	258.24	5.345e-07	1.885e-06	1.557e-07
9.500	9.598	1.000	0.858	1.267	2001	540.51	3.100e-07	1.094e-06	8.311e-08

TABLE 1. Example 1: Approach 1: Results for different numbers of iterations and various values of η_0, M_{int}=200, M = 300, N_{int} = 50, N = 70 and $c_0 = 1$.

$L = 50$ for the interior Halton test points. Figure 3 shows the exact solutions for $\varepsilon = 1, 0.25$ and 0.1 in (4.5), from which it is obvious that as ε decreases, it becomes increasingly difficult to approximate the solution, due to the presence of boundary layers.

η_0	η	$d_{\min}$	$d_{\max}$	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	$\mathtt{e}$	E	$\mathcal{E}$
1.500	1.856	0.500	3.000	0.489	3.001	101	30.33	8.788e-06	3.100e-05	2.080e-06
1.500	1.967	0.500	3.000	0.485	3.001	501	148.79	5.054e-06	1.783e-05	1.495e-06
1.500	2.017	0.500	3.000	0.483	3.001	1001	312.67	4.448e-06	1.569e-05	1.247e-06
1.500	2.179	0.500	3.000	0.485	2.994	2001	613.07	2.889e-06	1.019e-05	6.143e-07
3.000	3.118	0.500	3.000	0.497	3.000	101	30.33	1.657e-05	5.845e-05	3.362e-06
3.000	3.173	0.500	3.000	0.497	2.999	501	149.13	8.243e-06	2.907e-05	1.648e-06
3.000	3.217	0.500	3.000	0.490	2.998	1001	298.13	2.157e-06	7.610e-06	5.662e-07
3.000	3.325	0.500	3.000	0.474	2.998	2001	592.36	8.226e-07	2.902e-06	2.280e-07
5.000	5.022	0.500	3.000	0.500	3.000	101	29.60	4.300e-06	1.517e-05	1.169e-06
5.000	5.127	0.500	3.000	0.509	3.001	501	148.82	2.938e-06	1.036e-05	7.976e-07
5.000	5.177	0.500	3.000	0.510	3.001	1001	310.44	1.682e-06	5.932e-06	4.677e-07
5.000	5.311	0.500	3.000	0.507	2.999	2001	624.20	6.724e-07	2.372e-06	1.863e-07
6.500	6.517	0.500	3.000	0.501	3.000	101	26.01	4.289e-06	1.513e-05	1.182e-06
6.500	6.523	0.500	3.000	0.534	3.000	501	138.60	1.153e-06	4.066e-06	3.679e-06
6.500	6.573	0.500	3.000	0.544	3.000	1001	289.12	4.921e-07	1.736e-06	1.473e-07
6.500	6.615	0.500	3.000	0.556	3.001	2001	545.24	2.174e-07	7.667e-07	6.875e-08
8.000	8.000	0.500	3.000	0.499	3.000	101	32.16	1.031e-05	3.635e-05	2.979e-06
8.000	8.027	0.500	3.000	0.536	3.000	501	152.04	2.506e-06	8.840e-06	7.986e-07
8.000	8.048	0.500	3.000	0.535	3.000	1001	302.18	1.524e-06	5.377e-06	4.797e-07
8.000	8.064	0.500	3.000	0.536	3.000	2001	632.64	5.702e-07	2.011e-06	1.669e-07
9.500	9.506	0.500	3.000	0.513	3.000	101	29.32	3.703e-06	1.306e-05	9.945e-07
9.500	9.515	0.500	3.000	0.536	3.000	501	136.73	1.990e-06	7.018e-06	6.161e-07
9.500	9.539	0.500	3.000	0.537	3.000	1001	276.51	7.505e-07	2.647e-06	2.366e-07
9.500	9.579	0.500	3.000	0.540	3.000	2001	568.82	3.415e-07	1.204e-06	1.060e-07

TABLE 2. Example 1: Approach 2: Results for different numbers of iterations and various values of η_0, M_{int}=200, M = 300, N_{int} = 50, N = 70.



(a) $\varepsilon = 1$

(b) $\varepsilon = 0.25$

(c) $\varepsilon = 0.1$

FIGURE 3. Example 2: Analytical solutions for different ε.

4.2.1. *Case* $\varepsilon = 1$. This is the most straightforward case and Table 3 shows results calculated for different iteration numbers `niter` for Approaches 1 and 2 with M_{int}=200, $\mathsf{M} = 276$, $\mathsf{N}_{\text{int}} = 60$, and $\mathsf{N} = 80$. These results are significantly more precise than those in [11], see also [12,13], while the precision obtained with Approaches 1 and 2 is very similar.

(a) Approach 1

η_0	η	c_0	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	E	$\mathcal{E}$
1.500	1.504	4.000	2.818	4.213	101	35.69	7.805e-02	5.227e-04
1.500	1.874	4.000	0.742	4.586	501	176.04	4.386e-03	2.780e-05
1.500	2.069	4.000	0.607	4.838	1001	338.40	6.819e-04	4.964e-06
1.500	2.133	4.000	0.593	4.924	2001	675.50	4.63e-04	2.975e-06
4.000	5.128	4.000	2.554	4.147	101	32.95	2.071e-02	1.688e-04
4.000	5.693	4.000	1.945	4.288	501	143.47	1.421e-02	1.213e-04
4.000	6.160	4.000	1.465	4.635	1001	285.64	8.506e-03	7.053e-05
4.000	6.317	4.000	1.226	4.864	2001	589.93	5.553e-03	4.626e-05

