AIMS Mathematics, 11(11): 36904–36947.
DOI: 10.3934/math.20261426
Received: 25 August 2026
Revised: 12 October 2026
Accepted: 22 October 2026
Published: 09 November 2026

https://www.aimspress.com/journal/Math

Research article

A physics-informed singularity-regularized radial basis function extreme learning machine for Lane-Emden-Fowler equations

Neslişah İmamoğlu Karabaş*

Department of Mathematics, Izmir University of Economics, 35330 Izmir, Turkey

* **Correspondence:** Email: neslisah.karabas@ieu.edu.tr.

Abstract: A singularity-regularized radial basis function extreme learning machine (RBF-ELM) is proposed for second-order singular initial and boundary value problems of Lane-Emden-Fowler type. The trial solution is constructed so that the prescribed initial or boundary conditions hold exactly, and the collocation residual extends continuously to the origin, for every weight vector. The classical limiting form of the equation at the singular point therefore enters the least-squares problem as an ordinary collocation equation, with no penalty term and no offset parameter. The centers are fixed before training, and only the output weights are determined. The shape parameter is selected from a small candidate set by an off-node residual criterion. Linear problems reduce to a minimum-norm least-squares computation, and nonlinear problems reduce to a damped Newton iteration with an analytic Jacobian. A conditional convergence result is established for linear initial value problems under a residual-regularity hypothesis, for which a sufficient condition on the coefficient norm is given. For nonlinear problems, no convergence guarantee is available, and the results are empirical. Gaussian, multiquadric, and inverse multiquadric kernels are compared on six benchmark problems spanning linear and nonlinear, homogeneous and nonhomogeneous, initial value and boundary value formulations. Across the 18 kernel-problem combinations, the maximum error at 20 centers is of order 10^{-11} to 10^{-8}, and the limiting residual at the origin remains small at every refinement level. Under refinement, the error decreases rapidly and then levels off close to the rounding level of the computed coefficients. This plateau is consistent with a finite-precision floor set by rank truncation and rounding. The effective numerical rank never exceeds 11 of 20.

Keywords: Lane-Emden-Fowler equation; radial basis function neural network; extreme learning machine; physics-informed trial solution; singular boundary value problem; collocation method; shape parameter selection; convergence analysis
Mathematics Subject Classification: 65L05, 65L10, 65L20, 65L60, 34B16, 68T07

1. Introduction

Singular second-order ordinary differential equations of Lane-Emden-Fowler type are central to mathematical physics. They arise as the governing equations of polytropic stellar structure, isothermal gas spheres, thermionic emissions, and steady-state reaction-diffusion processes in biological media. In the general formulation by Fowler, these equations take the form

$$y''(x) + \frac{k}{x}\, y'(x) + p(x)\, q(y) = r(x), \qquad x \geq 0, \quad y(0) = a, \quad y'(0) = b, \tag{1.1}$$

where k, a, b are real constants, and p, q, r are prescribed functions. Throughout this work, attention is restricted to the physically regular subclass $b = 0$, for which $y'(0) = 0$. The classical Lane-Emden equation, recovered with $k = 2$, $p \equiv 1$, and $r \equiv 0$, was introduced independently by Lane and Emden [1, 2] to model a self-gravitating polytropic gas sphere. Extensions to general k and nonlinear q arise in the modeling of isothermal gas spheres, thermionic currents, and the diffusion of oxygen through living tissues [3–5].

The analytical difficulty of (1.1) lies in the term $(k/x)\, y'(x)$, which introduces a regular singularity at the origin. This singularity hinders the direct application of finite-difference and finite-element discretizations near $x = 0$, and it restricts classical power-series solutions to a local neighborhood of the singular point. Consequently, the development of globally accurate numerical methods has attracted sustained research attention. Classical finite-difference and collocation methods for singular boundary value problems of this kind are well established [6–8], and Chebyshev spectral methods have been developed for the Lane-Emden equation itself [9]. Solvers that evaluate the equation at the singular point use its limiting form $(1 + k)\, y''(0) + p(0)\, q(y(0)) = r(0)$ [8]. Semi-analytical approaches applied to these problems include the Adomian decomposition method of Wazwaz and its modified forms [10–12], homotopy perturbation and homotopy analysis methods [13–15], the differential transformation method [16], spectral collocation [17], Haar and Taylor wavelet methods [18, 19], and the boundary shape function method [20]. They achieve high accuracy but generally require basis construction, symbolic manipulation, or problem-specific iterative procedures that become demanding for singular nonlinear problems.

Scientific machine learning offers a mesh-free alternative. Lagaris, Likas, and Fotiadis [21] showed that a feedforward network minimizing the equation residual at collocation points can serve as an approximate solver. This principle was later formalized in the physics-informed neural network framework of Raissi, Perdikaris, and Karniadakis [22]. This framework has since been applied to a wide range of ordinary and partial differential equations, including forward and inverse heat conduction problems in multilayer media [23] and coupled Poisson-Nernst-Planck systems [24], and it has been combined with the method of fundamental solutions [25]. Deep networks of this kind rely on gradient-based optimization over all parameters, which can converge slowly and prove sensitive to initialization, particularly for singular or stiff operators. The extreme learning machine of Huang, Zhu, and Siew [26] is computationally lighter. Its hidden-layer parameters are fixed a priori, and only the output weights are trained. A physics-informed residual loss therefore reduces to a linear least-squares problem for linear equations and to a Newton-type iteration for nonlinear ones, and backpropagation is avoided entirely.

Extreme learning machines have been applied to differential equations along several lines. These include physics-informed formulations for partial differential equations [27, 28], local formulations

with domain decomposition and analyses of hyperparameter selection [29, 30], and constraint-satisfying formulations, such as the extreme theory of functional connections [31], which enforce the prescribed data through a trial expression. Sigmoidal and radial-basis networks have also been used with the boundary conditions imposed as additional collocation equations [32] and random-projection networks with the initial conditions embedded exactly [33]. Liu et al. [34] combined a radial basis architecture with extreme learning machine training for general ordinary differential equations but enforced boundary conditions through penalty terms and did not address the singular structure of Lane-Emden-Fowler operators.

A substantial body of work applies neural solvers specifically to Lane-Emden and Emden-Fowler equations, predominantly through functional-link networks with orthogonal polynomial bases. Examples include the Chebyshev neural networks [35–37] and the Legendre neural network [38] of Mall and Chakraverty, the combination of Legendre networks with improved extreme learning machine algorithms [39, 40], fractional Legendre networks [41], and multilayer perceptrons [42]. Kurmanç and Özavşar [43] directly compared Laguerre, Chebyshev, and Legendre bases. More recently, physics-informed formulations have been applied to the same problems. Zheng et al. [44] combined Pell-Lucas polynomials with a network trained by error backpropagation. Babaei et al. [45] introduced a matrix-based recurrent network that computes the derivatives in the loss through a finite-difference scheme. Both enforce the initial conditions through penalty terms and address the origin singularity numerically rather than analytically. Baty [46] trained a physics-informed neural network on an equivalent form of the equation without the singular coefficient and imposed the initial data as soft constraints. When the singular coefficient is removed by multiplying the equation by x, the multiplied equation reduces at $x = 0$ to the initial data and carries no information from the equation itself. Near the origin, the residual is also weighted by x.

Radial basis functions are a flexible alternative to polynomial bases. They require no orthogonality condition and yield closed-form derivatives of any order, both of which are attractive for collocation. The price is a problem-dependent shape parameter that balances approximation accuracy against ill-conditioning. Mai-Duy and Tran-Cong [47] demonstrated rapid convergence of multiquadric networks for smooth differential equations, and Parand and Hashemi [48] showed their advantage over polynomial collocation on nonlinear Lane-Emden problems. The underlying approximation theory is set out in [49]. In the proposed method, the centers and the shape parameter are fixed, and only the output weights are determined by the collocation equations. The linear algebra is therefore that of Kansa-type radial basis collocation. The extreme learning machine designation describes the architecture rather than the algebra, and what is proposed is a trial space rather than a training paradigm. The method differs from classical radial basis collocation in two respects. First, the prescribed data are embedded in the trial solution rather than entering as additional collocation rows, in the manner of Lagaris et al. [21]. Second, the differential equation is imposed at the singular endpoint itself rather than at a displaced node, which the first property makes possible.

The neural-network and extreme learning machine solvers surveyed above leave the following aspects of these problems unaddressed. First, in these solvers, the singular term is handled numerically. The equation is collocated away from the origin, rewritten without the singular coefficient, or left to the training procedure. The behavior of the residual at $x = 0$ is not analyzed. The limiting relation at the singular point, which is standard in the classical solvers cited above, has not been built into the trial space of such a solver. The trial construction used here makes the residual continuously extendible to

the origin for every weight vector, so that its limiting value can be included as an ordinary collocation equation, one row of an unconstrained least-squares problem. Second, the Gaussian, multiquadric, and inverse multiquadric kernels have not been compared systematically on this equation class, nor has the dependence of accuracy on the shape parameter been examined together with the conditioning of the resulting collocation matrices. Third, exact satisfaction of both initial and Robin-type boundary conditions within a single trial-solution framework, without penalty terms, has not been demonstrated for these problems in this class of solvers.

This paper introduces the RBF-ELM method, a radial basis function network trained as an extreme learning machine, in which the approximation space itself regularizes the singularity. The trial solution, of the form introduced by Lagaris et al. [21], satisfies the prescribed initial or Robin conditions exactly for every realization of the weight vector. For Lane-Emden-Fowler operators, it has a further property, shown in Section 4. The collocation residual extends continuously to the origin for every weight vector, and its value there is the classical limiting form of the equation. Two consequences follow. The training problem is unconstrained, since no penalty term enters the loss. The differential equation is imposed at the singular endpoint as an ordinary collocation equation, with no offset parameter. The novelty therefore lies not in the limiting relation, which is classical, but in realizing it through the trial space of a neural-network solver. The construction is established rigorously and is accompanied by a conditional convergence result for linear initial value problems. For the nonlinear problems and the Robin problem, no convergence result is claimed, and the numerical results are empirical. Three radial basis kernels are compared across six benchmark problems spanning linear, nonlinear, homogeneous, nonhomogeneous, initial value, and boundary value formulations. The conditioning and effective rank of the resulting collocation matrices are examined. The numerical study also quantifies the finite-precision error floor, relating it to the retained rank and to the coefficient norm, and compares the method with alternative formulations and solver choices.

The paper is organized as follows: Section 2 describes the network architecture and the three kernel families. Section 3 constructs the trial solution, derives the training procedure for the linear and nonlinear cases, and presents the shape parameter selection. Section 4 establishes the regularization of the singularity and develops the convergence analysis for the linear case. Section 5 reports the numerical results, including the conditioning and effective rank of the collocation matrices, and Section 6 draws conclusions.

2. Radial basis function neural network

The approximation architecture used throughout is a single-layer radial basis function neural network with one input node, a hidden layer of M radial units, and one linear output node. Given M fixed centers $\{c_j\}_{j=1}^{M} \subset [0, 1]$ and a shape parameter $\varepsilon > 0$, the network output at $x \in [0, 1]$ is

$$N(x, \mathbf{w}) = \sum_{j=1}^{M} w_j \, \phi(\varepsilon, \, x - c_j), \tag{2.1}$$

where $\mathbf{w} = (w_1, \ldots, w_M)^\top$ is the weight vector, and $\phi\colon \mathbb{R}_{>0} \times \mathbb{R} \to \mathbb{R}$ is the kernel function. The j-th hidden unit evaluates ϕ at the signed displacement $x - c_j$, and the hidden-layer outputs are combined at the linear output node through the trainable weights $w_1, \ldots, w_M$. No nonlinear activation is applied at

the output node, so N is linear in $\mathbf{w}$ for fixed centers and shape parameter. Consequently, the residual system derived in Section 3 is linear whenever the governing ordinary differential equation is linear, and has a closed-form Jacobian in the nonlinear case. Figure 1 shows the architecture together with the trial solution and the training of the output weights, which are constructed in Section 3.

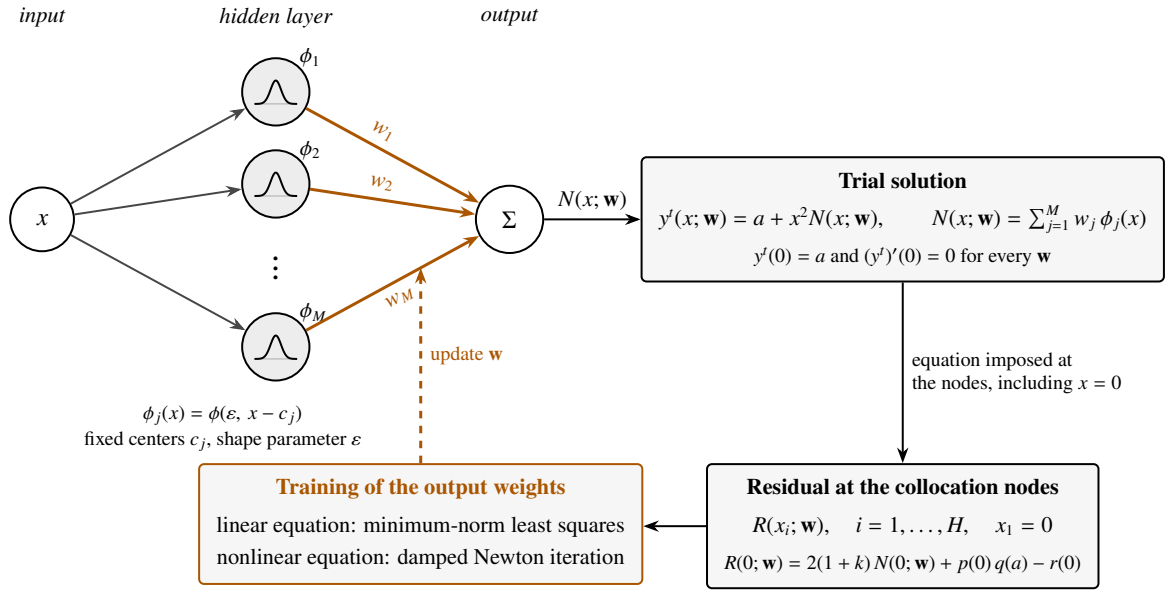


Figure 1. Structure of the radial basis function neural network.

Kernel functions and their analytical derivatives

Let $\zeta = x - c_j$ denote the signed displacement between the evaluation point and the j-th center. Since ζ depends on x with unit derivative, differentiation with respect to x coincides with differentiation with respect to ζ. The derivatives required to assemble the residual (3.5) and the Jacobian (3.11) are therefore available in closed form. Three kernel families are examined, all infinitely smooth, but with different growth and decay behavior, and hence with different approximation and conditioning characteristics.

The Gaussian (GA) kernel and its derivatives are

$$\phi_{\mathrm{GA}}(\zeta) = \exp\left(-\varepsilon^2\zeta^2\right), \tag{2.2}$$

$$\phi'_{\mathrm{GA}}(\zeta) = -2\varepsilon^2\zeta\exp\left(-\varepsilon^2\zeta^2\right), \qquad \phi''_{\mathrm{GA}}(\zeta) = \left(4\varepsilon^4\zeta^2 - 2\varepsilon^2\right)\exp\left(-\varepsilon^2\zeta^2\right). \tag{2.3}$$

It is strictly positive definite on $\mathbb{R}^n$ for every $\varepsilon > 0$ and every n, belongs to the Schwartz class, and is infinitely smooth, with spectral approximation properties for analytic functions. The condition number of the interpolation matrix it generates grows super-algebraically as $\varepsilon \to 0$.

The multiquadric (MQ) kernel of Hardy [50] and its derivatives are

$$\phi_{\mathrm{MQ}}(\zeta) = \sqrt{1 + \varepsilon^2\zeta^2}, \tag{2.4}$$

$$\phi'_{\mathrm{MQ}}(\zeta) = \frac{\varepsilon^2\zeta}{\sqrt{1 + \varepsilon^2\zeta^2}}, \qquad \phi''_{\mathrm{MQ}}(\zeta) = \frac{\varepsilon^2}{(1 + \varepsilon^2\zeta^2)^{3/2}}. \tag{2.5}$$

The kernel itself is conditionally negative definite of order one, that is, its negative is conditionally positive definite of order one. Micchelli [51] proved that the interpolation matrix it generates with M distinct centers is non-singular for all $\varepsilon > 0$. Since ϕ_{MQ} grows algebraically as $|\zeta| \to \infty$, the basis functions are global in character.

The inverse multiquadric (IMQ) kernel and its derivatives are

$$\phi_{\mathrm{IMQ}}(\zeta) = \frac{1}{\sqrt{1+\varepsilon^2\zeta^2}}, \tag{2.6}$$

$$\phi'_{\mathrm{IMQ}}(\zeta) = -\frac{\varepsilon^2\zeta}{(1+\varepsilon^2\zeta^2)^{3/2}}, \qquad \phi''_{\mathrm{IMQ}}(\zeta) = \frac{\varepsilon^2\left(2\varepsilon^2\zeta^2-1\right)}{(1+\varepsilon^2\zeta^2)^{5/2}}. \tag{2.7}$$

It is strictly positive definite on $\mathbb{R}^n$ for all $\varepsilon > 0$ and $n \geq 1$. Unlike the MQ kernel, it decays to zero as $|\zeta| \to \infty$, which gives the basis functions a quasi-local envelope.

3. A neural network method for solving differential equations

This section presents the neural network formulation for second-order ordinary differential equations. Following the general framework of Lagaris, Likas, and Fotiadis [21], the solution strategy rests on three components. These are: a trial solution that satisfies the prescribed initial or boundary conditions exactly for every choice of weights, a physics-informed loss function defined purely by the ODE residual at the collocation nodes in $[0, 1]$, and an algebraic or iterative scheme that reduces the residual loss.

3.1. Construction of the trial solution

Consider a second-order initial value problem of the form

$$y''(x) = f(x, y, y'), \qquad x \in [0,1], \quad y(0) = a, \quad y'(0) = b, \tag{3.1}$$

where f is a prescribed function of its arguments. Define the affine correction $A(x) = a + bx$, which satisfies $A(0) = a$ and $A'(0) = b$. The trial solution is constructed as

$$y^t(x;\mathbf{w}) = A(x) + x^2\,N(x;\mathbf{w}) = a + bx + x^2\,N(x;\mathbf{w}), \tag{3.2}$$

where $N(x;\mathbf{w})$ is the network output defined in (2.1). The factor x^2 ensures that both the network contribution and its first derivative vanish at the origin, so that $y^t(0;\mathbf{w}) = a$ and $(y^t)'(0;\mathbf{w}) = b$ hold for every $\mathbf{w} \in \mathbb{R}^M$. Differentiating (3.2) with respect to x yields

$$(y^t)'(x;\mathbf{w}) = b + 2x\,N(x;\mathbf{w}) + x^2\,N'(x;\mathbf{w}), \tag{3.3}$$

$$(y^t)''(x;\mathbf{w}) = 2\,N(x;\mathbf{w}) + 4x\,N'(x;\mathbf{w}) + x^2\,N''(x;\mathbf{w}), \tag{3.4}$$

where the network derivatives N' and N'' are computed analytically by differentiating (2.1) term by term with the kernel derivative formulas (2.3), (2.5), and (2.7).

