Journal of Mathematical Modeling
Vol. 7, No. 3, 2019, pp. 319-336

JMM

Numerical solution of sigularly perturbed parabolic problems by a local kernel-based method with an adaptive algorithm

Hossein Rafieayanzadeh, Maryam Mohammadi[*] and Esmail Babolian

Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran
Emails: std_rafieayan@khu.ac.ir, m.mohammadi@khu.ac.ir, babolian@khu.ac.ir

Abstract. Global approaches make troubles and deficiencies for solving singularly perturbed problems. In this work, a local kernel-based method is applied for solving singularly perturbed parabolic problems. The kernels are constructed by the Newton basis functions (NBFs) on stencils selected as thin regions of the domain of problem that leads to increasing accuracy with less computational costs. In addition, position of nodes may affect significantly on accuracy of the method, therefore, the adaptive residual subsampling algorithm is used to locate optimal position of nodes. Finally, some problems are solved by the proposed method and the accuracy and efficiency of the method is compared with results of some other methods.

Keywords: Local kernel-based method, Newton basis functions, adaptive residual subsampling algorithm, singularly perturbed parabolic problems, convection-diffusion problems.
AMS Subject Classification: 35B25, 65M99.

1 Introduction

Singularly perturbed differential equations are problems with having a small positive parameter, ε, in the highest derivative terms. Reformula-

[*] Corresponding author.

Received: 10 August 2019 / Revised: 24 August 2019 / Accepted: 24 August 2019.
DOI: 10.22124/jmm.2019.14093.1305

http://jmm.guilan.ac.ir

tion physical variables into dimensionless variables triggers these problems. We cosider the nonlinear singularly perturbed parabolic equation of the general form

$$u_t = \varepsilon u_{xx} + F(x,t,u,u_x), \quad (x,t) \in (0,1) \times (0,T), \tag{1}$$

with the initial condition

$$u(x,0) = u_0(x), \quad x \in (0,1), \tag{2}$$

and the boundary conditions

$$u(0,t) = h_1(t), \quad u(1,t) = h_2(t), \quad t \in (0,T), \tag{3}$$

where the function F is sufficiently smooth and $\frac{\partial F}{\partial u}$ and $\frac{\partial F}{\partial u_x}$ exist. We also consider the following singularly perturbed convection-diffusion problem

$$u_t - \varepsilon u_{xx} + a(x)u_x + b(x)u = f(x,t), \quad (x,t) \in (0,1) \times (0,T), \tag{4}$$

with the initial and boundary conditions (2)-(3). Many issues are modeled by these equations in various branches of science and engineering such as fluid dynamics, chemical kinetics, robotics, control theory, convective heat transport problem, financial modeling and turbulence model [11,22,24,29].

The solution of these problems often exhibits a thin transition layer, called a boundary layer, at one or both ends. Within the boundary layer regions the magnitude of the first derivative of the solution can be significantly larger compared to the size of the solution derivatives outside the boudary layers and it causes the solution varies rapidly in the boundary layer, while away from the layer, the solution behaves smoothly and varies gradually [15]. In these problems high accuracy can not be obtained unless an appropriate method is employed. It has been well established [4] that the classical numerical approaches, such as the centred finite difference method and the upwind finite difference operator on uniform mesh, to solve these problems have many deficiencies, especially in pointwise global approaches and can not be hoped to construct a layer resolving with using a uniform mesh. In [20] and [21] new two-level implicit variable mesh schemes have been developed for the solution of nonlinear parabolic equations. In [35], Yzbasi and Sahin suggest a collocation method by using the truncated Bessel series and constructing the matrix operations in a global approach. In [7] numerical approximations are generated using a classical finite difference operator on a piecewise uniform Shishkin mesh. In [6] the domain is divided by using a piecewise uniform adaptive mesh in the spatial direction. The domain decomposition method is applied in [34] by dividing the original domain of the problem into three overlapping subdomains and discretizing the problem by the backward Euler scheme in the time direction.

In this study, we use a local kernel-based method with an adaptive algorithm to solve singularly perturbed problems. The kernel-based methods have been applied in many papers for solving various problems, see, e.g. [8, 9, 16–19, 26, 36]. The representations of kernel-based approximants in terms of the standard basis of translated kernels are notoriously unstable. A more useful basis, so-called Newton basis, is offered in [23]. The Newton basis turns out to be orthonomal in the reproducing native Hilbert space, and it is complete, if infinitely many data locations are reasonably chosen. A timesaving calculation of Newton basis arising from a pivoted Cholesky factorization has been introduced in [25]. In this study, expansion in terms of the Newton basis functions (NBFs), as a spatial interpolation, is used to obtain a system of ODEs from the time-dependent PDEs. In this local scheme, using overlapping small sub-domains of the whole domain [1, 12], produces sparse operational matrices and consequently less computer storages are needed and computational cost significantly reduces. In addition, in order to increase the accuracy of the method, we employ the adaptive residual subsampling algorithm [3] to optimize location of nodes by generating non-uniform adapted mesh, specially in boundary layers and regions with rapid variation. In this adaptive method an interpolant is computed for the set of points and then based on interpolation residuals, some points can be added to or removed from the set of points to sample a finer nodes set.

This paper is structured as follows. In Section 2, the kernels spanned by the NBFs are constructed and used for approximating the spatial derivatives needed in the paper. In Section 3 the method implementation is provided. The adaptive residual subsampling algorithm is explained in Section 4. In Section 5, the numerical results and their comprison with the results of some other methods are given. Finally, a brief conclusion is given in the last section.

