Research Article

Exponentially Fitted Element-Free Galerkin Approach for Nonlinear Singularly Perturbed Problems

Jagbir Kaur and Vivek Sangwan

Thapar Institute of Engineering and Technology, Patiala 147004, Punjab, India

Correspondence should be addressed to Vivek Sangwan; sangwan.vivek@gmail.com

Received 7 September 2021; Accepted 11 November 2021; Published 1 December 2021

Academic Editor: M. M. Bhatti

Copyright © 2021 Jagbir Kaur and Vivek Sangwan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As it is well recognized that conventional numerical schemes are inefficient in approximating the solutions of the singularly perturbed problems (SPP) in the boundary layer region, in the present work, an effort has been made to propose a robust and efficient numerical approach known as element-free Galerkin (EFG) technique to capture these solutions with a high precision of accuracy. Since a lot of weight functions exist in the literature which plays a crucial role in the moving least square (MLS) approximations for generating the shape functions and hence affect the accuracy of the numerical solution, in the present work, due emphasis has been given to propose a robust weight function for the element-free Galerkin scheme for SPP. The key feature of nonrequirement of elements or node connectivity of the EFG method has also been utilized by proposing a way to generate nonuniformly distributed nodes. In order to verify the computational consistency and robustness of the proposed scheme, a variety of linear and nonlinear numerical examples have been considered and L_{∞} errors have been presented. Comparison of the EFG solutions with those available in the literature depicts the superiority of the proposed scheme.

1. Introduction

Singularly perturbed problems (SPP) and their numerical solutions have attained great attention from researchers from the past few decades. Pearson [1] is credited with being the first researcher to solve the SPP numerically using traditional three-point finite-difference techniques on nonuniform meshes. The problem of nonsmoothness in the solution was firstly overcome by utilizing layer adaptive mesh methodologies. Bakhvalov [2] was the first who proposed special type of meshes, known as Bakhvalov meshes, to capture the boundary layers found in reaction-diffusion problems' solutions, and later on, these meshes were used and modified by Gartland [3] and others for convection-diffusion problems. Later on, Shishkin [4] introduced another special type of meshes, called Shishkin meshes, to generate parameter-uniform numerical schemes under the finite-difference framework. Because of the simple structure of the Shishkin meshes, these meshes were utilized by many other researchers also for

generating parameter-uniform numerical schemes for singularly perturbed problems. With the passage of time, other numerical schemes, called fitted operator schemes, were also developed by mathematicians and other researchers for solving singularly perturbed differential equations. For instance, Roos et al. [5] proposed an exponentially fitted finite element method for solving singularly perturbed problems.

Recently, considering the benefits and advantages of mesh-free methods over other numerical schemes, researchers have paid considerable attention to the developments of mesh-free methods. To name a few, the diffuse element method by Nayroles et al. [6], the EFG method by Belytschko et al. [7], reproducing kernel particle method (RKPM) by Liu et al. [8], the local Petrov-Galerkin method by Atluri and Zhu [9] etc., are some very famous mesh-free methods. However, for singularly perturbed problems, rarely, any mesh-free method has been proposed. Therefore, the main motive of the present work is to propose a suitable mesh-free method for SPP.

Geng and Qian [10] employed the reproducing kernel method for obtaining numerical solutions of singularly perturbed turning point problems having dual boundary layers. The authors [11] devised a modified reproducing kernel method (RKM) to solve singularly perturbed delay boundary value problems later in 2015. The suggested approach was based on the reproducing kernel theory, and the method's error estimates were also discussed. Nadjafi and Ghassabzade [12] proposed another MLS-based meshless technique to deal with singularly perturbed delay differential equations (DDEs). Another meshless approach, namely, the radial basis collocation method with the coordinate stretching procedure was presented by Akhavan Ghassabzade et al. [13] for the mathematical treatment of singularly perturbed DDEs.

In this study, we have investigated the following non-linear singularly perturbed problem:

$$\epsilon v'' = f(v', v, x), \quad x \in \Omega = (a, b), \quad (1)$$

under essential boundary conditions defined by

$$\begin{aligned} v(a) &= v_a, \\ v(b) &= v_b, \quad v_a, v_b \in \mathbb{R}, \end{aligned} \quad (2)$$

where ϵ is a small positive parameter known as singular perturbation parameter with $0 < \epsilon \ll 1$, and we suppose that $f(v', v, x)$ is continuously differentiable function. These types of problems appear in almost all the fields of engineering and science, e.g., plasma dynamics, biochemistry, electric power, aerospace systems, geophysics, fluid mechanics, and heat transport problems [14].

In the present work, we have proposed the element-free Galerkin (EFG) method, one of the most popular and versatile mesh-free methods, for approximating nonlinear SPP solutions. The EFG technique is based on the theory of moving least squares (MLS) approximation and standard weight functions to construct the shape functions. The method employs background cells to compute numerical integrations and is based on the global weak form. Since the Kronecker delta condition is not satisfied by the MLS shape function approximants, as a result, the imposition of essential boundary conditions is not straightforward. To overcome this problem, Lagrange multiplier approach [7] has been utilized. Looking on the bright side, the EFG technique is one among the very efficient techniques for approximating the solutions of nonlinear differential equations. Due to the absence of element connectivity, adaptive refinement of the node particles can be achieved very efficiently and easily. These features make the EFG method more adaptable than the finite element method (FEM). On the contrary, the EFG technique still has a few downsides also in comparison to FEM. The EFG method is computationally more expensive and has a more complex implementation algorithm than FEM. Since the method has not been applied to SPP so far, therefore, after analyzing various weight functions in the present work, efficient weight functions have been proposed for SPP.

The paper is organized as follows.

The MLS approximation and construction of shape functions have been discussed in Section 2. Different weight functions and their properties have been presented in Section 3. In Section 4, the quasi-linearization technique and its convergence have been presented. Section 5 describes the element-free Galerkin (EFG) technique. Section 6 contains the algorithm of the proposed method and numerical results, followed by a conclusion and bibliography.

2. Moving Least Squares Approximation

The field function $v(x)$ is approximated by $v^h(x)$ using MLS approximants in the domain Ω . These approximants contain three components: basis functions which are usually polynomials, weight functions that are related with nodes present in the influence domain of a particular node, and the set of coefficients that depend on the position of nodes. The domain of influence of any nodal point x_I is defined as a local neighborhood of x_I or the domain that the particular node x_I exerts an influence upon. The MLS's approximation of the unknown function $v^h(x)$ is defined as follows:

$$v^h(x) = \sum_{i=1}^m c_i(x) y_i(x) = \mathbf{c}^T(x) \mathbf{y}(x), \quad (3)$$

where $\mathbf{c}(x)$ is a vector of monomial basis functions, m is the number of components in the basis vector $\mathbf{c}$, and $\mathbf{y}(x)$ is an unknown vector of coefficients of the basis functions.

The choice of basis functions $\mathbf{c}(x)$ is essential in the construction of moving least squares based shape functions. The following are the examples of the basic functions:

(i) Linear basis:

$$\begin{aligned} \mathbf{c}^T(x) &= (1, x), \quad \text{in } 1D, \\ \mathbf{c}^T(x, y) &= (1, x, y), \quad \text{in } 2D. \end{aligned} \quad (4)$$

(ii) Quadratic basis:

$$\begin{aligned} \mathbf{c}^T(x) &= (1, x, x^2), \quad \text{in } 1D, \\ \mathbf{c}^T(x, y) &= (1, x, y, x^2, xy, y^2), \quad \text{in } 2D. \end{aligned} \quad (5)$$

The field function's approximate value at any particular node x_I in the support domain of x is defined by

$$v^h(x, x_I) = \mathbf{c}^T(x_I) \mathbf{y}(x). \quad (6)$$

We define the weighted residual functional as follows to determine the unknown coefficient vector $\mathbf{y}(x)$ in the support domain of the point of interest x :

$$J = \sum_{I=1}^n w(x - x_I) [\mathbf{c}^T(x_I) \mathbf{y}(x) - v_I]^2, \quad (7)$$

where n denotes the number of nodes x_I in the support domain of the point of interest x . The nodes in the support domain of any point x are used to support or approximate

the function value at x . The dimension d_s of the support domain of the point x is defined by

$$d_s = \alpha_s d_c, \quad (8)$$

where α_s signifies the dimensionless size of the support domain and the characteristic length d_c denotes the nodal spacing for the point x . $w(x - x_I)$ in expression (7) is some weight function which is generally chosen as monotonically decreasing as $|x - x_I|$ increases. In the literature, numerous weight functions have been introduced and employed by researchers, as mentioned in Section 3. In MLS approximation, the condition $m > n$ generally leads to the ill-conditioned system of equations.