3.2. Physics-informed loss function

Let $\{x_i\}_{i=1}^{H}$ be a set of H equidistant collocation nodes in $[0, 1]$, with $x_1 = 0$. For each collocation node with $x_i > 0$, the ODE residual is

$$R_i(\mathbf{w}) = (y^t)''(x_i; \mathbf{w}) + \frac{k}{x_i}(y^t)'(x_i; \mathbf{w}) + p(x_i)\, q(y^t(x_i; \mathbf{w})) - r(x_i). \tag{3.5}$$

At the singular endpoint $x_1 = 0$, the residual is defined by its limiting form (4.1), so that (3.5) and (4.1) together specify $R_i(\mathbf{w})$ for every node. The physics-informed loss function to be minimized over $\mathbf{w}$ is then

$$E(\mathbf{w}) = \frac{1}{2}\left\|\mathbf{R}(\mathbf{w})\right\|^2 = \frac{1}{2}\sum_{i=1}^{H} R_i(\mathbf{w})^2. \tag{3.6}$$

Since the prescribed data are embedded in the trial solution (3.2) and do not appear in (3.6), the minimization problem is unconstrained with respect to $\mathbf{w}$. A weight vector $\hat{\mathbf{w}}$ satisfying $\mathbf{R}(\hat{\mathbf{w}}) = \mathbf{0}$ is a global minimizer of (3.6) with $E(\hat{\mathbf{w}}) = 0$.

3.3. Extreme learning machine training for linear equations

When $q(y)$ is linear in y, the residual $R_i(\mathbf{w})$ is an affine function of $\mathbf{w}$, and the condition $\mathbf{R}(\mathbf{w}) = \mathbf{0}$ reduces to the linear algebraic system

$$\mathbf{A}\,\mathbf{w} = \mathbf{d}, \tag{3.7}$$

where the design matrix $\mathbf{A} \in \mathbb{R}^{H\times M}$ and the right-hand side vector $\mathbf{d} \in \mathbb{R}^H$ are assembled by substituting the trial solution (3.2) and its derivatives (3.3) and (3.4) into the residual (3.5). The row corresponding to $x = 0$ is given by (4.1) and (4.2). The minimum-norm least-squares solution

$$\mathbf{w}^\star = \mathbf{A}^+\,\mathbf{d}, \tag{3.8}$$

where $\mathbf{A}^+$ denotes the Moore-Penrose pseudoinverse of $\mathbf{A}$, is the unique minimum-norm minimizer of (3.6) in the linear case. In practice, it is computed by a rank-revealing minimum-norm least-squares solver based on complete orthogonal factorization (MATLAB's lsqminnorm) rather than by forming $\mathbf{A}^+$ explicitly. The solver forms a column-pivoted QR factorization of $\mathbf{A}$, retains the r columns whose diagonal entries in the triangular factor exceed a threshold θ in absolute value, and returns the minimum-norm least-squares solution of the resulting rank-r problem. By default, θ is computed from the QR factor; its numerical value for the present matrices is given in Section 5. In the sensitivity study, the threshold is set to $\theta = \tau_{\text{rel}}\,\sigma_{\max}(\mathbf{A})$, where $\sigma_{\max}(\mathbf{A})$ is the largest singular value of $\mathbf{A}$ and τ_{rel} a relative tolerance. Since the computed solution depends on θ, the threshold is part of the numerical method rather than an implementation detail. When $r < M$, the computed vector minimizes the residual only over the retained subspace and is therefore not the minimizer (3.8) of the full problem. This computation constitutes the extreme learning machine training step. The centers $\{c_j\}$ and the shape parameter ε are fixed a priori, so no iterative optimization over the hidden-layer parameters is performed.

In exact arithmetic, (3.8) is the solution, and no further correction is possible, since

$$\mathbf{A}^+(\mathbf{A}\mathbf{A}^+\mathbf{d} - \mathbf{d}) = \mathbf{0}$$

by the Penrose identity $\mathbf{A}^{+}\mathbf{A}\mathbf{A}^{+} = \mathbf{A}^{+}$. In floating-point arithmetic, the collocation matrix may be numerically rank deficient, the rank-revealing solver does not return the exact pseudoinverse, and a single application of (3.8) may leave a residual well above the level attainable in double precision. The least-squares computation is therefore followed by a small number of correction steps. Setting $\mathbf{w}^{(0)} = \mathbf{0}$ and iterating

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \mathbf{A}^{+}\,\mathbf{R}(\mathbf{w}^{(t)}), \qquad t = 0, 1, 2, \ldots, \tag{3.9}$$

reproduces (3.8) at the first step, since $\mathbf{R}(\mathbf{0}) = -\mathbf{d}$. The effect of the subsequent steps is as follows: Let $\widehat{\mathbf{A}}^{+}$ denote the pseudoinverse actually applied by the rank-revealing solver and let $\mathbf{P} = \mathbf{A}^{+}\mathbf{A}$ be the projector formed from the exact rank-truncated pseudoinverse. Because the residual is affine, (3.9) satisfies

$$\mathbf{w}^{(t+1)} - \mathbf{w}^{(\infty)} = (\mathbf{I} - \widehat{\mathbf{A}}^{+}\mathbf{A})(\mathbf{w}^{(t)} - \mathbf{w}^{(\infty)}).$$

Were $\widehat{\mathbf{A}}^{+}$ exact, $\mathbf{I} - \mathbf{P}$ would be the orthogonal projector onto the discarded right-singular subspace and the iteration would terminate after a single step. In floating point, $\widehat{\mathbf{A}}^{+}\mathbf{A}$ differs from $\mathbf{P}$, and the later steps act as iterative refinement with an inexact solver. They reduce the residual component in the retained subspace, which is contaminated by rounding, but not the component in the discarded subspace, which is annihilated by $\widehat{\mathbf{A}}^{+}$. The correction therefore removes rounding error but not rank-truncation error, and it does not alter the exact minimum-norm solution. The procedure is not iterative refinement in the classical sense, which requires the residual to be evaluated in higher precision. The iteration stops as soon as $\|\mathbf{R}\|_{\infty}$ falls below the prescribed tolerance or ceases to decrease. A step that does not decrease $\|\mathbf{R}\|_{\infty}$ is rejected, and the previous vector is kept. At most $T_{\max}$ steps are taken. The correction is inexpensive because $\mathbf{A}$ is assembled once and only a few steps are needed; the number of steps required for the linear benchmark is reported in Section 5. The vector returned by the last accepted step is denoted $\mathbf{w}_c$; for nonlinear problems, it is the last accepted Newton iterate of Section 3.4. It is the vector used in all tables and figures, and r denotes the rank used in computing it.

Since $\mathbf{A}$ is the Jacobian of an affine residual, (3.9) is the Newton iteration (3.10) of the next subsection, taken with unit step length. The linear case is therefore the special case of the general scheme in which the Jacobian is constant and no line search is required.

3.4. Damped Newton iteration for nonlinear equations

When $q(y)$ is nonlinear, the residual $\mathbf{R}(\mathbf{w})$ is a nonlinear function of $\mathbf{w}$, and the linear system (3.7) must be replaced by the nonlinear system $\mathbf{R}(\mathbf{w}) = \mathbf{0}$. This system is solved by the damped Newton iteration

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \lambda_t[\mathbf{J}(\mathbf{w}^{(t)})]^{+}\mathbf{R}(\mathbf{w}^{(t)}), \qquad t = 0, 1, 2, \ldots, \tag{3.10}$$

where $\lambda_t \in (0, 1]$ is a step length determined by Armijo backtracking on the least-squares merit function, and $\mathbf{J}(\mathbf{w}) \in \mathbb{R}^{H\times M}$ is the exact Jacobian matrix of the residual vector. The (i, j)-entry of $\mathbf{J}$ is obtained by differentiating (3.5) with respect to w_j and applying the analytical kernel derivatives,

$$J_{ij}(\mathbf{w}) = \frac{\partial (y^t)''}{\partial w_j}(x_i) + \frac{k}{x_i}\,\frac{\partial (y^t)'}{\partial w_j}(x_i) + p(x_i)\,q'\big(y^t(x_i)\big)\,\frac{\partial y^t}{\partial w_j}(x_i), \tag{3.11}$$

where the partial derivatives with respect to w_j of the trial solution and of its first two derivatives follow directly from (3.2)–(3.4),

$$\begin{aligned}\frac{\partial y^t}{\partial w_j}(x) &= x^2\phi_j(x),\\ \frac{\partial (y^t)'}{\partial w_j}(x) &= 2x\phi_j(x) + x^2\phi_j'(x),\\ \frac{\partial (y^t)''}{\partial w_j}(x) &= 2\phi_j(x) + 4x\phi_j'(x) + x^2\phi_j''(x).\end{aligned} \tag{3.12}$$

Substituting (3.12) into (3.11) yields the Jacobian in closed form, so no finite-difference approximation of $\mathbf{J}$ is required. The row corresponding to the singular endpoint is given by (4.2).

The iteration (3.10) is initialized at $\mathbf{w}^{(0)} = \mathbf{0}$. At each step, the search direction is the Newton direction

$$\mathbf{p}^{(t)} = -[\mathbf{J}(\mathbf{w}^{(t)})]^{+}\,\mathbf{R}(\mathbf{w}^{(t)}),$$

where the pseudoinverse is applied through the same rank-revealing minimum-norm least-squares solver as in Section 3.3. The step length λ_t is then chosen by Armijo backtracking on

$$f(\mathbf{w}) = \frac{1}{2}\|\mathbf{R}(\mathbf{w})\|_2^2.$$

Beginning from $\lambda_t = 1$, the step length is halved until

$$\begin{aligned} f(\mathbf{w}^{(t)} + \lambda_t\mathbf{p}^{(t)}) &\le f(\mathbf{w}^{(t)}) + c_1\lambda_t\nabla f(\mathbf{w}^{(t)})^{\mathsf{T}}\mathbf{p}^{(t)},\\ \nabla f(\mathbf{w})^{\mathsf{T}}\mathbf{p} &= \mathbf{R}(\mathbf{w})^{\mathsf{T}}\mathbf{J}(\mathbf{w})\mathbf{p},\end{aligned} \tag{3.13}$$

for a fixed constant $c_1 \in (0, 1)$. The test is imposed on the same function for which descent can be shown. For the exact rank-truncated pseudoinverse, with $\mathbf{P_J} = \mathbf{J}\mathbf{J}^{+}$ the orthogonal projector onto the retained column space of $\mathbf{J}$,

$$\nabla f(\mathbf{w})^{\mathsf{T}}\mathbf{p}^{(t)} = -\|\mathbf{P_J}\mathbf{R}\|_2^2 \le 0.$$

Hence, $\mathbf{p}^{(t)}$ is a descent direction for f whenever $\mathbf{P_J}\mathbf{R} \neq \mathbf{0}$, and (3.13) can then be satisfied for small λ_t. Equality holds when $\mathbf{R}$ is orthogonal to the retained column space. This is stationarity with respect to the retained directions only. The gradient $\nabla f = \mathbf{J}^{\mathsf{T}}\mathbf{R}$ may still have components along the discarded directions, so a vanishing truncated direction does not imply that $\mathbf{w}^{(t)}$ is a stationary point of f. In floating point, and with the QR-based truncation of Section 3.3, the computed slope may fail to be negative once the residual is at the level set by rounding; the iteration then stops without a line search, as described below. The stopping tolerance is measured in $\|\mathbf{R}\|_\infty$, the largest nodal residual.

Where the reaction term is defined only for positive arguments, as for a logarithmic nonlinearity, a trial step is rejected before the residual is evaluated whenever $\min_{x\in X_{\mathrm{pos}}} y^t(x;\mathbf{w}) \le y_{\min}$, a small positive floor, where

$$X_{\mathrm{pos}} = X_{\mathrm{col}} \cup X_{\mathrm{val}} \cup X_{\mathrm{ev}},$$

and X_{ev} is the grid on which accuracy is reported. Positivity is thus verified on this finite set, not on the whole interval $[0, 1]$. The test depends only on the trial solution and uses no reference values, so

the reported accuracy remains independent of the exact solution. Rejection is preferable to clipping the argument of the nonlinearity, which would replace the residual by a piecewise-modified function whose derivative is no longer (3.11). With rejection, the differential equation is never modified, and the Jacobian remains the exact derivative of the residual.

The iteration is said to have converged when $\|\mathbf{R}(\mathbf{w}^{(t)})\|_\infty$ falls below the prescribed tolerance. It is said to have stagnated when the computed slope $\nabla f(\mathbf{w}^{(t)})^\top \mathbf{p}^{(t)}$ is not negative, or when no step length down to a prescribed minimum $\lambda_{\min}$ satisfies (3.13); the last accepted iterate is then returned. Otherwise, it stops after a prescribed maximum number of iterations. Stagnation indicates either a nonnegative computed slope or failure of the backtracking search to find an acceptable step; it does not imply that no further decrease of f is possible. It occurs, for example, when the residual has reached the floor set by rank truncation and rounding. The terminal states of the computations are reported in Section 5.

3.5. Shape parameter selection

The shape parameter ε governs both the approximation power of the network and the numerical conditioning of the design matrix $\mathbf{A}$. As $\varepsilon \to 0$, all kernel functions tend to a common constant, and the columns of $\mathbf{A}$ become nearly linearly dependent. As $\varepsilon \to \infty$, the Gaussian and inverse multiquadric kernels localize around their centers, whereas the multiquadric kernel approaches $\varepsilon|\zeta|$ away from its center. Its second derivative (2.5) is then concentrated within a distance of order $1/\varepsilon$ of the center, and the network output N becomes nearly piecewise linear between the centers. In both cases, the kernels remain smooth for every finite ε, but the quality of the approximation between the nodes deteriorates. This trade-off [49] implies an optimal intermediate range of ε. Here, the shape parameter is selected from a finite candidate set $\mathcal{E}$ by minimizing $\|\mathbf{R}\|_\infty$ on the off-node validation grid $\{\xi_\ell\}$, obtained from the L staggered points

$$\frac{\ell - 1/2}{L}, \qquad \ell = 1, \ldots, L$$

by removing every point that coincides with a collocation node

$$x_i = (i - 1)/(H - 1).$$

The criterion depends only on the differential equation residual; the exact or reference solution is used solely to report errors after training, so the reported accuracy is not circular.

The complete RBF-ELM procedure, comprising the shape-parameter selection, the linear least-squares computation with correction, and the nonlinear damped Newton iteration, is summarized in Algorithm 1.

Algorithm 1 Singularity-regularized RBF-ELM for Lane-Emden-Fowler equations

Require: kernel ϕ; centers $c_j = (j-1)/(M-1)$, $j = 1, \dots, M$; collocation nodes $x_i = (i-1)/(H-1)$, $i = 1, \dots, H$, so that $x_1 = 0$; candidate set $\mathcal{E}$; off-node validation grid $\{\xi_\ell\}_{\ell=1}^{L}$; data a (or the Robin coefficients), $b = 0$; tolerance tol; maximum iterations $I_{\max}$ (Newton) and $T_{\max}$ (residual correction); minimum step $\lambda_{\min}$; Armijo constant c_1; positivity floor $y_{\min}$ (or $-\infty$ if unconstrained) enforced on X_{pos}; rank threshold θ of the least-squares solver (default of lsqminnorm, Section 3.3).

Ensure: $y^t(x) = a + x^2 N(x; \mathbf{w}_c)$, with $a = a(\mathbf{w}_c)$ for a Robin problem.

function RESIDUAL($\mathbf{w}, \varepsilon, \{x_i\}$)
  if Robin problem **then**
    $a \leftarrow a(\mathbf{w})$ from (5.14) ▷ Robin condition holds for every $\mathbf{w}$
  end if
  $R_1 \leftarrow 2(1+k)N(0;\mathbf{w}) + p(0)q(a) - r(0)$ ▷ limiting form (4.1) at $x_1 = 0$
  $R_i \leftarrow \mathcal{L}y^t(x_i;\mathbf{w}) - r(x_i)$ for $i = 2, \dots, H$ ▷ (3.5), valid for $x_i > 0$
  $\mathbf{J} \leftarrow$ (3.11) with row 1 given by (4.2)
  if Robin problem **then**
    $J_{ij} \leftarrow J_{ij} + p(x_i)\, q'(y^t(x_i;\mathbf{w}))\, \partial a/\partial w_j$ for $i = 2, \dots, H$ ▷ from (5.15); row 1 is already (4.2)
  end if
  return $\mathbf{R}$, $\mathbf{J}$
end function
for each $\varepsilon \in \mathcal{E}$ **do**
  if the equation is linear **then**
    $\mathbf{w} \leftarrow \mathbf{0}$; $(\mathbf{R}, \mathbf{A}) \leftarrow$ RESIDUAL($\mathbf{w}, \varepsilon, \{x_i\}$); status $\leftarrow$ *limit*
    for $t = 1, 2, \dots, T_{\max}$ **do**
      $\mathbf{w}_{\text{try}} \leftarrow \mathbf{w} - \mathbf{A}^{+}\mathbf{R}$; $\mathbf{R}_{\text{try}} \leftarrow \mathbf{R}(\mathbf{w}_{\text{try}})$ ▷ $t = 1$ gives (3.8); $t > 1$ is finite-precision residual correction
      if $\|\mathbf{R}_{\text{try}}\|_\infty \geq \|\mathbf{R}\|_\infty$ **then** status $\leftarrow$ *stagnated*; **break** ▷ step rejected, $\mathbf{w}$ kept
      $(\mathbf{w}, \mathbf{R}) \leftarrow (\mathbf{w}_{\text{try}}, \mathbf{R}_{\text{try}})$; **if** $\|\mathbf{R}\|_\infty <$ tol **then** status $\leftarrow$ *converged*; **break**
    end for
  else
    $\mathbf{w} \leftarrow \mathbf{0}$; status $\leftarrow$ *limit*
    for $t = 0, 1, \dots, I_{\max} - 1$ **do**
      $(\mathbf{R}, \mathbf{J}) \leftarrow$ RESIDUAL($\mathbf{w}, \varepsilon, \{x_i\}$)
      if $\|\mathbf{R}\|_\infty <$ tol **then** status $\leftarrow$ *converged*; **break**
      $\mathbf{p} \leftarrow -\mathbf{J}^{+}\mathbf{R}$
      $s \leftarrow \mathbf{R}^{\top}\mathbf{J}\mathbf{p}$; **if** $s \geq 0$ **then** status $\leftarrow$ *stagnated*; **break** ▷ no descent in floating point
      $\lambda \leftarrow 1$; accepted $\leftarrow$ FALSE
      while $\lambda \geq \lambda_{\min}$ **do**
        $\mathbf{w}^{+} \leftarrow \mathbf{w} + \lambda\mathbf{p}$
        if $\min_{x \in X_{\text{pos}}} y^t(x;\mathbf{w}^{+}) > y_{\min}$ **and** $\frac{1}{2}\|\mathbf{R}(\mathbf{w}^{+})\|_2^2 \leq \frac{1}{2}\|\mathbf{R}(\mathbf{w})\|_2^2 + c_1\lambda s$ **then**
          $\mathbf{w} \leftarrow \mathbf{w}^{+}$; accepted $\leftarrow$ TRUE; **break**
        end if
        $\lambda \leftarrow \lambda/2$ ▷ rejects both non-decreasing and non-positive steps
      end while
      if $\neg$ accepted **then** status $\leftarrow$ *stagnated*; **break**
    end for
  end if
  if status $\in \{$*converged*, *stagnated*$\}$ **and** $\mathbf{R}$ finite **then** $\gamma(\varepsilon) \leftarrow \max_\ell |R(\xi_\ell;\mathbf{w})|$ **else** $\gamma(\varepsilon) \leftarrow \infty$ ▷ failed candidates excluded
end for
$\varepsilon^{\star} \leftarrow \arg\min_{\varepsilon \in \mathcal{E}} \gamma(\varepsilon)$; $\mathbf{w}_c \leftarrow \mathbf{w}_{\varepsilon^{\star}}$
return y^t, together with $\varepsilon^{\star}$, the terminal status, and the iteration and step-halving counts

4. Theoretical analysis

This section establishes two analytical properties of the method. The first, in Section 4.1, is that the trial-solution construction makes the residual continuous at the origin, so that the differential equation can be imposed there directly. The second, in Section 4.2, is that the collocation error vanishes as the fill distance tends to zero, under explicit assumptions and for linear initial value problems. No convergence

result is established for the nonlinear problems or for the Robin problem, and the corresponding results of Section 5 are empirical. The obstacles are stated at the end of Section 4.2. The numerical rank deficiency of the resulting collocation matrices is examined empirically in Section 5.