2 Method description

Let Ω be a nonempty set. A function $K : \Omega \times \Omega \to \mathbb{R}$ is called a positive definite kernel on Ω, that is $\sum_{i,j=1}^{n} K(y_i, y_j)\xi_i\xi_j \geq 0$ for any finite set of points $\{y_1, ..., y_n\} \in \Omega$ and any real numbers $\xi_1, ..., \xi_n$. Let $K : \Omega \times \Omega \to \mathbb{R}$ be a symmetric positive definite kernel on Ω. This means that for all finite sets $\{x_i \in \Omega, i = 1, ..., n\}$ the kernel matrix $A = [K(x_i, x_j)]_{i,j=1,...,n}$ is symmetric and positive definite. These kernels are *reproducing* in "native" Hilbert space $\mathcal{N}_K = \overline{\text{span}\{K(x, \cdot) \mid x \in \Omega\}}$ of functions on Ω in the sense

$$\langle f, K(x, \cdot)\rangle_{\mathcal{N}_K} = f(x) \quad \text{for all } x \in \Omega, f \in \mathcal{N}_K.$$

The most important examples are the Whittle-Matern kernels $r^{m-d/2}\mathrm{K}_{m-d/2}(r)$, $r = \|x - y\|$, $x, y \in \mathbb{R}^d$, reproducing in the Sobolev space $W_2^m(\mathbb{R}^d)$ for $m > d/2$, where K_ν is the modified Bessel function of the second kind [32]. For scattered nodes $\{x_i \in \overline{\Omega}, i = 1, ..., N\}$ the *translates* $K_j(x) = K(x, x_j)$ are the trial functions. The Newton basis functions (NBFs) $\{N_k(x)\}_{k=1}^n$ can be expressed by

$$N_k(x) = \sum_{j=1}^{n} K(x, x_j)c_{jk}, \quad k = 1, ..., n, \tag{5}$$

where $K(x, x_j)$ must be a positive definite kernel like some kinds of radial basis functions (RBFs) such as Gaussian (GA), Inverse Multiquadric (IMQ) and Mattern Sobolov (MS), repectively given by $\phi(r) = \exp(-\frac{r^2}{2\alpha^2})$, $\phi(r) = (1+\frac{r^2}{\alpha^2})^{-\frac{1}{2}}$ and $\phi(r) = (\frac{r}{\alpha})^\nu \mathrm{K}_\nu(\frac{r}{\alpha})$, where K_ν is the modified Bessel function of second kind and $r = \|x - x_j\|_2$. In addition, there is a free parameter α in the RBFs called the shape parameter which can change the accuracy and the condition number of the interpolation matix.

Considering $N = [N_j(x_i)]_{i,j=1,...,n}$, $A = [K(x_i, x_j)]_{i,j=1,...n}$ and $C = [c_{jk}]_{j,k=1,...,n}$ in (5) gives $N = A \cdot C$. It has been proved [25] that the Cholesky decomposition $A = L \cdot L^T$ with a nonsingular lower triangular matrix L leads to the Newton basis

$$N(x) = T(x) \cdot C, \tag{6}$$

with

$$N = L, \quad C = (L^T)^{-1}, \tag{7}$$

where $N(x) = [N_1(x) \cdots N_n(x)]$, $T(x) = [K(x, x_1) \cdots K(x, x_n)]$. The condition number of the collocation matrix corresponding to the NBFs is smaller than the one corresponding to translated kernels which leads to more stability rather than using the translated kernels.

For using the NBFs in a local method, a local set of points is needed. For this purpose a set of pionts $X = \{x_i \in \overline{\Omega}, i = 1, ..., n\}$ consists of discrete pionts is considered. In order to construct a local domain, the center $x_i \in X$ and its $m - 1$ nearest neighboring points are considered as a stencil $\Omega_i = \{x_k^i\}_{k=1}^m$ and the NBFs $N_1^i, ..., N_m^i$ corresponding to the stencil are obtained by (6). Now, a solution function $u(x,t)$ can be approximated by NBFs on the local domain in the form

$$u^i(x,t) = \sum_{k=1}^{m} N_k^i(x)\ \alpha_k^i(t). \tag{8}$$

Substituting $x_i \in \Omega_i$ in (8) results in

$$u^i(x_i,t) = N^i \cdot \alpha^i, \quad x_i \in \Omega_i, \tag{9}$$

where, $N^i = [N_1^i(x_i) \cdots N_m^i(x_i)]$ and $\alpha^i = [\alpha_1^i(t) \cdots \alpha_m^i(t)]^T$ is an unknown vector. For obtaining the unknown vector α^i, putting the points $\{x_k^i\}_{k=1}^m$ in (8) leads to the linear system

$$U^i = \mathbf{N}^i \cdot \alpha^i,$$

where $U^i = [u^i(x_1^i,t) \cdots u^i(x_m^i,t)]^T$, and $\mathbf{N}^i = [N_k^i(x_p^i)]_{k,p=1,\ldots,m}$. Therefore

$$\alpha^i = \left(\mathbf{N}^i\right)^{-1} \cdot U^i. \tag{10}$$

Consequently (9) and (10) result in

$$u^i(x_i,t) = N^i \cdot \left(\mathbf{N}^i\right)^{-1} \cdot U^i. \tag{11}$$

Moreover, the spatial partial derivatives are needed to be implemented. The s-th order spatial partial derivatives are in the following form

$$\frac{\partial^s}{\partial x^s} u^i(x_i,t) = N_{(s)}^i \cdot \left(\mathbf{N}^i\right)^{-1} \cdot U^i, \quad s = 1, 2, \ldots \tag{12}$$