(b) Approach 2

η_0	η	$d_{\min}$	$d_{\max}$	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	E	$\mathcal{E}$
1.500	1.701	1.000	5.000	0.933	5.000	101	28.84	9.237e-03	9.008e-05
1.500	1.850	1.000	5.000	0.769	5.001	501	141.58	2.691e-03	2.243e-05
1.500	1.910	1.000	5.000	0.602	5.003	1001	284.40	2.281e-03	1.601e-05
1.500	2.044	1.000	5.000	0.332	5.010	2001	574.63	3.744e-04	3.161e-06
4.000	3.924	1.000	5.000	0.779	5.001	101	31.97	2.055e-02	1.921e-04
4.000	3.910	1.000	5.000	0.598	5.005	501	183.94	2.629e-03	1.285e-05
4.000	3.914	1.000	5.000	0.581	5.007	1001	371.06	2.054e-03	1.054e-05
4.000	3.941	1.000	5.000	0.552	5.007	2001	745.83	2.351e-04	1.441e-06

TABLE 3. Example 2, $\varepsilon = 1$: Results with M_{int}=200, $\mathsf{M} = 276$, $\mathsf{N}_{\text{int}} = 60$, $\mathsf{N} = 80$.

4.2.2. *Case* $\varepsilon = 0.25$. This case is more complicated and numerical results are presented in Table 4 for different iteration numbers `niter` for Approaches 1 and 2, with M_{int}=200, $\mathsf{M} = 276$, $\mathsf{N}_{\text{int}} = 60$, and $\mathsf{N} = 80$. We once more notice a significant improvement in precision compared with the corresponding results of [11], also see [12, 13], whereas Approaches 1 and 2 result in similar precision.

4.2.3. *Case* $\varepsilon = 0.1$. This case is the most difficult as we require to take more degrees and standard results are shown in Table 5 for different iteration numbers `niter` for Approaches 1 and 2, with M_{int}=400, $\mathsf{M} = 476$, $\mathsf{N}_{\text{int}} = 150$, and $\mathsf{N} = 190$. These results are more precise than those in [11], see also [12,13], and, in contrast to the above cases, the precision achieved with Approach 2 is superior

(a) Approach 1

η_0	η	c_0	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	E	$\mathcal{E}$
1.500	2.029	4.000	2.623	5.660	101	35.34	3.618e-03	6.159e-04
1.500	2.083	4.000	1.924	6.548	501	175.05	1.746e-03	2.986e-04
1.500	2.110	4.000	1.637	6.968	1001	326.36	1.243e-03	2.217e-04
1.500	2.213	4.000	1.127	8.230	2001	736.14	7.369e-04	1.030e-04
2.000	2.219	4.000	3.121	5.133	101	29.33	1.049e-02	1.893e-03
2.000	2.404	4.000	1.492	5.665	501	145.53	2.562e-03	3.381e-04
2.000	2.545	4.000	1.179	6.484	1001	294.65	1.850e-03	2.281e-04
2.000	2.672	4.000	1.080	7.310	2001	675.97	1.357e-03	1.638e-04

(b) Approach 2

η_0	η	$d_{\min}$	$d_{\max}$	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	E	$\mathcal{E}$
1.500	2.297	1.000	6.000	0.656	7.098	101	28.79	3.841e-03	7.288e-04
1.500	2.444	1.000	6.000	0.601	7.806	501	142.31	2.365e-03	4.968e-04
1.500	2.501	1.000	6.000	0.548	8.203	1001	342.47	1.921e-03	3.908e-04
1.500	2.562	1.000	6.000	0.480	8.654 2001	630.05	1.837e-03	2.806e-04	
2.000	2.443	1.000	6.000	0.978	6.516	101	33.77	4.935e-03	8.470e-04
2.000	2.528	1.000	6.000	0.933	6.938	501	181.91	2.245e-03	3.338e-04
2.000	2.582	1.000	6.000	0.898	7.229	1001	350.87	2.130e-03	2.913e-04
2.000	2.774	1.000	6.000	0.710	8.444	2001	615.00	1.108e-03	1.620e-04

TABLE 4. Example 2, $\varepsilon = 0.25$: Results with M_{int}=200, $\mathsf{M} = 276$, $\mathsf{N}_{\text{int}} = 60$, $\mathsf{N} = 80$.

to that of Approach 1. Remarkably, in comparison to the current approach, the results obtained in [11] for this value of ε were computed by enhancing approximation (2.2) with polynomial basis functions, therefore more degrees of freedom were required as well as much greater initial values for the shape parameters, to give adequate precision.

4.3. **Example 3.** We consider the fourth order BVP [13] consisting the following governing equation in the peanut–shaped domain Ω

$$\mathcal{N}u = \Delta^2 u - 96u^5 = 0, \tag{4.6}$$

subject to the boundary conditions $u = g_1$ and $\partial u/\partial n = g_2$ (*cf.* (2.6b)), where g_1 and g_2 refer to the exact solution (4.2). See also Figure 1 and the domain Ω described by (3.5) and (4.3). The domain discretization is similar to that of Example 1. For r specified in (4.3),

$$\boldsymbol{n} = \frac{1}{\sqrt{r^2(\vartheta) + r'^2(\vartheta)}} \Big(r'(\vartheta)\sin\vartheta + r(\vartheta)\cos\vartheta,\ r(\vartheta)\sin\vartheta - r'(\vartheta)\cos\vartheta \Big)$$

(a) Approach 1

η_0	η	c_0	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	E	$\mathcal{E}$
1.200	1.990	15.000	12.348	15.860	51	124.56	1.675e-02	3.521e-03
1.200	2.021	15.000	10.735	16.414	101	245.46	1.045e-02	2.649e-03
1.200	2.032	15.000	8.801	16.913	201	489.99	7.166e-03	2.155e-03
1.500	1.701	15.000	14.474	15.265	51	150.49	1.963e-02	7.152e-03
1.500	1.759	15.000	13.690	15.881	101	314.25	7.294e-03	2.258e-03
1.500	1.792	15.000	11.371	16.505	201	551.19	6.461e-03	2.058e-03