4.1. Regularization of the singularity

Throughout this section, the regularity condition $y'(0) = 0$ is assumed, so that $b = 0$ and the affine correction reduces to $A(x) = a$. This condition holds for the regular radially symmetric problems considered here. For $k > 0$, the restriction is necessary rather than convenient, since the term $(k/x)(y^t)'$ contains kb/x, which is unbounded at the origin unless $b = 0$. The trial solution then takes the form $y^t(x; \mathbf{w}) = a + x^2 N(x; \mathbf{w})$. Its defining feature is that it makes the residual continuous at the origin despite the regular singularity of the operator. The following proposition makes this precise.

Proposition 4.1. *Let the trial solution be $y^t(x; \mathbf{w}) = a + x^2 N(x; \mathbf{w})$, that is,* (3.2) *with $b = 0$, with*

$$N(x; \mathbf{w}) = \sum_{j=1}^{M} w_j \, \phi(\varepsilon, x - c_j),$$

and suppose the kernel ϕ is twice continuously differentiable on $[-1, 1]$*, as holds for the Gaussian, multiquadric, and inverse multiquadric kernels. Then, for every $\mathbf{w} \in \mathbb{R}^M$ and every $k \geq 0$, the map*

$$x \mapsto \frac{k}{x} (y^t)'(x; \mathbf{w})$$

extends to a continuous function on $[0, 1]$*, with limiting value $2k\,N(0; \mathbf{w})$ at $x = 0$. Consequently, if p, q, r are continuous, the residual*

$$R(x; \mathbf{w}) = (y^t)''(x; \mathbf{w}) + \frac{k}{x} (y^t)'(x; \mathbf{w}) + p(x)\, q(y^t(x; \mathbf{w})) - r(x)$$

is continuous on the closed interval $[0, 1]$*; in particular, it is bounded at the origin, so that the regular singularity of the operator is regularized by construction.*

Proof. Since $x - c_j \in [-1, 1]$ for $x, c_j \in [0, 1]$ and $\phi \in C^2[-1, 1]$, the network output $N(\cdot; \mathbf{w}) \in C^2[0, 1]$. Differentiating $y^t = a + x^2 N$ gives

$$(y^t)'(x) = 2xN(x) + x^2 N'(x) = x(2N(x) + xN'(x)),$$

so that for $x \in (0, 1]$,

$$\frac{k}{x} (y^t)'(x) = k(2N(x) + xN'(x)).$$

The right-hand side is continuous on all of $[0, 1]$ and equals $2k\,N(0)$ at $x = 0$; hence, $\frac{k}{x}(y^t)'$ extends continuously to $[0, 1]$. Moreover,

$$(y^t)''(x) = 2N(x) + 4xN'(x) + x^2 N''(x)$$

is continuous on $[0, 1]$. Adding the continuous terms $p\, q(y^t) - r$ shows that R is continuous on $[0, 1]$. □

For $k = 0$, the operator has no singular term, and the proposition is immediate. The limiting form (4.1) below then reduces to the ordinary residual at $x = 0$, and the construction coincides with standard collocation at the endpoint; the case of interest is $k > 0$.

A direct consequence is that, for $b = 0$, the residual admits an explicit closed form at the singular endpoint. Letting $x \to 0$ in $R(x; \mathbf{w})$ and using $(y^t)''(0; \mathbf{w}) = 2N(0; \mathbf{w})$ together with the limiting value $2k\,N(0; \mathbf{w})$ established above gives

$$R(0; \mathbf{w}) = 2(1 + k)\,N(0; \mathbf{w}) + p(0)\,q(a) - r(0), \tag{4.1}$$

with

$$\frac{\partial R(0; \mathbf{w})}{\partial w_j} = 2(1 + k)\,\phi_j(0) + p(0)\,q'(a)\,\frac{\partial a}{\partial w_j}. \tag{4.2}$$

For an initial value problem, a is prescribed, so $\partial a/\partial w_j = 0$ and the row (4.2) is assembled once. Where $a = a(\mathbf{w})$ is eliminated from a Robin condition, the second term is present, and (4.1) is nonlinear in $\mathbf{w}$. Equation (4.1) is imposed as the first collocation equation throughout, so that the differential equation is enforced at the singular endpoint itself rather than at a nearby interior node. Setting $R(0; \mathbf{w}) = 0$ in (4.1) gives the classical limiting relation $(1+k)y''(0) + p(0)q(a) = r(0)$, which every regular solution satisfies. Solvers that evaluate the equation at the singular point use this relation there [8]. For neural-network and extreme learning machine solvers, Proposition 4.1 adds that in the trial space $a + x^2 V_{\varepsilon,h}$, this relation is the exact residual for every weight vector. It therefore enters the unconstrained least-squares problem as an ordinary collocation equation.

As an illustration, consider the linear equation

$$y'' + \frac{k}{x}y' + \beta(x)\,y = r(x)$$

with $y(0) = a$ and $y'(0) = 0$, so that $p = \beta$ and $q(y) = y$. The trial solution is $y^t = a + x^2 N$, and substituting (3.3) and (3.4) gives, for $x > 0$,

$$R(x; \mathbf{w}) = x^2 N''(x; \mathbf{w}) + (4 + k)\,x\,N'(x; \mathbf{w}) + \big(2 + 2k + \beta(x)\,x^2\big)N(x; \mathbf{w}) + \beta(x)\,a - r(x).$$

The factor $1/x$ has been canceled, so the right-hand side is defined at $x = 0$, where it equals

$$2(1 + k)N(0; \mathbf{w}) + \beta(0)\,a - r(0),$$

in agreement with (4.1). Since R is affine in $\mathbf{w}$, the collocation equations form the system (3.7) with

$$A_{ij} = x_i^2\phi_j''(x_i) + (4 + k)\,x_i\phi_j'(x_i) + \big(2 + 2k + \beta(x_i)\,x_i^2\big)\phi_j(x_i)$$

and $d_i = r(x_i) - \beta(x_i)\,a$, where $\phi_j = \phi(\varepsilon, \cdot - c_j)$. The first row, at $x_1 = 0$, is simply $A_{1j} = 2(1 + k)\phi_j(0)$. The initial conditions do not appear in the system, because $y^t(0) = a$ and $(y^t)'(0) = 0$ hold for every $\mathbf{w}$.

4.2. Convergence of the RBF-ELM collocation scheme

This section presents a conditional convergence result for the proposed RBF-ELM collocation scheme in the linear case, for the initial value problem. The regularity condition $y'(0) = 0$ is still assumed. Consider the linear Lane-Emden-Fowler operator

$$\mathcal{L}y = y'' + \frac{k}{x}\,y' + \beta(x)\,y, \qquad x \in (0, 1],$$

where $\beta \in C[0,1]$, together with the model problem

$$\mathcal{L}y = r, \qquad y(0) = a, \qquad y'(0) = 0.$$

Let y^* denote the exact solution. The RBF-ELM trial approximation is

$$y^t(x;\mathbf{w}) = a + x^2 N(x;\mathbf{w}), \qquad N(x;\mathbf{w}) = \sum_{j=1}^{M} w_j\, \phi(\varepsilon, x - c_j),$$

where the shape parameter $\varepsilon > 0$ is fixed, the centers $C = \{c_j\}_{j=1}^{M}$ and the collocation nodes $X = \{x_i\}_{i=1}^{H}$ lie in $[0,1]$, and $R(x;\mathbf{w}) = \mathcal{L}y^t(x;\mathbf{w}) - r(x)$ is the residual. Both sets are assumed to form quasi-uniform families under refinement, with $M = H$ as in the implemented scheme. The single fill distance

$$h = \sup_{x\in[0,1]} \min_{j} |x - c_j|$$

then describes both sets up to a fixed constant, and refinement means $h \to 0$.

The argument has three steps. Stability bounds the error in terms of the residual, so that bounding the error reduces to bounding the residual. The residual is small at the collocation nodes because the trial space can reproduce the reduced solution there. A sampling inequality then carries that smallness from the nodes to the whole interval, which requires the residual not to oscillate too rapidly in between. The three hypotheses below are what these steps need, one for each.

Assumption A1. There is a constant $C_s > 0$, independent of ρ, such that for every $\rho \in C[0,1]$, the solution z of $\mathcal{L}z = \rho$, $z(0) = z'(0) = 0$, satisfies

$$\|z\|_{L^\infty(0,1)} \le C_s\, \|\rho\|_{L^\infty(0,1)}.$$

In other words, a solution that satisfies the homogeneous initial conditions can be no larger than a fixed multiple of the corresponding right-hand side.

Proposition 4.2. *Let $k \ge 0$ and $\beta \in C[0,1]$ with $\|\beta\|_\infty < 2(k+1)$, and let z satisfy $\mathcal{L}z = \rho$ with $z(0) = z'(0) = 0$ and $\rho \in C[0,1]$. Then*

$$\|z\|_{L^\infty(0,1)} \;\le\; \frac{\|\rho\|_{L^\infty(0,1)}}{2(k+1) - \|\beta\|_\infty},$$

so that Assumption A1 holds with

$$C_s = \big(2(k+1) - \|\beta\|_\infty\big)^{-1}.$$

Proof. Multiplying $\mathcal{L}z = \rho$ by x^k gives $(x^k z')' = x^k(\rho - \beta z)$. Integrating from 0 to x and using $z'(0) = 0$,

$$x^k z'(x) = \int_0^x t^k\big(\rho(t) - \beta(t) z(t)\big)\, dt,$$

so

$$|z'(x)| \le \big(\|\rho\|_\infty + \|\beta\|_\infty \|z\|_\infty\big) x/(k+1).$$

Integrating again and using $z(0) = 0$,

$$|z(x)| \le \big(\|\rho\|_\infty + \|\beta\|_\infty \|z\|_\infty\big) \frac{x^2}{2(k+1)}.$$

Taking the supremum over $[0,1]$ and rearranging, which the hypothesis $\|\beta\|_\infty < 2(k+1)$ permits, gives the bound. □

Assumption A2. The exact solution satisfies $y^* \in C^4[0,1]$ with $y^*(0) = a$ and $(y^*)'(0) = 0$, and the reduced solution $v^*(x) = \big(y^*(x) - a\big)/x^2$, extended to the origin by

$$v^*(0) = \frac{1}{2}(y^*)''(0),$$

is approximable in the trial space,

$$\eta(h) := \inf_{s \in V_{\varepsilon,h}} \|v^* - s\|_{C^2[0,1]} \longrightarrow 0 \qquad \text{as } h \to 0,$$

where

$$V_{\varepsilon,h} = \operatorname{span}\{\phi(\varepsilon, \cdot - c_j)\}_{j=1}^{M}$$

is the trial space spanned by the kernels at the centers.

In other words, the trial space must be rich enough to reproduce the reduced solution as the centers are refined. That the extension of v^* is of class $C^2[0,1]$ follows from Taylor expansion about the origin,

$$y^*(x) - a = \frac{1}{2}(y^*)''(0)\, x^2 + \frac{1}{6}(y^*)'''(0)\, x^3 + O(x^4).$$

The convergence of $\eta(h)$, by contrast, is a hypothesis on the trial space and not a consequence of the smoothness of y^* alone. Rates are available under native-space conditions on v^* [49, Ch. 11], which are strictly stronger than $v^* \in C^2[0,1]$. Rates are also available under Sobolev conditions [52]. These give only algebraic decay and, for the multiquadric kernel, require an augmentation of the trial space that is not used here. No rate is assumed, and Theorem 4.6 is stated in terms of $\eta(h)$ itself. Whether the centers $c_j = (j-1)/(M-1)$ and the shape parameters used here satisfy such conditions is not verified. Assumption A2 therefore remains an open hypothesis for each of the three kernels, and Theorem 4.6 gives convergence only conditionally. The computed errors at fixed ε are discussed in this respect in Section 5.

Assumption A3. There exist $\tau > 1/2$ and a constant $\Lambda > 0$, independent of h, such that the residual of the least-squares approximation satisfies

$$\big\|R(\cdot; \mathbf{w}^\star)\big\|_{H^\tau(0,1)} \le \Lambda,$$

where $\mathbf{w}^\star$ is the minimum-norm minimizer (3.8) of

$$\sum_{i=1}^{H} R(x_i; \mathbf{w})^2,$$

taken in exact arithmetic.

In other words, the residual must not oscillate arbitrarily fast between the collocation nodes, since otherwise it could be small at every node and large in between. Because the trial space is finite dimensional and explicitly generated, a sufficient condition for the required H^τ bound is a uniform bound on $\sqrt{M}\,\|\mathbf{w}^\star\|_2$, the scaled norm of the least-squares coefficient vector, as the next proposition shows; whether such a bound holds is not established.

Proposition 4.3. *Let $\tau \in (1/2, 2]$, let $\beta \in C^2[0,1]$ and $\beta a - r \in H^\tau(0,1)$, and suppose*

$$\phi(\varepsilon, \cdot) \in H^{\tau+2}(-1,1),$$

as holds for the Gaussian, multiquadric, and inverse multiquadric kernels. Set

$$K_\phi(\varepsilon, \tau) = \max_{1 \le j \le M} \|\phi(\varepsilon, \cdot - c_j)\|_{H^{\tau+2}(0,1)}.$$

Then for every $\mathbf{w} \in \mathbb{R}^M$,

$$\|R(\cdot; \mathbf{w})\|_{H^\tau(0,1)} \le C'_{\mathcal{L}} K_\phi(\varepsilon, \tau) \sqrt{M} \|\mathbf{w}\|_2 + \|\beta a - r\|_{H^\tau(0,1)},$$

where $C'_{\mathcal{L}}$ depends only on k and $\|\beta\|_{C^2[0,1]}$. Moreover,

$$K_\phi(\varepsilon, \tau) \le \bar{K}_\phi(\varepsilon, \tau) := \|\phi(\varepsilon, \cdot)\|_{H^{\tau+2}(-1,1)},$$

which depends on neither M nor the centers. In particular, along a refinement family, Assumption A3 holds if $\sup_h \sqrt{M}\|\mathbf{w}^\star\|_2 < \infty$, with

$$\Lambda = C'_{\mathcal{L}} \bar{K}_\phi \sup_h \sqrt{M}\|\mathbf{w}^\star\|_2 + \|\beta a - r\|_{H^\tau(0,1)};$$

a finite bound for each fixed M does not suffice.

Proof. By Lemma 4.4 with $v = N(\cdot; \mathbf{w})$,

$$R(\cdot; \mathbf{w}) = x^2 N'' + (4 + k)xN' + (2 + 2k + \beta x^2)N + \beta a - r.$$

The coefficients x^2, $(4 + k)x$ and $2 + 2k + \beta(x)x^2$ are smooth and bounded on $[0, 1]$, and pointwise multiplication by such a function is bounded on $H^\tau(0, 1)$ for $\tau \in [0, 2]$, whence

$$\|R(\cdot; \mathbf{w})\|_{H^\tau} \le C'_{\mathcal{L}}(\|N''\|_{H^\tau} + \|N'\|_{H^\tau} + \|N\|_{H^\tau}) + \|\beta a - r\|_{H^\tau} \le C'_{\mathcal{L}}\|N\|_{H^{\tau+2}} + \|\beta a - r\|_{H^\tau}.$$

Since $N = \sum_j w_j \phi(\varepsilon, \cdot - c_j)$, the triangle and Cauchy-Schwarz inequalities give

$$\|N\|_{H^{\tau+2}} \le K_\phi \|\mathbf{w}\|_1 \le K_\phi \sqrt{M}\|\mathbf{w}\|_2,$$

which is the stated bound. □

Proposition 4.3 replaces an unverifiable hypothesis by a sufficient condition on the coefficient norm. Assumption A3 holds with Λ independent of h whenever $\sqrt{M}\,\|\mathbf{w}^\star\|_2$ remains uniformly bounded under refinement. Only sufficiency is asserted, and no converse is claimed. For $\mathbf{w}^\star$, this norm cannot be obtained in floating-point arithmetic, since $\mathbf{A}$ is numerically rank-deficient. The computed vector is discussed after Theorem 4.6. The condition is sufficient but not necessary. The bound uses only the triangle inequality and ignores cancellation between the terms of $N(\cdot; \mathbf{w})$, which is substantial when the coefficients are large. There is no theoretical guarantee that $\sqrt{M}\|\mathbf{w}\|_2$ stays bounded under refinement, and for nearly flat bases, one expects it to grow. The computed values are compared with this condition in Section 5.

Lemma 4.4. *Let $v \in C^2[0,1]$ and set $u(x) = a + x^2 v(x)$. Then*

$$\mathcal{L}u = x^2 v'' + (4+k)\, x\, v' + (2 + 2k + \beta(x)x^2)v + \beta(x)a.$$

In particular, $\mathcal{L}u$ extends continuously to $[0,1]$, and there exists a constant $C_{\mathcal{L}} > 0$ such that

$$\|\mathcal{L}(x^2 v)\|_{L^\infty(0,1)} \le C_{\mathcal{L}}\, \|v\|_{C^2[0,1]}.$$

Proof. Differentiating $u = a + x^2 v$ and substituting into

$$\mathcal{L}u = u'' + \frac{k}{x}u' + \beta u$$

gives the stated identity. The term

$$\frac{k}{x}u' = k(2v + xv')$$

extends continuously to $x = 0$ by the computation in the proof of Proposition 4.1. The coefficients x^2, $(4+k)x$ and $2 + 2k + \beta(x)x^2$ are continuous and bounded on $[0,1]$, which gives the operator-norm bound. □

Lemma 4.5. *Let $s \in V_{\varepsilon,h}$ satisfy $\|v^* - s\|_{C^2[0,1]} \le 2\eta(h)$, and set $y_b^t(x) = a + x^2 s(x)$. Then*

$$\|R(\cdot; \mathbf{w}^\star)\|_{\ell_\infty(X)} \le 2C_{\mathcal{L}}\, H^{1/2}\, \eta(h).$$

Proof. Since $y_b^t - y^* = x^2(s - v^*)$ and $\mathcal{L}y^* = r$, Lemma 4.4 gives

$$R_b := \mathcal{L}y_b^t - r = \mathcal{L}(y_b^t - y^*), \qquad \|R_b\|_{L^\infty(0,1)} \le C_{\mathcal{L}}\, \|s - v^*\|_{C^2[0,1]} \le 2C_{\mathcal{L}}\, \eta(h).$$

Because $\mathbf{w}^\star$ minimizes the discrete least-squares residual and $\|f\|_{\ell_2(X)} \le H^{1/2}\|f\|_{L^\infty(0,1)}$ for any f,

$$\|R(\cdot; \mathbf{w}^\star)\|_{\ell_2(X)} \le \|R_b\|_{\ell_2(X)} \le H^{1/2}\|R_b\|_{L^\infty(0,1)} \le 2C_{\mathcal{L}}\, H^{1/2}\, \eta(h).$$

The claim follows from $\|f\|_{\ell_\infty(X)} \le \|f\|_{\ell_2(X)}$. □

Theorem 4.6. *Suppose that Assumptions A1–A3 hold, and let τ be the exponent in Assumption A3. For every α with*

$$0 < \alpha < \min\{\tau - \frac{1}{2}, 1\}$$

there is a constant $C > 0$, independent of h, such that the least-squares RBF-ELM approximation $y_\star^t(x) = a + x^2 N(x; \mathbf{w}^\star)$ satisfies

$$\|y_\star^t - y^*\|_{L^\infty(0,1)} \le C(h^\alpha \Lambda + H^{1/2}\eta(h)).$$

Consequently, if $H^{1/2}\eta(h) \to 0$ as $h \to 0$, then $\|y_\star^t - y^*\|_{L^\infty(0,1)} \to 0$.