where $N_{(s)}^i = [\frac{\partial^s}{\partial x^s} N_1^i(x_i) \cdots \frac{\partial^s}{\partial x^s} N_m^i(x_i)]$ that is clearly calculated by differentiation of (6). Considering $B_{(s)}^i = N_{(s)}^i \cdot \left(\mathbf{N}^i\right)^{-1}$, (12) results

$$\frac{\partial^s}{\partial x^s} u^i(x_i,t) = B_{(s)}^i \cdot U^i. \tag{13}$$

3 Implementation

Consider the set of points $X = \{x_i, i = 1, \ldots, n\}$ on $[0,1]$ which $X_I = \{x_i, i = 1, \ldots, n-2\}$ in $(0,1)$ is the set of interior points and $X_B = \{x_{n-1}, x_n\} = \{0,1\}$ is the set of boundary points. Substituting $x_i \in X_I$ in (1) leads to system of equations

$$u^i{}_t(x_i,t) = \varepsilon u^i{}_{xx}(x_i,t) + F\big(x_i, t, u^i(x_i,t), u^i{}_x(x_i,t)\big), \quad i = 1, \ldots, n-2. \tag{14}$$

By substituting (13) in (14), we have

$$u^i{}_t(x_i,t) = \varepsilon B^i_{(2)} U^i + F\big(x_i, t, u(x_i,t), B^i_{(1)} U^i\big), \quad i = 1, ..., n-2. \tag{15}$$

To attain an explicit matrix form for system of equations (15) and to change the local derivative matrices into global matrices, a $(n-2)$-by-n matrix $D_{(s)}$ according to derivatives appeared in (14) is defined as follows

$$D_{(s)}(i, \mathcal{I}_i) = B^i_{(s)}, \tag{16}$$

where $\mathcal{I}_i$ is a vector contains the indices of center x_i and its $m-1$ nearest neighboring points in its related stencil. The local property of the stencils leads to sparsity of $D_{(s)}$ which in turn decrease computational cost. Applying (16) for (15) leads to

$$U'_I = \varepsilon D_{(2)} \cdot U_I + F\big(x_i, t, U_I, D_{(1)} \cdot U_I\big), \tag{17}$$

where $U'_I = [u^i{}_t(x_1,t) \cdots u^i{}_t(x_{n-2},t)]^T$ and $U_I = [u^i(x_1,t) \cdots u^i(x_{n-2},t)]^T$. Similarly, substituting $x_i \in X_I$ in the convection-diffusion equation (4) and using (13) and (16) give the following matrix form

$$U'_I = \varepsilon D_{(2)} \cdot U_I - A.*D_{(1)} \cdot U_I - B.*U_I + F, \tag{18}$$

where $A = [a(x_1) \cdots a(x_{n-2})]^T$, $B = [b(x_1) \cdots b(x_{n-2})]^T$, $F = [f(x_1,t) \cdots f(x_{n-2},t)]^T$ and $.*$ denotes the pointwise product between two matrices or vectors. The boundary conditions (3) are of the following vector form

$$U_B = H, \tag{19}$$

where $U_B = [U^{n-1} \;\; U^n]^T = [u(0,t) \;\; u(1,t)]^T$ and $H = [h_1(t) \;\; h_2(t)]^T$. In addition, the initial condition (2) gives

$$U(0) = U_0, \tag{20}$$

where $U(0) = [u(x_1,0) \cdots u(x_n,0)]^T$ and $U_0 = [u_0(x_1) \cdots u_0(x_n)]^T$. Finally, Eqs. (17) and (19) or Eqs. (18)-(19) lead to systems of ODEs with the initial condition (20) that can be solved by an appropriate ODE solver uses automatically a reasonable time-stepping and detects stiffness of the ODE system.

4 Adaptive algorithm

The kernel functions are constructed by using a set of center points which can set anywhere of the given domain independently and also positions of the center points make influence on approximation quality and stability of interpolation [31]. Since kernel-based methods are completely meshfree, some adaptive algorithms [13, 14] for finding optimal set of point may be devised. For example, in problems with rapid variations in given domain, such as boundary layers, steep gradients and corners, adaptive methods may be preferred over fixed grid methods. In order to achieve accuracy and stability, adaptive methods select optimal centers by moving, adding or removing points.

In adaptive residual subsampling algorithm, some points may be added or removed by using computed residuals [3]. In this section, we describe the adaptive residual subsampling algorithm for the local kernel-based method which spatial trial space is spanned by the NBFs.

Implementation of the adaptive residual subsampling technique for time-dependent PDEs is to alternate time stepping with adaptation. First, initial center points $\{x_i, i = 1, ..., n\}$ are generated using n equally spaced points in the given domain. Then, using NBF interpolation at center points and initial condition of the PDE, unknown coefficients vector $\alpha = [\alpha_1 \cdots \alpha_n]^T$ is calculated from the linear system

$$N\alpha = U_0,$$

where, $N = [N_j(x_i)]_{i,j=1,...,n}$ is the NBF matrix, $U_0 = [u_0(x_1) \cdots u_0(x_n)]^T$ and $u_0(x)$ is the initial condition function of the PDE. Now, the set $\{y_i = \frac{1}{2}(x_{i+1} - x_i), i = 1, ..., n-1\}$ is considered halfway between the center points. The residuals vector r is calculated by