On simplification, equation (7) can be rewritten as

$$J = (\mathbf{C}\mathbf{y} - \mathbf{v})^T \mathbf{W} (\mathbf{C}\mathbf{y} - \mathbf{v}), \quad (9)$$

where

$$\mathbf{v}^T(x) = (v_1, v_2, v_3, \dots, v_n),$$

$$\mathbf{C} = \begin{bmatrix} c_1(\mathbf{x}_1) & c_1(\mathbf{x}_2) & \dots & c_1(\mathbf{x}_n) \\ c_2(\mathbf{x}_1) & c_2(\mathbf{x}_2) & \dots & c_2(\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ c_n(\mathbf{x}_1) & c_n(\mathbf{x}_2) & \dots & c_n(\mathbf{x}_n) \end{bmatrix}, \quad (10)$$

and $\mathbf{W}$ is the diagonal matrix defined as

$$\mathbf{W} = \begin{bmatrix} w(\mathbf{x} - \mathbf{x}_1) & 0 & \dots & 0 \\ 0 & w(\mathbf{x} - \mathbf{x}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w(\mathbf{x} - \mathbf{x}_n) \end{bmatrix}. \quad (11)$$

The minimization of J with respect to $\mathbf{y}(x)$ leads to

$$\frac{\partial J}{\partial \mathbf{y}} = 0. \quad (12)$$

This gives

$$\mathbf{y}(x) = \mathbf{P}^{-1}(x) \mathbf{S}(x) \mathbf{v}, \quad (13)$$

where

$$\mathbf{P}(x) = \sum_{I=1}^n w(x - x_I) \mathbf{c}(x_I) \mathbf{c}^T(x_I),$$

$$\mathbf{S}(x) = [w(x - x_1) \mathbf{c}(x_1), w(x - x_2) \mathbf{c}(x_2), \dots, w(x - x_n) \mathbf{c}(x_n)]. \quad (14)$$

Therefore, from (3), the approximated field function v^h can be computed as

$$v^h(x) = \sum_{I=1}^n \sum_{j=1}^m c_j(x) (\mathbf{P}^{-1}(x) \mathbf{S}(x))_{jI} v_I$$

$$= \sum_{I=1}^n \phi_I(x) v_I, \quad (15)$$

where $\phi_I(x)$, $I = 1, 2, 3, \dots, n$, are the shape functions defined by

$$\phi_I(x) = \sum_{j=1}^m c_j(x) (\mathbf{P}^{-1}(x) \mathbf{S}(x))_{jI} = \mathbf{c}^T \mathbf{P}^{-1} \mathbf{S}_I. \quad (16)$$

The graphs of typical shape functions for different nodes have been plotted in Figure 1. We can observe that the shape functions are bell-shaped. The derivatives of the MLS shape functions can be obtained by differentiating (16) with respect to the spatial coordinates.

3. Choice of Weight Functions

Weight function corresponding to any node provides weights for the residuals at different nodes within the support domain of the node of interest, and it ensures that nodes should enter and leave the support domain smoothly so that the shape functions satisfy the compatibility condition and the approximation is globally continuous. The weight functions must be continuous and positive in the corresponding support domains. The weight functions should satisfy the following properties:

- (i) $w(x - x_I) > 0$ in the support domain Ω_x of the node x
- (ii) $w(x - x_I) = 0$ outside the support domain Ω_x
- (iii) $w(|x - x_I|)$ is a monotonically decreasing function
- (iv) $w(|x - x_I|) \rightarrow \delta(|x - x_I|)$ as $h \rightarrow 0$

$\delta(|x - x_I|)$ represents the Dirac delta function and h is measure of the size of the support domain Ω_x . The weight functions are usually based on a normalized distance $s = (|x - x_I|/d_s)$, where d_s denotes the radius of the influence domain. In [15–23], there are variety of weight functions, which are widely used in literature, are defined below. Though these weight functions are utilized by many researchers, here we have referenced only a few of them.

- (i) Quadratic spline: the weight function of a quadratic spline is defined as follows:

$$w(s) = \begin{cases} 1 - s^2, & 0 \leq s \leq 1, \\ 0, & s > 1. \end{cases} \quad (17)$$

Singh [15] compared different weight functions (exponential, elliptical, cosine, and quadratic) under the EFG method for heat transfer problems and found that elliptical, quadratic, and cosine weight functions provide acceptable results for the support domain with radius $d_s = \alpha_s * d_c$, where the scaling parameter d_c satisfies $0 < d_c < 1.5$, whereas exponential weight functions provide good results for the scaling parameter $1 < d_c < 2.0$. In overall comparison, the author found that the exponential weight functions provide much better results in comparison to the other weight functions (i.e., elliptical, cosine, and quadratic) for heat transfer problems.

- (ii) Cubic spline: the weight function of a cubic spline is defined by

FIGURE 1: Shape functions at different nodes.

$$w(s) = \begin{cases} \frac{2}{3} - 4s^2 + 4s^3, & s \leq \frac{1}{2}, \\ \frac{4}{3} - 4s + 4s^2 - \frac{4}{3}s^3, & \frac{1}{2} < s \leq 1, \\ 0, & s > 1. \end{cases} \quad (18)$$

Cubic spline weight functions were firstly proposed by Dolbow and Belytschko [16] in the EFG method. The authors constructed cubic spline weight functions to impose C^2 continuity requirements while solving linear elastostatic problems. These are some of the most frequent weight functions which are widely adopted by many researchers while employing the element-free Galerkin methodology, e.g., Park and Leap [17] utilized the EFG method with cubic spline weight functions for solving groundwater flow problems. Yang and He [18] utilized these weight functions in the EFG method for heat transfer problems. Juan et al. [19] used cubic spline weight functions to solve the topology optimization problem of the continuum structures. Zhang et al. [24] also applied the EFG method with cubic spline weight functions for two-dimensional elastodynamics problems. Rajesh Sharma [25] also successfully applied these weight functions along with the EFG method to study the heat transfer fluid flow problems over a stretching sheet.

- (iii) Quartic spline: the quartic spline weight function is given by

$$w(s) = \begin{cases} 1 - 6s^2 + 8s^3 - 3s^4, & s \leq 1, \\ 0, & s > 1. \end{cases} \quad (19)$$

These weight functions are also widely used by researchers as they provide C^2 continuity. Shibata and Murakami [26] presented a stability procedure for soil-water coupled problems using the EFG method with quartic spline weight function. He et al. [27] established the hybrid EFG method and the scaled boundary method known as the EFG scaled boundary method. The authors employed quartic spline weight function in EFG formulation to carry out numerical results. Jaberzadeh et al. [28] also considered quartic weight function in EFG methodology to study thermal buckling of trapezoidal plates. Cheng et al. demonstrated improved the complex variable EFG method for advection-diffusion [29]. The impact of cubic and quartic spline weight functions on computational precision has been discussed in both the articles. The proposed method has higher computational accuracy while employing quartic spline weight functions. The quartic spline weight functions in the EFG technique have been adopted by various other researchers also for solving different types of problems such as nonlinear shallow water equations [20] and static and buckling analysis of the thin plate [30].

- (iv) Gaussian exponential spline weight function: the Gaussian exponential spline weight function is defined as

$$w(s) = \begin{cases} \frac{e^{-(s/\alpha)^2} - e^{-(1/\alpha)^2}}{1 - e^{-(1/\alpha)^2}}, & s \leq 1, \\ 0, & s > 1. \end{cases} \quad (20)$$

Here, the parameter α is generally chosen from the interval $[0.3, 0.5]$. The Gaussian weighting function

of exponential type is one of the most commonly used weight functions. These weight functions have been initially utilized by Belytschko et al. to solve elasticity and heat conduction problems [7], crack propagation problems [31], and fracture problems [23]. Following these, many researchers employed Gaussian exponential weight functions for solving various types of other problems such as fractional cable equation [32], quasi-three-dimensional free vibration analysis of multilayered composite plates [33], and bimaterial problems [34]. The performance of Gaussian weight functions has been analyzed by comparison with other weight functions in [35, 36].

- (v) Conical weight function: conical weight function is defined as

$$w(s) = \begin{cases} 1 - s^{2k}, & s \leq 1, \\ 0, & s > 1. \end{cases} \quad (21)$$

Liu et al. [37] performed EFG formulation with conical weight function to obtain the numerical solutions of 3D electromagnetic problems. The authors compared the performance of different weight functions (cubic spline, rational formula, conical formula, negative exponential, and quartic spline) by correlating the EFG method results with the exact one. The graphical results indicate that error estimation of conical formula increases by increasing the value of scaling parameter d_c . Zhuang et al. [35] highlighted issues and sources of errors for meshless methods. The control of errors has been shown by considering three types of weight functions, i.e., a rational function, an exponential function, and a conical function in the element-free Galerkin methodology. Graphical results of curve fitting problems show that the conical weight function exhibits lower error in comparison to the exponential weight function. The steady heat transfer analysis for orthotropic structure has been carried out by Zhang et al. [21] based on the proposed EFG technique. Different weight functions such as cubic spline, parabola function, conical function, triangle weight function, and exponential weight function have been considered. The numerical results suggest preferring cubic spline and parabola weight functions in comparison to the conical weight function for the heat transfer problems.

- (vi) Cosine weight function: the cosine weight function is defined as

$$w(s) = \begin{cases} \cos\left(\frac{\pi s}{2}\right), & 0 \leq s \leq 1. \\ 0, & s > 1. \end{cases} \quad (22)$$

Though only a few researchers have utilized the cosine weight functions in the element-free

Galerkin formulation, these weight functions provide C^1 continuous approximations. Bouillard and Suleau [22] successfully applied these weight functions for numerical assessment of the pollution effect governed by the Helmholtz model problem. After that, Singh [15] also mentioned cosine weight function along with some other weight functions for EFG formulation to solve three-dimensional heat transfer problems.

- (vii) Hyperbolic and elliptical weight functions: hyperbolic and elliptical weight functions are given by

$$w(s) = \begin{cases} \text{sech}(s + 3), & 0 \leq s \leq 1, \\ 0, & s > 1, \end{cases} \quad (23)$$

$$w(s) = \begin{cases} 2\sqrt{1-s^2}, & 0 \leq s \leq 1, \\ 0, & s > 1, \end{cases}$$

respectively. Elliptical weight functions have been referenced by Singh in [15]. The author also considered exponential and cosine weight functions to carry out error and steady-state analysis for 3D heat transfer problems under the EFG framework.

The shapes of different weight functions at any point x in the domain have been presented in Figure 2.

4. Quasi-Linearization Technique

Since the problem under consideration is nonlinear in nature, and quasi-linearization is a standard technique that is widely used for approximating the nonlinear problems by their linearized version for obtaining the approximate solutions; in this section, we will discuss the quasi-linearization process and its convergence for the problem under consideration, i.e., equation (1).

Using the quasi-linearization process by Bellman and Kalaba [38], equation (1) can be linearized as follows:

$$\epsilon v''^{(k+1)} = f\left(v'^{(k)}, v^{(k)}, x\right) + \left(v'^{(k+1)} - v'^{(k)}\right) \frac{\partial f}{\partial v'}\left(v'^{(k)}, v^{(k)}, x\right) + \left(v^{(k+1)} - v^{(k)}\right) \frac{\partial f}{\partial v}\left(v'^{(k)}, v^{(k)}, x\right), \quad (24)$$

where $v^{(k+1)}$ stands for the approximate solution at $(k+1)$ th iteration level. For example, the nonlinear problem 3 in Section 6 can be linearized by applying the quasi-linearization process (24) as below.