Proof. Let $e = y_\star^t - y^*$. Since both functions satisfy the initial conditions exactly, $e(0) = e'(0) = 0$ and $\mathcal{L}e = R(\cdot; \mathbf{w}^\star)$. Applying Assumption A1 with $z = e$ and $\rho = R(\cdot; \mathbf{w}^\star)$,

$$\|e\|_{L^\infty(0,1)} \le C_s\, \|R(\cdot; \mathbf{w}^\star)\|_{L^\infty(0,1)}. \tag{4.3}$$

The residual is carried from the nodes to the whole interval as follows. Let

$$h_X = \sup_{x\in[0,1]} \min_{1\le i\le H} |x - x_i|$$

be the fill distance of the collocation set. By the quasi-uniformity assumed above, $h_X \le c_X h$, with c_X independent of h. Since $\tau > 1/2$ and

$$0 < \alpha < \min\{\tau - \frac{1}{2}, 1\},$$

the Sobolev embedding $H^\tau(0,1) \hookrightarrow C^{0,\alpha}[0,1]$ [53] gives a constant C_E, depending only on τ and α, such that

$$|g(x) - g(z)| \le C_E \, \|g\|_{H^\tau(0,1)} \, |x - z|^\alpha$$

for all $g \in H^\tau(0,1)$ and $x, z \in [0,1]$. Let $\mathbf{w} \in \mathbb{R}^M$ and $g = R(\cdot;\mathbf{w})$. For $x \in [0,1]$, there is a node $x_i \in X$ with $|x - x_i| \le h_X$, and therefore,

$$|g(x)| \le |g(x_i)| + |g(x) - g(x_i)| \le \|g\|_{\ell_\infty(X)} + C_E \, h_X^\alpha \, \|g\|_{H^\tau(0,1)}.$$

Taking the supremum over x gives

$$\|R(\cdot;\mathbf{w})\|_{L^\infty(0,1)} \le C_1 \, h^\alpha \, \|R(\cdot;\mathbf{w})\|_{H^\tau(0,1)} + C_2 \, \|R(\cdot;\mathbf{w})\|_{\ell_\infty(X)}, \qquad C_1 = C_E \, c_X^\alpha, \quad C_2 = 1. \tag{4.4}$$

Applying (4.4) with $\mathbf{w} = \mathbf{w}^\star$, Assumption A3 to the first term, and Lemma 4.5 to the second gives

$$\|R(\cdot;\mathbf{w}^\star)\|_{L^\infty(0,1)} \le C_1 \, h^\alpha \Lambda + 2 C_2 C_{\mathcal{L}} \, H^{1/2} \, \eta(h).$$

Substituting into (4.3) and absorbing all constants into C yields the stated estimate. The final convergence statement follows immediately from the condition $H^{1/2}\eta(h) \to 0$. □

Theorem 4.6 concerns the exact minimizer $\mathbf{w}^\star$; its minimizing property enters the proof only through Lemma 4.5. The vector $\mathbf{w}_c$ computed in floating-point arithmetic (Section 3.3) retains $r < M$ directions and is not a minimizer of the full problem, so that lemma does not apply to it. The estimates (4.3) and (4.4), together with Proposition 4.3, do not use the minimizing property. They hold for every coefficient vector, since every trial solution satisfies the initial conditions exactly. For the computed solution, they give

$$\|y^t(\cdot;\mathbf{w}_c) - y^*\|_{L^\infty(0,1)} \le C_s(C_1 h^\alpha \Lambda_c + C_2 \|R(\cdot;\mathbf{w}_c)\|_{\ell_\infty(X)}),$$

where Λ_c is the bound of Proposition 4.3 with $\mathbf{w} = \mathbf{w}_c$. The two quantities that depend on $\mathbf{w}_c$ (the factor $\sqrt{M}\|\mathbf{w}_c\|_2$ in Λ_c and the nodal residual) are reported in Section 5. The constants are not computed, so this bound is not used as an error estimate. Rather, it identifies the quantities through which the computed solution enters the analysis.

For kernels such as the Gaussian and the inverse multiquadric, native-space error estimates decay faster than algebraically [49, Ch. 11]. If such a rate holds for $\eta(h)$, the condition $H^{1/2}\eta(h) \to 0$ is satisfied. In that case, the residual-sampling term $h^\alpha\Lambda$ may determine the asymptotic rate of the conditional bound, since the approximation term $H^{1/2}\eta(h)$ is the faster-decaying of the two.

Theorem 4.6 is a statement in exact arithmetic and does not account for rounding. By Proposition 4.3, growth of the computed coefficient norm enters Λ_c directly, so the bound for the computed solution need not decrease under refinement. The theorem is also stated at fixed ε, whereas in practice, ε is selected by the off-node residual. The constants depend on ε both through the native space and through $K_\phi(\varepsilon, \tau)$.

The proof uses $e(0) = e'(0) = 0$, which holds because the initial data are embedded in the trial solution. For a Robin problem, where $a = a(\mathbf{w})$ is determined by the boundary condition, $y^t(0)$ in general differs from $y^*(0)$, so the argument does not apply. The Robin problem of Section 5 therefore illustrates the construction and the collocation procedure, not Theorem 4.6.

Theorem 4.6 is restricted to linear problems, and two steps of its proof do not carry over to the nonlinear case. First, Lemma 4.5 uses the fact that $\mathbf{w}^\star$ minimizes the discrete residual over all weight vectors. For nonlinear q, the least-squares problem is nonconvex, and the damped Newton iteration of Section 3.4 need not reach such a minimizer; termination by stagnation does not establish stationarity. Second, the error equation acquires a term that is quadratic in the error,

$$\mathcal{L}_* e = R - p\,\rho(e),$$

where $\mathcal{L}_*$ is the operator linearized about the exact solution and

$$\rho(e) = q(y^* + e) - q(y^*) - q'(y^*)\,e.$$

If q is smooth enough, and $\mathcal{L}_*$ is stable in the sense of Assumption A1, the stability estimate (4.3) closes only under an a priori bound on the error that is uniform in h and small enough to absorb the quadratic remainder. The sampling inequality (4.4) is unaffected, since it applies to the residual of any trial solution. A convergence result for nonlinear problems would therefore require an a priori bound on the error, together with an analogue of Proposition 4.3 for the nonlinear term. This is a natural direction for further work. The nonlinear results of Section 5 are accordingly presented as numerical evidence. Similarly, the line search (3.13) ensures that the least-squares objective decreases along the Newton iterates. No convergence statement is made for the iteration itself, however, because the standard theory of damped Gauss-Newton methods assumes Jacobians of full rank, which is not the case here.

5. Numerical results

The RBF-ELM method with Gaussian (GA), multiquadric (MQ) and inverse multiquadric (IMQ) kernels is applied to six benchmark problems of Lane-Emden-Fowler type, linear and nonlinear, homogeneous and nonhomogeneous. All computations were carried out in MATLAB R2024a on an Intel Core i9-13900K CPU at 3.00 GHz with 32 GB RAM. Unless stated otherwise, $M = H = 20$ uniformly spaced centers and collocation nodes, $c_j = (j-1)/(M-1)$ and $x_i = (i-1)/(H-1)$, are used on $[0, 1]$, and the node $x_1 = 0$ carries the limiting residual (4.1). The differential equation is thus enforced at the singular endpoint itself, no displaced node enters the formulation, and $|R(0; \mathbf{w})|$ is reported in each result table. The shape parameter is selected as in Section 3.5 from the candidate set $\mathcal{E} = \{0.1, 0.5, 1, 2, 5\}$ on the off-node grid

$$\xi_\ell = (\ell - \frac{1}{2})/500,$$

that is, with $L = 500$ and node-coincident points removed (there are none for $H = 20$; among the refinement levels only $H = 25$ has them, at $1/8$, $3/8$, $5/8$, and $7/8$). Accuracy is measured on the separate grid

$$x_i^* = (i-1)/499, \quad i = 1, \ldots, 500,$$

used only for reporting.

The Newton iteration uses a tolerance of 10^{-10}, a maximum iteration count of 200, a minimum step length of 10^{-10}, and the Armijo constant $c_1 = 10^{-4}$. The maximum iteration count was not reached in any of the computations reported in the tables. The residual correction for the linear problem uses the same tolerance and at most $T_{\max} = 20$ steps. At $M = H = 20$, every computation terminated by stagnation rather than by reaching the residual tolerance, and the terminal state at other refinement levels is noted where it differs. The least-squares computations use the default threshold of lsqminnorm, whose influence is examined below. Relative to $\sigma_{\max}(\mathbf{A})$, this threshold lies between 5.9×10^{-16} and 1.1×10^{-15} for the matrices of Example 1. Computing times are medians of 15 repeated runs at the selected shape parameter, with the first call discarded, and exclude the shape-parameter search. The tables also give the total time of the complete five-candidate search, which includes the computation at the selected value and is measured once rather than as a median. This total lies between 6.7 and 112 ms, or 2 to 36 times the computing time, so the search dominates the cost.

Exact solutions are available for Examples 1, 2, and 5. Elsewhere, the reference is computed with ode45 or bvp4c at a tolerance of 10^{-13} on the truncated interval $[x_0, 1]$ with $x_0 = 10^{-6}$, since ode45 cannot evaluate the singular term $(k/x)y'$ at the origin, and is initialized from the local series expansion given with each example. The truncation point is not a free parameter. Because the initial data are taken from that expansion rather than from the values at the origin, they differ from the exact local behavior by $O(x_0^3)$ or less. At $x_0 = 10^{-6}$, this difference is several orders of magnitude below the integrator tolerance. The accuracy of each reference is therefore governed by that tolerance and not by the position of the truncation point. The agreement between independent solvers provides supporting evidence for the accuracy of the numerical reference solutions, although it does not constitute a rigorous error bound. Tightening the tolerance to 3×10^{-14} changes the references by at most 2.0×10^{-14}. For Examples 3 and 4, an independent integrator, ode113, agrees with ode45 to 6.6×10^{-14} and 1.0×10^{-14}, respectively. For Example 6, bvp4c posed on $[0, 1]$ itself agrees with the truncated computation to 1.1×10^{-16}. Let

$$e_i = |y^t(x_i^*) - y_{\mathrm{ref}}(x_i^*)|$$

denote the deviation of the trial solution from the exact or reference solution. The reported measures are

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{500}\sum_{i=1}^{500} e_i, \\ \mathrm{RMSE} &= \Big(\frac{1}{500}\sum_{i=1}^{500} e_i^2\Big)^{1/2}, \\ L_\infty &= \max_{1\le i\le 500} e_i. \end{aligned} \tag{5.1}$$

5.1. Example 1

The first benchmark is the classical Lane-Emden equation of polytropic index $m = 1$,

$$y''(x) + \frac{2}{x}y'(x) + y(x) = 0, \qquad x \in [0, 1], \tag{5.2}$$

subject to $y(0) = 1$ and $y'(0) = 0$. Substituting $a = 1$ and $b = 0$ into the general trial solution (3.2) gives

$$y^t(x; \mathbf{w}) = 1 + x^2 N(x; \mathbf{w}), \tag{5.3}$$

which satisfies $y^t(0; \mathbf{w}) = 1$ and $(y^t)'(0; \mathbf{w}) = 0$ exactly for every $\mathbf{w} \in \mathbb{R}^M$. The equation is linear and homogeneous, so the residual system reduces to the algebraic form (3.7), and the weights follow from a single least-squares computation (3.8), followed, when needed, by residual correction. The exact solution is [4]

$$y(x) = \frac{\sin x}{x}. \tag{5.4}$$

Table 1 collects values reported in the literature for three neural solvers on the same equation. Since the error measures, node counts, and stopping criteria differ between studies, the table is not a controlled comparison, and the orders of magnitude separating the entries do not measure relative accuracy. The computing times likewise refer to the present method alone, because matched timings for the other methods are unavailable. What the table does show is a difference in how the weights are obtained. MB-PIRNN used 2500 Adam and 500 L-BFGS iterations on this problem. Because the equation is linear, the RBF-ELM weights follow from a single least-squares computation with at most two residual correction steps and no line search. In Table 1 and in the other result tables (Tables 3, 5–7, and 9), Time is the median computing time at the selected ε. Total is the time of the complete five-candidate shape-parameter search, which includes the computation at the selected value. For this linear problem, Iter. is the number of accepted least-squares computations, including the first one.

Table 1. Example 1. Literature values and RBF-ELM results at $M = H = 20$. Not a controlled comparison (see text).

Method	Error measure	Error	Iter.	**Time** (ms)	**Total** (ms)
LeNN [43]	MAE	3.437×10^{-4}	–	–	–
PL-PINN [44]	rel. L_∞	5.85×10^{-3}	–	–	–
MB-PIRNN [45]	L_∞	1.66×10^{-3}	–	–	–
GA-RBF-ELM	MAE	2.500×10^{-11}	3	1.05	7.6
MQ-RBF-ELM	MAE	2.995×10^{-11}	1	0.45	6.8
IMQ-RBF-ELM	MAE	4.379×10^{-12}	1	0.52	6.7

Figure 2 shows that the approximate solutions agree with the exact solution $\sin(x)/x$.

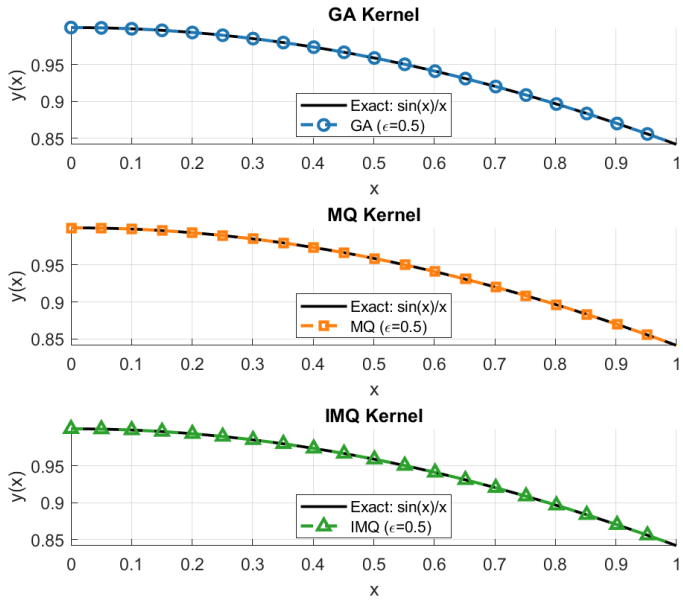


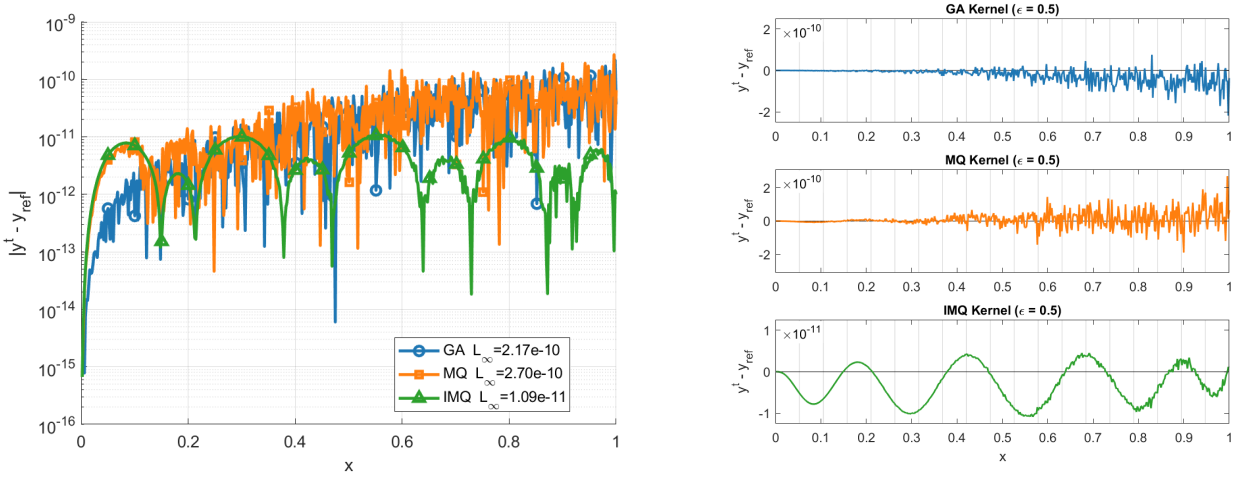
Figure 2. Approximate solutions obtained by the GA (top), MQ (middle), and IMQ (bottom) kernels compared with the exact solution for Example 1.

Table 2 reports the behavior under joint refinement $M = H \in \{5, \ldots, 30\}$, with ε reselected at each M by the off-node residual. In this table and in Tables 4 and 8, r is the rank used by the solver, and $\sqrt{M}\|\mathbf{w}\|_2$ is the factor of Proposition 4.3 that varies with the discretization at fixed ε, evaluated at the computed weights. Two features of the table matter. The error decreases with M until it reaches a floor near 10^{-11}, beyond which further refinement changes the selected shape parameter and the retained subspace without improving the approximation. The level of this floor follows the coefficient norm, which is consistent with a finite-precision effect. The last column holds $\varepsilon = 0.5$ fixed, the setting of Theorem 4.6, whereas the other columns test the procedure used in practice, in which ε is reselected. The last column coincides with the L_∞ column wherever 0.5 is also the reselected value. At fixed ε, the error falls by three to four orders of magnitude between $M = 5$ and $M = 10$ and then levels off in the same way. The limiting residual at the singular endpoint remains small at every refinement level, which confirms that the differential equation is enforced at $x = 0$ itself rather than approached from a nearby interior node. At the coarsest refinement, the accuracy is already comparable to that reported in [43] at the same basis size.

Table 2. Convergence of the RBF-ELM method for Example 1 under joint refinement $M = H$, with ε reselected at each M by the off-node residual.

