Journal of Prime Research in Mathematics Vol. **13***(2017), 90-106*

ADAPTIVE RADIAL BASIS FUNCTION FOR TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

SYEDA LAILA NAQVI[1], JEREMY LEVESLEY[2], SALMA ALI[3]

Abstract. We propose a meshless adaptive solution of the time-dependent partial differential equations (PDE) using radial basis functions (RBFs). The approximate solution to the PDE is obtained using multiquadrics (MQ). We choose MQ because of its exponential convergence for sufficiently smooth functions. The solution of partial differential equations arising in science and engineering frequently have large variations occurring over small portion of the physical domain. The challenge then is to resolve the solution behaviour there. For the sake of efficiency we require a finer grid in those parts of the physical domain whereas a much coarser grid can be used otherwise. Local scattered data reconstruction is used to compute an error indicator to decide where nodes should be placed. We use polyharmonic spline approximation in this step. The performance of the method is shown for numerical examples of one dimensional Kortweg-de-Vries equation, Burger's equation and Allen-Cahn equation.

1. Introduction

Partial differential equations (PDEs) with solutions that have highly localized properties appear in many areas of application, such as combustion, shock hydrodynamics and transport in porous media. The solutions of such PDEs sometimes exhibit steep gradients, corners, and rapid topological changes. Such examples include, shock hydrodynamics [24, 7], transport in turbulent flow fields, moving front problems [9, 4], combustion processes [1], reactive or non-reactive flows [18], and singularities in interface flows [5]. The solution, if exits, is difficult to obtain analytically so one needs to use a numerical approach. There are cases when even numerical solution becomes hard to achieve, such as PDEs solved in complex geometries with nonlinearities.

[1]Abdus Salam School of Mathematical Sciences, Lahore. Email: naqvi_maths@yahoo.com
[2]Department of Mathematics, University of Leicester, UK. Email: jl1@le.ac.uk
[3]Shaheed Benazir Bhutto Women University , Pakistan. Email: salma.mathematics@gmail.com .

An ongoing research area, in most of the engineering and science problems is numerical simulation. Most of the techniques for these simulations in complex geometries depends upon discretizing the domain using a grid or triangulation. The major methods include the finite difference (FDM), finite volume (FVM), finite element (FEM), and spectral methods. The discretization in all of these traditional methods involves some sort of mesh generation or triangulation of the region of interest. This may result in an increase in computational cost, especially in high dimensions.

The use of a uniform mesh for problems with localized features can be computationally expensive especially in multidimensions, where the required degrees of freedom (DoF) can be prohibitively large. An alternative to the uniform grid, called *mesh adaptivity* is to flag more points in the region of high variation, and few points in rest of the domain. Mesh adaptivity is preferred over a uniform grid for the sake of efficiency. The adaptive grid should reflect the profile of the solution. Information concerning adaptive methods using different approaches can be found in [12, 8, 14, 13, 9, 19, 22], and references therein.

Radial basis function (RBF) methods are not tied to a grid and in turn belong to a category of the aforementioned meshfree methods.

2. Radial basis function

Given a set of centers $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \subset \mathbb{R}^d$, the RBF interpolant takes the form

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(\| \mathbf{x} - \mathbf{x}_j \|_2), \quad \mathbf{x} \in \mathbb{R}^d, \tag{1}$$

where $\phi(r)$ is defined for $r \geq 0$ and the coefficients, λ, are chosen by enforcing the interpolation condition $s(\mathbf{x}_i) = f(\mathbf{x}_i)$ at a set of nodes that typically coincides with the centers. Enforcing the interpolation condition at N centers results in a $N \times N$ linear system.