$$r = |N^y\alpha - U_0^y|,$$

where, N^y is the NBF matrix for the points $\{y_i\}_{i=1}^{n-1}$, which is calculated by (6)-(7), $U_0^y = [u_0(y_1) \cdots u_0(y_{n-1})]^T$, and $r = [r_1 \cdots r_{n-1}]^T$. Points at which the residual exceeds a threshold θ_r are to become center points, and center points that lie between two points whose error is below a smaller threshold θ_c are removed. This means, if $r_i > \theta_r$, then y_i will be added to set of center points, and if $r_i < \theta_c$ and $r_{i+1} < \theta_c$, then x_{i+1} will be removed. This process is called coarse–refine (coarse for removing center points and refine for adding new points). Therefore, a new set of center points is given and the coarse–refine process is repeated while any new point can not be added. After ending coarse–refine processes, a new set of center points is obtained. These new center points are used to advance the discrete solution up to a predetermined time $t = \tau$ by using the local method, which was described in the previous sections. τ must be large enough to avoid excessive adaptation steps, while keeping it small enough that the adaptation can keep up with emerging or changing features in the solution. So the solution is obtained at the time $t = \tau$. Now residual subsampling algorithm is applied by using the solution at this time level as a new initial state for further time. This process continues to achieve $t = T$.

5 Numerical results

In this section, to verify the described method, the numerical results of our scheme for several singularly perturbed ploblems, of the general form (1) and also convection-diffusion problems (4), with different values of the singular perturbation parameter ε are represented. In some cases the exact solution does not exist and the pointwise error is calculated. The pointwise error is defined by

$$e_\varepsilon^n(x_i) = |u_n(x_i) - u_{2n}(x_i)|,$$

and the maximum pointwise error is given by

$$E_\varepsilon^n = \max_i (e_\varepsilon^n(x_i)),$$

where u_n and u_{2n} are the approximation solutions resulted by n and $2n$ points, respectively. Also, in some cases which the exact solution exists, the following well-known maximum absolute error (MAE) and root mean square error (RMSE) are used:

$$\max_i |u(x_i) - u_n(x_i)|,$$

and

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(u(x_i) - u_n(x_i)\big)^2},$$

where u is the exact solution. In the following examples, the MS and IMQ RBFs are taken for constructing the NBFs and $\tau = 0.01$.

Example 1. Consider the following problem [7]

$$u_t = \varepsilon u_{xx} - (1 + x + t)u + 4^3x^3t^2(1 - x)^3, \quad (x, t) \in (0, 1) \times (0, T], \tag{21}$$

with the boundary conditions

$$u(0, t) = u(1, t) = t^3, \tag{22}$$

and the initial condition

$$u(x, 0) = \big(4x(1 - x)\big)^3. \tag{23}$$

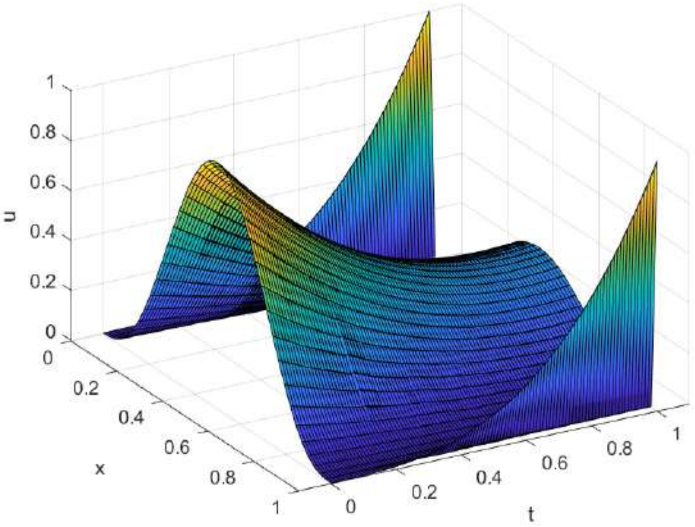

Figure 1: Solution of Example 1 for $\varepsilon = 2^{-20}$ at $T = 1$.

The computed solution of Eqs. (21)-(23) for $\varepsilon = 2^{-20}$ and using MS RBFs with the shape parameter 1, thresholds $\theta_r = 10^{-2}$ and $\theta_c = 10^{-5}$ at $T = 1$ is displayed in Figure 1. As seen in Figure 1, there are boundary layers near $x = 0$ and $x = 1$. Because of using the adaptive method the number of nodes starts with 30 and finally grows to 46 at $T = 1$ and as shown in Figure 2 due to the adaptive method, number of nodes increases near the boundary layers that leads to more accuracy of the method. In Table 1 the maximum pointwise errors E_ε^{30} for different values of ε is shown.

Table 1: The maximum pointwise error E_ε^{30} for Example 1.

$\varepsilon = 2^0$	$\varepsilon = 2^{-1}$	$\varepsilon = 2^{-2}$	$\varepsilon = 2^{-3}$	$\varepsilon = 2^{-4}$	$\varepsilon = 2^{-5}$	$\varepsilon = 2^{-6}$
1.16e-02	1.74e-02	2.41e-02	2.98e-02	3.56e-2	5.50e-02	8.25e-02
$\varepsilon = 2^{-7}$	$\varepsilon = 2^{-8}$	$\varepsilon = 2^{-9}$	$\varepsilon = 2^{-10}$	$\varepsilon = 2^{-11}$	$\varepsilon = 2^{-12}$	$\varepsilon = 2^{-13}$
9.03e-02	9.25e-02	5.57e-02	3.29e-02	1.79e0-2	1.72e-02	7.30e-03
$\varepsilon = 2^{-14}$	$\varepsilon = 2^{-15}$	$\varepsilon = 2^{-16}$	$\varepsilon = 2^{-17}$	$\varepsilon = 2^{-18}$	$\varepsilon = 2^{-19}$	$\varepsilon = 2^{-20}$
6.40e-03	6.48e-04	6.42e-04	7.55e-04	9.50e-04	3.09e-04	7.43e-04