Kernel	$M = H$	ε	MAE	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0;\mathbf{w})\|$	r	$\sqrt{M}\|\mathbf{w}\|_2$	L_∞, $\varepsilon = 0.5$
GA	5	0.1	5.682×10^{-9}	9.587×10^{-9}	1.630×10^{-9}	1.630×10^{-9}	5	3.3×10^{6}	3.376×10^{-7}
	10	0.5	1.015×10^{-10}	2.034×10^{-10}	2.119×10^{-8}	8.085×10^{-10}	8	3.4×10^{4}	2.034×10^{-10}
	15	0.5	1.705×10^{-11}	1.549×10^{-10}	7.134×10^{-10}	4.657×10^{-10}	9	5.7×10^{6}	1.549×10^{-10}
	20	0.5	2.500×10^{-11}	2.175×10^{-10}	8.001×10^{-10}	2.910×10^{-10}	9	5.2×10^{6}	2.175×10^{-10}
	25	1.0	5.364×10^{-11}	1.986×10^{-10}	1.161×10^{-8}	9.313×10^{-10}	11	3.3×10^{6}	1.073×10^{-10}
	30	1.0	1.102×10^{-11}	7.218×10^{-11}	1.322×10^{-8}	2.125×10^{-9}	11	3.2×10^{6}	1.057×10^{-10}
MQ	5	0.1	6.868×10^{-10}	1.739×10^{-9}	1.164×10^{-10}	1.164×10^{-10}	5	1.2×10^{6}	2.520×10^{-6}
	10	0.1	1.233×10^{-10}	2.579×10^{-10}	3.275×10^{-8}	1.027×10^{-8}	6	9.6×10^{5}	1.304×10^{-9}
	15	0.5	5.793×10^{-11}	4.238×10^{-10}	7.187×10^{-9}	1.281×10^{-9}	11	7.8×10^{6}	4.238×10^{-10}
	20	0.5	2.995×10^{-11}	2.701×10^{-10}	9.155×10^{-9}	2.328×10^{-10}	11	6.0×10^{6}	2.701×10^{-10}
	25	0.5	2.716×10^{-11}	1.867×10^{-10}	1.123×10^{-8}	3.027×10^{-9}	11	5.6×10^{6}	1.867×10^{-10}
	30	0.1	9.797×10^{-11}	2.254×10^{-10}	4.607×10^{-8}	4.607×10^{-8}	6	1.1×10^{6}	1.602×10^{-10}
IMQ	5	0.1	2.324×10^{-9}	4.118×10^{-9}	1.735×10^{-11}	3.638×10^{-12}	5	5.1×10^{4}	1.195×10^{-6}
	10	0.1	1.020×10^{-10}	1.988×10^{-10}	3.023×10^{-8}	8.964×10^{-9}	6	8.8×10^{5}	1.834×10^{-10}
	15	0.5	1.115×10^{-11}	1.740×10^{-11}	3.885×10^{-9}	1.100×10^{-10}	11	6.0×10^{4}	1.740×10^{-11}
	20	0.5	4.379×10^{-12}	1.090×10^{-11}	6.297×10^{-9}	6.414×10^{-10}	11	5.0×10^{4}	1.090×10^{-11}
	25	0.5	4.278×10^{-12}	9.291×10^{-12}	6.482×10^{-9}	1.612×10^{-9}	12	4.6×10^{4}	9.291×10^{-12}
	30	0.5	4.331×10^{-12}	9.075×10^{-12}	8.110×10^{-9}	2.578×10^{-9}	11	4.6×10^{4}	9.075×10^{-12}

Figure 3 shows the pointwise absolute error on a logarithmic scale. For all three kernels, the error remains below 3×10^{-10} across the interval. The error vanishes at $x = 0$ because the initial data are embedded exactly in (5.3). This zero carries no information about the equation, and the point is omitted from the logarithmic plot. How well the equation is satisfied at the origin is measured by $|R(0;\mathbf{w})|$, reported in Table 2. The inverse multiquadric error consists of smooth arches, whereas the Gaussian and multiquadric errors are irregular and grow toward $x = 1$, in line with their much larger coefficient norms (about 10^6 against 10^4, Table 11). The irregular part varies from point to point and is of the size of $\epsilon_{\text{mach}}\|\mathbf{w}_c\|_2$, about 2.6×10^{-10} for the Gaussian and 3.0×10^{-10} for the multiquadric kernel, as expected for rounding errors. The arches are consistent with the smooth truncation error of the retained subspace, which the irregular part hides for those two kernels. Panel (b) shows the same errors with their sign on a linear scale, each kernel on its own axis, with the collocation nodes marked. The smooth component changes sign seven or eight times, about two to three node spacings apart, and its zeros do not coincide with the nodes. The irregular component grows roughly like x^2, as expected for rounding errors in $N(x;\mathbf{w}_c)$, which enter the trial solution multiplied by x^2. Every sign change of the error, in either component, appears in panel (a) as a downward spike.



(a) (b)

Figure 3. Pointwise error for the GA, MQ, and IMQ kernels applied to Example 1. Panel (a) shows $|y^t - y_{\mathrm{ref}}|$ on a logarithmic scale, and panel (b) shows the signed error $y^t - y_{\mathrm{ref}}$ on a linear scale, in units of the factor shown in each panel, with grey lines at the collocation nodes. The error at $x = 0$ is exactly zero, since the initial value is built into the trial solution, and is not shown in (a).

5.2. *Example 2*

The second benchmark is the Lane-Emden equation of polytropic index $m = 5$,

$$y''(x) + \frac{2}{x}y'(x) + y^5(x) = 0, \qquad x \in [0, 1], \tag{5.5}$$

subject to $y(0) = 1$ and $y'(0) = 0$. Unlike Example 1, this problem is nonlinear, so the Newton iteration (3.10) with the analytically computed Jacobian (3.11) and (3.12) replaces the single least-squares computation. A closed-form solution exists for $m = 5$ and is given by Davis [4] as

$$y(x) = \left(1 + \frac{x^2}{3}\right)^{-1/2}, \tag{5.6}$$

which is monotonically decreasing and bounded away from zero on $[0, 1]$.

Table 3 lists published results for the same problem. As in Example 1, the entries come from different studies, with different error measures, node counts, and stopping criteria, so the table is not a controlled comparison and the differences between entries do not measure relative accuracy. Times are given for the present method only. For this and the other nonlinear problems, Iter. is the number of Newton steps. Halv. gives the step halvings in accepted steps / in total. The total also counts the final unsuccessful line search, if any, after which the iteration stops by stagnation. Time and Total are as in Table 1. The nonlinearity is felt in the solver rather than in the accuracy. The Newton iteration stops after seven or eight steps, with between 53 and 85 step halvings depending on the kernel, so the Armijo safeguard of Section 3.4 is active throughout. The reported accuracy is nevertheless of the same order as on the linear problem of Example 1.

Table 3. Example 2. Literature values and RBF-ELM results at $M = H = 20$. Not a controlled comparison (see text).

Method	Error measure	Error	Iter.	Halv.	**Time** (ms)	**Total** (ms)
LeNN [43]	MAE	5.752×10^{-4}	–	–	–	–
PL-PINN [44]	rel. L_∞	5.12×10^{-3}	–	–	–	–
MB-PIRNN [45]	L_∞	1.83×10^{-3}	–	–	–	–
GA-RBF-ELM	MAE	8.022×10^{-12}	7	27/61	2.64	13.3
MQ-RBF-ELM	MAE	1.015×10^{-11}	8	19/53	4.22	14.6
IMQ-RBF-ELM	MAE	4.890×10^{-12}	7	51/85	8.45	25.4

Figure 4 compares the approximate solutions with the exact solution (5.6). The solution decreases monotonically from $y(0) = 1$ to $(4/3)^{-1/2} \approx 0.866$ at $x = 1$, and in all three panels, the numerical and exact curves coincide to within the resolution of the figure.

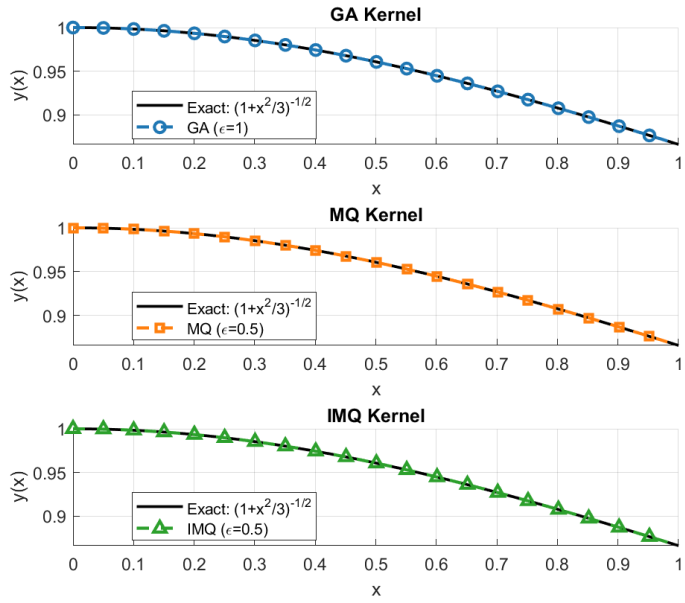


Figure 4. Approximate solutions obtained by the GA, MQ, and IMQ kernels compared with the exact solution (5.6) for Example 2.

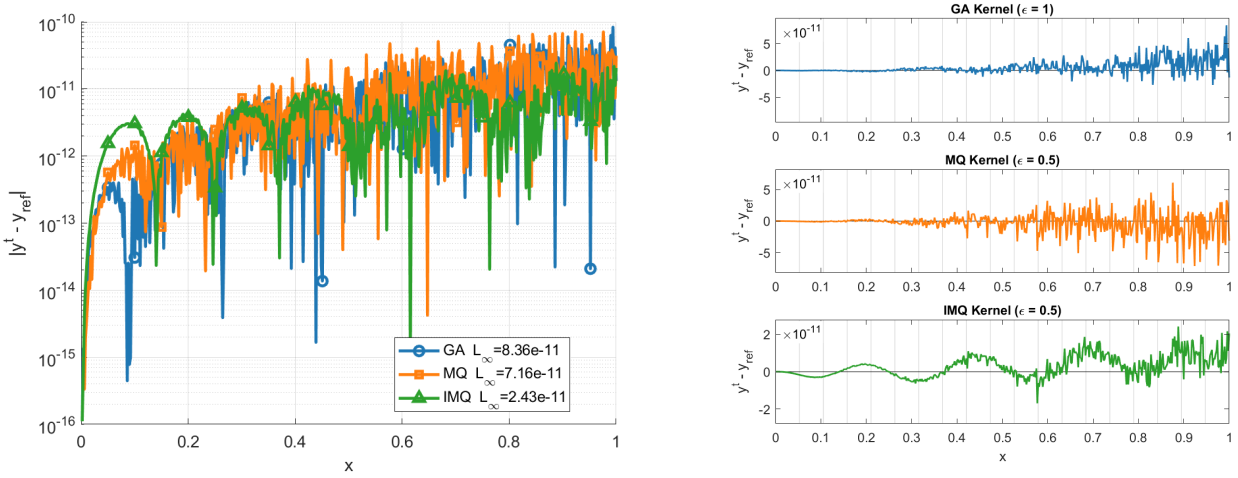
Table 4 reports the same refinement study for the nonlinear problem. The error decreases rapidly and then saturates at the same floor. One trend is easily misread. The limiting residual at the origin grows with M, though not monotonically. This growth does not indicate a loss of accuracy at the singular endpoint. Row 1 is one of H equations, and $|R(0; \mathbf{w})|$ follows the nodal residual as a whole. While the retained rank equals M ($M = 5$ and 10), the system is solved to the residual tolerance, with $\|\mathbf{R}\|_\infty \leq 9 \times 10^{-11}$, and $|R(0; \mathbf{w})|$ lies between 2×10^{-15} and 6×10^{-11}. Once $r < M$ ($M \geq 15$), the discarded directions leave a least-squares residual, and $\|\mathbf{R}\|_\infty$ jumps to between 1×10^{-9} and 4×10^{-8}. The limiting residual $|R(0; \mathbf{w})|$ rises with it and lies between 0.002 and 0.4 times $\|\mathbf{R}\|_\infty$. The row at

$x = 0$ is not singled out, since its Euclidean norm is between 0.67 and 1.01 times the median norm of the other rows in all 18 kernel-example cases. Accordingly, the coarser refinements reach the residual tolerance and the finer ones stagnate.

Table 4. Convergence of the RBF-ELM method for Example 2 under joint refinement $M = H$, with ε reselected at each M by the off-node residual.

Kernel	$M = H$	ε	MAE	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0; \mathbf{w})\|$	r	$\sqrt{M}\|\mathbf{w}\|_2$	L_∞, $\varepsilon = 0.5$
GA	5	0.5	1.573×10^{-6}	2.866×10^{-6}	4.625×10^{-12}	4.441×10^{-15}	5	3.8×10^{1}	2.866×10^{-6}
	10	1.0	1.070×10^{-9}	2.217×10^{-9}	2.617×10^{-11}	2.274×10^{-11}	10	1.3×10^{5}	4.126×10^{-9}
	15	1.0	1.262×10^{-11}	8.173×10^{-11}	1.736×10^{-9}	7.276×10^{-11}	11	2.7×10^{6}	3.542×10^{-9}
	20	1.0	8.022×10^{-12}	8.362×10^{-11}	3.600×10^{-9}	5.821×10^{-11}	11	2.0×10^{6}	2.686×10^{-9}
	25	1.0	8.790×10^{-12}	5.695×10^{-11}	4.298×10^{-9}	2.910×10^{-11}	11	1.8×10^{6}	2.628×10^{-9}
	30	1.0	5.087×10^{-12}	2.956×10^{-11}	4.591×10^{-9}	7.276×10^{-12}	11	1.8×10^{6}	2.576×10^{-9}
MQ	5	0.5	6.881×10^{-7}	1.181×10^{-6}	4.843×10^{-12}	7.105×10^{-15}	5	2.2×10^{2}	1.181×10^{-6}
	10	0.5	1.869×10^{-10}	3.737×10^{-10}	8.919×10^{-11}	1.455×10^{-11}	10	4.6×10^{5}	3.737×10^{-10}
	15	0.5	1.648×10^{-11}	1.378×10^{-10}	1.375×10^{-9}	2.328×10^{-10}	11	2.8×10^{6}	1.378×10^{-10}
	20	0.5	1.015×10^{-11}	7.160×10^{-11}	2.246×10^{-9}	2.328×10^{-10}	11	2.2×10^{6}	7.160×10^{-11}
	25	0.5	9.282×10^{-12}	7.072×10^{-11}	2.838×10^{-9}	5.821×10^{-11}	11	2.0×10^{6}	7.072×10^{-11}
	30	0.5	2.504×10^{-11}	5.124×10^{-11}	4.050×10^{-8}	1.534×10^{-8}	11	1.4×10^{5}	5.124×10^{-11}
IMQ	5	0.5	5.144×10^{-7}	1.744×10^{-6}	4.976×10^{-12}	1.776×10^{-15}	5	3.4×10^{1}	1.744×10^{-6}
	10	0.5	4.501×10^{-10}	9.632×10^{-10}	5.684×10^{-11}	5.684×10^{-11}	10	9.3×10^{4}	9.632×10^{-10}
	15	0.5	4.577×10^{-12}	3.141×10^{-11}	4.661×10^{-9}	5.821×10^{-11}	11	5.4×10^{5}	3.141×10^{-11}
	20	0.5	4.890×10^{-12}	2.431×10^{-11}	7.642×10^{-9}	4.075×10^{-10}	11	4.2×10^{5}	2.431×10^{-11}
	25	0.5	5.555×10^{-12}	2.465×10^{-11}	8.252×10^{-9}	9.750×10^{-10}	11	3.9×10^{5}	2.465×10^{-11}
	30	0.5	4.394×10^{-12}	1.665×10^{-11}	9.793×10^{-9}	1.433×10^{-9}	11	3.8×10^{5}	1.665×10^{-11}

The pointwise error in Figure 5 behaves as in Example 1, again vanishing at $x = 0$ because of the embedded initial data. The inverse multiquadric kernel attains the smallest maximum error, $L_\infty = 2.431 \times 10^{-11}$, followed by the multiquadric and Gaussian kernels at 7.160×10^{-11} and 8.362×10^{-11}. Panel (b) shows the same two components as in Example 1.



(a) (b)

Figure 5. Pointwise error for the GA, MQ, and IMQ kernels applied to Example 2. Panel (a) shows $|y^t - y_{\text{ref}}|$ on a logarithmic scale, and panel (b) shows the signed error $y^t - y_{\text{ref}}$ on a linear scale, in units of the factor shown in each panel, with grey lines at the collocation nodes. The error at $x = 0$ is exactly zero, since the initial value is built into the trial solution, and is not shown in (a).

5.3. Example 3

Consider the nonlinear homogeneous Lane-Emden-Fowler equation with singularity coefficient $k = 3$,

$$y''(x) + \frac{3}{x}y'(x) + 2x^2y^2(x) = 0, \qquad x \in [0, 1], \tag{5.7}$$

subject to $y(0) = 1$ and $y'(0) = 0$. The trial solution is therefore again $y^t(x; \mathbf{w}) = 1 + x^2N(x; \mathbf{w})$ as in (5.3), which satisfies the initial conditions for every $\mathbf{w} \in \mathbb{R}^M$. No exact solution is available, so ode45 serves as the reference. Substituting $y = 1 + c_4x^4$ into (5.7) gives $c_4 = -1/12$, and the integrator is started from $y = 1 - x^4/12$ and $y' = -x^3/3$ at $x_0 = 10^{-6}$. The quantities reported for this example, and for Examples 4 and 6, are therefore measured against a numerical reference rather than against a closed-form solution.

Table 5 gives the error metrics; the Newton iteration stops by stagnation after four to six steps. One column requires interpretation. For this problem, $p(0) = 0$ and $r(0) = 0$, so the limiting residual (4.1) reduces to the homogeneous condition $N(0; \mathbf{w}) = 0$. The small reported values of $|R(0; \mathbf{w})|$ therefore show that the differential equation is enforced at the singular endpoint itself, with no offset parameter and no extrapolation.

Table 5. Error metrics for Example 3, with $M = H = 20$.

Method	ε	**MAE**	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0; \mathbf{w})\|$	**Iter.**	**Halv.**	**Time** (ms)	**Total** (ms)
GA-RBF-ELM	1.0	3.198×10^{-11}	2.266×10^{-10}	1.399×10^{-8}	1.164×10^{-10}	6	33/67	2.12	12.7
MQ-RBF-ELM	0.5	2.963×10^{-10}	7.853×10^{-10}	4.346×10^{-7}	5.774×10^{-8}	4	0/0	1.09	11.2
IMQ-RBF-ELM	0.5	1.254×10^{-10}	3.114×10^{-10}	2.071×10^{-7}	2.026×10^{-8}	4	1/1	1.42	18.8

Figure 6 compares the approximate solutions with the reference; the solution decays monotonically from $y(0) = 1$ to approximately 0.921 at $x = 1$.

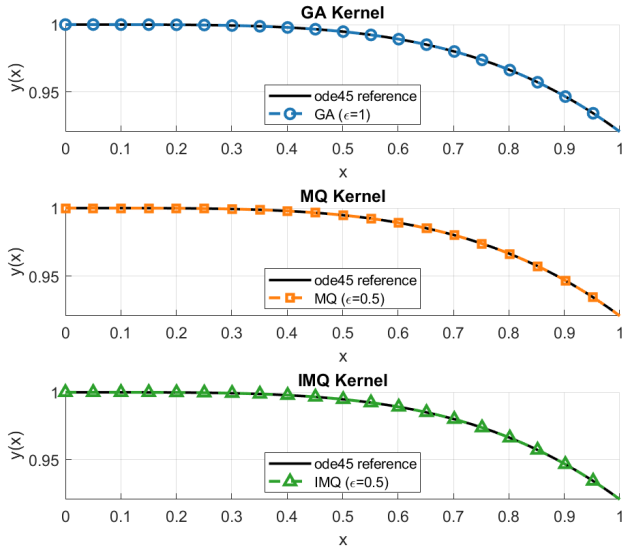
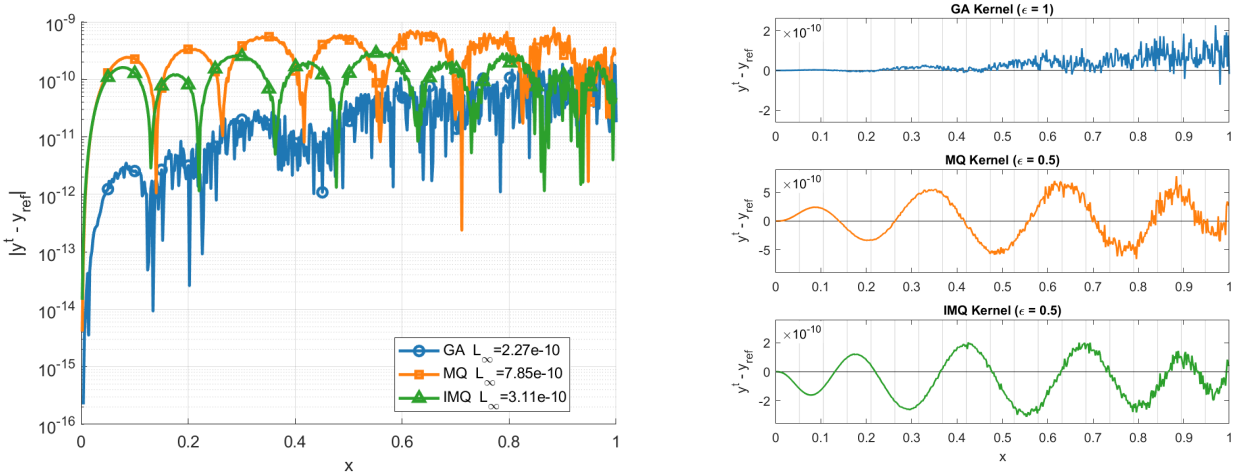


Figure 6. Approximate solutions obtained by the GA, MQ, and IMQ kernels compared with the ode45] reference solution for Example 3.