The considered nonlinear differential equation in Example 3, given by

$$\epsilon v'' = -e^v - 2v', \quad (25)$$

is linearized as

$$\epsilon v''^{(k+1)} + 2v'^{(k+1)} + e^{v^{(k)}} v^{(k+1)} = e^{v^{(k)}} v^{(k)} - e^{v^{(k)}}. \quad (26)$$

Similarly, the nonlinear singularly perturbed differential equation in Example 4,

FIGURE 2: Shapes of different weight functions for x .

$$\epsilon v'' = -v v' + v, \quad (27)$$

is linearized as

$$\epsilon v''^{(k+1)} + v^{(k)} v'^{(k+1)} + v'^{(k)} v^{(k+1)} - v^{(k+1)} = v^{(k)} v'^{(k)}. \quad (28)$$

4.1. Convergence of Quasi-Linearization Process. In order to prove the convergence of the quasi-linearization process, we consider

$$\begin{aligned} \epsilon v'' &= f(v), \quad a < x < b, \\ v(a) &= v_a, \\ v(b) &= v_b. \end{aligned} \quad (29)$$

By using quasi-linearization process, the following recurrence relation provides a sequence of linear differential equations as

$$\begin{aligned} \epsilon v''^{(k+1)} &= f(v^{(k)}) + (v^{(k+1)} - v^{(k)}) \\ &\quad \cdot \frac{\partial f(v^{(k)})}{\partial v}, \quad a < x < b, \end{aligned} \quad (30)$$

$$\begin{aligned} v^{(k+1)}(a) &= v_a, \\ v^{(k+1)}(b) &= v_b, \end{aligned} \quad (31)$$

where (k) denotes the iteration level. We assume $v^{(k=0)}$ as the initial guess which also incorporates the boundary conditions. Rewriting the above relation at the previous iteration step, we obtain

$$\epsilon v''^{(k)} = f(v^{(k-1)}) + (v^{(k)} - v^{(k-1)}) \frac{\partial f(v^{(k-1)})}{\partial v}, \quad a < x < b. \quad (32)$$

Subtracting equation (32) from equation (30), we obtain

$$\begin{aligned} \epsilon (v''^{(k+1)} - v''^{(k)}) &= f(v^{(k)}) - f(v^{(k-1)}) + (v^{(k+1)} - v^{(k)}) \\ &\quad \frac{\partial f(v^{(k)})}{\partial v} - (v^{(k)} - v^{(k-1)}) \frac{\partial f(v^{(k-1)})}{\partial v}, \quad a < x < b. \end{aligned} \quad (33)$$

The above differential equation is of second order in $(v^{(k+1)} - v^{(k)})$. Next, with the help of Green's function, equation (33) can be converted into an integral equation as

$$\begin{aligned} \epsilon (v^{(k+1)} - v^{(k)}) &= \int_a^b G(x, r) \left[f(v^{(k)}) - f(v^{(k-1)}) \right. \\ &\quad \left. + (v^{(k+1)} - v^{(k)}) \frac{\partial f(v^{(k)})}{\partial v} \right. \\ &\quad \left. - (v^{(k)} - v^{(k-1)}) \frac{\partial f(v^{(k-1)})}{\partial v} \right] ds, \quad a < x < b, \end{aligned} \quad (34)$$

where Green's function $G(x, r)$ is defined as

$$G(x, r) = \begin{cases} x(r-1), & 0 \leq x \leq r \leq 1, \\ (x-1)r, & 0 \leq r \leq x \leq 1, \end{cases} \quad (35)$$

$$\max_{x,r} G(x, r) = \frac{1}{4}. \quad (36)$$

Now, by using mean value theorem, we obtain

$$f(v^{(k)}) - f(v^{(k-1)}) = (v^{(k)} - v^{(k-1)}) \frac{\partial f(v^{(k-1)})}{\partial v} + \frac{(v^{(k)} - v^{(k-1)})^2}{2} \frac{\partial^2 f(\theta)}{\partial v^2}, \quad (37)$$

where $v^{(k-1)} < \theta < v^{(k)}$.

Next, using equation (37) in (34), we get the following expression:

$$\epsilon(v^{(k+1)} - v^{(k)}) = \int_a^b G(x, r) \left[\frac{(v^{(k)} - v^{(k-1)})^2}{2} \frac{\partial^2 f(\theta)}{\partial v^2} + (v^{(k+1)} - v^{(k)}) \frac{\partial f(v^{(k)})}{\partial v} \right] ds, \quad a < x < b. \quad (38)$$

Suppose

$$\begin{aligned} \max_{|v^{(k-1)}| \leq 1} \left| \frac{\partial^2 f(\theta)}{\partial v^2} \right| &= L, \\ \max_{|v^{(k)}| \leq 1} \left| \frac{\partial f(v^{(k)})}{\partial v} \right| &= R. \end{aligned} \quad (39)$$

Using maximum norm over the spatial domain and using equations (36) and (39) in (38), we obtain

$$\|v^{(k+1)} - v^{(k)}\| \leq \frac{1}{4\epsilon} \int_a^b \left[\frac{L}{2} (v^{(k)} - v^{(k-1)})^2 + R \|v^{(k+1)} - v^{(k)}\| \right] ds. \quad (40)$$

Simplifying the above inequality, we obtain

$$\begin{aligned} \|v^{(k+1)} - v^{(k)}\|_{\Omega} &\leq \frac{L}{8\epsilon - 2R} \|v^{(k)} - v^{(k-1)}\|_{\Omega}^2 \\ &\leq Q \|v^{(k)} - v^{(k-1)}\|_{\Omega}^2, \end{aligned} \quad (41)$$

where $Q = L/(8\epsilon - 2R)$.

Thus, the quasi-linearization process converges quadratically by choosing appropriate initial iterative approximation $v^{(k=a)}$. Therefore, in order to find the solution of the nonlinear problem (1), it is sufficient to find the solution of its approximate linearized problem (24). Next, we will discuss the proposed numerical strategy to approximate the solution of the linearized problem (24).

5. Element-Free Galerkin Formulation

In this section, element-free Galerkin formulation has been discussed in detail.

5.1. Node Generation. The node generation has been done in a way so that more points can be put in the boundary layer region. As discussed by Roos et al. [5], when the singular perturbation parameter ϵ tends to 0, boundary layers appear in the solution, and we need to adopt some special technique to capture these boundary layers. In the present work, we have utilized Shishkin's approach to generate more nodes in the boundary layer region and to capture these layers nicely. For boundary layers at $x = a$, the distribution of nodes is carried out by dividing the spatial domain $\bar{\Omega} = [a, b]$ into two subintervals $[a, a + \delta]$ and $[a + \delta, b]$, where the transition parameter δ is chosen as

$$\delta = \min\{0.5, M\epsilon \ln N\}, \quad (42)$$

where M is a constant chosen appropriately and $(N + 1)$ denotes the total number of nodes in the spatial domain. In each of these subintervals, the nodes are generated as follows.

We define

$$x_i = \begin{cases} i * h_i, & i = 0, 1, \dots, \frac{N}{2}, \\ \delta + \left(i - \frac{N}{2}\right) * h_i, & i = \frac{N}{2} + 1, \dots, N, \end{cases} \quad (43)$$

where the mesh spacings h_i 's are given by

$$h_i = \begin{cases} \frac{2\delta}{N}, & i = 0, 1, \dots, \frac{N}{2}, \\ \frac{2(1-\delta)}{N}, & i = \frac{N}{2} + 1, \dots, N. \end{cases} \quad (44)$$

For 32 and 64 number of nodes and $\epsilon = 2^{-5}$ and $\epsilon = 2^{-10}$, the node distribution is given in Figure 3. It can be easily

observed that nodes condense near $x = a$ as ϵ becomes smaller. A similar procedure can be carried out for boundary layers appearing at $x = 1$.

Remark 1. Note that here Shishkin's idea is used only to generate nodes and not the mesh as the present method does not require element's connectivity.

Next, we will discuss the element-free Galerkin weak formulation of (24).

5.2. Weak Formulation. Using the above discussed moving least squares methodology, the element-free Galerkin (EFG) weak formulation for the linearized problem (24) is given by

$$\begin{aligned} & -\epsilon \int_{\Omega} \phi_i v''^{(k+1)} d\Omega - \int_{\Omega} \frac{\partial f}{\partial v'}(v'^{(k)}, v^{(k)}, x) \phi_i v'^{(k+1)} d\Omega - \int_{\Omega} \frac{\partial f}{\partial v}(v'^{(k)}, v^{(k)}, x) \phi_i v^{(k+1)} d\Omega \\ & -\delta \lambda (v - v_a)|_{x=a} - \delta v \lambda|_{x=a} - \delta \rho (v - v_b)|_{x=b} - \delta v \rho|_{x=b} = \int_{\Omega} \phi_i f(x) d\Omega \\ & - \int_{\Omega} \frac{\partial f}{\partial v'}(v'^{(k)}, v^{(k)}, x) \phi_i v'^{(k)} d\Omega \\ & - \int_{\Omega} \frac{\partial f}{\partial v}(v'^{(k)}, v^{(k)}, x) \phi_i v^{(k)} d\Omega, \end{aligned} \quad (45)$$

where ϕ_i 's are the test functions generated by moving least square approach as discussed in Section 2. The last terms in equation (45) are introduced due to the method of Lagrange

multipliers to enforce the essential boundary conditions (2). Substituting the EFG approximation of field function $v(x)$ and simplifying equation (45), we obtain

$$\begin{aligned} & \sum_j \left[-\epsilon \int_{\Omega} \phi'_i \phi_j'^{(k+1)} d\Omega - \int_{\Omega} \frac{\partial f}{\partial v'}(v'^{(k)}, v^{(k)}, x) \phi_i \phi_j'^{(k+1)} d\Omega \right. \\ & \quad \left. - \int_{\Omega} \frac{\partial f}{\partial v}(v'^{(k)}, v^{(k)}, x) \phi_i \phi_j^{(k+1)} d\Omega \right] v_j^{(k+1)} \\ & + \sum_l \sum_j \delta \lambda_l (-N_l \phi_j v_j) \Big|_{x=a} - \sum_l \delta \lambda_l (-N_l v_a) \Big|_{x=a} \\ & + \sum_l \sum_j \delta v_j (-\phi_j N_l) \lambda_l \Big|_{x=a} + \sum_l \sum_j \delta \rho_l (-R_l \phi_j) v_j \Big|_{x=b} \\ & - \sum_l \delta \rho_l (-R_l v_b) \Big|_{x=b} + \sum_l \sum_j \delta v_j (-\phi_j R_l) \rho_l \Big|_{x=b} \\ & = \left[\int_{\Omega} \phi_i f(x) d\Omega - \int_{\Omega} \frac{\partial f}{\partial v'}(v'^{(k)}, v^{(k)}, x) \phi_i \phi_j'^{(k)} d\Omega \right. \\ & \quad \left. - \int_{\Omega} \frac{\partial f}{\partial v}((v'^{(k)}, v^{(k)}, x)) v^{(k)} \phi_i \phi_j^{(k)} d\Omega + \epsilon \phi_i(b) \frac{dv}{dx} \Big|_{x=b} - \epsilon \phi_i(a) \frac{dv}{dx} \Big|_{x=a} \right]. \end{aligned} \quad (46)$$