(b) Approach 2

η_0	η	$d_{\min}$	$d_{\max}$	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	E	$\mathcal{E}$
1.200	1.404	8.000	20.000	6.907	20.011	51	128.09	3.538e-03	1.175e-03
1.200	1.414	8.000	20.000	6.066	20.157	101	246.82	2.707e-03	8.898e-04
1.200	1.425	8.000	20.000	4.394	20.318	201	628.54	2.370e-03	6.829e-04
1.500	1.636	8.000	20.000	6.444	20.014	51	125.07	1.300e-02	2.698e-03
1.500	1.651	8.000	20.000	5.691	20.023	101	250.98	1.101e-02	2.124e-03
1.500	1.671	8.000	20.000	3.540	20.504	201	487.72	8.180e-03	1.577e-03

TABLE 5. Example 2, $\varepsilon = 0.1$: Results with M_{int}=400, $\mathsf{M} = 476$, $\mathsf{N}_{\text{int}} = 150$, $\mathsf{N} = 190$.

gives the outward unit normal vector $\boldsymbol{n}$ to the boundary that is needed for the application of the proposed methodology. Tables 6, 7 and 8 show results for different numbers of iterations `niter` in Methods 1, 2 and 3, respectively, and various initial values η_0, ξ_0, for Approach 2 with $d_{\min} = 0.5$ and $d_{\max} = 3$, M_{int}=200, $\mathsf{M} = 300$, $\mathsf{N}_{\text{int}} = 50$, and $\mathsf{N} = 70$. Once more, a considerable improvement in precision is noted as compared to the corresponding results in [11], see also [12,13]. Furthermore, Method 3 provides the most precise results, whereas Method 2 outperforms Method 1.

η_0	η	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	e	E	$\mathcal{E}$
2.000	2.570	0.499	2.996	101	67.58	1.054e-04	3.719e-04	4.697e-05
2.000	2.911	0.490	3.003	501	324.15	1.139e-04	4.017e-04	4.153e-05
2.000	3.117	0.460	3.010	1001	634.88	2.777e-05	9.797e-05	1.037e-05
2.000	3.166	0.454	3.007	2001	1262.58	2.177e-05	7.679e-05	8.844e-06
4.000	4.218	0.500	3.000	101	67.71	6.922e-05	2.442e-04	2.676e-05
4.000	4.635	0.510	3.001	501	316.37	1.770e-05	6.242e-05	7.440e-06
4.000	4.857	0.473	3.002	1001	627.86	1.474e-05	5.200e-05	6.743e-06
4.000	5.055	0.436	3.007	2001	1290.45	1.373e-05	4.843e-05	6.191e-06
7.000	7.008	0.527	3.000	101	77.57	2.718e-05	9.588e-05	1.038e-05
7.000	7.049	0.527	3.000	501	351.15	1.647e-05	5.808e-05	5.940e-06
7.000	7.279	0.527	3.000	1001	712.13	9.448e-06	3.333e-05	3.693e-06
7.000	7.467	0.560	3.037	2001	1408.78	8.470e-06	2.988e-05	3.288e-06
7.500	7.505	0.527	3.000	101	71.66	3.025e-05	1.067e-04	1.101e-05
7.500	7.520	0.507	3.000	501	351.70	9.242e-06	3.260e-05	3.709e-06
7.500	7.598	0.527	3.000	1001	705.01	8.411e-06	2.967e-05	3.391e-06
7.500	7.716	0.548	3.000	2001	1402.62	8.045e-06	2.838e-05	3.174e-06
9.000	8.968	0.507	3.000	101	67.33	2.703e-05	9.533e-05	1.188e-05
9.000	8.935	0.497	3.000	501	327.06	1.550e-05	5.469e-05	7.097e-06
9.000	8.938	0.494	3.000	1001	627.78	1.497e-05	5.280e-05	7.042e-06
9.000	9.122	0.488	3.000	2001	1252.72	8.882e-06	3.133e-05	3.660e-06

TABLE 6. Example 3: Method 1: Results with M_{int}=200, $\mathsf{M} = 300$, $\mathsf{N}_{\text{int}} = 50$, $\mathsf{N} = 70$.

4.4. **Example 4.** Finally, we examine the Navier–Stokes equations in a wavy chanel [11, 13, 15, 21, 25] for a fourth-order BVP with

$$\mathcal{N}\psi = \Delta^2\psi - \mathrm{Re}\left(\frac{\partial\psi}{\partial y}\frac{\partial\Delta\psi}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial\Delta\psi}{\partial y}\right) = 0 \quad \text{in} \quad \Omega, \tag{4.7a}$$

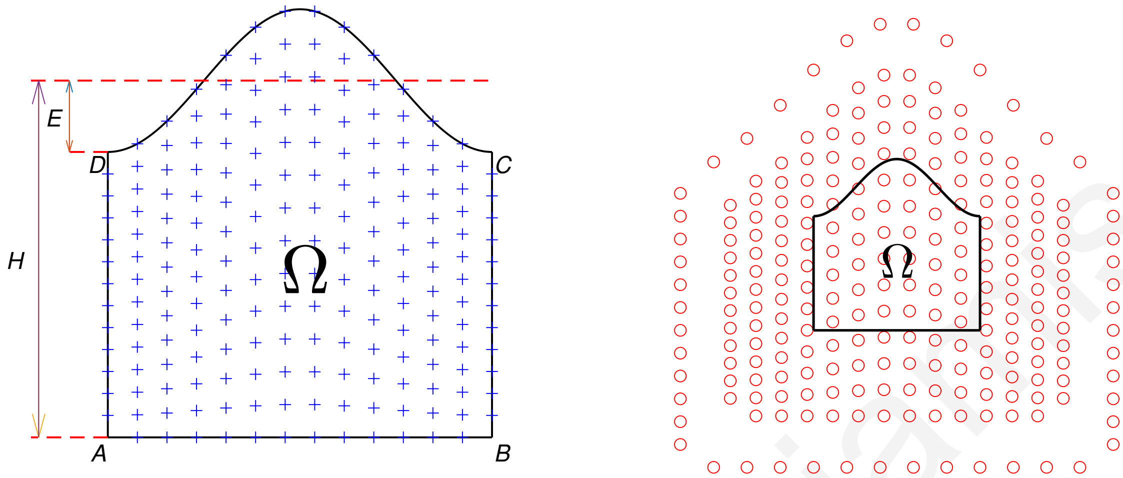
subject to

$$\psi = 0 \quad \text{and} \quad \frac{\partial^2\psi}{\partial y^2} = 0 \quad \text{on} \quad AB, \tag{4.7b}$$