Figure 7 shows the pointwise absolute error on a logarithmic scale. The Gaussian kernel has the smallest error over most of the interval, and all three errors grow toward $x = 1$. In panel (b), the multiquadric and inverse multiquadric errors are smooth oscillations of the same kind as in Example 1, whereas the Gaussian error is irregular.



(a) (b)

Figure 7. Pointwise error for the GA, MQ, and IMQ kernels applied to Example 3. Panel (a) shows $|y^t - y_{\text{ref}}|$ on a logarithmic scale, and panel (b) shows the signed error $y^t - y_{\text{ref}}$ on a linear scale, in units of the factor shown in each panel, with grey lines at the collocation nodes. The error at $x = 0$ is exactly zero, since the initial value is built into the trial solution, and is not shown in (a).

5.4. Example 4

Consider the nonlinear nonhomogeneous Lane-Emden-Fowler equation with singularity coefficient $k = 8$,

$$y''(x) + \frac{8}{x}y'(x) + x\,y^2(x) = x^4 + x^5, \qquad x \in [0, 1], \tag{5.8}$$

subject to $y(0) = 1$ and $y'(0) = 0$, so that the trial solution is again $y^t(x; \mathbf{w}) = 1 + x^2 N(x; \mathbf{w})$ as in (5.3). Substituting $y = 1 + c_3 x^3$ into (5.8) gives $c_3 = -1/30$, so the ode45 reference is started from $y = 1 - x^3/30$ and $y' = -x^2/10$. This problem differs from Example 3 in the stronger singularity, $k = 8$, and in the nonhomogeneous term $x^4 + x^5$ that must also be resolved.

The errors in Table 6 are roughly an order of magnitude larger than in Example 3, consistent with the stronger singularity and the additional forcing term. The Newton iteration stops after two to five steps, and the coefficient norms, between 7.7×10^6 and 1.0×10^7, are the largest of any example.

Table 6. Error metrics for Example 4, with $M = H = 20$.

Method	ε	**MAE**	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0; \mathbf{w})\|$	**Iter.**	**Halv.**	**Time** (ms)	**Total** (ms)
GA-RBF-ELM	1.0	2.893×10^{-10}	1.431×10^{-9}	5.397×10^{-7}	3.772×10^{-8}	3	1/35	1.23	11.6
MQ-RBF-ELM	0.5	2.812×10^{-9}	7.047×10^{-9}	5.234×10^{-6}	6.454×10^{-7}	2	0/0	0.68	13.1
IMQ-RBF-ELM	0.5	1.248×10^{-9}	3.753×10^{-9}	2.880×10^{-6}	1.844×10^{-7}	5	10/44	4.65	22.0

Figure 8 compares the kernel solutions with the reference. The solution decreases only slightly, from $y(0) = 1$ to a minimum of about 0.9877 near $x \approx 0.89$, and then rises to about 0.990 at $x = 1$. The forcing term largely offsets the damping effect of xy^2.

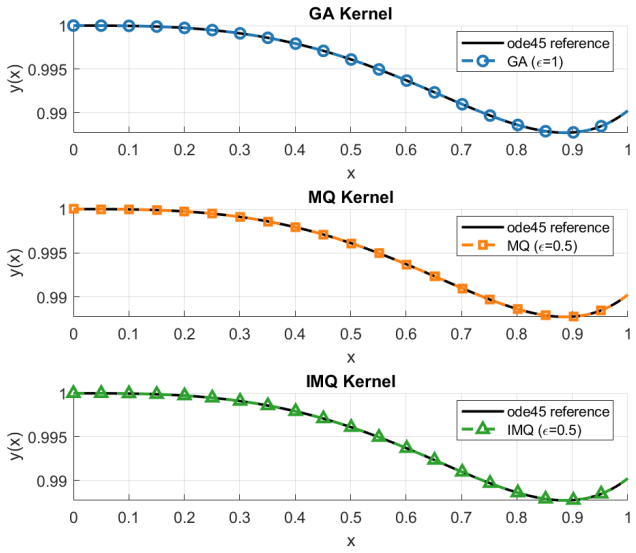
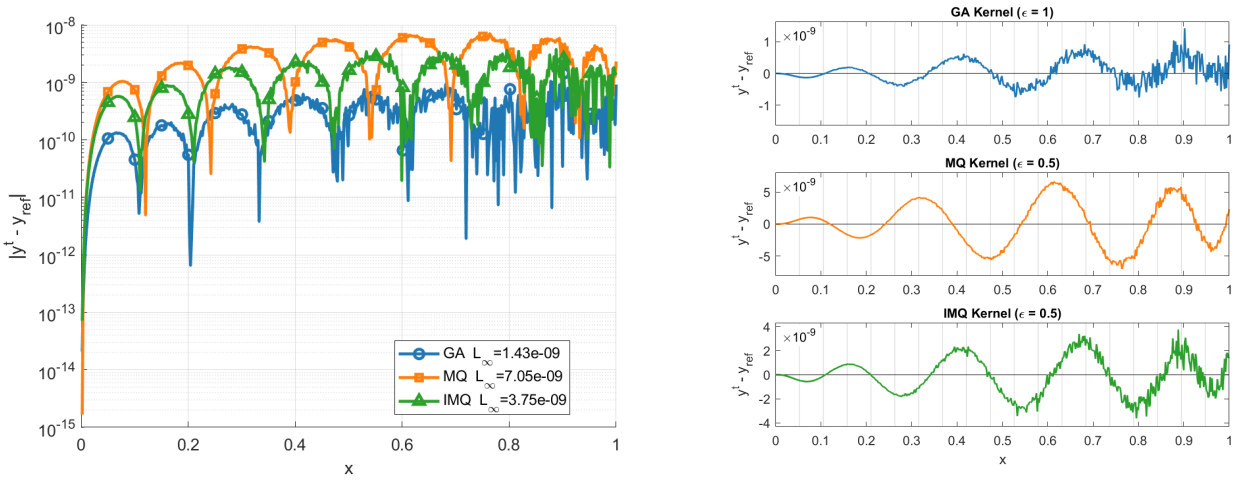


Figure 8. Approximate solutions obtained by the GA, MQ, and IMQ kernels compared with the ode45 reference solution for Example 4.

Figure 9 shows the pointwise absolute error. It increases over the first half of the interval and stays close to its maximum on $[0.6, 1]$. The Gaussian kernel is the most accurate over most of the interval.

Panel (b) shows smooth oscillations for all three kernels, with an irregular component in the second half of the interval for the Gaussian and inverse multiquadric kernels.



(a) (b)

Figure 9. Pointwise error for the GA, MQ, and IMQ kernels applied to Example 4. Panel (a) shows $|y^t - y_{\text{ref}}|$ on a logarithmic scale, and panel (b) shows the signed error $y^t - y_{\text{ref}}$ on a linear scale, in units of the factor shown in each panel, with grey lines at the collocation nodes. The error at $x = 0$ is exactly zero, since the initial value is built into the trial solution, and is not shown in (a).

5.5. *Example 5*

Consider the Lane-Emden-Fowler equation with logarithmic nonlinearity and singularity coefficient $k = 6$,

$$y''(x) + \frac{6}{x}\,y'(x) + 14\,y(x) + 4\,y(x)\ln y(x) = 0, \qquad x \in [0, 1], \tag{5.9}$$

subject to $y(0) = 1$ and $y'(0) = 0$, so that the trial solution is again $y^t(x; \mathbf{w}) = 1 + x^2 N(x; \mathbf{w})$ as in (5.3), which satisfies the initial conditions exactly for every $\mathbf{w} \in \mathbb{R}^M$. The exact solution is

$$y(x) = \exp\left(-x^2\right), \tag{5.10}$$

which decays from $y(0) = 1$ to $\exp(-1) \approx 0.368$ at $x = 1$. This example extends the benchmark suite to a transcendental nonlinearity not covered by the polynomial cases of Examples 2–4. The term $y \ln y$ is smooth for all $y > 0$ but undefined otherwise. Clipping the argument of the logarithm would modify the residual and invalidate the analytic Jacobian. Instead, positivity is maintained by rejecting any line-search step whose trial solution falls to or below $y_{\min} = 10^{-12}$ at a point of X_{pos} (Section 3.4). For $M = H = 20$, the set X_{pos} has 1018 points, and the minimum of the computed solution on it is 0.3679, attained at $x = 1$.

Table 7 reports the error metrics and residual norms. Unlike Examples 3 and 4, the limiting residual here does not reduce to a homogeneous condition. With $p(0)q(a) = 14$, the origin condition (4.1) reads $14\,N(0; \mathbf{w}) + 14 = 0$, so $N(0; \mathbf{w})$ must reproduce the value -1 taken by the reduced exact solution

$v^*(x) = (y(x) - 1)/x^2$ at the origin. It does so to at least eight decimal places. In this sense, the singular endpoint is handled correctly here. The largest computing time reported in this paper is that of the inverse multiquadric kernel on this example, about 48 ms, against 6.6 ms for the Gaussian kernel with a comparable number of Newton steps and step halvings (Table 7). The measured timings alone do not identify the operations responsible for this difference.

Table 7. Error metrics for Example 5, with $M = H = 20$.

Method	ε	**MAE**	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0;\mathbf{w})\|$	**Iter.**	**Halv.**	**Time** (ms)	**Total** (ms)
GA-RBF-ELM	1.0	1.765×10^{-11}	1.352×10^{-10}	6.863×10^{-9}	4.075×10^{-10}	8	39/73	6.59	26.1
MQ-RBF-ELM	0.5	1.593×10^{-10}	3.608×10^{-10}	2.417×10^{-7}	3.922×10^{-8}	4	0/0	2.25	81.0
IMQ-RBF-ELM	0.5	2.321×10^{-11}	1.747×10^{-10}	2.836×10^{-8}	2.445×10^{-9}	7	47/81	47.69	112.1

Figure 10 compares the approximate solutions with the exact solution (5.10).

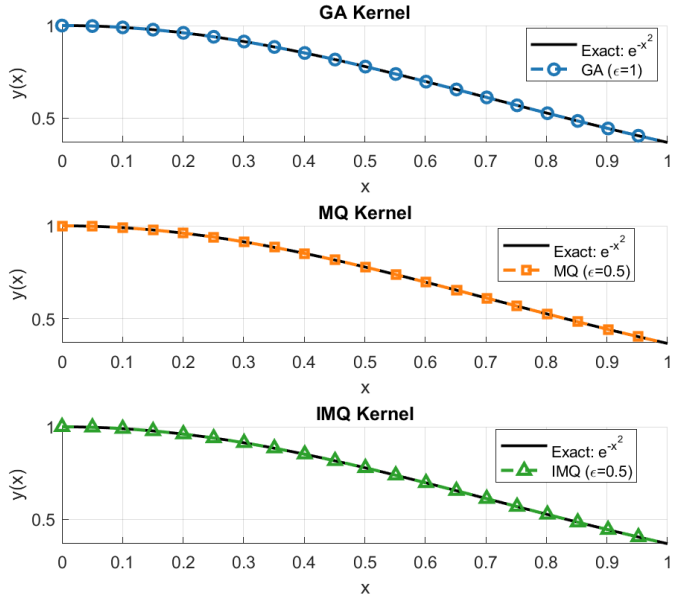


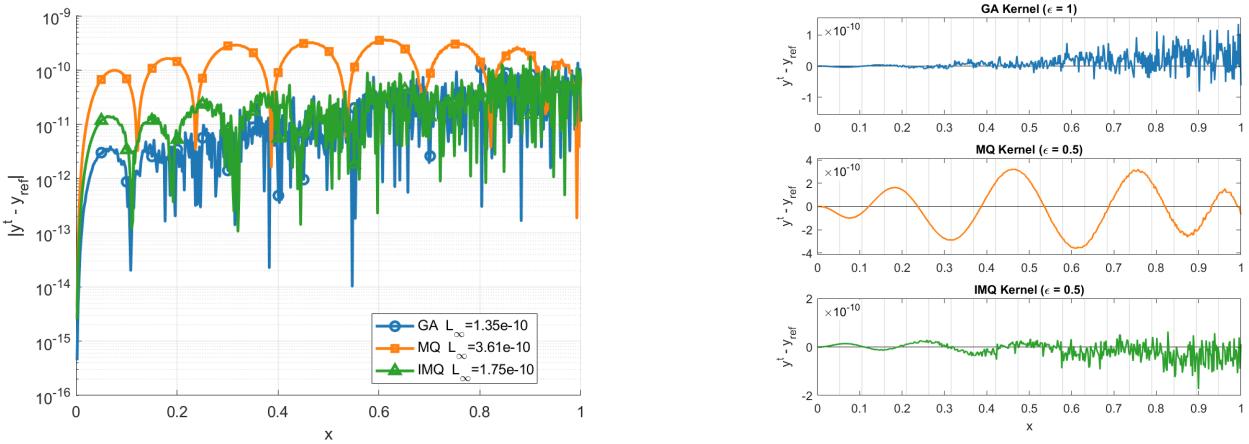
Figure 10. Approximate solutions obtained by the GA, MQ, and IMQ kernels compared with the exact solution for Example 5.

Table 8 shows the same refinement behavior as in the linear and polynomially nonlinear problems, with a rapid decrease followed by saturation at the same floor. The pattern is unchanged for a transcendental nonlinearity, which indicates that it is a property of the discretization rather than of a particular equation. The limiting residual at the origin behaves as in Example 2. It stays below $\|\mathbf{R}\|_\infty$ and rises with it once the retained rank falls below M.

Table 8. Convergence of the RBF-ELM method for Example 5 under joint refinement $M = H$, with ε reselected at each M by the off-node residual.

Kernel	$M = H$	ε	MAE	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0;\mathbf{w})\|$	r	$\sqrt{M}\|\mathbf{w}\|_2$	L_∞, $\varepsilon = 0.5$
GA	5	0.5	1.108×10^{-5}	3.559×10^{-5}	2.065×10^{-12}	7.958×10^{-13}	5	1.3×10^{3}	3.559×10^{-5}
	10	1.0	7.482×10^{-10}	4.636×10^{-9}	7.062×10^{-11}	2.547×10^{-11}	10	1.8×10^{5}	2.126×10^{-8}
	15	1.0	2.752×10^{-11}	2.098×10^{-10}	3.744×10^{-9}	8.149×10^{-10}	11	6.1×10^{6}	1.689×10^{-8}
	20	1.0	1.765×10^{-11}	1.352×10^{-10}	6.863×10^{-9}	4.075×10^{-10}	11	4.6×10^{6}	1.862×10^{-8}
MQ	5	0.5	7.033×10^{-6}	2.329×10^{-5}	1.034×10^{-12}	5.969×10^{-13}	5	1.1×10^{3}	2.329×10^{-5}
	10	0.5	3.737×10^{-10}	1.604×10^{-9}	7.644×10^{-10}	2.547×10^{-10}	10	1.5×10^{6}	1.604×10^{-9}
	15	0.5	1.365×10^{-10}	3.182×10^{-10}	1.670×10^{-7}	1.100×10^{-8}	10	7.7×10^{5}	3.182×10^{-10}
	20	0.5	1.593×10^{-10}	3.608×10^{-10}	2.417×10^{-7}	3.922×10^{-8}	10	7.3×10^{5}	3.608×10^{-10}
IMQ	5	0.5	8.018×10^{-6}	1.964×10^{-5}	4.907×10^{-13}	9.948×10^{-14}	5	3.7×10^{2}	1.964×10^{-5}
	10	0.5	1.111×10^{-9}	5.846×10^{-9}	2.274×10^{-10}	1.019×10^{-10}	10	5.8×10^{5}	5.846×10^{-9}
	15	0.5	3.213×10^{-11}	1.681×10^{-10}	1.642×10^{-8}	2.037×10^{-10}	11	4.0×10^{6}	1.681×10^{-10}
	20	0.5	2.321×10^{-11}	1.747×10^{-10}	2.836×10^{-8}	2.445×10^{-9}	11	3.1×10^{6}	1.747×10^{-10}

Figure 11 shows the pointwise absolute error on a logarithmic scale. As in Example 1, the multiquadric error consists of smooth arches, while the Gaussian and inverse multiquadric errors are smaller and irregular. The Gaussian kernel attains the smallest maximum, 1.352×10^{-10}. In panel (b), the zeros of the multiquadric error lie at almost the same positions as in Examples 3 and 4, which use the same kernel, shape parameter, and nodes. This suggests that the shape of this error is set mainly by the discretization rather than by the equation.



(a) (b)

Figure 11. Pointwise error for the GA, MQ, and IMQ kernels applied to Example 5. Panel (a) shows $|y^t - y_{\text{ref}}|$ on a logarithmic scale, and panel (b) shows the signed error $y^t - y_{\text{ref}}$ on a linear scale, in units of the factor shown in each panel, with grey lines at the collocation nodes. The error at $x = 0$ is exactly zero, since the initial value is built into the trial solution, and is not shown in (a).

5.6. Example 6

Consider the steady-state oxygen diffusion equation in a spherical cell [54],

$$y''(x) + \frac{2}{x}\, y'(x) = \frac{\delta\, y(x)}{\kappa + y(x)}, \qquad x \in [0, 1], \tag{5.11}$$

with $\delta = 0.76129$ and $\kappa = 0.03119$. Unlike Examples 1–5, this is a boundary value problem. The symmetry condition $y'(0) = 0$ holds at the origin, and the Robin condition

$$y'(1) + 5y(1) = 5 \tag{5.12}$$

is prescribed at $x = 1$. Because (5.12) couples $y(1)$ and $y'(1)$, the standard trial solution (3.2) cannot be applied directly and is replaced by

$$y^t(x; \mathbf{w}) = a(\mathbf{w}) + x^2\, N(x; \mathbf{w}), \tag{5.13}$$

which satisfies $(y^t)'(0) = 0$ for every $\mathbf{w}$. Substituting (5.13) into (5.12) and solving for a gives

$$a(\mathbf{w}) = 1 - \frac{1}{5}(7\, N(1; \mathbf{w}) + N'(1; \mathbf{w})), \tag{5.14}$$

so the Robin condition holds exactly for every $\mathbf{w}$. Since a now depends on the weights, the first partial derivative of (5.13) acquires a term absent from the initial value case (3.12),

$$\begin{aligned} \frac{\partial y^t}{\partial w_j}(x) &= \frac{\partial a}{\partial w_j} + x^2\, \phi_j(x), \\ \frac{\partial a}{\partial w_j} &= -\frac{1}{5}\Big(7\, \phi_j(1) + \phi_j'(1)\Big), \end{aligned} \tag{5.15}$$

while $\partial(y^t)'/\partial w_j$ and $\partial(y^t)''/\partial w_j$ retain the forms in (3.12), since $a(\mathbf{w})$ is constant in x. The Jacobian (3.11) is assembled with this modified first partial derivative. The system has the same structure as in the initial value problems. There are H residual equations for the M output weights. Row 1 is the limiting form (4.1) with a replaced by $a(\mathbf{w})$, and rows $2, \ldots, H$ are the interior residuals (3.5). The node $x_H = 1$ carries an ordinary collocation equation rather than a boundary condition, since the elimination (5.14) satisfies the Robin condition identically for every $\mathbf{w}$.