FIGURE 3: Node distribution for $\epsilon = 2^{-5}$ and 2^{-10} .

On solving the above nodal weak formulations and assembling, the resulted algebraic linear equations can be written in the matrix form as

$$\begin{bmatrix} K & G & H \\ G^T & 0 & 0 \\ H^T & 0 & 0 \end{bmatrix} \begin{bmatrix} V \\ \lambda \\ \rho \end{bmatrix} = \begin{bmatrix} F \\ q \\ q_c \end{bmatrix}, \quad (47)$$

$$K_{ij} = -\epsilon \int_{\Omega} \phi'_i \phi'_j \, d\Omega - \int_{\Omega} \frac{\partial f}{\partial v'}(v'^{(k)}, v^{(k)}, x) \phi_i \phi'_j \, d\Omega - \int_{\Omega} \frac{\partial f}{\partial v}(v'^{(k)}, v^{(k)}, x) \phi_i \phi_j^{(k+1)} \, d\Omega,$$

$$G_{lj}^T = -\delta \lambda_l (N_l \phi_j)|_{x=a},$$

$$H_{lj}^T = -\delta \rho_l (R_l \phi_j)|_{x=b},$$

$$F_i = \int_{\Omega} \phi_i f(x) \, d\Omega - \int_{\Omega} \frac{\partial f}{\partial v'}(v'^{(k)}, v^{(k)}, x) \phi_i \phi'_j \, d\Omega - \int_{\Omega} \frac{\partial f}{\partial v}((v'^{(k)}, v^{(k)}, x)) v^{(k)} \phi_i \phi_j^{(k)} \, d\Omega,$$

$$q_l = -\delta \lambda_l (N_l v_a)|_{x=a},$$

$$q_{c_l} = -\delta \rho_l (R_l v_b)|_{x=b}.$$

(48)

$\lambda = \sum_{l=1}^n N_l \lambda_l$ and $\rho = \sum_{l=1}^n R_l \rho_l$ are Lagrange multipliers, and N_l and R_l are shape functions for l^{th} node on the essential boundary.

6. Numerical Results and Discussion

In the present section, before proceeding to numerical examples, firstly, we present below the algorithm of the proposed scheme to help the interested researchers.

6.1. Algorithm for the EFG Scheme. We mention below the step-by-step algorithm for the proposed scheme:

- (1) Enter the given parameter values and material properties.
- (2) Set up the adaptive nodal coordinates in the domain of interest.
- (3) Set up the Gauss quadrature background points (to carry out numerical integrations), weights, and Jacobian.
- (4) Define the support domain of nodal points.
- (5) Set parameters for weight functions and calculate the weight functions for each node.
- (6) Define loop over Gauss integration points and do
 - (a) Construct shape functions and their derivatives at each node.
 - (b) Assemble the contributions from each node to matrix K and the source vector F based on the weak formulation as defined in the last section.
 - (c) Incorporate the essential boundary conditions using the Lagrange multiplier method (G and H matrices defined above) or using the penalty method.
- (7) Solve the global system using some efficient solver to get the nodal parameter values v_i .

Next, we will consider some linear and nonlinear singularly perturbed problems to test the efficiency and robustness of the proposed scheme. In order to demonstrate the potentials of the present approach, the numerical results obtained using the proposed numerical methodology have been compared with those cited in the literature. The pointwise errors and maximum absolute errors have been presented for the considered problems. These errors have been estimated by

$$\begin{aligned} e_{e,N}^i &= |v(x_i) - v_N(x_i)|, \\ E_{e,N} &= \max_i e_{e,N}^i. \end{aligned} \quad (49)$$

The accuracy of the proposed method significantly depends on the choice of the size (radius) of the support domain. Therefore, a lot of experiments have been carried out to find the range of the size of the support domain for singularly perturbed problems. It has been found that, for $2.8 \leq \alpha_c \leq 3.4$, the EFGM provides better numerical approximation for SPP. Computations have been performed using **MATLABR2016a** software on an HP machine with Intel Core i3 processor running at 1.70 GHz speed with 4 GB RAM.

Example 1. Consider the following variable coefficients' singularly perturbed problem [39]:

$$\epsilon v'' + (1+x)^2 v' + 2(1+x)v = f(x), \quad (50)$$

under boundary conditions

$$\begin{aligned} v(a) &= 0, \\ v(b) &= e^{-1/2} - e^{-7/3\epsilon}, \end{aligned} \quad (51)$$

where $f(x)$ is so chosen to satisfy the exact solution:

$$v(x) = e^{-x/2} - e^{-x(x^2+3x+3)/3\epsilon}. \quad (52)$$

The proposed element-free Galerkin scheme has been employed to solve the abovementioned problem. The problem has been solved by considering different weight functions and by using both the linear and the quadratic basis functions. Comparisons have been made for different weight functions in both the cases, and the results have been tabulated in Tables 1 and 2 for 128 number of nodes and different values of singular perturbation parameter ϵ . One can easily observe that the present scheme provides more accurate results by utilizing Gaussian exponential spline and cubic spline weight functions with shape parameter $\alpha = 0.3$ for both linear and quadratic basis functions. Also, as expected, the numerical results are more accurate in the case of quadratic basis functions in comparison to the linear basis functions.

The problem has been solved again for different values of ϵ and the different numbers of nodes and by considering quadratic basis and Gaussian exponential spline weight functions. The L_∞ errors have been presented in Table 3. It can clearly be seen that, for each ϵ , the maximum pointwise absolute errors are decreasing as the number of nodes are increasing. This clearly depicts that the proposed scheme converges numerically. The results obtained from the proposed scheme have also been compared with those cited in the literature in Table 4. It can be noticed that the proposed scheme provides comparative better accurate results. Figure 4 also depicts that the EFG scheme is very much efficient in capturing the sharp boundary layers.

Example 2. For the second example, the following variable coefficients convection-diffusion singularly perturbed problem [39] has been considered:

$$\epsilon v'' + (1+x)^3 v' = f(x), \quad (53)$$

under boundary conditions

$$\begin{aligned} v(0) &= 2, \\ v(1) &= \frac{e^{-15/4\epsilon}}{8} + e^{-1/2}. \end{aligned} \quad (54)$$

TABLE 1: Maximum absolute errors for Example 1 for different values of ϵ with linear basis functions.

ϵ	Quadratic spline	Cubic spline	Quartic spline	Gaussian exp. spline	Cosine function	Conical function	Hyperbolic function	Elliptic function
2^{-2}	1.51×10^{-2}	1.60×10^{-4}	1.54×10^{-2}	3.57×10^{-5}	1.18×10^{-2}	2.01×10^{-2}	1.45×10^{-1}	7.10×10^{-3}
2^{-4}	6.08×10^{-2}	3.48×10^{-3}	5.32×10^{-2}	5.06×10^{-4}	4.10×10^{-2}	9.63×10^{-2}	5.60×10^{-1}	3.45×10^{-2}
2^{-6}	5.85×10^{-2}	5.62×10^{-3}	5.49×10^{-2}	8.49×10^{-4}	5.30×10^{-2}	9.58×10^{-2}	6.83×10^{-1}	4.51×10^{-2}
2^{-8}	9.08×10^{-2}	5.57×10^{-3}	5.23×10^{-2}	8.54×10^{-4}	5.86×10^{-2}	9.73×10^{-2}	6.91×10^{-1}	4.60×10^{-2}
2^{-10}	8.57×10^{-2}	5.58×10^{-3}	5.14×10^{-2}	8.80×10^{-4}	5.46×10^{-2}	9.83×10^{-2}	6.76×10^{-1}	4.57×10^{-2}
2^{-12}	7.41×10^{-2}	5.58×10^{-3}	5.07×10^{-2}	8.60×10^{-4}	5.36×10^{-2}	9.26×10^{-2}	6.74×10^{-1}	4.72×10^{-2}
2^{-14}	7.39×10^{-2}	5.59×10^{-3}	5.08×10^{-2}	8.08×10^{-4}	5.34×10^{-2}	9.18×10^{-2}	6.73×10^{-1}	4.69×10^{-2}
2^{-16}	7.44×10^{-2}	5.60×10^{-3}	5.47×10^{-2}	8.24×10^{-4}	5.33×10^{-2}	9.18×10^{-2}	6.73×10^{-1}	4.64×10^{-2}
2^{-18}	7.48×10^{-2}	5.60×10^{-3}	5.37×10^{-2}	8.33×10^{-4}	5.33×10^{-2}	9.18×10^{-2}	6.73×10^{-1}	4.64×10^{-2}
2^{-20}	7.49×10^{-2}	5.60×10^{-3}	5.34×10^{-2}	8.36×10^{-4}	5.33×10^{-2}	9.18×10^{-2}	6.73×10^{-1}	4.64×10^{-2}
2^{-22}	7.49×10^{-2}	5.60×10^{-3}	5.33×10^{-2}	8.36×10^{-4}	5.33×10^{-2}	9.18×10^{-2}	6.73×10^{-1}	4.64×10^{-2}
2^{-24}	7.49×10^{-2}	5.60×10^{-3}	5.33×10^{-2}	8.36×10^{-4}	5.33×10^{-2}	9.18×10^{-2}	6.73×10^{-1}	4.64×10^{-2}

TABLE 2: Maximum absolute errors for Example 1 for different values of ϵ with quadratic basis functions.