$$\psi = 1,\ \frac{\partial\psi}{\partial n} = 0 \quad \text{on} \quad CD, \tag{4.7c}$$

$$\psi = \frac{3\,(H-E)^2\,y - y^3}{2\,(H-E)^3} \quad \text{and} \quad \frac{\partial\psi}{\partial x} = 0 \quad \text{on} \quad BC \text{ and } DA, \tag{4.7d}$$

where Re is the Reynolds number in (4.7a), ψ is the stream–function and the computational domain Ω is shown in Figure 4(a).



(a) Collocation points (b) Centres

FIGURE 4. Example 4: Computational domain and standard distributions of collocation points (+) and centres (o).

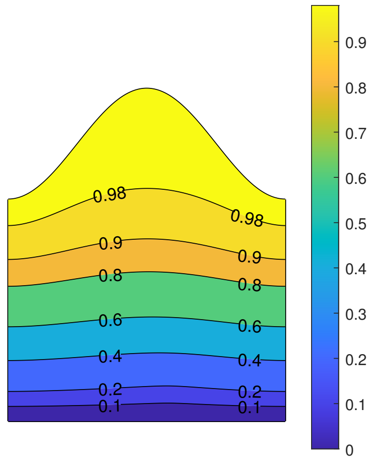
We took the length $AB = 1$, and the boundary's CD profile was chosen to be $y = H - E\cos(2\pi x)$, while $E = 1/5$, $H = 1$, as in previous studies. As a consequence, in boundary condition (4.7c), the normal derivative is

$$\frac{\partial \psi}{\partial n} = \frac{1}{\sqrt{1 + 4\pi^2 E^2 \sin^2(2\pi x)}} \left(\frac{\partial \psi}{\partial y} - 2\pi E \sin(2\pi x) \frac{\partial \psi}{\partial x} \right).$$

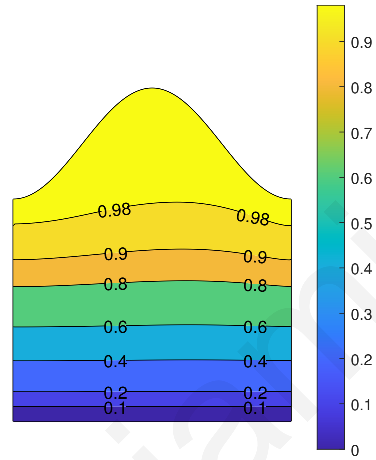
We used $\mathsf{M}_{\text{int}} = m^2$ interior and $\mathsf{M}_{\text{bry}} = 4m$ boundary collocation points for a specified number m, seen in Figure 4(a). For a specified n, the centres are computed using (2.3), yielding $\mathsf{N}_{\text{int}} = n^2$ interior and $\mathsf{N}_{\text{bry}} = 4n$ boundary centres; see Figure 4(b).
Here, we utilized the shifted polyharmonic spline (PS) [3] (see also the Appendix) as in [11],

$$\Phi(c_{\mathsf{n}}, r_{\mathsf{n}}) = r_{\mathsf{n}}^2 \log\left(\sqrt{r_{\mathsf{n}}^2 + c_{\mathsf{n}}^2}\right). \tag{4.8}$$

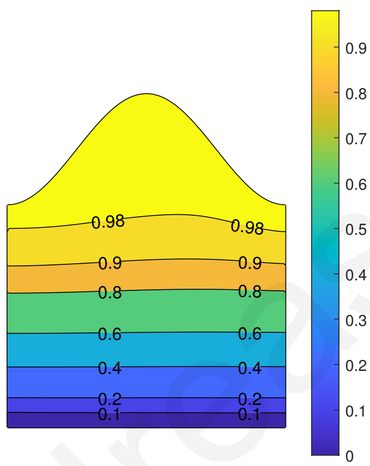
Figure 5 shows the stream–function contour plots for Re $= 0, 20, 40$ and 80 obtained utilizing Approach 2 and Method 3, with $m = 13, n = 8$, $\xi_0 = 1.2$, $\eta_0 = 2$, $d_{\min} = 0.01$, $d_{\max} = 0.1$, and 5000 iterations. (It should be noted that 2000 iterations were required to get satisfactory results for Re=20.) These findings agree very well with those found in [11], also obtained and compared with the COMSOL Multiphysics® finite element package [7], see also [13, 15]. It is worth noting that, in comparison to the results in [11] and [13], ours were acquired without adding any polynomial basis functions in the RBF approximation (2.2).



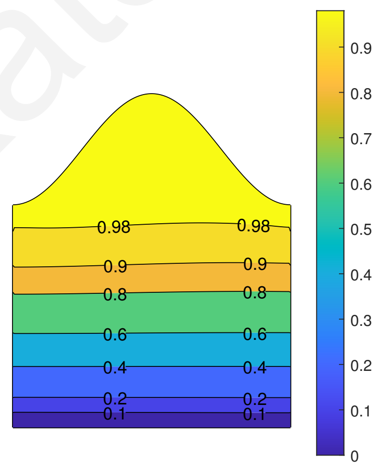
(a) Re = 0



(b) Re = 20



(c) Re = 40



(d) Re = 80

FIGURE 5. Example 4: Streamlines for Re $= 0, 20, 40, 80$.