The reference solution is obtained with bvp4c on $[x_0, 1]$. The condition $y'(0) = 0$ is there replaced by its linearization from the local expansion of (5.11), imposed at x_0, and the value at the origin is recovered from the same expansion. Imposing $y'(x_0) = 0$ instead is consistent only to $O(x_0^2)$, which would exceed the differences reported here. Solving on $[0, 1]$ itself with the same solver gives the same reference to within 1.1×10^{-16}, and both computations give $y(0) = 0.828483290359689$.

Table 9 shows that the three kernels give tightly clustered results. The small shape parameters selected for the multiquadric and inverse multiquadric kernels are consistent with the slowly varying solution. This is the only problem in which the offset a depends on the weights, so the limiting residual (4.1) is nonlinear in $\mathbf{w}$, and the Jacobian acquires the extra term of (4.2). The small values of $|R(0; \mathbf{w})|$ confirm that the singular endpoint is handled by the same construction as in the initial value problems.

Table 9. Error metrics for Example 6, with $M = H = 20$.

Method	ε	**MAE**	L_∞	$\|\mathbf{R}\|_\infty$	$\|R(0;\mathbf{w})\|$	**Iter.**	**Halv.**	**Time** (ms)	**Total** (ms)
GA-RBF-ELM	0.5	1.810×10^{-10}	5.164×10^{-10}	1.541×10^{-9}	5.254×10^{-10}	3	14/48	1.41	18.5
MQ-RBF-ELM	0.1	1.258×10^{-10}	2.943×10^{-10}	4.249×10^{-9}	1.755×10^{-9}	4	5/39	2.85	14.9
IMQ-RBF-ELM	0.1	6.936×10^{-11}	1.354×10^{-10}	1.191×10^{-8}	9.082×10^{-9}	4	5/39	4.28	21.8

Figure 12 compares the three kernel solutions with the bvp4c reference.

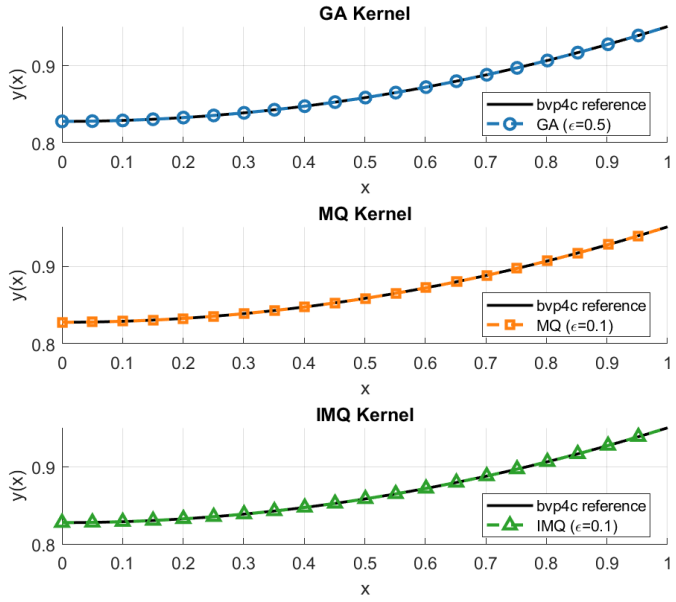
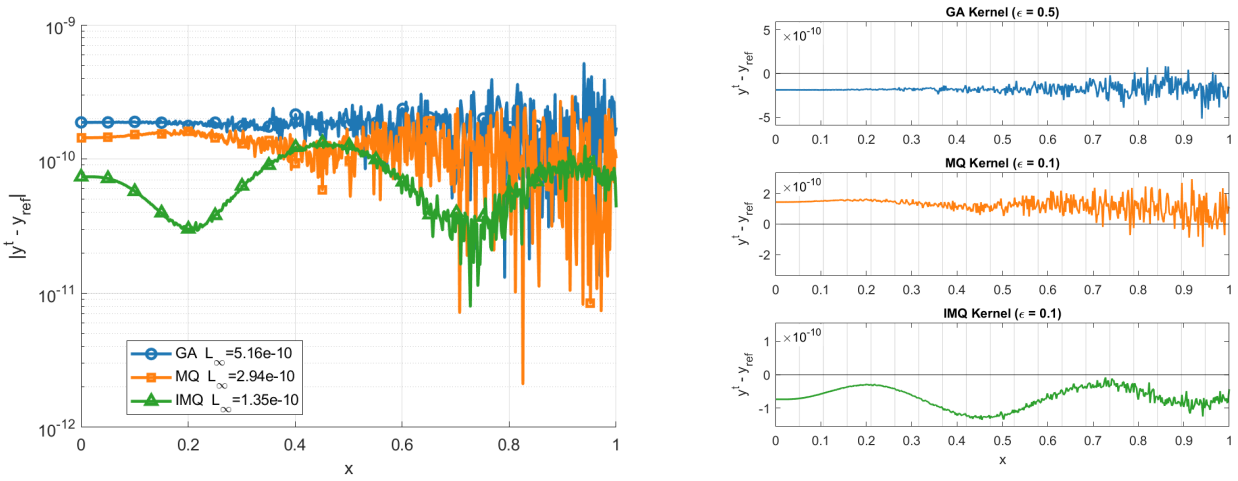


Figure 12. Approximate solutions obtained by the GA, MQ, and IMQ kernels compared with the bvp4c reference solution for Example 6.

Figure 13 shows the pointwise absolute error on a logarithmic scale. The inverse multiquadric kernel attains the smallest maximum. Here, the error does not vanish at $x = 0$, since $y'(0) = a(\mathbf{w})$ is determined by the Robin condition rather than prescribed. Panel (b) shows this offset for all three kernels.



(a) (b)

Figure 13. Pointwise error for the GA, MQ, and IMQ kernels applied to Example 6. Panel (a) shows $|y^t - y_{\text{ref}}|$ on a logarithmic scale, and panel (b) shows the signed error $y^t - y_{\text{ref}}$ on a linear scale, in units of the factor shown in each panel, with grey lines at the collocation nodes.

5.7. Shape-parameter sensitivity, conditioning, and effective rank

Table 10 shows how the maximum error depends on the shape parameter. The pattern is the accuracy-stability trade-off that is characteristic of radial basis approximation. The error is large at both ends of the tested range and smallest at an intermediate value. Small ε makes the basis nearly linearly dependent, whereas large ε localizes the Gaussian and inverse multiquadric kernels and, for the multiquadric kernel, makes the network output N nearly piecewise linear between the centers (Section 3.5). In both cases, this comes at the expense of the approximation between the nodes. The location of the optimum depends on the problem as well as on the kernel, so no single value can be recommended in advance. Since only five candidates are examined, this pattern describes the tested values and is not a claim of unimodality in ε on $(0, \infty)$. The value selected by the off-node residual criterion is shown in bold. It is the error-minimizing candidate in 10 of the 18 cases and an adjacent candidate otherwise, and its error is at most 2.05 times the smallest error in the row (last column of Table 10). The selection uses the off-node residual as a proxy for the error. The stability estimate (4.3), which bounds the error by the residual up to the constant C_s, is available only for the linear initial value problems, so the factor 2.05 is an empirical observation. Larger values were also tested. For $\varepsilon = 10$, 20, and 50, the product of ε and the node spacing $1/19$ is at least 0.5. The collocation matrices are then well conditioned (condition numbers between 4×10^2 and 4×10^5, full rank), but the trial space is too poor between the nodes. The Gaussian and inverse multiquadric kernels are too localized for the node spacing, and the second derivative of the multiquadric kernel is concentrated near the centers. The off-node residual rises to between 4×10^{-2} and 4×10^3 and the maximum error to between 3×10^{-4} and 1. Such values are rejected by the selection criterion. The limiting factor at large ε is therefore approximation rather than conditioning, in line with the trade-off between error and conditioning [55]. The search over five candidates is a deliberately simple choice, and it cannot locate a global optimum. A finer candidate set could identify a smaller off-node residual, but this need not

yield a smaller solution error. For the present candidate set, the selected solution error is at most 2.05 times the smallest tested error. More systematic strategies from radial basis function interpolation, such as leave-one-out cross-validation computed with Rippa's algorithm [56] and the criteria compared by Fasshauer and Zhang [57], could replace the grid search once adapted to residual-based collocation. This adaptation is left to future work.

Table 10. Maximum error L_∞ as a function of the shape parameter, for each kernel and benchmark problem.

Problem	**Kernel**	$\varepsilon = 0.1$	0.5	1	2	5	**Ratio**
Example 1	GA	5.625×10^{-9}	$\mathbf{2.175 \times 10^{-10}}$	1.062×10^{-10}	2.878×10^{-10}	5.069×10^{-6}	2.05
	MQ	2.313×10^{-10}	$\mathbf{2.701 \times 10^{-10}}$	2.793×10^{-10}	1.260×10^{-7}	4.807×10^{-5}	1.17
	IMQ	2.230×10^{-10}	$\mathbf{1.090 \times 10^{-11}}$	2.350×10^{-10}	3.907×10^{-7}	2.654×10^{-3}	1.00
Example 2	GA	3.810×10^{-6}	2.686×10^{-9}	$\mathbf{8.362 \times 10^{-11}}$	1.449×10^{-10}	3.508×10^{-6}	1.00
	MQ	4.732×10^{-7}	$\mathbf{7.160 \times 10^{-11}}$	2.834×10^{-10}	1.078×10^{-7}	5.038×10^{-5}	1.00
	IMQ	2.766×10^{-7}	$\mathbf{2.431 \times 10^{-11}}$	2.348×10^{-11}	1.375×10^{-7}	1.755×10^{-3}	1.04
Example 3	GA	1.061×10^{-5}	1.794×10^{-8}	$\mathbf{2.266 \times 10^{-10}}$	3.290×10^{-10}	3.721×10^{-6}	1.00
	MQ	1.569×10^{-6}	$\mathbf{7.853 \times 10^{-10}}$	4.323×10^{-10}	3.429×10^{-8}	1.329×10^{-5}	1.82
	IMQ	4.932×10^{-7}	$\mathbf{3.114 \times 10^{-10}}$	1.071×10^{-9}	7.594×10^{-7}	4.030×10^{-3}	1.00
Example 4	GA	7.943×10^{-6}	1.247×10^{-8}	$\mathbf{1.431 \times 10^{-9}}$	1.024×10^{-9}	9.934×10^{-7}	1.40
	MQ	1.008×10^{-6}	$\mathbf{7.047 \times 10^{-9}}$	3.789×10^{-9}	2.165×10^{-7}	4.390×10^{-5}	1.86
	IMQ	2.065×10^{-7}	$\mathbf{3.753 \times 10^{-9}}$	8.689×10^{-9}	6.854×10^{-7}	6.424×10^{-4}	1.00
Example 5	GA	6.876×10^{-5}	1.862×10^{-8}	$\mathbf{1.352 \times 10^{-10}}$	3.543×10^{-10}	1.354×10^{-5}	1.00
	MQ	7.567×10^{-6}	$\mathbf{3.608 \times 10^{-10}}$	8.297×10^{-10}	5.008×10^{-7}	3.416×10^{-4}	1.00
	IMQ	3.381×10^{-6}	$\mathbf{1.747 \times 10^{-10}}$	1.937×10^{-9}	2.373×10^{-7}	8.926×10^{-3}	1.00
Example 6	GA	4.494×10^{-9}	$\mathbf{5.164 \times 10^{-10}}$	3.847×10^{-10}	3.964×10^{-9}	4.913×10^{-5}	1.34
	MQ	$\mathbf{2.943 \times 10^{-10}}$	2.128×10^{-10}	1.812×10^{-9}	8.834×10^{-7}	5.414×10^{-5}	1.38
	IMQ	$\mathbf{1.354 \times 10^{-10}}$	2.931×10^{-10}	4.662×10^{-9}	4.103×10^{-6}	2.835×10^{-3}	1.00

Across the benchmarks, the raw condition number at the selected shape parameter lies between 2.9×10^{17} and 1.7×10^{19}, beyond the double-precision limit for a direct solver. The ratio $\sigma_{\max}/\sigma_r$, where σ_r is the r-th singular value and r the rank used by the solver, is three to five orders of magnitude smaller, between 2.7×10^{13} and 1.2×10^{15}. The raw value therefore comes from singular values beyond the retained rank. This ratio is the condition number of the best rank-r approximation of the matrix. Since the solver truncates by column-pivoted QR rather than by singular values, the ratio indicates the conditioning of the retained problem only approximately. The effective numerical rank never exceeds 11 of the $M = 20$ directions and falls as low as 6, so the basis is substantially redundant. The coefficient norms are correspondingly large, between 1.1×10^4 and 1.0×10^7, and no bounded weight norm is claimed. Of the factors in the bound Λ_c for the computed solution (Section 4.2), only $\sqrt{M}\,\|\mathbf{w}_c\|_2$ varies with the discretization at fixed ε, and at $M = 20$, it lies between 5.0×10^4 and 4.6×10^7. Tables 2, 4, and 8

report its values under refinement. From $M = 15$ on, it does not increase in any of the nine kernel-example pairs, but over the same range, the retained rank no longer follows M and stays between 6 and 12. The leveling off therefore reflects the saturation of the computed subspace and provides no evidence about the limit $h \to 0$. Whether this factor stays bounded as $h \to 0$ remains open. The computed values are large, yet the residual of the computed solution is small between the nodes. At the selected parameters, the off-node residual lies between 2×10^{-9} and 1×10^{-5}. This suggests that the sufficient condition of Proposition 4.3 is pessimistic. However, H^τ norms have not been computed, and the gap between this condition and the observations remains open. Table 11 examines the dependence on the rank threshold, which Section 4.2 does not predict. In this table, r is the rank used by the solver and σ_r is the r-th singular value. The default rows give the value of the solver's default tolerance relative to $\sigma_{\max}$.

Table 11. Sensitivity of the computed solution to the rank tolerance for Example 1 at the selected shape parameters.

Kernel	τ_{rel}	r	σ_r	$\|\mathbf{w}\|_2$	L_∞	$\|\mathbf{R}\|_\infty$	iter
	10^{-8}	5	6.499×10^{-5}	5.046×10^{1}	4.332×10^{-8}	8.868×10^{-6}	1
	10^{-10}	6	5.694×10^{-7}	5.122×10^{1}	4.340×10^{-8}	7.937×10^{-6}	2
	10^{-12}	7	3.880×10^{-9}	6.221×10^{3}	1.402×10^{-10}	6.705×10^{-8}	2
GA	10^{-14}	8	2.122×10^{-11}	6.749×10^{3}	1.115×10^{-10}	4.494×10^{-8}	1
	10^{-15}	9	9.603×10^{-14}	1.160×10^{6}	2.175×10^{-10}	8.001×10^{-10}	3
	default (6.2×10^{-16})	9	9.603×10^{-14}	1.160×10^{6}	2.175×10^{-10}	8.001×10^{-10}	3
	10^{-16}	9	9.603×10^{-14}	1.160×10^{6}	2.175×10^{-10}	8.001×10^{-10}	3
	10^{-8}	6	3.861×10^{-6}	1.321×10^{2}	3.324×10^{-7}	6.675×10^{-5}	3
	10^{-10}	7	1.502×10^{-7}	1.321×10^{3}	2.246×10^{-8}	9.850×10^{-6}	1
	10^{-12}	9	1.684×10^{-10}	4.080×10^{4}	4.807×10^{-10}	3.003×10^{-7}	3
MQ	10^{-14}	10	5.005×10^{-12}	1.305×10^{5}	1.503×10^{-10}	8.960×10^{-8}	3
	default (1.1×10^{-15})	11	1.630×10^{-13}	1.350×10^{6}	2.701×10^{-10}	9.155×10^{-9}	1
	10^{-15}	11	1.630×10^{-13}	1.350×10^{6}	2.701×10^{-10}	9.155×10^{-9}	1
	10^{-16}	11	1.630×10^{-13}	1.350×10^{6}	2.701×10^{-10}	9.155×10^{-9}	1
	10^{-8}	7	1.130×10^{-6}	5.318×10^{1}	9.794×10^{-9}	3.874×10^{-6}	3
	10^{-10}	8	4.503×10^{-8}	1.925×10^{2}	1.344×10^{-9}	6.945×10^{-7}	2
	10^{-12}	9	1.763×10^{-9}	8.131×10^{2}	2.978×10^{-10}	1.540×10^{-7}	7
IMQ	10^{-14}	11	2.195×10^{-12}	1.112×10^{4}	1.090×10^{-11}	6.297×10^{-9}	1
	10^{-15}	11	2.195×10^{-12}	1.112×10^{4}	1.090×10^{-11}	6.297×10^{-9}	1
	default (5.9×10^{-16})	11	2.195×10^{-12}	1.112×10^{4}	1.090×10^{-11}	6.297×10^{-9}	1
	10^{-16}	12	6.885×10^{-14}	2.273×10^{5}	5.537×10^{-11}	3.599×10^{-10}	1

Tightening the tolerance from 10^{-8} adds directions, increases the coefficient norm, and, at first, reduces the error, so these directions carry solution content. The reduction stops when the error has fallen to about $\epsilon_{\text{mach}}\|\mathbf{w}\|_2$, where $\epsilon_{\text{mach}} \approx 2.2\times10^{-16}$. This is the size of the rounding errors to be expected when large coefficients of opposite sign are combined. This scale is a heuristic. The standard bound

for evaluating $\sum_j w_j \phi_j(x)$ in floating point is about $M\epsilon_{\text{mach}} \sum_j |w_j||\phi_j(x)|$, and the error in the computed $\mathbf{w}$ itself adds to this. Hence, $\epsilon_{\text{mach}}\|\mathbf{w}\|_2$ indicates only the order of magnitude. For the Gaussian kernel, $r = 8$ at $\tau_{\text{rel}} = 10^{-14}$ gives $L_\infty = 1.1 \times 10^{-10}$, about seventy times $\epsilon_{\text{mach}}\|\mathbf{w}\|_2$. The ninth direction, retained at the default, raises $\|\mathbf{w}\|_2$ from 6.7×10^3 to 1.2×10^6 and the error to 2.2×10^{-10}, which is $0.84\,\epsilon_{\text{mach}}\|\mathbf{w}\|_2$. The multiquadric kernel behaves in the same way between $r = 10$ and $r = 11$, and the inverse multiquadric kernel between $r = 11$ and $r = 12$ at $\tau_{\text{rel}} = 10^{-16}$. At every default row, the ratio $L_\infty/(\epsilon_{\text{mach}}\|\mathbf{w}\|_2)$ lies between 0.84 and 4.4. The default threshold is therefore within a factor of two of the most accurate tested value, and tightening it further gives essentially the same or a less accurate result. The same study for Examples 2 and 6 shows the same behavior, with retained ranks between 3 and 12 and a ratio between 0.66 and 3.6 at the default.