$$\mathbf{A}\ \lambda = f,$$

or in expanded form,

$$\begin{bmatrix} \phi(\| x_1 - x_1 \|_2) & \phi(\| x_1 - x_2 \|_2) & \dots & \phi(\| x_1 - x_N \|_2) \\ \phi(\| x_2 - x_1 \|_2) & \phi(\| x_2 - x_2 \|_2) & \dots & \phi(\| x_2 - x_N \|_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\| x_N - x_1 \|_2) & \phi(\| x_N - x_2 \|_2) & \dots & \phi(\| x_N - x_N \|_2) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix},$$

which has to solved for λ Sometimes the assumption on the form of equation (1), for solution to the scattered data interpolation problem is extended by

adding certain polynomials to the expansion, i.e., $s(\mathbf{x})$ is now assumed to be of the form

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_i \phi(\| \mathbf{x} - \mathbf{x}_j \|) + p(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \tag{2}$$

where $p \in \pi_{m-1}^d$, the polynomials of degree $m-1$ in $\mathbb{R}^d$. The dimension of π_{m-1}^d is

$$M = \binom{d+m-1}{m-1}.$$

The choice of m is motivated by the degree of *conditional positive definiteness* of the function ϕ; see [23]. Equation (2) can be written as

$$s(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi(||\mathbf{x} - \mathbf{x}_j||) + \sum_{l=1}^{M} d_l p_l(\mathbf{x}). \tag{3}$$

Enforcing the interpolation conditions $s(\mathbf{x}_i) = f(\mathbf{x}_i), i = 1, \cdots, N$, leads to a system of N linear equations in $N + M$ unknowns λ_j and d_l, one usually adds M additional conditions to ensure a unique solution. A popular choice of conditions, as with the univariate natural boundary conditions, is

$$\sum_{j=1}^{N} \lambda_j p_l(\mathbf{x}_j) = 0, \quad l = 1, \cdots, M.$$

The classical choice for radial basis function ϕ along with their order m is given in the following table .

Table 1. Radial Basis Functions.

RadialBasisFunction	$\varphi(r) =$	*Parameters*	*Order* m
Polyharmonic Splines	r^ν	$\nu > 0, \nu \notin 2\mathbb{N}$	$m \geq \lceil \frac{\nu}{2} \rceil$
Thin Plate Splines (TPS)	$r^{2k} log(r)$	$k \in \mathbb{N}$	$m \geq k+1$
Gaussians	$e^{-(cr)^2}$	$c > 0$	$m \geq 0$
Multiquadrics(MQs)	$(c^2 + r^2)^{\frac{\nu}{2}}$	$\nu > 0,\ \nu \notin 2\mathbb{N},\ c > 0$	$m \geq \lceil \frac{\nu}{2} \rceil$
Inverse Multiquadrics(IMQs)	$(c^2 + r^2)^{\frac{\nu}{2}}$	$v < 0,\ c > 0$	$m \geq 0$

Our numerical approximations are based on *multiquadrics* (MQ). The method was introduced by Hardy in 1971 and the exponential convergence rate of the MQ interpolation for smooth functions was proved by Madych and Nelson in 1992 [15]. The RBF methods suffer with ill-conditioning while giving good accuracy. This relation is described as the uncertainty principal in RBF interpolation and was documented by Schaback in [20]. The infinitely smooth RBFs can gives exponential convergence as compared to the piecewise smooth

RBFs where the convergence is of algebraic order. The convergence of the RBFs containing the shape parameter, can be discussed in *stationary* and *non-stationary* settings. In stationary approximation, the ratio of the shape parameter to the number of centers remains more or less fixed, whereas in a non-stationary approximation keeps the ratio will either decrease or increase. In the non-stationary setting with increasing N the error behaves as

$$|f(\mathbf{x}) - s(\mathbf{x})| \leq e^{-\frac{K(c)}{h}}, \tag{4}$$

where $K(c)$ is a constant depends on the value of the shape parameter and h is the fill distance. However, since condition numbers can be high, errors occur due to rounding, and it is often difficult to obtain highly accurate results. In the stationary setting, the estimate (4) cannot be used since the exponential exponent remains more or less fixed.