ϵ	Quadratic spline	Cubic spline	Quartic spline	Gaussian exp. spline	Cosine function	Conical function	Hyperbolic function	Elliptic function
2^{-2}	2.27×10^{-2}	1.78×10^{-3}	3.20×10^{-1}	3.35×10^{-4}	5.90×10^{-3}	4.50×10^{-3}	4.02×10^{-2}	1.98×10^{-2}
2^{-4}	9.69×10^{-2}	2.55×10^{-3}	1.47×10^{-1}	9.30×10^{-4}	2.10×10^{-2}	2.07×10^{-2}	1.24×10^{-1}	7.23×10^{-2}
2^{-6}	1.18×10^{-1}	2.83×10^{-3}	3.96×10^{-1}	7.66×10^{-4}	2.73×10^{-2}	2.72×10^{-2}	1.28×10^{-1}	8.28×10^{-2}
2^{-8}	1.18×10^{-1}	1.09×10^{-3}	7.82×10^{-1}	7.09×10^{-4}	2.60×10^{-2}	2.51×10^{-2}	1.27×10^{-1}	8.48×10^{-2}
2^{-10}	1.19×10^{-1}	6.78×10^{-4}	8.18×10^{-1}	7.36×10^{-4}	2.94×10^{-2}	2.48×10^{-2}	1.25×10^{-1}	8.52×10^{-2}
2^{-12}	1.19×10^{-1}	7.03×10^{-4}	8.01×10^{-1}	7.34×10^{-4}	2.32×10^{-2}	2.44×10^{-2}	1.27×10^{-1}	8.05×10^{-2}
2^{-14}	1.19×10^{-1}	7.06×10^{-4}	8.60×10^{-1}	7.34×10^{-4}	2.37×10^{-2}	2.44×10^{-2}	1.27×10^{-1}	8.05×10^{-2}
2^{-16}	1.19×10^{-1}	7.13×10^{-4}	8.00×10^{-1}	7.34×10^{-4}	2.37×10^{-2}	2.43×10^{-2}	1.27×10^{-1}	8.05×10^{-2}
2^{-18}	1.19×10^{-1}	7.22×10^{-4}	7.75×10^{-1}	7.35×10^{-4}	2.37×10^{-2}	2.43×10^{-2}	1.27×10^{-1}	8.05×10^{-2}
2^{-20}	1.19×10^{-1}	7.42×10^{-4}	7.62×10^{-1}	7.35×10^{-4}	2.37×10^{-2}	2.43×10^{-2}	1.27×10^{-1}	8.05×10^{-2}
2^{-22}	1.19×10^{-1}	7.48×10^{-4}	7.84×10^{-1}	7.35×10^{-4}	2.37×10^{-2}	2.43×10^{-2}	1.27×10^{-1}	8.05×10^{-2}
2^{-24}	1.19×10^{-1}	7.50×10^{-4}	7.41×10^{-1}	7.35×10^{-4}	2.37×10^{-2}	2.43×10^{-2}	1.27×10^{-1}	8.05×10^{-2}

TABLE 3: Maximum absolute errors for Example 1 for different values of ϵ .

ϵ	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
2^{-2}	4.40×10^{-3}	2.90×10^{-3}	5.00×10^{-4}	3.35×10^{-4}	1.35×10^{-3}
2^{-4}	7.90×10^{-3}	7.40×10^{-3}	8.00×10^{-4}	9.30×10^{-4}	1.54×10^{-3}
2^{-6}	1.18×10^{-2}	6.70×10^{-3}	1.00×10^{-3}	7.66×10^{-4}	4.87×10^{-4}
2^{-8}	1.76×10^{-2}	6.10×10^{-3}	1.10×10^{-3}	7.09×10^{-4}	2.14×10^{-4}
2^{-10}	2.17×10^{-2}	6.30×10^{-3}	1.10×10^{-3}	7.36×10^{-4}	2.20×10^{-4}
2^{-12}	2.20×10^{-2}	6.70×10^{-3}	1.10×10^{-3}	7.34×10^{-4}	2.22×10^{-4}
2^{-14}	2.20×10^{-2}	6.70×10^{-3}	1.10×10^{-3}	7.34×10^{-4}	2.24×10^{-4}
2^{-16}	2.20×10^{-2}	6.70×10^{-3}	1.10×10^{-3}	7.34×10^{-4}	2.27×10^{-4}

TABLE 4: Comparison of L_{∞} errors for the fitted mesh finite-difference method (FMFDM) [39] and the EFG method for Example 1.

ϵ	$N = 64$		$N = 128$		$N = 256$		$N = 512$	
	FMFDM	EFGM	FMFDM	EFGM	FMFDM	EFGM	FMFDM	EFGM
2^{-4}	0.29×10^{-2}	0.81×10^{-3}	0.73×10^{-3}	0.93×10^{-3}	0.18×10^{-3}	0.15×10^{-3}	0.46×10^{-4}	0.30×10^{-3}
2^{-8}	0.39×10^{-1}	0.11×10^{-2}	0.12×10^{-1}	0.70×10^{-3}	0.38×10^{-2}	0.21×10^{-3}	0.12×10^{-2}	0.19×10^{-2}
10^{-4}	0.38×10^{-1}	0.10×10^{-2}	0.11×10^{-1}	0.73×10^{-3}	0.37×10^{-2}	0.21×10^{-3}	0.12×10^{-2}	0.19×10^{-2}
10^{-5}	0.38×10^{-1}	0.11×10^{-2}	0.11×10^{-1}	0.73×10^{-3}	0.37×10^{-2}	0.22×10^{-3}	0.12×10^{-2}	0.19×10^{-2}
10^{-6}	0.38×10^{-1}	0.11×10^{-2}	0.11×10^{-1}	0.73×10^{-3}	0.37×10^{-2}	0.22×10^{-3}	0.12×10^{-2}	0.19×10^{-2}

FIGURE 4: (a) Exact and EFG solution plot. (b) Pointwise absolute error plot for $\epsilon = 10^{-4}$ and $N = 128$ for Example 1 using linear basis and cubic spline weight functions.

TABLE 5: Maximum absolute errors for Example 2 for different values of ϵ with linear basis functions.

ϵ	Quadratic spline	Cubic spline	Quartic spline	Gaussian exp. spline	Cosine function	Conical function	Hyperbolic function	Elliptic function
2^{-2}	7.14×10^{-1}	4.36×10^{-2}	5.05×10^{-2}	4.37×10^{-2}	5.74×10^{-2}	3.55×10^{-2}	2.71×10^{-1}	3.72×10^{-2}
2^{-4}	2.60×10^{-1}	6.17×10^{-3}	5.55×10^{-2}	7.51×10^{-3}	4.90×10^{-2}	3.61×10^{-2}	4.98×10^{-1}	4.53×10^{-2}
2^{-6}	2.90×10^{-1}	4.38×10^{-3}	4.11×10^{-2}	5.10×10^{-4}	5.95×10^{-2}	3.80×10^{-2}	4.35×10^{-1}	3.55×10^{-2}
2^{-8}	4.41×10^{-1}	4.55×10^{-3}	4.24×10^{-2}	4.24×10^{-4}	4.90×10^{-2}	3.71×10^{-2}	4.18×10^{-1}	3.36×10^{-2}
2^{-10}	4.41×10^{-1}	4.55×10^{-3}	4.38×10^{-2}	5.88×10^{-4}	5.17×10^{-2}	3.86×10^{-2}	4.18×10^{-1}	3.29×10^{-2}
2^{-12}	4.40×10^{-1}	4.55×10^{-3}	4.11×10^{-2}	7.09×10^{-4}	5.31×10^{-2}	3.79×10^{-2}	4.73×10^{-1}	3.26×10^{-2}
2^{-14}	4.40×10^{-1}	4.56×10^{-3}	4.63×10^{-2}	8.13×10^{-4}	5.34×10^{-2}	3.78×10^{-2}	4.21×10^{-1}	3.27×10^{-2}
2^{-16}	4.40×10^{-1}	4.56×10^{-3}	5.85×10^{-2}	8.57×10^{-4}	5.35×10^{-2}	3.78×10^{-2}	4.21×10^{-1}	3.27×10^{-2}
2^{-18}	4.40×10^{-1}	4.56×10^{-3}	5.96×10^{-2}	8.70×10^{-4}	5.35×10^{-2}	3.78×10^{-2}	4.20×10^{-1}	3.27×10^{-2}
2^{-20}	4.40×10^{-1}	4.56×10^{-3}	5.35×10^{-2}	8.74×10^{-4}	5.35×10^{-2}	3.78×10^{-2}	4.20×10^{-1}	3.27×10^{-2}
2^{-22}	4.40×10^{-1}	4.56×10^{-3}	5.91×10^{-2}	8.74×10^{-4}	5.35×10^{-2}	3.78×10^{-2}	4.20×10^{-1}	3.27×10^{-2}
2^{-24}	4.40×10^{-1}	4.56×10^{-3}	5.99×10^{-2}	8.74×10^{-4}	5.35×10^{-2}	3.78×10^{-2}	4.20×10^{-1}	3.27×10^{-2}

TABLE 6: Maximum absolute errors for Example 2 for different values of ϵ with quadratic basis functions.