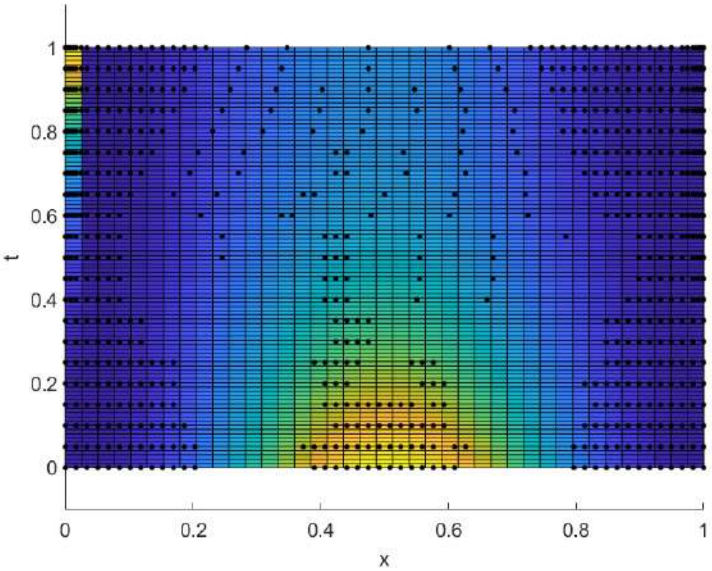

Figure 2: Adaption of nodes for solution of Example 1.

Example 2. Consider the following equation [28]

$$u_t = \varepsilon u_{xx} - (u_x)^2 + u^2 + f(x,t), \quad (x,t) \in (0,1) \times (0,T], \tag{24}$$

with the boundary conditions

$$u(0,t) = u(1,t) = 0, \tag{25}$$

and the initial condition

$$u(x,0) = \varepsilon \sin(\pi x)\cos(\pi x), \tag{26}$$

where $f(x,t)$ is given by the exact solution

$$u(x,t) = \varepsilon \mathrm{e}^{-\varepsilon^2 t} \sin(\pi x)\cos(\pi x).$$

The MAE and RMSE for computed solution of the Eqs. (24)-(26) are given in Table 2 for different values of ε and various number of points n at $T=1$ by using IMQ RBF, which shows accuracy of the method for small values of ε. Moreover, the results are reported at $T=3$ in Table 3 which shows the method remain accurate with increasing time. For more clarity the absolute errors of the solutin for $\varepsilon = 10^{-2}$ and different values of n at $T=1$ and $T=3$ are displayed in Figures 3-6.

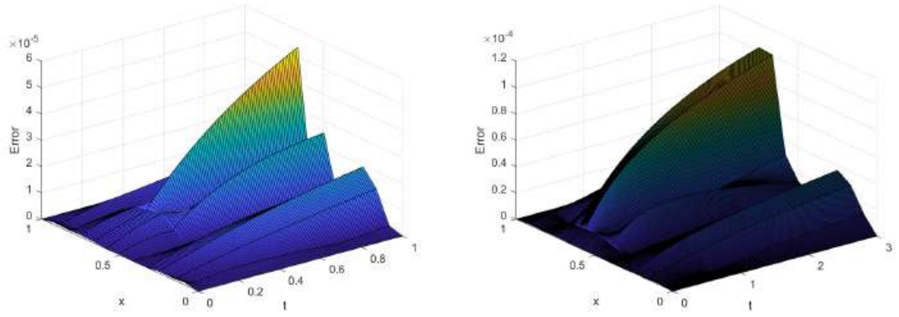

Figure 3: Absolute error for solution of Example 2 with $n = 8$, (left) at $T = 1$, (right) at $T = 3$.

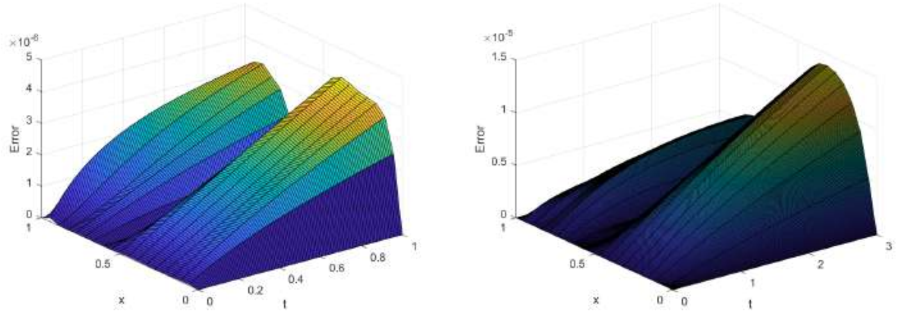

Figure 4: Absolute error for solution of Example 2 with $n = 16$, (left) at $T = 1$, (right) at $T = 3$.