5. CONCLUSIONS

In order to solve nonlinear boundary value problems of second and fourth order utilizing fictitious centres, novel Kansa–RBF collocation methodologies have been created. Based on a magnification parameter, the fictitious centres are dispersed around a region resembling and containing the problem's physical domain. In the suggested methodology, this magnification parameter is considered to be one of the unknowns in the nonlinear system of equations obtained by the Kansa–RBF discretization, along with the coefficients in the RBF expansion and the shape parameters

related with each RBF. The MATLAB© routines `fsolve` and `lsqnonlin` were used to solve these nonlinear systems. The initial shape parameter values were once again analyzed and two initial shape parameter distributions were studied, as in [11]. All the initial values were considered equal in the first approach, while they were uniformly distributed over a given interval in the second. Also, an analysis is carried out to select a second collection of boundary centres in fourth-order problems as these have two boundary conditions. The results of numerous numerical experiments demonstrated that the suggested formulation results in more precise approximations than the corresponding ones (with or without additional polynomial basis functions) in [13] and [11] in which the centres were defined within the physical domain of the problem. Furthermore, the proposed methodology can be used when the expansion includes any RBF (with or without a shape parameter), and it is simple to implement. Eventually, a nonlinear BVP described by higher-order partial differential equations can also be modeled with the current methodology.
It is our intention to additionaly analyze the performance of the routines `fsolve` and `lsqnonlin` when the nonlinear system's Jacobian is provided. We also plan to apply the proposed methodology to solve three–dimensional nonlinear BVPs, by allowing the use of a localized RBF method [27] rather than the global strategy.

Appendix

The derivatives of the MQ RBF (3.4) that we used are given by:

$$\left(\frac{\partial \Phi}{\partial x}(c_{\mathsf{n}},\ r_{\mathsf{n}}, x-\mathrm{x}_{\mathsf{n}}),\ \frac{\partial \Phi}{\partial y}(c_{\mathsf{n}}, r_{\mathsf{n}}, y-\mathrm{y}_{\mathsf{n}})\right) \;=\; \frac{c_{\mathsf{n}}^2}{\sqrt{(c_{\mathsf{n}}\, r_{\mathsf{n}})^2+1}}\ (x-\mathrm{x}_{\mathsf{n}},\ y-\mathrm{y}_{\mathsf{n}})\ , \tag{A.1}$$

$$\frac{\partial^2 \Phi}{\partial x^2}(c_{\mathsf{n}}, r_{\mathsf{n}}, y-\mathrm{y}_{\mathsf{n}}) \;=\; \frac{c_{\mathsf{n}}^2\,(c_{\mathsf{n}}^2\,(y-\mathrm{y}_{\mathsf{n}})^2+1)}{\left((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+1\right)^{3/2}}\ , \quad \frac{\partial^2 \Phi}{\partial y^2}(c_{\mathsf{n}}, r_{\mathsf{n}}, x-\mathrm{x}_{\mathsf{n}}) \;=\; \frac{c_{\mathsf{n}}^2\,(c_{\mathsf{n}}^2\,(x-\mathrm{x}_{\mathsf{n}})^2+1)}{\left((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+1\right)^{3/2}}\ , \tag{A.2}$$

$$\Delta\Phi(c_{\mathsf{n}}, r_{\mathsf{n}}) \;=\; \frac{c_{\mathsf{n}}^2\ ((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+2)}{\left((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+1\right)^{3/2}}\ , \tag{A.3}$$

$$\left(\frac{\partial \Delta\Phi}{\partial x}(c_{\mathsf{n}}, r_{\mathsf{n}}, x-\mathrm{x}_{\mathsf{n}}),\ \frac{\partial \Delta\Phi}{\partial y}(c_{\mathsf{n}}, r_{\mathsf{n}}, y-\mathrm{y}_{\mathsf{n}})\right) \;=\; -\frac{c_{\mathsf{n}}^4\ ((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+4)}{\left((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+1\right)^{5/2}}\ (x-\mathrm{x}_{\mathsf{n}},\ y-\mathrm{y}_{\mathsf{n}})\ , \tag{A.4}$$

$$\Delta^2\Phi(c_{\mathsf{n}}, r_{\mathsf{n}}) = \frac{c_{\mathsf{n}}^4\ ((c_{\mathsf{n}}\, r_{\mathsf{n}})^4+8(c_{\mathsf{n}}\, r_{\mathsf{n}})^2-8)}{\left((c_{\mathsf{n}}\, r_{\mathsf{n}})^2+1\right)^{7/2}}\ . \tag{A.5}$$

We have also used the PS RBF (4.8) derivatives given below:

$$\left(\frac{\partial \Phi}{\partial x}(c_{\mathsf{n}}, r_{\mathsf{n}}, x-\mathrm{x}_{\mathsf{n}}),\ \frac{\partial \Phi}{\partial y}(c_{\mathsf{n}}, r_{\mathsf{n}}, y-\mathrm{y}_{\mathsf{n}})\right) \;=\; \left(\log\left(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2\right)+\frac{r_{\mathsf{n}}^2}{r_{\mathsf{n}}^2+c_{\mathsf{n}}^2}\right)\ (x-\mathrm{x}_{\mathsf{n}},\ y-\mathrm{y}_{\mathsf{n}})\ , \tag{A.6}$$

$$\frac{\partial^2 \Phi}{\partial x^2}(c_{\mathsf{n}}, r_{\mathsf{n}}, x-\mathrm{x}_{\mathsf{n}}) \;=\; \log\left(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2\right)+\frac{r_{\mathsf{n}}^2}{r_{\mathsf{n}}^2+c_{\mathsf{n}}^2}+\frac{2\,(x-\mathrm{x}_{\mathsf{n}})^2\,(r_{\mathsf{n}}^2+2c_{\mathsf{n}}^2)}{(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2)^2}\,, \tag{A.7}$$