Interpolation by radial basis functions is a viable choice in several adaptive methods in a number of time-independent and time-dependent settings. For instance, Driscoll and Heryudono [9] have proposed the resudial subsampling method for interpolation, boundary-value, and initial-boundary-value problems with high degrees of localization. Schaback et al. [11] have used a greedy adaptive algorithm where the method is of linear convergence. Behrens et al.[4] have combined a semi-Lagrangian method with local RBF approximation, specifically, thin plate splines, for linear transport equations. An extension of the method can be in seen in [2] for the non-linear transport equations i.e., Burgers equation and the Buckley-Leverett equation which describes a two-phase fluid flow in a porous medium. Davydov [8] has proposed an adaptive meshless discretization based on generalized finite difference stencils. To generate the finite difference stencils they preferred RBFs due to their robustness on highly irregular data over the local polynomial least square method which is the best known approach considered in literature for the generation of the stencils. Iske et al.[13] combined the adaptive particle methods with the scattered data reconstruction through polyharmonic splines, which plays a key role in their method. The numerical stability and approximation behaviour of the polyharmonic splines have been discussed. Their method performed well on real world problems, for instance, tracer transport problem in the arctic stratosphere and a popular test case scenario from hydrocarbon reservoir modelling, termed the five-spot problem. Sarra [19] modified a simple moving grid algorithm from finite differences to RBF methods.

3. Adaptive method

The essence of most of the adaptive methods for time-dependent PDEs is a cyclic procedure [9]: solve $\rightarrow$ error indicator $\rightarrow$ refine/coarsen, and the cycle terminate when a stopping criterion is satisfied. The procedure *solve* in the time-dependent sense means that the solutions to the PDEs are obtained by

marching in time using any explicit/implicit ODE solver. By space discretization using MQ-RBF in a method of lines scheme, we get a system of ODEs to advance in time for the solution of underlying PDE. Numerical experiments shows that adjustment of shape parameters with the number of centers can produce invertible, well-conditioned interpolation matrices in finite precision arithmetic. The adjustment of shape parameters means that every center will have its own shape parameter and the size of shape parameter will be related to the distance to its nearest neighbours. Once the approximate solutions are obtained the information is then passed to the *error indicator* which gathers the information about the local approximation quality of the the interpolation around $\mathbf{x} \in \mathbf{X}$.

The error indicator can be considered as a function of the node set $\mathbf{X}$ i.e., $\eta : \mathbf{X} \to [0, \infty]$ which assigns a significance value to each node $\mathbf{x} \in \mathbf{X}$. Let $s_{\mathcal{N}_\mathbf{x}}$ be the polyharmonic spline reconstruction which matches the solution values at a node set $\mathcal{N}_\mathbf{x} \subset \mathbf{X} \backslash \mathbf{x}$ in a neighbourhood around $\mathbf{x}$ but not at $\mathbf{x}$:

$$s_{\mathcal{N}_\mathbf{x}}(v) = u(v) \quad \forall v \in \mathcal{N}_\mathbf{x}. \tag{5}$$

The adaption relies on the error indicator defined as

$$\eta(\mathbf{x}) = |u(\mathbf{x}) - s_{\mathcal{N}_\mathbf{x}}(\mathbf{x})| \text{ for all } \mathbf{x} \in \mathbf{X}, \tag{6}$$

where u is the approximation to the true solution, obtained by the MQ-RBF.

The error indicator will be high if the approximate solution and a second approximation deviate significantly in a region of interest. Then we will add new points into this region. If there is little deviation we will consider removing points.