Example 3. Consider the following problem

$$u_t = \varepsilon u_{xx} - (2 - x^2)u_x - xu + 10t^2 \mathrm{e}^{-t} x(1 - x), \quad (x, t) \in [0, 1] \times [0, T], \tag{27}$$

with the boundary conditions

$$u(0, t) = u(1, t) = 0, \tag{28}$$

and the initial condition

$$u(x, 0) = 0. \tag{29}$$

There is not an exact solution for this problem. In [35], the Eqs. (27)-(29) have been solved by the Bessel collocation method and the maximum pointwise error E^n_ε compared with some other methods. In Table 4, the maximum pointwise errors E^n_ε of the Eqs. (27)-(29) for various values of

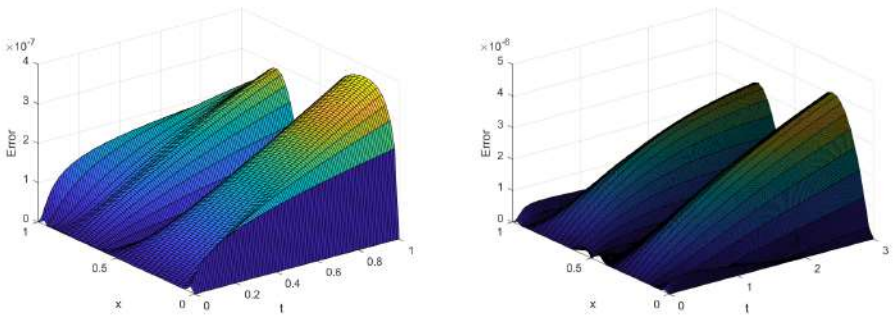

Figure 5: Absolute error for solution of Example 2 with $n = 32$, (left) at $T = 1$, (right) at $T = 3$.

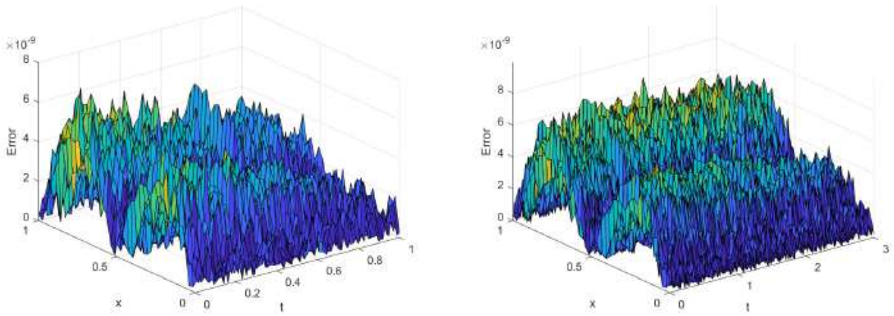

Figure 6: Absolute error for solution of Example 2 with $n = 64$, (left) at $T = 1$, (right) at $T = 3$.

the perturbation parameter ε at $T = 1$ are reported by applying the peresnt method and using the MS RBF with the shape parameter 1, and compared with results of [35].

Example 4. Consider the following problem with a singular coefficient

$$u_t = \varepsilon u_{xx} - \frac{1}{x} u u_x - u^2 + f(x,t), \quad (x,t) \in (0,1) \times (0,T),$$

with the boundary conditions

$$u(0,t) = u(1,t) = 0,$$

and the initial condition

$$u(x,0) = 0,$$

Table 2: The MAE and RMSE for Example 2 at $T = 1$.

n		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$
8	MAE	5.257e-05	2.185e-07	2.420e-09	2.319e-11	2.216e-13
	RMSE	1.9719e-05	1.0907e-07	1.5208e-09	1.0942e-11	1.0464e-13
16	MAE	1.978e-06	9.278e-08	6.714e-10	6.644e-12	6.266e-14
	RMSE	1.8926e-06	6.4810e-08	1.9080e-10	2.0566e-12	1.9390e-14
32	MAE	2.333e-06	3.790e-08	7.868e-11	9.254e-13	8.322e-15
	RMSE	1.5020e-06	4.0278e-09	1.7985e-11	2.1147e-13	1.9283e-15
64	MAE	7.269e-09	3.177e-10	1.539e-11	9.228e-14	9.471e-16
	RMSE	4.7335e-09	7.8427e-11	1.0885e-11	4.1699e-14	4.1840e-16

Table 3: The MAE and RMSE for solution of Example 2 at $T = 3$.

n		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$
8	MAE	1.099e-04	6.281e-07	7.551e-09	7.268e-11	6.904e-13
	RMSE	4.5892e-05	3.0234e-07	4.7093e-09	3.4289e-11	3.2807e-13
16	MAE	1.319e-05	2.136e-07	1.875e-09	2.076e-11	1.964e-13
	RMSE	6.2391e-06	9.9790e-08	5.7850e-10	6.4258e-12	6.0775e-14
32	MAE	5.105e-06	3.790e-08	2.142e-10	2.858e-12	2.640e-14
	RMSE	3.2171e-06	6.7590e-09	4.9474e-11	6.5295e-13	6.0365e-15
64	MAE	9.983e-09	3.608e-10	2.586e-10	2.715e-13	2.951e-15
	RMSE	6.1701e-09	1.3077e-10	4.4739e-11	1.2981e-13	1.3107e-15

where $f(x,t)$ is given by the exact solution

$$u(x,t) = tx(1-x).$$

This problem is solved by the present method for various values of ε and different number of points n by using IMQ RBF at $T = 1$. The muximum pointwise errors E_ε^n are computed for each case and compared with results of the Sinc collocation method [28] and are given in Table 5. The results show that errors decrease while n increases.

Table 4: Comparison of the maximum pointwise error E_{ε}^{n} for Example 3 at $T = 1$.