$$\frac{\partial^2 \Phi}{\partial y^2}(c_{\mathsf{n}}, r_{\mathsf{n}}, y-\mathrm{y}_{\mathsf{n}}) \;=\; \log\left(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2\right)+\frac{r_{\mathsf{n}}^2}{r_{\mathsf{n}}^2+c_{\mathsf{n}}^2}+\frac{2\,(y-\mathrm{y}_{\mathsf{n}})^2\,(r_{\mathsf{n}}^2+2c_{\mathsf{n}}^2)}{(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2)^2}\,, \tag{A.8}$$

$$\Delta\Phi(c_{\mathsf{n}}, r_{\mathsf{n}}) \;=\; 2\,\log\left(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2\right)+\frac{2\,r_{\mathsf{n}}^2\,(2r_{\mathsf{n}}^2+3c_{\mathsf{n}}^2)}{(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2)^2}\,, \tag{A.9}$$

$$\left(\frac{\partial \Delta\Phi}{\partial x}(c_{\mathsf{n}}, r_{\mathsf{n}}, x-\mathrm{x}_{\mathsf{n}}),\ \frac{\partial \Delta\Phi}{\partial y}(c_{\mathsf{n}}, r_{\mathsf{n}}, y-\mathrm{y}_{\mathsf{n}})\right) \;=\; \frac{4\,(r_{\mathsf{n}}^4+3r_{\mathsf{n}}^2c_{\mathsf{n}}^2+4c_{\mathsf{n}}^4)}{(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2)^3}\ (x-\mathrm{x}_{\mathsf{n}},\ y-\mathrm{y}_{\mathsf{n}})\ , \tag{A.10}$$

$$\Delta^2\Phi(c_{\mathsf{n}}, r_{\mathsf{n}}) \;=\; \frac{16\,c_{\mathsf{n}}^4\ (2c_{\mathsf{n}}^2-r_{\mathsf{n}}^2)}{(r_{\mathsf{n}}^2+c_{\mathsf{n}}^2)^4}\ . \tag{A.11}$$

References

[1] F. Afiatdoust and M. Esmaeilbeigi, *Optimal variable shape parameters using genetic algorithm for radial basis function approximation*, Ain Shams Eng. J. **6** (2015), 639–647.

[2] M. Benalili and K. Tahri, Nonlinear elliptic fourth order equations existence and multiplicity results, Nonlinear Differ. Equ. Appl. Vol. 18, pp. 539-556, 2011.

[3] W. Chen, Z.-J. Fu, and C. S. Chen, *Recent Advances in Radial Basis Function Collocation Methods*, Springer Briefs in Applied Sciences and Technology, Springer, Heidelberg, 2014.

[4] C. S. Chen, A. Karageorghis and F. Dou, *A novel RBF collocation method using fictitious centres*, Appl. Math. Lett. **101** (2020), 106069.

[5] C. S. Chen, A. Karageorghis and H. Zheng, *Improved RBF collocation methods for fourth order boundary value problems*, Commun. Comput. Phys. **27** (2020), 1530–1549.

[6] P. T. Church, E. N. Dancer and J. G. Timourian, The structure of a nonlinear elliptic operator, Transactions of the American Mathematical Society, Vol. 338, pp. 1-42, 1993.

[7] COMSOL Multiphysics® v. 5.2. `www.comsol.com` , COMSOL AB, Stockholm, Sweden.

[8] G. E. Fasshauer, *Newton iteration with multiquadrics for the solution of nonlinear PDEs*, Comput. Math. Appl. **43** (2002), 423–438.

[9] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences, vol. 6, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

[10] G. E. Fasshauer, E. C. Gartland and J. W. Jerome, *Newton iteration for partial differential equations and the approximation of the identity*, Numer. Algor. **25** (2000), 181–195.

[11] M. A. Jankowska and A. Karageorghis, *Variable shape parameter Kansa RBF method for the solution of nonlinear boundary value problems*, Eng. Anal. Bound. Elem. **103** (2019), 32–40.

[12] M. A. Jankowska, A. Karageorghis and C. S. Chen, *Kansa RBF method for nonlinear problems*, Int. J. Comp. Meth. Exp. Meas. **6** (2018), 1000–1007.

[13] M. A. Jankowska, A. Karageorghis and C. S. Chen, *Improved Kansa RBF method for the solution of nonlinear boundary value problems*, Eng. Anal. Bound. Elem. **87** (2018), 173–183.

[14] E. J. Kansa, *Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appl. **19** (1990), 147–161.

[15] J. A. Kołodziej and J. K. Grabski, *Application of the method of fundamental solutions and the radial basis functions for viscous laminar flow in wavy channel*, Eng. Anal. Bound. Elem. **57** (2015), 58–65.

[16] M. Li, C. S. Chen, and A. Karageorghis, *The MFS for the solution of harmonic boundary value problems with non-harmonic boundary conditions*, Comput. Math. Appl. **66** (2013), 2400–2424.

[17] C. S. Liu, *An algorithm with m-step residual history for solving linear equa- tions: Data interpolation by a multi-shape-factors RBF*, Eng. Anal. Bound. Elem. **51** (2015), 123–135.

[18] C. S. Liu and D. Liu, *Optimal shape parameter in the MQ-RBF by minimizing an energy gap functional*, Appl. Math. Lett. **86** (2018), 157–165.

[19] The MathWorks, Inc., 3 Apple Hill Dr., Natick, MA, *Matlab*.

[20] S. A. Sarra and D. Sturgill, *A random variable shape parameter strategy for radial basis function approximation methods*, Eng. Anal. Bound. Elem. **33** (2009), 1239–1245.

[21] I. J. Sobey, *On flow through furrowed channels. Part 1. Calculated flow patterns*, J. Fluid Mech. **96** (1980), 1–26.