ϵ	Quadratic spline	Cubic spline	Quartic spline	Gaussian exp. spline	Cosine function	Conical function	Hyperbolic function	Elliptic function
2^{-2}	8.26×10^{-2}	4.33×10^{-2}	2.87×10^{-1}	4.36×10^{-2}	4.60×10^{-2}	4.28×10^{-2}	7.32×10^{-2}	4.41×10^{-2}
2^{-4}	1.88×10^{-1}	7.46×10^{-3}	4.21×10^{-1}	7.80×10^{-3}	2.91×10^{-2}	2.12×10^{-2}	1.73×10^{-1}	1.17×10^{-2}
2^{-6}	1.91×10^{-1}	2.71×10^{-3}	3.43×10^{-1}	8.78×10^{-4}	2.63×10^{-2}	2.32×10^{-2}	1.55×10^{-1}	7.60×10^{-3}
2^{-8}	1.77×10^{-1}	1.07×10^{-3}	6.67×10^{-1}	5.57×10^{-4}	2.37×10^{-2}	2.42×10^{-2}	1.51×10^{-1}	6.10×10^{-3}
2^{-10}	1.61×10^{-1}	7.09×10^{-4}	8.38×10^{-1}	2.74×10^{-4}	2.84×10^{-2}	2.46×10^{-2}	1.50×10^{-1}	5.90×10^{-3}
2^{-12}	1.86×10^{-1}	7.19×10^{-4}	8.37×10^{-1}	2.83×10^{-4}	2.41×10^{-2}	2.45×10^{-2}	1.54×10^{-1}	6.10×10^{-3}
2^{-14}	1.86×10^{-1}	7.23×10^{-4}	8.85×10^{-1}	3.05×10^{-4}	2.39×10^{-2}	2.47×10^{-2}	1.53×10^{-1}	5.90×10^{-3}
2^{-16}	1.86×10^{-1}	7.21×10^{-4}	8.21×10^{-1}	3.02×10^{-4}	2.38×10^{-2}	2.47×10^{-2}	1.53×10^{-1}	6.10×10^{-3}
2^{-18}	1.86×10^{-1}	7.19×10^{-4}	8.15×10^{-1}	3.01×10^{-4}	2.38×10^{-2}	2.47×10^{-2}	1.53×10^{-1}	6.20×10^{-3}
2^{-20}	1.86×10^{-1}	7.14×10^{-4}	8.17×10^{-1}	3.01×10^{-4}	2.38×10^{-2}	2.47×10^{-2}	1.53×10^{-1}	6.20×10^{-3}
2^{-22}	1.86×10^{-1}	7.14×10^{-4}	8.14×10^{-1}	3.01×10^{-4}	2.38×10^{-2}	2.47×10^{-2}	1.53×10^{-1}	6.20×10^{-3}
2^{-24}	1.86×10^{-1}	7.14×10^{-4}	8.20×10^{-1}	3.01×10^{-4}	2.38×10^{-2}	2.47×10^{-2}	1.53×10^{-1}	6.20×10^{-3}

TABLE 7: L_∞ errors for Example 2 for different values of ϵ for quadratic basis and Gaussian weight functions.

ϵ	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
2^{-2}	4.25×10^{-2}	4.31×10^{-2}	4.37×10^{-2}	4.36×10^{-2}	4.22×10^{-2}
2^{-4}	1.45×10^{-2}	8.40×10^{-3}	7.70×10^{-3}	9.80×10^{-3}	6.40×10^{-3}
2^{-6}	1.37×10^{-2}	2.30×10^{-3}	1.40×10^{-3}	8.78×10^{-4}	1.43×10^{-4}
2^{-8}	1.81×10^{-2}	3.40×10^{-2}	7.00×10^{-4}	5.57×10^{-4}	1.30×10^{-4}
2^{-10}	1.92×10^{-2}	3.80×10^{-3}	7.00×10^{-4}	2.74×10^{-4}	2.10×10^{-4}
2^{-12}	1.95×10^{-2}	4.00×10^{-3}	7.00×10^{-4}	2.83×10^{-4}	3.00×10^{-4}
2^{-14}	1.96×10^{-2}	4.00×10^{-3}	7.00×10^{-4}	3.05×10^{-4}	3.00×10^{-4}
2^{-16}	1.96×10^{-2}	4.00×10^{-3}	8.00×10^{-4}	3.02×10^{-4}	3.00×10^{-4}

TABLE 8: Comparison of L_∞ errors for the FMFDM [39] and EFG method for Example 2.

ϵ	$N = 64$		$N = 128$		$N = 256$		$N = 512$	
	FMFDM	EFGM	FMFDM	EFGM	FMFDM	EFGM	FMFDM	EFGM
2^{-8}	0.39×10^{-1}	0.90×10^{-3}	0.12×10^{-1}	0.57×10^{-3}	0.39×10^{-2}	0.13×10^{-3}	0.12×10^{-2}	0.11×10^{-2}
10^{-4}	0.38×10^{-1}	0.80×10^{-3}	0.11×10^{-1}	0.21×10^{-3}	0.37×10^{-2}	0.46×10^{-3}	0.12×10^{-2}	0.42×10^{-3}
10^{-5}	0.38×10^{-1}	0.81×10^{-3}	0.11×10^{-1}	0.21×10^{-3}	0.37×10^{-2}	0.46×10^{-3}	0.12×10^{-2}	0.41×10^{-3}
10^{-6}	0.38×10^{-1}	0.82×10^{-3}	0.11×10^{-1}	0.20×10^{-3}	0.37×10^{-2}	0.47×10^{-3}	0.12×10^{-2}	0.41×10^{-3}
10^{-7}	0.38×10^{-1}	0.82×10^{-3}	0.11×10^{-1}	0.20×10^{-3}	0.37×10^{-2}	0.47×10^{-3}	0.12×10^{-2}	0.38×10^{-3}
10^{-8}	0.38×10^{-1}	0.82×10^{-3}	0.11×10^{-1}	0.20×10^{-3}	0.37×10^{-2}	0.47×10^{-3}	0.12×10^{-2}	0.38×10^{-3}

FIGURE 5: (a) Exact and EFG solution plot. (b) Pointwise absolute error plot for $\epsilon = 2^{-7}$ and $N = 128$ for Example 2 using linear basis and cubic weight functions.

Its exact solution is given by

$$v(x) = \frac{e^{-[(x+1)^4 - 1]/4\epsilon}}{(x+1)^3} + e^{-x/2}. \quad (55)$$

Problem 2 has boundary layers at $x = 0$ for small values of the singular perturbation parameter ϵ . For this problem also, it can be easily noticed in both the cases, i.e., linear basis

functions (Table 5) and quadratic basis functions (Table 6), that the Gaussian exponential spline weight functions provide better results in comparison to the other weight functions for all values of the singular perturbation parameter ϵ . The maximum absolute errors obtained by the proposed method for different values of ϵ and N have been presented in Table 7. From all these tables, one can easily conclude the numerical convergence of the proposed scheme. Comparison of the EFGM results in Table 8 with

TABLE 9: Maximum absolute errors for Example 3 for different values of ϵ with linear basis functions.

ϵ	Quadratic spline	Cubic spline	Quartic spline	Gaussian exp. spline	Cosine function	Conical function	Hyperbolic function	Elliptic function
2^{-2}	3.40×10^{-2}	2.83×10^{-2}	8.09×10^{-2}	2.83×10^{-2}	3.35×10^{-2}	3.03×10^{-2}	1.14×10^{-1}	2.71×10^{-1}
2^{-4}	2.79×10^{-1}	1.88×10^{-2}	7.19×10^{-2}	1.55×10^{-2}	3.15×10^{-2}	1.18×10^{-1}	3.60×10^{-1}	5.40×10^{-1}
2^{-6}	1.03×10^{-1}	6.77×10^{-3}	6.01×10^{-2}	5.72×10^{-3}	8.49×10^{-2}	1.32×10^{-1}	4.14×10^{-1}	5.26×10^{-1}
2^{-8}	1.07×10^{-1}	3.38×10^{-3}	6.26×10^{-2}	1.78×10^{-3}	8.78×10^{-2}	1.35×10^{-1}	4.45×10^{-1}	5.09×10^{-1}
2^{-10}	1.08×10^{-1}	3.39×10^{-3}	6.33×10^{-2}	6.64×10^{-4}	8.87×10^{-2}	1.36×10^{-1}	4.28×10^{-1}	5.04×10^{-1}
2^{-12}	1.08×10^{-1}	3.39×10^{-3}	6.26×10^{-2}	5.53×10^{-4}	8.89×10^{-2}	1.36×10^{-1}	4.41×10^{-1}	5.06×10^{-1}
2^{-14}	1.08×10^{-1}	3.40×10^{-3}	6.29×10^{-2}	5.51×10^{-4}	8.86×10^{-2}	1.36×10^{-1}	3.91×10^{-1}	4.96×10^{-1}
2^{-16}	1.08×10^{-1}	3.41×10^{-3}	6.51×10^{-2}	5.63×10^{-4}	8.84×10^{-2}	1.36×10^{-1}	3.96×10^{-1}	4.99×10^{-1}
2^{-18}	1.08×10^{-1}	3.41×10^{-3}	6.42×10^{-2}	5.67×10^{-4}	8.83×10^{-2}	1.36×10^{-1}	3.97×10^{-1}	5.00×10^{-1}
2^{-20}	1.08×10^{-1}	3.41×10^{-3}	6.41×10^{-2}	5.68×10^{-4}	8.83×10^{-2}	1.36×10^{-1}	3.97×10^{-1}	5.00×10^{-1}
2^{-22}	1.08×10^{-1}	3.41×10^{-3}	6.41×10^{-2}	5.68×10^{-4}	8.83×10^{-2}	1.36×10^{-1}	3.97×10^{-1}	5.00×10^{-1}
2^{-24}	1.08×10^{-1}	3.41×10^{-3}	6.41×10^{-2}	5.68×10^{-4}	8.82×10^{-2}	1.36×10^{-1}	3.97×10^{-1}	5.00×10^{-1}

TABLE 10: Maximum absolute errors for Example 3 for different values of ϵ with quadratic basis functions.