The same balance is consistent with the error floor in the refinement tables. Here, the floor denotes the regime that begins at about $M = 10$ to 15 and in which the error no longer decreases systematically with M. For $M \geq 15$ the maximum error of each kernel varies within a factor of about three in Tables 2, 4, and 8, without a trend. In this regime, the error is close to $\epsilon_{\text{mach}}\|\mathbf{w}_c\|_2$. At $M = H = 20$, the ratio $L_\infty/(\epsilon_{\text{mach}}\|\mathbf{w}_c\|_2)$ lies between 0.6 and 4.4 in 17 of the 18 kernel-example pairs and is 9.9 in the remaining one. By contrast, L_∞ itself ranges over almost three orders of magnitude, from 1.1×10^{-11} to 7.0×10^{-9}. For $M \geq 15$ in the refinement tables, the ratio lies between 0.4 and 9.9. Together with Table 11, this is consistent with a finite-precision floor. The ratio is an empirical diagnostic, not a proven bound. Retaining fewer directions lowers $\|\mathbf{w}_c\|_2$ but leaves a truncation error well above $\epsilon_{\text{mach}}\|\mathbf{w}_c\|_2$, and retaining more raises $\|\mathbf{w}_c\|_2$ and the error with it. The conditioning of the collocation matrix enters through this balance by limiting the number of directions that can be retained and fixing the size of the resulting coefficients. The reference solutions are unlikely to be the cause, since the independent solvers agree to about 10^{-13}. At fixed $\varepsilon = 0.5$, the Gaussian error levels off at higher values in Examples 2 and 5, about 3×10^{-9} and 2×10^{-8}, respectively, with only eight or nine directions retained. There, truncation rather than rounding sets the level. In all three refinement tables, the error at fixed ε falls by about three to four orders of magnitude between $M = 5$ and $M = 10$. This is consistent with a rapidly decaying $\eta(h)$ in Assumption A2. However, the floor is reached before the asymptotic regime, so the decay of $\eta(h)$ cannot be measured in double precision.

Tables 1 and 3 are complemented by a comparison with alternative formulations and solver choices. Examples 1 and 2 were solved at $M = H = 20$ with the kernel, centers, and nodes unchanged. Table 12 lists the maximum error of the proposed method and, for each alternative, the ratio

$$\rho = L_\infty^{\text{alternative}} / L_\infty^{\text{proposed}},$$

calculated from the unrounded errors. A value $\rho < 1$ indicates a smaller error for the alternative. These comparisons do not isolate the individual ingredients. In the first alternative, the initial conditions are imposed as two additional least-squares equations with unit, untuned weights, the equation at $x = 0$ is dropped, and the shape parameter is reselected by the same criterion. The maximum error then becomes 7 to 835 times larger, but since the equation set changes as well as the trial space, the increase cannot be attributed to the embedding alone. Replacing the row at $x = 0$ by the residual at a displaced first node $x_1 = 10^{-3}$ or $x_1 = 10^{-6}$ gives ratios between 0.65 and 1.13, 9 of the 12 below one. These comparisons show no systematic accuracy advantage from retaining the origin row. What it provides is consistency, since it includes the exact limiting form (4.1) as a collocation equation, without introducing an offset parameter. Solving without rank truncation gives values of ρ between 10 and 185.

Table 12. Maximum error of the proposed method and error ratio ρ of each alternative, $M = H = 20$.

			Ratio ρ to the proposed method			
Example	Kernel	L_∞, proposed	Not embedded	$x_1 = 10^{-3}$	$x_1 = 10^{-6}$	No truncation
1	GA	2.2×10^{-10}	7.1	0.81	0.77	10
	MQ	2.7×10^{-10}	187	1.13	0.91	13
	IMQ	1.1×10^{-11}	835	0.99	0.97	11
2	GA	8.4×10^{-11}	379	0.95	0.65	33
	MQ	7.2×10^{-11}	24	1.02	1.13	130
	IMQ	2.4×10^{-11}	19	0.82	0.70	185

6. Conclusions

This paper introduces the RBF-ELM method for singular second-order problems of Lane-Emden-Fowler type. The method is a singularity-regularized radial basis function network trained within an extreme learning machine framework. The trial solution satisfies the prescribed initial or Robin conditions exactly for every realization of the weight vector, and the loss is the differential equation residual alone. The central analytical result is that the x^2 construction renders the residual continuous at the origin. The limiting equation (4.1) may therefore be imposed as the first collocation equation, and no displaced node enters the formulation. A conditional convergence result is established for linear initial value problems, together with a sufficient condition on the coefficient norm for the residual-regularity hypothesis (Proposition 4.3). Whether this condition holds uniformly under refinement remains open.

The six benchmarks span linear, nonlinear, homogeneous, nonhomogeneous, initial value, and boundary value formulations. For linear equations, the weights follow from a single minimum-norm least-squares computation with residual correction, and for nonlinear equations from a damped Newton iteration with a fully analytic Jacobian. The accuracy is of the same order in both cases. The most accurate kernel is problem-dependent, and the shape parameter strongly influences accuracy. Under refinement, the error decreases rapidly and then levels off at a value close to $\epsilon_{\text{mach}}\|\mathbf{w}_c\|_2$, which is consistent with a finite-precision floor set by rank truncation and rounding. This behavior occurs for linear, polynomial-nonlinear, and transcendental-nonlinear problems alike, which indicates that the floor is a property of the discretization rather than of a particular equation. At the selected shape parameters, the collocation matrices are numerically rank deficient. The reported accuracy is therefore attained on a retained subspace whose dependence on the rank threshold is quantified rather than assumed away. The Newton iteration usually terminates by stagnation, that is, without an acceptable step, rather than by reaching the residual tolerance. The Robin boundary value problem shows that the same construction applies outside the initial value setting, although the convergence result is not claimed for it. The comparisons of Section 5 do not isolate the individual ingredients. They show that rank truncation strongly affects the accuracy and that imposing the initial conditions as additional equations with unit, untuned weights was markedly less accurate than embedding them. Retaining the limiting row gives no systematic accuracy advantage. Its role is to include the exact limiting form of the equation at the singular endpoint as a collocation equation, without an offset parameter.

The convergence result does not extend to the nonlinear case, for the reasons stated in Section 4.2. The bound it provides does not certify the accuracy actually attained, because the computed coefficient norms are large. The shape parameter is selected using the off-node residual as a proxy for solution error. A theoretically justified restriction of the search interval could reduce the search cost; cross-validation strategies adapted to residual-based collocation provide another direction for investigation. Stable formulations such as RBF-QR [58] offer a possible route past the conditioning barrier. Radially symmetric problems in d dimensions lead to the same operator with $k = d - 1$. Coupled systems and fractional or delay variants are possible extensions that require further analysis. For nonlocal operators in particular, it is not clear that the residual of the trial solution remains regular at the origin.

Data and code availability

No external data are used, and all results are deterministic. The MATLAB code that generates every table and figure is available from the author on reasonable request.

Use of Generative-AI tools declaration

The author used an AI assistant (Claude, Anthropic) for language editing.

Conflict of interest

The author declares no conflict of interest in this paper.

References

1. J. H. Lane, On the theoretical temperature of the sun, *Am. J. Sci. Arts*, **50** (1870), 57–74. https://doi.org/10.2475/ajs.s2-50.148.57
2. R. Emden, *Gaskugeln: Anwendungen der mechanischen Wärmetheorie auf kosmologische und meteorologische Probleme*, Leipzig und Berlin: B. G. Teubner, 1907. https://doi.org/10.1007/BF01736734
3. S. Chandrasekhar, *An introduction to the study of stellar structure*, New York: Dover Publications, 1957.
4. H. T. Davis, *Introduction to nonlinear differential and integral equations*, Washington: US Government Printing Office, 1961. https://doi.org/10.1017/S0008439500027120
5. S. H. Lin, Oxygen diffusion in a spherical cell with nonlinear oxygen uptake kinetics, *J. Theor. Biol.*, **60** (1976), 449–457. https://doi.org/10.1016/0022-5193(76)90071-0
6. R. D. Russell, L. F. Shampine, Numerical methods for singular boundary value problems, *SIAM J. Numer. Anal.*, **12** (1975), 13–36. https://doi.org/10.1137/0712002
7. F. R. de Hoog, R. Weiss, Collocation methods for singular boundary value problems, *SIAM J. Numer. Anal.*, **15** (1978), 198–217. https://doi.org/10.1137/0715013
8. L. F. Shampine, Singular boundary value problems for ODEs, *Appl. Math. Comput.*, **138** (2003), 99–112. https://doi.org/10.1016/S0096-3003(02)00111-X

9. J. P. Boyd, Chebyshev spectral methods and the Lane-Emden problem, *Numer. Math. Theory Methods Appl.*, **4** (2011), 142–157. https://doi.org/10.4208/nmtma.2011.42s.2
10. A. M. Wazwaz, A new algorithm for solving differential equations of Lane-Emden type, *Appl. Math. Comput.*, **118** (2001), 287–310. https://doi.org/10.1016/S0096-3003(99)00223-4
11. A. M. Wazwaz, Adomian decomposition method for a reliable treatment of the Emden-Fowler equation, *Appl. Math. Comput.*, **161** (2005), 543–560. https://doi.org/10.1016/j.amc.2003.12.048
12. A. M. Wazwaz, R. Rach, J. S. Duan, A study on the systems of the Volterra integral forms of the Lane-Emden equations by the Adomian decomposition method, *Math. Methods Appl. Sci.*, **37** (2014), 10–19. https://doi.org/10.1002/mma.2776
13. M. S. H. Chowdhury, I. Hashim, Solutions of a class of singular second-order IVPs by homotopy-perturbation method, *Phys. Lett. A*, **365** (2007), 439–447. https://doi.org/10.1016/j.physleta.2007.02.002
14. M. S. H. Chowdhury, I. Hashim, Solutions of Emden-Fowler equations by homotopy-perturbation method, *Nonlinear Anal. Real World Appl.*, **10** (2009), 104–115. https://doi.org/10.1016/j.nonrwa.2007.08.017
15. A. S. Bataineh, M. S. M. Noorani, I. Hashim, Homotopy analysis method for singular IVPs of Emden-Fowler type, *Commun. Nonlinear Sci. Numer. Simul.*, **14** (2009), 1121–1131. https://doi.org/10.1016/j.cnsns.2008.02.004
16. H. Demir, İ. Ç. Süngü, Numerical solution of a class of nonlinear Emden-Fowler equations by using differential transform method, *Çankaya Univ. J. Sci. Eng.*, **12** (2009), 75–82.
17. A. H. Bhrawy, A. S. Alofi, A Jacobi-Gauss collocation method for solving nonlinear Lane-Emden type equations, *Commun. Nonlinear Sci. Numer. Simul.*, **17** (2012), 62–70. https://doi.org/10.1016/j.cnsns.2011.04.025
18. R. Singh, V. Guleria, M. Singh, Haar wavelet quasilinearization method for numerical solution of Emden-Fowler type equations, *Math. Comput. Simul.*, **174** (2020), 123–133. https://doi.org/10.1016/j.matcom.2020.02.004
19. S. Gümgüm, Taylor wavelet solution of linear and nonlinear Lane-Emden equations, *Appl. Numer. Math.*, **158** (2020), 44–53. https://doi.org/10.1016/j.apnum.2020.07.019
20. C. S. Liu, E. R. El-Zahar, C. W. Chang, A boundary shape function iterative method for solving nonlinear singular boundary value problems, *Math. Comput. Simul.*, **187** (2021), 614–629. https://doi.org/10.1016/j.matcom.2021.03.030
21. I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Netw.*, **9** (1998), 987–1000. https://doi.org/10.1109/72.712178
22. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045
23. B. Zhang, G. Wu, Y. Gu, X. Wang, F. Wang, Multi-domain physics-informed neural network for solving forward and inverse problems of steady-state heat conduction in multilayer media, *Phys. Fluids*, **34** (2022), 116116. https://doi.org/10.1063/5.0116038

24. X. Huang, F. Wang, B. Zhang, H. Liu, Enriched physics-informed neural networks for dynamic Poisson-Nernst-Planck systems, *Math. Comput. Simul.*, **237** (2025), 231–246. https://doi.org/10.1016/j.matcom.2025.04.037
25. F. Wang, X. Li, H. Liu, L. Qiu, X. Yue, An adaptive method of fundamental solutions using physics-informed neural networks, *Eng. Anal. Bound. Elem.*, **178** (2025), 106295. https://doi.org/10.1016/j.enganabound.2025.106295
26. G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: theory and applications, *Neurocomputing*, **70** (2006), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126
27. V. Dwivedi, B. Srinivasan, Physics informed extreme learning machine (PIELM)—a rapid method for the numerical solution of partial differential equations, *Neurocomputing*, **391** (2020), 96–118. https://doi.org/10.1016/j.neucom.2019.12.099
28. F. Calabrò, G. Fabiani, C. Siettos, Extreme learning machine collocation for the numerical solution of elliptic PDEs with sharp gradients, *Comput. Methods Appl. Mech. Eng.*, **387** (2021), 114188. https://doi.org/10.1016/j.cma.2021.114188
29. S. Dong, Z. Li, Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations, *Comput. Methods Appl. Mech. Eng.*, **387** (2021), 114129. https://doi.org/10.1016/j.cma.2021.114129
30. S. Dong, J. Yang, On computing the hyperparameter of extreme learning machines: algorithm and application to computational PDEs, and comparison with classical and high-order finite elements, *J. Comput. Phys.*, **463** (2022), 111290. https://doi.org/10.1016/j.jcp.2022.111290
31. E. Schiassi, R. Furfaro, C. Leake, M. De Florio, H. Johnston, D. Mortari, Extreme theory of functional connections: a fast physics-informed neural network method for solving ordinary and partial differential equations, *Neurocomputing*, **457** (2021), 334–356. https://doi.org/10.1016/j.neucom.2021.06.015
32. G. Fabiani, F. Calabrò, L. Russo, C. Siettos, Numerical solution and bifurcation analysis of nonlinear partial differential equations with extreme learning machines, *J. Sci. Comput.*, **89** (2021), 44. https://doi.org/10.1007/s10915-021-01650-5
33. G. Fabiani, E. Galaris, L. Russo, C. Siettos, Parsimonious physics-informed random projection neural networks for initial value problems of ODEs and index-1 DAEs, *Chaos*, **33** (2023), 043128. https://doi.org/10.1063/5.0135903
34. M. Liu, W. Peng, M. Hou, Z. Tian, Radial basis function neural network with extreme learning machine algorithm for solving ordinary differential equations, *Soft Comput.*, **27** (2023), 3955–3964. https://doi.org/10.1007/s00500-022-07529-3
35. S. Mall, S. Chakraverty, Chebyshev neural network based model for solving Lane-Emden type equations, *Appl. Math. Comput.*, **247** (2014), 100–114. https://doi.org/10.1016/j.amc.2014.08.085
36. S. Mall, S. Chakraverty, Numerical solution of nonlinear singular initial value problems of Emden-Fowler type using Chebyshev neural network method, *Neurocomputing*, **149** (2015), 975–982. https://doi.org/10.1016/j.neucom.2014.07.036

37. S. Chakraverty, S. Mall, Single layer Chebyshev neural network model with regression-based weights for solving nonlinear ordinary differential equations, *Evol. Intell.*, **13** (2020), 687–694. https://doi.org/10.1007/s12065-020-00383-y

38. S. Mall, S. Chakraverty, Application of Legendre neural network for solving ordinary differential equations, *Appl. Soft Comput.*, **43** (2016), 347–356. https://doi.org/10.1016/j.asoc.2015.10.069

39. Y. Yang, M. Hou, J. Luo, A novel improved extreme learning machine algorithm in solving ordinary differential equations by Legendre neural network methods, *Adv. Differ. Equ.*, **2018** (2018), 469. https://doi.org/10.1186/s13662-018-1927-x

40. Y. Yang, Y. Wu, M. Hou, J. Luo, X. Xie, Solving Emden-Fowler equations using improved extreme learning machine algorithm based on block Legendre basis neural network, *Neural Process. Lett.*, **55** (2023), 7135–7154. https://doi.org/10.1007/s11063-023-11254-9

41. A. H. Hadian-Rasanan, D. Rahmati, S. Gorgin, K. Parand, A single layer fractional orthogonal neural network for solving various types of Lane-Emden equation, *New Astron.*, **75** (2020), 101307. https://doi.org/10.1016/j.newast.2019.101307

42. A. Verma, M. Kumar, Numerical solution of Lane-Emden type equations using multilayer perceptron neural network method, *Int. J. Appl. Comput. Math.*, **5** (2019), 141. https://doi.org/10.1007/s40819-019-0728-6

43. S. N. Kurmanç, M. Özavşar, Single-layer Laguerre neural network model for solving Lane-Emden-Fowler type equations, *Math. Commun.*, **30** (2025), 243–258. https://doi.org/10.64785/mc.30.2.8

44. Z. Zheng, H. Yuan, J. He, A physics-informed neural network model combined Pell-Lucas polynomials for solving the Lane-Emden type equation, *Eur. Phys. J. Plus*, **139** (2024), 223. https://doi.org/10.1140/epjp/s13360-024-04999-2

45. M. Babaei, H. Dana Mazraeh, K. Parand, MB-PIRNN: matrix-based physics-informed recurrent neural network for solving Lane-Emden type equation, *Soft Comput.*, **30** (2026), 4427–4446. https://doi.org/10.1007/s00500-025-10966-5

46. H. Baty, Modelling Lane-Emden type equations using physics-informed neural networks, *Astron. Comput.*, **44** (2023), 100734. https://doi.org/10.1016/j.ascom.2023.100734

47. N. Mai-Duy, T. Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, *Neural Netw.*, **14** (2001), 185–199. https://doi.org/10.1016/S0893-6080(00)00095-2

48. K. Parand, S. Hashemi, RBF-DQ method for solving non-linear differential equations of Lane-Emden type, *Ain Shams Eng. J.*, **9** (2018), 615–629. https://doi.org/10.1016/j.asej.2016.03.010

49. H. Wendland, *Scattered data approximation*, Cambridge: Cambridge University Press, 2005. https://doi.org/10.1017/CBO9780511617539

50. R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.*, **76** (1971), 1905–1915. https://doi.org/10.1029/JB076i008p01905

51. C. A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constr. Approx.*, **2** (1986), 11–22. https://doi.org/10.1007/BF01893414

52. F. J. Narcowich, J. D. Ward, H. Wendland, Sobolev error estimates and a Bernstein inequality for scattered data interpolation via radial basis functions, *Constr. Approx.*, **24** (2006), 175–186. https://doi.org/10.1007/s00365-005-0624-7

53. R. A. Adams, J. J. F. Fournier, *Sobolev spaces*, Amsterdam: Academic Press / Elsevier, 2003.

54. A. M. Wazwaz, The variational iteration method for solving nonlinear singular boundary value problems arising in various physical models, *Commun. Nonlinear Sci. Numer. Simul.*, **16** (2011), 3881–3886. https://doi.org/10.1016/j.cnsns.2011.02.026

55. R. Schaback, Error estimates and condition numbers for radial basis function interpolation, *Adv. Comput. Math.*, **3** (1995), 251–264. https://doi.org/10.1007/BF02432002

56. S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, *Adv. Comput. Math.*, **11** (1999), 193–210. https://doi.org/10.1023/A:1018975909870

57. G. E. Fasshauer, J. G. Zhang, On choosing "optimal" shape parameters for RBF approximation, *Numer. Algorithms*, **45** (2007), 345–368. https://doi.org/10.1007/s11075-007-9072-8

58. B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, *SIAM J. Sci. Comput.*, **33** (2011), 869–892. https://doi.org/10.1137/09076756X



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)