[22] D. Tappoura, *Kansa RBF methods for the solution of second and fourth order boundary value problems*, MSc thesis, Department of Mathematics and Statistics, University of Cyprus, 2020.

[23] C.-C. Tsai, *Homotopy method of fundamental solutions for solving certain nonlinear partial differential equations*, Eng. Anal. Bound. Elem. **36** (2012), 1226–1234.

[24] C.-C. Tsai, C.-S. Liu and W.-C. Yeih, *Fictitious time integration method of fundamental solutions with Chebyshev polynomials for solving Poisson-type nonlinear PDEs*, CMES Comput. Model. Eng. Sci. **56** (2010), 131–151.
[25] C. C. Wang and C. K. Chen, *Forced convection in a wavy-wall channel*, Int. J. Heat Mass Transf. **45** (2002), 2587–2595.
[26] S. Xiang, K-M. Wang, Y-T. Ai, Y-D. Sha and H. Shi, *Thrigonometric variable shape parameter and exponent strategy for generalized multiquadric radial basis function approximation*, Appl. Math. Model. **36** (2012), 1931 – 1938.
[27] G. Yao, J. Kolibal, and C. S. Chen, *A localized approach for the method of approximate particular solutions*, Comput. Math. Appl. **61** (2011), 2376–2387.

ξ_0	η_0	η	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	`e`	E	$\mathcal{E}$
2.000	2.000	2.177	0.490	3.000	101	65.73	7.007e-05	2.471e-04	2.834e-05
2.000	2.000	2.442	0.469	3.002	501	370.48	6.083e-05	2.146e-04	2.943e-05
2.000	2.000	2.600	0.465	3.000	1001	684.67	2.497e-05	8.806e-05	9.720e-06
2.000	2.000	2.738	0.486	3.000	2001	1249.20	1.570e-05	5.537e-05	5.878e-06
2.000	4.000	4.065	0.487	3.000	101	64.88	3.170e-05	1.118e-04	1.278e-05
2.000	4.000	4.057	0.448	3.000	501	348.99	1.809e-05	6.379e-05	7.090e-06
2.000	4.000	4.062	0.410	3.000	1001	945.29	1.748e-05	6.165e-05	6.558e-06
2.000	4.000	4.089	0.335	3.000	2001	1252.82	6.099e-06	2.151e-05	2.350e-06
2.000	7.000	6.996	0.500	3.000	101	79.31	4.568e-06	1.611e-05	1.956e-06
2.000	7.000	7.042	0.459	3.000	501	332.10	2.932e-06	1.034e-05	1.104e-06
2.000	7.000	7.135	0.460	3.000	1001	923.23	2.257e-06	7.960e-06	8.266e-07
2.000	7.000	7.230	0.411	3.000	2001	1239.11	1.584e-06	5.586e-06	5.798e-07
2.000	7.500	7.501	0.500	3.000	101	87.29	1.301-05	4.591-05	7.184-06
2.000	7.500	7.491	0.477	3.000	501	434.52	3.826-06	1.349-05	1.417-06
2.000	7.500	7.503	0.482	3.000	1001	805.02	3.058-06	1.079-05	1.081-06
2.000	7.500	7.643	0.488	3.000	2001	1465.85	1.444-06	5.106-06	4.954-07
2.000	9.000	9.030	0.501	3.000	101	64.16	3.188e-05	1.124e-04	1.589e-05
2.000	9.000	9.077	0.504	3.000	501	329.51	1.748e-05	6.164e-05	8.148e-06
2.000	9.000	9.077	0.454	3.000	1001	888.44	3.581e-06	1.263e-05	1.236e-06
2.000	9.000	9.071	0.428	3.000	2001	1236.95	2.411e-06	8.503e-06	9.127e-07
4.000	2.000	2.223	0.499	3.000	101	64.44	3.093e-05	1.091e-04	1.782e-05
4.000	2.000	2.423	0.498	3.033	501	333.46	1.380e-05	4.868e-05	7.773e-06
4.000	2.000	2.542	0.496	3.083	1001	726.04	1.332e-05	4.699e-05	4.978e-06
4.000	2.000	2.628	0.496	3.125	2001	1402.65	1.341e-05	4.729e-05	4.631e-06
4.000	4.000	4.062	0.500	3.000	101	64.46	2.388e-06	8.425e-06	9.087e-07
4.000	4.000	4.164	0.502	3.000	501	315.86	1.199e-06	4.230e-06	4.059e-07
4.000	4.000	4.229	0.503	3.000	1001	630.17	1.121e-06	3.954e-06	3.613e-07
4.000	4.000	4.277	0.509	3.001	2001	1361.01	8.070e-07	2.847e-06	2.562e-07
4.000	7.000	6.992	0.494	3.000	101	76.52	2.672-05	9.426-05	1.269-05
4.000	7.000	6.969	0.498	3.000	501	397.62	1.044-05	3.682-06	3.985-06
4.000	7.000	6.930	0.525	3.000	1001	809.88	4.478-06	1.580-05	1.489-06
4.000	7.000	6.890	0.493	3.001	2001	1759.24	1.080-06	3.810-06	2.866-07
4.000	7.500	7.494	0.483	3.000	101	76.12	2.913e-05	1.027e-04	1.397e-05
4.000	7.500	7.469	0.477	3.000	501	311.89	1.433e-05	5.055e-05	5.905e-06
4.000	7.500	7.426	0.480	3.000	1001	672.25	7.167e-06	2.528e-05	2.290e-06
4.000	7.500	7.385	0.503	3.002	2001	1235.25	1.429e-06	5.040e-06	4.517e-07
4.000	9.000	8.845	0.388	3.001	101	64.25	1.036e-05	3.655e-05	5.143e-06
4.000	9.000	8.893	0.314	3.001	501	315.92	7.461e-06	2.632e-05	2.508e-06
4.000	9.000	8.909	0.307	3.002	1001	622.67	6.873e-06	2.424e-05	2.216e-06
4.000	9.000	8.896	0.312	3.003	2001	1238.96	5.437e-06	1.918e-05	1.704e-06

TABLE 7. Example 3: Method 2: Results with $\mathsf{M}_{\rm int}$=200, $\mathsf{M} = 300$, $\mathsf{N}_{\rm int} = 50$, $\mathsf{N} = 70$.