$n \rightarrow$	3	4	16	32
$\varepsilon \downarrow$	Present method		Present method	
2^{-2}	1.7606e-05	1.1500e-05	4.3164e-06	3.6481e-06
2^{-4}	7.0420e-05	4.5998e-05	1.0852e-05	9.5800e-06
2^{-6}	2.8158e-04	1.8392e-04	4.3407e-05	3.8318e-05
2^{-8}	9.3327e-04	8.3034e-04	1.7360e-04	1.5324e-04
2^{-10}	9.1980e-04	5.3611e-04	6.9403e-04	6.1240e-04
2^{-12}	7.3757e-04	9.2937e-05	4.6151e-05	1.1725e-05
2^{-14}	7.3757e-04	1.0170e-05	3.6481e-05	4.3164e-06

$n \rightarrow$	3	4	16	32
$\varepsilon \downarrow$	Euler implicit method [2]		Piecewise-analytical method [27]	
2^{-2}	1.1249e-2	6.3202e-3	2.6e-3	9.922e-4
2^{-4}	1.6783e-2	8.1043e-3	1.15e-2	5.1e-3
2^{-6}	3.090e-2	1.5221e-2	2.25e-2	1.67e-2
2^{-8}	3.5742e-2	1.9347e-2	1.52e-2	1.44e-2
2^{-10}	3.6717e-2	2.0475e-2	1.33e-2	7.9e-3
2^{-12}	3.6931e-2	2.0732e-2	1.4e-2	6.7e-3
2^{-14}	3.6982e-2	2.0794e-2	1.41e-2	6.9e-3

$n \rightarrow$	3	4	16	32
$\varepsilon \downarrow$	Bessel collocation method [35]		B-spline collocation method with Shishkin mesh [10]	
2^{-2}	1.7917e-2	1.0909e-3	5.5489e-3	1.4220e-3
2^{-4}	2.4545e-2	1.1414e-2	4.9629e-3	2.8234e-3
2^{-6}	4.2727e-2	1.1875e-2	3.7566e-3	1.2640e-3
2^{-8}	5.1250e-2	2.9091e-2	1.2536e-2	2.6440e-3
2^{-10}	5.3750e-2	3.2500e-2	1.6606e-2	5.3053e-3
2^{-12}	5.5000e-2	3.3333e-2	1.7491e-2	6.3301e-3
2^{-14}	5.6250e-2	3.4545e-2	1.7860e-2	6.4345e-3

6 Conclusion

In this paper, a local kernel-based method was studied to solve singularly perturbed parabolic problems and successfully applied for some numerical examples. In this method, kernels were constructed by spanning the

Table 5: The maximum pointwise errors E_ε^n for Example 4 at $T = 1$.

$n \rightarrow$	4	8	4	8
$\varepsilon \downarrow$	Present method		Sinc collocation method [28]	
10^{-2}	7.3070e-04	5.7647e-05	6.53e-04	1.51e-04
10^{-3}	5.6685e-04	1.8079e-06	1.79e-04	1.37e-05
10^{-4}	9.3726e-05	1.2870e-05	2.03e-04	6.13e-05
10^{-5}	7.3819e-04	6.5612e-05	9.12e-04	3.11e-05
10^{-6}	6.6060e-05	6.6060e-05	1.65e-03	3.13e-04

$n \rightarrow$	16	32	16	32
$\varepsilon \downarrow$	Present method		Sinc collocation method [28]	
10^{-2}	2.2888e-05	3.0124e-07	1.12e-05	4.65e-07
10^{-3}	9.5467e-06	5.1231e-07	3.44e-06	5.96e-07
10^{-4}	5.4929e-06	7.3578e-07	9.67e-06	7.62e-07
10^{-5}	5.6034e-07	8.1148e-07	8.48e-06	1.34e-06
10^{-6}	6.9021e-06	9.3689e-07	7.81e-05	8.51e-06

Newton basis Functions on stencils which led to low computational costs. Moreover, in order to optimize nodes, specially in boudary layers, the adaptive residual subsampling algorithm was employed which in turn gave more accurate results.

Acknowledgements

The authors would like to thank the referee for his/her helpful comments and suggestions.

References

[1] C.S. Chen, M. Ganesh, M.A. Golberg and A.H.D. Cheng, *Multilevel compact radial basis functions based computational scheme for some elliptic problems*, Comput. Math. Appl. **43** (2002) 359-378.

[2] C. Clavero, J.C. Jorge and F. Lisbona, *A uniformly convergent scheme on a nonuniform mesh for convectiondiffusion parabolic problems*, J. Comput. Appl. Math. **154** (2003) 415-429.

[3] T.A. Driscoll and A.R.H. Heryudono, *Adaptive residual subsampling methods for radial basis function interpolation and collocation problems*, Comput. Math. Appl. **53** (2007) 927-939.

[4] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O'Riordan and G.I. Shishkin, *Robust Computational Techniques for Boundary Layers*, Chapman and Hall, 2000.

[5] D. Gong, C. Wei, L. Wang, L. Feng and L. Wang, *adaptive methods for center choosing of radial basis function interpolation: a review*, Lecture Notes in Comput. Sci. **6377** (2010) 573-580.

[6] S. Gowrisankar and S. Natesan, *ε-Uniformly convergent numerical scheme for singularly perturbed delay parabolic partial differential equations*, Int. J. Comput. Math. **94** (2017) 902-921.

[7] J.L. Gracia and E. ÓRiordan, *Numerical approximation of solution derivatives in the case of singularly perturbed time dependent reaction-diffusion problems*, J. Comput. Appl. Math. **273** (2015) 13-24.