ϵ	Quadratic spline	Cubic spline	Quartic spline	Gaussian exp. spline	Cosine function	Conical function	Hyperbolic function	Elliptic function
2^{-2}	2.88×10^{-2}	2.83×10^{-2}	2.87×10^{-2}	2.83×10^{-2}	2.86×10^{-2}	3.03×10^{-2}	3.02×10^{-2}	4.04×10^{-2}
2^{-4}	2.01×10^{-2}	1.61×10^{-2}	3.29×10^{-1}	1.53×10^{-2}	1.95×10^{-2}	6.08×10^{-2}	5.67×10^{-2}	6.63×10^{-2}
2^{-6}	2.31×10^{-2}	7.76×10^{-3}	3.15×10^{-1}	5.15×10^{-3}	1.99×10^{-2}	6.04×10^{-2}	6.18×10^{-2}	8.89×10^{-2}
2^{-8}	2.40×10^{-2}	2.15×10^{-3}	2.10×10^{-1}	1.88×10^{-3}	2.07×10^{-2}	6.04×10^{-2}	6.43×10^{-2}	7.18×10^{-2}
2^{-10}	2.40×10^{-2}	8.66×10^{-4}	2.58×10^{-1}	9.85×10^{-4}	2.09×10^{-2}	6.03×10^{-2}	6.43×10^{-2}	6.17×10^{-2}
2^{-12}	2.40×10^{-2}	8.25×10^{-4}	2.33×10^{-1}	9.23×10^{-4}	2.09×10^{-2}	5.96×10^{-2}	6.53×10^{-2}	5.79×10^{-2}
2^{-14}	2.40×10^{-2}	8.20×10^{-4}	2.36×10^{-1}	9.19×10^{-4}	2.75×10^{-2}	5.96×10^{-2}	6.64×10^{-2}	5.67×10^{-2}
2^{-16}	2.40×10^{-2}	8.21×10^{-4}	1.65×10^{-1}	9.16×10^{-4}	2.20×10^{-2}	5.96×10^{-2}	6.67×10^{-2}	5.66×10^{-2}
2^{-18}	2.40×10^{-2}	8.29×10^{-4}	1.19×10^{-1}	9.12×10^{-4}	2.00×10^{-2}	5.96×10^{-2}	6.68×10^{-2}	5.65×10^{-2}
2^{-20}	2.40×10^{-2}	8.25×10^{-4}	1.03×10^{-1}	9.14×10^{-4}	2.09×10^{-2}	5.96×10^{-2}	6.69×10^{-2}	5.64×10^{-2}
2^{-22}	2.40×10^{-2}	8.21×10^{-4}	1.03×10^{-1}	9.14×10^{-4}	2.09×10^{-2}	5.96×10^{-2}	6.69×10^{-2}	5.64×10^{-2}
2^{-24}	2.40×10^{-2}	8.21×10^{-4}	1.03×10^{-1}	9.14×10^{-4}	2.09×10^{-2}	5.96×10^{-2}	6.69×10^{-2}	5.64×10^{-2}

those cited in the literature for different values of ϵ also proves the efficiency of the proposed EFG scheme. In plots (a) and (b) of Figure 5, the comparison of the EFG and the exact solutions and the L_∞ errors has been plotted. The figure shows that very sharp boundary layers appear in the solution at $x = 0$, and the proposed EFG scheme is efficient enough to capture these layers efficiently even for very small values of ϵ . The L_∞ error plot also depicts the stability of the proposed scheme.

Example 3. Consider the following nonlinear singularly perturbed problem [40]:

$$\epsilon v'' + 2v' + e^v = 0, \quad (56)$$

under boundary conditions,

$$\begin{aligned} v(0) &= 0, \\ v(1) &= 0. \end{aligned} \quad (57)$$

The exact solution of the above problem is not available; however, a close approximation of the exact solution of the problem is considered to be

$$v(x) = -\ln\left(\frac{x+1}{2}\right) + \ln\left(\frac{1}{2}\right)e^{-2x/\epsilon}. \quad (58)$$

Alike the above test problems, comparisons of the EFG solutions obtained using different weight functions for both the linear and the quadratic basis functions have been presented in Tables 9 and 10, respectively. It can clearly be observed that, for the nonlinear singularly perturbed problem, Gaussian exponential spline weight functions perform much better in comparison to the other weight functions. The L_∞ errors have been presented in Table 11 for distinct values of ϵ and the different number of nodes. The comparison of the EFG solution with that given by (58), depicting the sharp boundary layer, has been displayed in Figure 6(a). The pointwise absolute error plot in Figure 6(b) clearly demonstrates the compatibility and robustness of the proposed scheme.

Example 4. For the fourth example, we have considered the following nonlinear singularly perturbed problem [41, 42]:

$$\begin{aligned} \epsilon v'' + vv' - v &= 0, \quad x \in (0, 1), \\ v(0) &= -1, \\ v(1) &= 3.9995, \end{aligned} \quad (59)$$

with uniformly valid approximation:

$$v(x) = x + c_1 \tanh\left(\frac{c_1}{2}\left(\frac{x}{\epsilon} + c_2\right)\right), \quad (60)$$

TABLE 11: L_∞ errors for Example 3 for different values of ϵ .

ϵ	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
2^{-2}	2.76×10^{-2}	2.81×10^{-2}	2.82×10^{-2}	2.82×10^{-2}	2.69×10^{-2}
2^{-4}	2.02×10^{-2}	1.76×10^{-2}	1.64×10^{-2}	1.53×10^{-2}	1.42×10^{-2}
2^{-6}	8.50×10^{-3}	5.60×10^{-3}	5.00×10^{-3}	5.15×10^{-3}	4.92×10^{-3}
2^{-8}	1.03×10^{-2}	4.80×10^{-3}	2.80×10^{-3}	1.88×10^{-3}	1.76×10^{-3}
2^{-10}	1.24×10^{-2}	5.20×10^{-3}	2.70×10^{-3}	9.85×10^{-4}	8.57×10^{-4}
2^{-12}	1.31×10^{-2}	5.40×10^{-3}	2.80×10^{-3}	9.23×10^{-4}	8.70×10^{-4}
2^{-14}	1.33×10^{-2}	5.50×10^{-3}	2.80×10^{-3}	9.19×10^{-4}	8.04×10^{-4}
2^{-16}	1.34×10^{-2}	5.50×10^{-3}	2.80×10^{-3}	9.16×10^{-4}	8.37×10^{-4}
2^{-18}	1.34×10^{-2}	5.60×10^{-3}	2.80×10^{-3}	9.12×10^{-4}	8.65×10^{-4}
2^{-20}	1.34×10^{-2}	5.60×10^{-3}	2.80×10^{-3}	9.14×10^{-4}	8.63×10^{-4}

FIGURE 6: (a) Exact and EFG solution plot. (b) Pointwise absolute error plot for $\epsilon = 10^{-6}$ and $N = 128$ for Example 3 using linear basis and cubic weight functions.

where $c_1 = 2.9995$ and $c_2 = (1/c_1)\log_e[(c_1 - 1)/(c_1 + 1)]$. From Table 12, one can analyze the validity and accuracy of the considered EFG scheme. The pointwise EFG solution of the considered problem and the solutions obtained by different methods [41, 42] have also been tabulated in this table. From the table, it is clear that the EFG results are in

good agreement with those cited in the literature. The comparison of the EFG solution and the approximated solution given by equation (60) has been made in Figure 7. It can be easily seen that the EFG solution approximates the close approximation of the exact one up to a good precision of accuracy.

TABLE 12: Comparison of the EFG solution with exact and other scheme solutions of Example 4 for $\epsilon = 10^{-3}$ and $N = 10^3$.

x	Exact solution	EFG method	[41]	[42]
0.00	-1.00000	-1.00000	-1.00000	-1.00000
0.01	3.009500	3.009567	3.009400	—
0.02	3.019500	3.019534	3.019500	3.019465
0.03	3.029500	3.029753	3.029500	—
0.04	3.039500	3.039727	3.039500	3.039476
0.05	3.049500	3.049770	3.049500	—
0.06	3.059500	3.059949	—	3.059494
0.08	3.079500	3.079712	—	3.079436
0.10	3.099500	3.099492	3.099500	3.099503
0.20	3.199500	3.199682	—	3.199500
0.30	3.299500	3.299199	3.299500	—
0.40	3.399500	3.399990	—	3.399476
0.50	3.499500	3.499973	3.499500	—
0.60	3.599500	3.599599	—	3.599431
0.70	3.699500	3.699656	3.699500	—
0.80	3.799500	3.799260	—	3.799473
0.90	3.899500	3.899348	3.899500	—
1.00	3.999500	3.999500	3.999500	3.999499

FIGURE 7: Exact and EFG solution plot for $\epsilon = 10^{-5}$ and $N = 256$ for Example 4.

7. Conclusion

Meshless methods have rarely been implemented for solving singularly perturbed problems in literature. In particular, the element-free Galerkin method has not been utilized for solving SPP to the best of the authors' knowledge. In the present work, an effort has been made to propose the appropriate weight functions in order to implement the EFG method for SPP. Since boundary layers appear in the solutions of SPP, non-equidistributed nodes have been

generated which condense in the boundary layer region and thus utilize the benefit of the proposed scheme. As the choice of weight functions affects the resulting approximation in the EFG method, a variety of weight functions have been presented and tested in the EFG formulation for solving both the linear and nonlinear singularly perturbed problems. It has been concluded that the exponential weight functions are most reliable and produce the most accurate results for SPP in comparison to other weight functions in the case of both the linear and quadratic basis functions. Moving least

square approximations and the Lagrange multiplier method have been utilized for generating the shape functions and to implement the boundary conditions respectively. For nonlinear problems, the quasi-linearization process has been utilized and its convergence has been discussed. Since the radius of the support domain also affects the EFG solution a lot, after a lot of experiments, it has been concluded that, for the range $\alpha_s \in (2.8, 3.2)$, the proposed method provides better approximate results. The convergence of the proposed method has been shown numerically for different problems. The EFG results have been compared with those cited in the literature and are found in good agreement for both the linear and nonlinear singularly perturbed problems.