ξ_0	ξ	η_0	η	$c_{\min}$	$c_{\max}$	`niter`	CPU (s)	$\mathtt{e}$	E	$\mathcal{E}$
2.000	2.079	2.000	2.336	0.468	3.001	101	65.24	5.071e-05	1.789e-04	2.345e-05
2.000	2.161	2.000	2.464	0.478	2.999	501	381.88	4.812e-05	1.697e-04	2.094e-05
2.000	2.284	2.000	2.462	0.502	2.996	1001	715.88	2.590e-05	9.136e-05	9.990e-06
2.000	2.449	2.000	2.471	0.502	2.996	2001	1351.62	1.460e-05	5.150e-05	5.336e-06
2.000	1.966	4.000	4.030	0.479	3.000	101	82.47	2.786e-05	9.827e-05	1.125e-05
2.000	1.870	4.000	4.043	0.469	3.000	501	398.63	1.687e-05	5.952e-05	5.904e-06
2.000	1.809	4.000	4.107	0.454	3.000	1001	734.47	7.522e-06	2.653e-05	2.945e-06
2.000	1.861	4.000	4.210	0.419	3.000	2001	1370.32	5.172e-06	1.824e-05	2.181e-06
2.000	2.046	7.000	7.006	0.498	3.000	101	65.98	1.667e-05	5.881e-05	8.880e-06
2.000	2.190	7.000	7.028	0.486	3.000	501	390.64	2.549e-06	8.991e-06	9.537e-07
2.000	2.289	7.000	7.128	0.448	3.000	1001	714.63	1.997e-06	7.043e-06	7.123e-07
2.000	2.356	7.000	7.196	0.437	3.000	2001	1379.98	1.216e-06	4.289e-06	4.156e-07
2.000	2.044	7.500	7.505	0.500	3.000	101	82.40	2.528e-05	8.917e-05	1.270e-05
2.000	2.231	7.500	7.485	0.477	3.000	501	384.41	1.431e-06	5.048e-06	7.126e-07
2.000	2.285	7.500	7.536	0.443	3.000	1001	758.90	1.659e-06	5.851e-06	5.837e-07
2.000	2.373	7.500	7.633	0.421	3.000	2001	1805.44	1.251e-06	4.411e-06	4.174e-07
2.000	2.026	9.000	9.006	0.501	3.000	101	65.74	2.662e-05	9.390e-05	1.366e-05
2.000	2.102	9.000	9.032	0.499	3.000	501	379.75	1.915e-05	6.755e-05	9.549e-06
2.000	2.150	9.000	9.041	0.488	3.000	1001	809.77	1.097e-05	3.870e-05	4.869e-06
2.000	2.205	9.000	9.047	0.420	3.000	2001	1379.72	3.808e-06	1.343e-05	1.176e-06
4.000	4.131	2.000	2.353	0.497	3.007	101	65.30	2.335e-05	8.235e-05	1.333e-05
4.000	4.166	2.000	2.497	0.497	3.052	501	325.49	1.265e-05	4.464e-05	5.111e-06
4.000	4.182	2.000	2.539	0.497	3.075	1001	765.85	1.162e-05	4.097e-05	4.191e-06
4.000	4.205	2.000	2.572	0.497	3.101	2001	1775.20	1.113e-05	3.926e-05	3.962e-06
4.000	4.029	4.000	4.058	0.498	3.000	101	65.29	1.629e-06	5.746e-06	7.825e-07
4.000	4.063	4.000	4.117	0.500	3.000	501	335.44	1.922e-06	6.779e-06	8.298e-07
4.000	4.126	4.000	4.215	0.507	3.000	1001	744.18	1.241e-06	4.377e-06	4.532e-07
4.000	4.180	4.000	4.299	0.501	3.001	2001	1796.60	8.838e-07	3.117e-06	2.692e-07
4.000	3.973	7.000	6.988	0.486	3.000	101	72.76	2.125e-05	7.497e-05	1.017e-05
4.000	3.892	7.000	6.943	0.476	3.000	501	384.00	5.735e-06	2.023e-05	1.803e-06
4.000	3.798	7.000	6.858	0.490	3.001	1001	747.56	4.958e-07	1.749e-06	2.013e-07
4.000	3.783	7.000	6.855	0.473	3.001	2001	1655.33	5.198e-07	1.834e-06	1.861e-07
4.000	3.976	7.500	7.491	0.481	3.000	101	66.89	3.055e-05	1.077e-04	1.532e-05
4.000	3.904	7.500	7.453	0.442	3.000	501	329.82	7.784e-06	2.745e-05	2.712e-06
4.000	3.819	7.500	7.415	0.417	3.001	1001	658.00	5.769e-06	2.035e-05	1.886e-06
4.000	3.761	7.500	7.378	0.400	3.001	2001	1691.03	3.660e-06	1.291e-05	1.275e-06
4.000	3.926	9.000	8.914	0.435	3.000	101	64.96	3.262e-05	1.151e-04	1.678e-05
4.000	3.844	9.000	8.856	0.359	3.000	501	334.83	1.488e-05	5.249e-05	7.324e-06
4.000	3.802	9.000	8.848	0.328	3.000	1001	699.41	1.155e-05	4.075e-05	5.029e-06
4.000	3.746	9.000	8.840	0.304	3.001	2001	2003.20	9.673e-06	3.412e-05	3.480e-06

TABLE 8. Example 3: Method 3: Results with M_{int}=200, M = 300, N_{int} = 50, N = 70.