[8] Z.D. Han and S.N. Atluri, *On the (meshless local Petrov-Galerkin) MLPG-Eshelby method in computational finite deformation solid mechanics - Part II*, Comput. Model. Eng. Sci. **97** (2014) 199-237.

[9] Y.C. Hon and R. Schaback, *Solvability of partial differential equations by meshless kernel methods*, Adv. Comput. Math. **28** (2008) 283-299.

[10] M.K. Kadalbajoo, V. Gupta and A. Awasthi, *A uniformly convergent B-spline collocation method on a nonuniform mesh for singularly perturbed one-dimensional time-dependent linear convection-diffusion problem*, J. Comput. Appl. Math. **220** (2008) 271-289.

[11] B.E. Launder and B. Spalding, *Mathematical Models of Turbulence*, Academic Press, 1972.

[12] C.K. Lee, X. Liu and S.C. Fan, *Local multiquadric approximation for solving boundary value problems*, Comput. Mech. **30** (2003) 396-409.

[13] L. Ling, R. Opfe and R. Schaback, *Results on meshless collocation techniques*, Eng. Anal. Bound. Elem. **30** (2006) 247-253.

[14] L. Ling and R. Schaback, *An improved subspace selection algorithm for meshless collocation methods*, Int. J. Numer. Meth. Engng. **80** (2009) 1623-1639.

[15] P. Mishra, K.K. Sharma, A.K. Pani and G. Fairweather, *Orthogonal Spline Collocation for Singularly Perturbed reaction diffusion problems in one dimension*, Int. J. Numer. Anal. Mod. **16** (2019) 647-667.

[16] M. Mohammadi and R. Mokhtari, *A reproducing kernel method for solving a class of nonlinear systems of PDEs*, Math. Model. Anal. **19** (2014) 180-198.

[17] M. Mohammadi, R. Mokhtari and H. Panahipour, *A Galerkin-reproducing kernel method: application to the 2D nonlinear coupled Burgers' equations*, Eng. Anal. Bound. Elem. **37** (2013) 1642-1652.

[18] M. Mohammadi, R. Mokhtari and H. Panahipour, *Solving two parabolic inverse problems with a nonlocal boundary condition in the reproducing kernel space*, Appl. Comput. Math. **13** (2014) 91-106.

[19] M. Mohammadi, F.S. Zafarghandi, E. Babolian and S. Javadi, *A local reproducing kernel method accompanied by some different edge improvement techniques: application to the Burgers equation*, Iran J. Sci. Technol. Trans. Sci. **42** (2018) 857-871.

[20] R.K. Mohanty, *An implicit high accuracy variable mesh scheme for 1-D nonlinear singular parabolic partial differential equations*, Appl. Math. Comput. **186** (2007) 219-229.

[21] R.K. Mohanty and S. Singh, *Non-uniform mesh arithmetic average discretization for parabolic initial boundary value problems*, Neural Parallel Scient. Comput. **13** (2005) 401-416.

[22] K.W. Morton, *Numerical Solution of Convection Diffusion Problems*, Chapman and Hall, 1995.

[23] S. Muller and R. Schaback, *A Newton basis for kernel spaces*, J. Approx. Theory. **161** (2009) 645-655.

[24] R.E. ÓMalley, *Singular Perturbation Methods for Ordinary Differential Equations*, Springer-Verlag, 1991.

[25] M. Pazouki and R. Schaback, *Bases for kernel-based spaces*, J. Comput. Appl. Math. **236** (2011) 575-588

[26] H. Rafieayanzadeh, M. Mohammadi and E. Babolian, *Solving a class of PDEs by a local reproducing kernel method with an adaptive residual subsampling technique*, Comput. Model. Eng. Sci. **108** (2015) 375-396.

[27] J.I. Ramos, *A piecewise-analytical method for singularly perturbed parabolic problems*, Appl. Math. Comput. **161** (2005) 501-512.

[28] J. Rashidinia and A. Barati, *Numerical solutions of one-dimensional non-linear parabolic equations using Sinc collocation method*, Ain Shams Eng. J. **6** (2015) 381-389.

[29] H. Roos, M. Stynes and L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations Convection-Diffusion-Reaction and Flow Problems*, Springer-Verlag, 2008.

[30] K. Sayevand and K. Pichaghchi, *A novel operational matrix method for solving singularly perturbed boundary value problems of fractional multi-order*, Int. J. Comput. Math. **95** (2018) 767-796.

[31] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math. **3** (1995) 251-264.

[32] R. Schaback, *Kernel-based meshless methods, Lecture Note*, Gottingen, http://num.math.unigoettingen.de/schaback/teaching/AV2.pdf, 2011.

[33] R. Schaback, *A computational tool for comparing all linear PDE solvers*, Adv. Comput. Math. **41** (2015) 333-355.

[34] J. Singh, S. Kumar and M. Kumar, *A domain decomposition method for solving singularly perturbed parabolic reactiondiffusion problems with time delay*, Numer. Methods Partial Differential Eq. **34** (2018) 1849-1866.

[35] S. Yüzbasi and N. Sahin, *Numerical solutions of singularly perturbed one-dimensional parabolic convectiondiffusion problems by the Bessel collocation method*, Appl. Math. Comput. **220** (2013) 305-315.

[36] F.S. Zafarghandi, M. Mohammadi, E. Babolian and S. Javadi, *A localized Newton basis functions meshless method for the numerical solution of the 2D nonlinear coupled Burgers equations*, Internat. J. Numer. Methods Heat Fluid Flow. **27** (2017) 2582-2602.