The polyharmonic splines are of the form,

$$\Phi_{d,k}(r) = \begin{cases} r^{2k-d} \log(r), & \text{if } d \text{ is even,} \\ r^{2k-d}, & \text{if } d \text{ is odd,} \end{cases} \tag{7}$$

where k is required to satisfy $2k > d$. We will take the order $m = k$ for $\Phi_{d,k}(r) \in CPD_d(m)$. This particular choice of m rather than the minimal choice $m = k - \lceil d/2 \rceil + 1$ means that, the Beppo Levi space $BL^k(\mathbb{R}^d)$, given by

$$BL^k(\mathbb{R}^d) = \{f \in C(\mathbb{R}^d) : |f|_k = \sum_{|\alpha|=k} \frac{k!}{\alpha!} \|D^\alpha u\|_2 < \infty\}, \tag{8}$$

is the optimal recovery space for the polyharmonic splines kernel $\Phi_{d,k}$; see the seminal papers of Meinguet [16, 17]. With this choice of $m = k$ the interpolant in (2) takes the form

$$s(\mathbf{x}) = \sum_{j=1}^{N} c_j \Phi_{d,k}(\|\mathbf{x} - \mathbf{x}_j\|) + \sum_{|\alpha|<k} d_\alpha \mathbf{x}^\alpha. \tag{9}$$

Scattered data approximation by polyharmonic splines is optimal in the native reproducing kernel semi-Hilbert space, which in the case of polyharmonic splines, is the above Beppo Levi space; see [10].

The local residual is the criteria for mesh refinement or coarsening. The decision is then passed to the *refine/coarse* stage. Refinement is 1-D context is, adding two nodes (left and right) around the node which lies in the region of high activity. We found that removal of all of the nodes which are less than the coarsening threshold θ_{crs} is not an effective approach in our experiments. This strategy can lower approximation quality too much in the smooth regions. We therefore sort the residuals and remove a percentage i.e., residuals are ranked from worse to least worse and we only remove points from those nodes associated with a certain percentage at the poor end of the scale. This percentage depends on the underlying problem. Coarsening a percentage is a crucial step in our adaptive method.

4. Numerical experiments for time-dependent PDEs

We will now present numerical results for one-dimensional time-dependent PDEs .The thresholds used for the refining and coarsening are denoted by θ_{ref} and θ_{crs} respectively. The choice of the thresholds is problem dependent. We will be using the classical Runge-Kutta (RK4) method for intergrading in time unless otherwise stated. All the numerical experiments are run in MATLAB on Windows 7 system running at 3.10 Ghz with 8 GB memory.

4.1. **Korteweg-de Vries (KdV) propagation of a single soliton.**

The KdV equation first arose in connection with water waves, but it was later realized that it is a model equation which balances nonlinearity and dispersion. The equation can be used to study nonlinear waves with an effect of dispersion in any medium. Korteweg and de Vries showed that unidirectional propagation of small-amplitude long waves in water of relatively shallow depth could be described approximately by a nonlinear partial differential equation. A solitary wave is one of the solutions which exist for this model. The conventional, nondimensional version of the third-order nonlinear KdV equation is

$$u_t + \epsilon u u_x + \mu u_{xxx} = 0, \tag{10}$$

The boundary and initial values can be derived from the exact solution given as

$$u(x,t) = 2\sec h^2(x - 4t). \tag{11}$$

Here u represents the height from the average water surface and x is the coordinate moving with the velocity of a linearized wave with $u \to 0$ as $|x| \to \infty$, where u is dependent variable, x is a spatial independent variable, t is the time independent variable. The numbers ϵ, μ are real constants.

In this example, we study a single soliton solution (10) with an exact solution given in (11). We choose $\epsilon = 6$ and $\mu = 1$. The initial condition $u(x, 0)$ and boundary functions f, g can be obtained from the exact solution in (11). The computational domain is $[-10, 40] \times [0, 5]$. The adaptive profiles of the single soliton for $t = 0, 1, 2, 3, 4, 5$ can be seen in Figures 1-3. The thresholds for refining $\theta_{ref} = 10^{-3}$ and for coarsening $\theta_{crs} = 10^{-6}$. The method is advanced in time using the classical RK4 method. Initially we started with 70 equally spaced points and adapted the solution at every time level. For the one-dimensional problems we have used the polyharmonic splines which is the natural spline of order $2k$. For instance, for KdV equation the polyharmonic spline will be of the form