Data Availability

The datasets supporting the results presented in the study have been generated using the MATLAB code.

Conflicts of Interest

The authors declare no conflicts of interest in this study.

Acknowledgments

The authors take this opportunity to acknowledge the use of Lab facilities provided by the School of Mathematics, Thapar Institute of Engineering and Technology, Patiala, due to the DST-FIST grant SR/FST/MS-1/2017/13.

References

- [1] C. E. Pearson, "On a differential equation of boundary layer type," *Journal of Mathematics and Physics*, vol. 47, no. 1–4, pp. 134–154, 1968.
- [2] N. Bakhvalov, "Towards optimization of methods for solving boundary value problems in the presence of boundary layers," *Zh. Vychisl. Mat. i Mat. Fiz.* vol. 9, pp. 841–859, 1969.
- [3] E. C. Gartland, "Graded-mesh difference schemes for singularly perturbed two-point boundary value problems," *Mathematics of Computation*, vol. 51, no. 184, pp. 631–657, 1988.
- [4] G. I. Shishkin, "A method of improving the accuracy of the solution of difference schemes for parabolic equations with a small parameter in the highest derivative," *USSR Computational Mathematics and Mathematical Physics*, vol. 24, no. 3, pp. 150–157, 1984.
- [5] H.-G. Roos, M. Stynes, and L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems*, Springer Science & Business Media, Berlin, Germany, 2008.
- [6] B. Nayroles, G. Touzot, and P. Villon, "Generalizing the finite element method: diffuse approximation and diffuse elements," *Computational Mechanics*, vol. 10, no. 5, pp. 307–318, 1992.
- [7] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994.
- [8] W. K. Liu, S. Jun, and Y. F. Zhang, "Reproducing kernel particle methods," *International Journal for Numerical Methods in Fluids*, vol. 20, no. 8–9, pp. 1081–1106, 1995.
- [9] S. N. Atluri and T. Zhu, "A new meshless local petrov-galerkin (mlpg) approach in computational mechanics," *Computational Mechanics*, vol. 22, no. 2, pp. 117–127, 1998.
- [10] F. Z. Geng and S. P. Qian, "Reproducing kernel method for singularly perturbed turning point problems having twin boundary layers," *Applied Mathematics Letters*, vol. 26, no. 10, pp. 998–1004, 2013.
- [11] F. Z. Geng and S. P. Qian, "Modified reproducing kernel method for singularly perturbed boundary value problems with a delay," *Applied Mathematical Modelling*, vol. 39, no. 18, pp. 5592–5597, 2015.
- [12] J. Saberi-Nadjafi and F. A. Ghassabzade, "The numerical solution of the singularly perturbed differential-difference equations based on the meshless method," *International Journal of Applied Mathematics Research*, vol. 3, no. 2, p. 116, 2014.
- [13] F. Akhavan Ghassabzade, J. Saberi_Nadjafi, and A. R. Soheili, "A method based on the meshless approach for singularly perturbed differential-difference equations with boundary layers," *Computational Methods for Differential Equations*, vol. 6, no. 3, pp. 295–311, 2018.
- [14] K. Chang and F. A. Howes, *Nonlinear Singular Perturbation Phenomena: Theory and Applications*, Springer Science & Business Media, Berlin, Germany, 2012.
- [15] I. V. Singh, "Meshless efg method in three-dimensional heat transfer problems: a numerical comparison, cost and error analysis," *Numerical Heat Transfer, Part A: Applications*, vol. 46, no. 2, pp. 199–220, 2004.
- [16] J. Dolbow and T. Belytschko, "An introduction to programming the meshless Element Free Galerkin method," *Archives of Computational Methods in Engineering*, vol. 5, no. 3, pp. 207–241, 1998.
- [17] Y.-C. Park and D. I. Leap, "Modeling groundwater flow by the element-free galerkin (efg) method," *Geosciences Journal*, vol. 4, no. 3, pp. 231–241, 2000.
- [18] H. Yang and Y. He, "Solving heat transfer problems with phase change via smoothed effective heat capacity and element-free galerkin methods," *International Communications in Heat and Mass Transfer*, vol. 37, no. 4, pp. 385–392, 2010.
- [19] Z. Juan, L. Shuyao, and L. Guangyao, "The topology optimization design for continuum structures based on the element free galerkin method," *Engineering Analysis with Boundary Elements*, vol. 34, no. 7, pp. 666–672, 2010.
- [20] X.-h. Yuan, "A well-balanced element-free galerkin method for the nonlinear shallow water equations," *Applied Mathematics and Computation*, vol. 331, pp. 46–53, 2018.
- [21] J.-P. Zhang, G.-Q. Zhou, S.-G. Gong, S.-S. Wang, and S. Hu, "Steady heat transfer analysis of orthotropic structure based on element-free galerkin method," *International Journal of Thermal Sciences*, vol. 121, pp. 163–181, 2017.
- [22] P. Bouillard and S. Suleaub, "Element-free galerkin solutions for helmholtz problems: formulation and numerical assessment of the pollution effect," *Computer Methods in Applied Mechanics and Engineering*, vol. 162, no. 1–4, pp. 317–335, 1998.
- [23] M. Fleming, Y. A. Chu, B. Moran, and T. Belytschko, "Enriched element-free galerkin methods for crack tip fields," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 8, pp. 1483–1504, 1997.
- [24] L. W. Zhang, D. M. Li, and K. M. Liew, "An element-free computational framework for elastodynamic problems based on the imls-ritz method," *Engineering Analysis with Boundary Elements*, vol. 54, pp. 39–46, 2015.
- [25] R. Sharma, "Effect of viscous dissipation and heat source on unsteady boundary layer flow and heat transfer past a

- stretching surface embedded in a porous medium using element free galerkin method,” *Applied Mathematics and Computation*, vol. 219, no. 3, pp. 976–987, 2012.
- [26] T. Shibata and A. Murakami, “A stabilization procedure for soil-water coupled problems using the element-free galerkin method,” *Computers and Geotechnics*, vol. 38, no. 5, pp. 585–597, 2011.
- [27] Y. He, H. Yang, and A. J. Deeks, “An element-free galerkin (efg) scaled boundary method,” *Finite Elements in Analysis and Design*, vol. 62, pp. 28–36, 2012.
- [28] E. Jaberzadeh, M. Azhari, and B. Boroomand, “Thermal buckling of functionally graded skew and trapezoidal plates with different boundary conditions using the element-free galerkin method,” *European Journal of Mechanics-A/Solids*, vol. 42, pp. 18–26, 2013.
- [29] H. Cheng, M. J. Peng, and Y. M. Cheng, “A hybrid improved complex variable element-free galerkin method for three-dimensional advection-diffusion problems,” *Engineering Analysis with Boundary Elements*, vol. 97, pp. 39–54, 2018.
- [30] H. Mousavi, M. Azhari, and M. M. Saadatpour, “A novel formulation for static and buckling analysis of plates using coupled element free galerkin-finite strip (efg-fs),” *Applied Mathematical Modelling*, vol. 70, pp. 264–284, 2019.
- [31] T. Belytschko, Y. Y. Lu, and L. Gu, “Crack propagation by element-free galerkin methods,” *Engineering Fracture Mechanics*, vol. 51, no. 2, pp. 295–315, 1995.
- [32] M. Dehghan and M. Abbaszadeh, “Analysis of the element free galerkin (efg) method for solving fractional cable equation with dirichlet boundary condition,” *Applied Numerical Mathematics*, vol. 109, pp. 208–234, 2016.
- [33] C.-P. Wu and K.-H. Chiu, “Rmvt-based meshless collocation and element-free galerkin methods for the quasi-3d free vibration analysis of multilayered composite and fgm plates,” *Composite Structures*, vol. 93, no. 5, pp. 1433–1448, 2011.
- [34] T. Pannachet, S. Triyotee, and M. Boonpichetvong, “Numerical instability in the element-free galerkin method for bi-material modeling,” *Applied and Computational Mechanics*, vol. 12, no. 2, 2019.
- [35] X. Zhuang, C. Heaney, and C. Augarde, “On error control in the element-free galerkin method,” *Engineering Analysis with Boundary Elements*, vol. 36, no. 3, pp. 351–360, 2012.
- [36] T. Most, “A natural neighbour-based moving least-squares approach for the element-free galerkin method,” *International Journal for Numerical Methods in Engineering*, vol. 71, no. 2, pp. 224–252, 2007.
- [37] S. Liu, Q. Yang, H. Chen, W. Yang, and W. Yan, “Application of element free galerkin method to three dimensional electromagnetic computations,” in *Proceedings of the 2005 International Conference on Electrical Machines and Systems*, pp. 2109–2111, Beijing, China, September 2005.
- [38] R. E. Bellman and R. K. Quasilinearization, *Nonlinear Boundary-Value Problems*, American Elsevier Publishing Co., Inc., New York, NY, USA, 1965.
- [39] M. K. Kadalbajoo and K. C. Patidar, “ ϵ -Uniformly convergent fitted mesh finite difference methods for general singular perturbation problems,” *Applied Mathematics and Computation*, vol. 179, no. 1, pp. 248–266, 2006.
- [40] R. K. Lodhi and H. K. Mishra, “Quintic b-spline method for solving second order linear and nonlinear singularly perturbed two-point boundary value problems,” *Journal of Computational and Applied Mathematics*, vol. 319, pp. 170–187, 2017.
- [41] A. Kaushik, V. Kumar, and A. K. Vashishth, “An efficient mixed asymptotic-numerical scheme for singularly perturbed convection diffusion problems,” *Applied Mathematics and Computation*, vol. 218, no. 17, pp. 8645–8658, 2012.
- [42] Y. N. Reddy and P. Pramod Chakravarthy, “Method of reduction of order for solving singularly perturbed two-point boundary value problems,” *Applied Mathematics and Computation*, vol. 136, no. 1, pp. 27–45, 2003.