$$\phi_{1,k} = r^{2k-1} \tag{12}$$

with the order $m = k \geq 1$. In our examples we have used $k = 3$ which will be of the form

$$\phi_{1,3} = \| x - x_j \|^5 + d_0 + d_1 x + d_2 x^2. \tag{13}$$

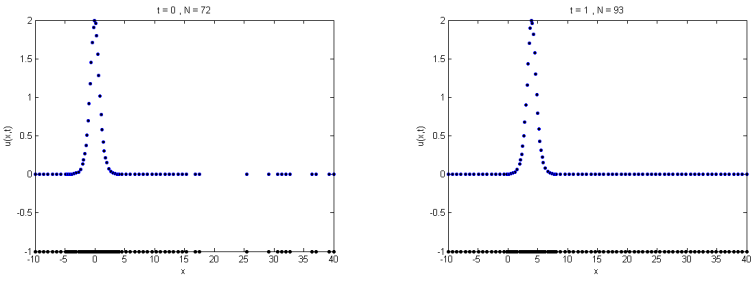

FIGURE 1. Left: Initial adaptive discretization at time $t = 0$, Right: Soliton moving from left to right, adaptive solution at time $t = 1$.

TABLE 2. Comparison of Adaptive and Non-Adaptive methods for the single soliton at final time $t = 5$

Adaptive/Non-Adaptive RBF	DoF	Max-Error	RMS-Error
Non-Adaptive MQs	151	$5.8845E-3$	$2.9854E-3$
Non-Adaptive Guassian	151	$1.1698E-3$	$5.78964E-4$
Non-Adaptive IMQs	251	$3.8734E-3$	$1.1961E-3$
Adaptive MQ	94	$8.6922E-3$	$3.2295E-3$

On the initial discretization of nodes, we adaptively select the nodes and the shape parameters $t_0 = 0$ (Figure 1). We then marched in time to obtain

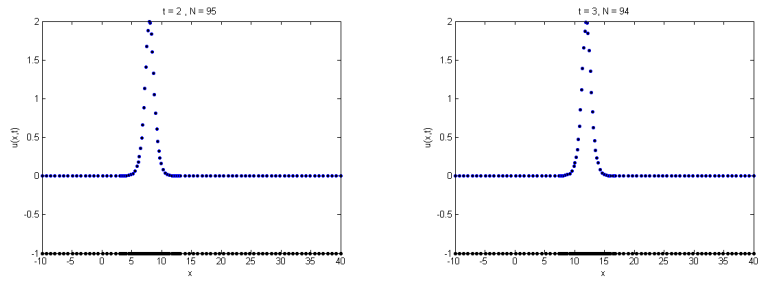

FIGURE 2. The profile of the adaptive solution at time $t = 2$ and $t = 3$

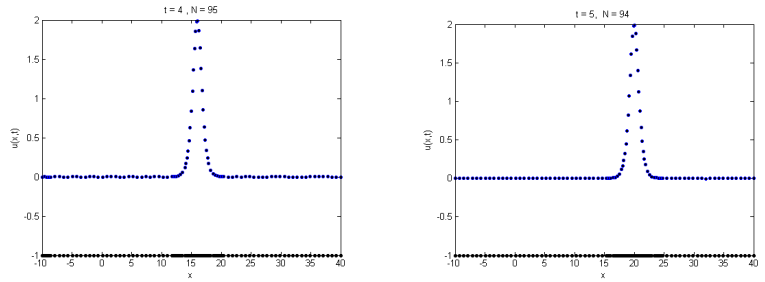

FIGURE 3. The profile of the adaptive solution at time $t = 4$ and $t = 5$

the solution at time $t = t_0 + \Delta t$, using explicit RK4. Once the solution is obtained we compute an RBF interpolant to approximate the solutions at time $t = t_0 + \Delta t$, we can now apply the adaptive algorithm which will give us a new set of adaptively selected nodes and shape parameters. The procedure will continue for every time level and will stop when the final time arrives. The adaptive method uses almost half the number of points of the non-adaptive method for the KdV (see Table 2). The performance of the method is compared using the maximum error, $\| u^N - u \|_{max} = \max\limits_{1 \le i \le N} \mid u_i^N - u_i \mid$ and the root mean square (RMS), $\sqrt{\frac{\sum_i^N (u_i^N - u_i)^2}{N}}$. The adaptive method just shifts the points with increasing time, which is due to movement of the soliton solution of the problem. We can see from Table 2 that the adaptive method has efficiently solved the problem with the same accuracy compared to non-adaptive RBF but with less number of nodes.

4.2. **Interaction of two solitons.** This example studies the interaction of the two solitons solution of Equation 10 with $\epsilon = 6$ and $\mu = 1$. The initial

condition is:

$$u(x,t_0) = 12\frac{3+4\cosh(2x-8t_0)+\cosh(4x-64t_0)}{(3\cosh(x-28t_0)+\cosh(3x-36t_0)^2}, \tag{14}$$

where t_0 is the initial time. The exact solution of 10 for interaction of two soliton is given by

$$u(x,t) = 12\frac{3+4\cosh(2x-8t)+\cosh(4x-64t)}{(3\cosh(x-28t)+\cosh(3x-36t)^2}. \tag{15}$$

The boundary functions f, g can be obtained from the exact solution.

The computational domain is $[-15,15]\times[-0.3,0.3]$. The profile of interaction of the two solitons shows that the adaptive method is able to track the developing features of the profile of the solution. We have recorded the results for $t = -0.3$, -0.25, -0.2, -0.15, -0.1, -0.05, 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 in Figures (4-9).

We started the adaptive interpolation with 80 equally spaced points where the refining threshold $\theta_{ref} = 10^{-3}$ whereas the coarsening $\theta_{crs} = 10^{-7}$. To maintain the balance in refining and deleting the number of points we coarsened 70% of the sorted errors. We solved this problem with MATLAB built-in ode15s ODE solver with $\Delta t = 0.0001$. The adaptation occurred at every time step.

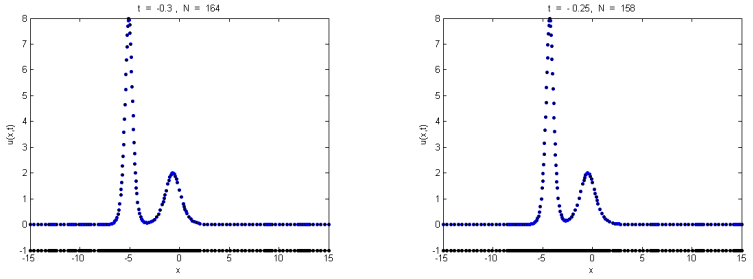

FIGURE 4. Interaction of two solitons moving from right to left, the profile at time $t = -0.3$, $t = -0.25$

Figure 4 is the profile of solitons moving from right to the left. We recorded this before the interaction actually happens. We can see that the adaptive algorithm is using a finer set of points where required.

Interaction of the two waves can be seen in Figure (5). We observed and recorded the results at the time levels when the interaction actually happened. We can see that at $t = -0.05$ the shorter wave passes through the bigger wave towards the left. At time $t = 0.1$ it interacted with a total number of adaptively adjusted points $N = 152$.

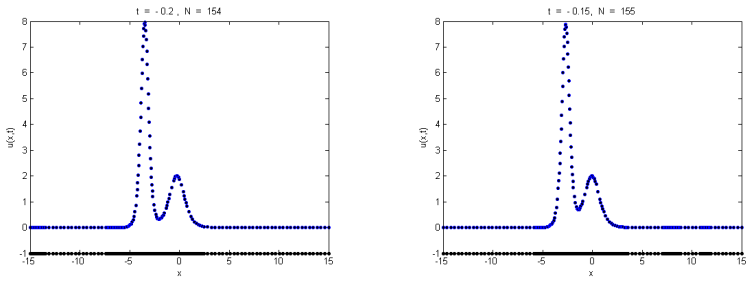

FIGURE 5. Interaction of two solitons moving from right to left, this profile is just before the interaction and recorded at time $t = -0.2$ and -0.15

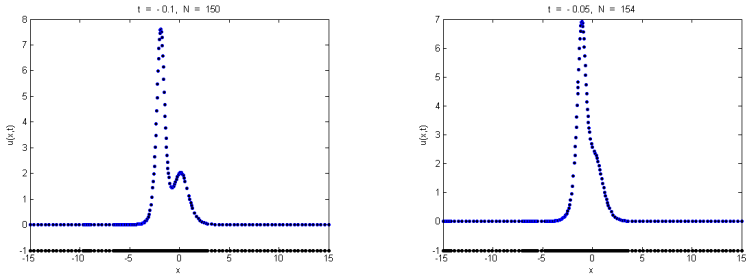

FIGURE 6. The smaller solitary wave is interacting with the larger wave at time $t = -0.1$ and $t = -0.05$

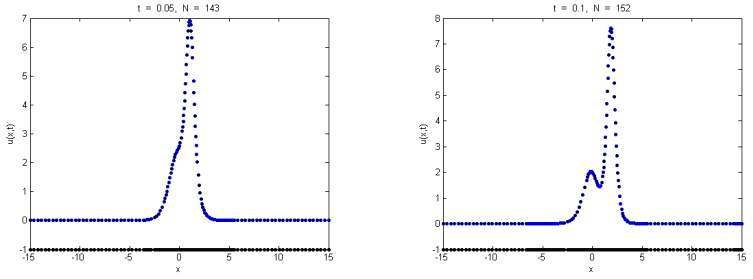

FIGURE 7. The smaller solitary wave is interacting with the larger wave at time $t = -0.1$, $t = -0.05$, $t = 0.05$ and $t = 0.1$.

Figure (9) shows that taller wave catches up, interacts with the shorter one and then passes towards the left. We can see that this is like almost no effect of interaction at all. Adaptive algorithm performed well to place a finer grid

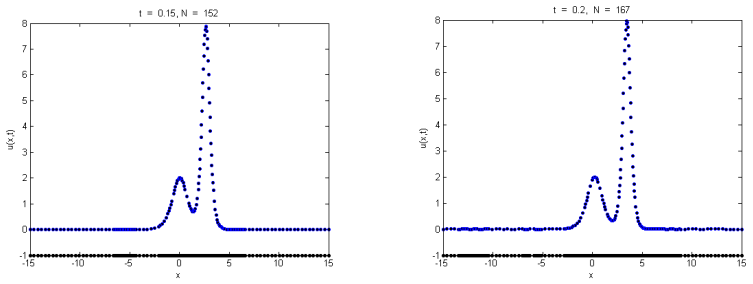

FIGURE 8. At time $t = 0.15$ and $t = 0.2$ the effects of interaction

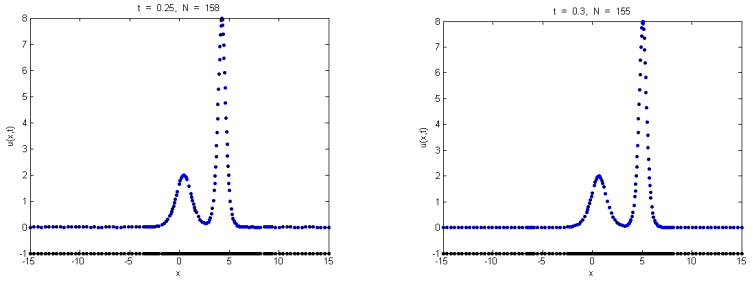

FIGURE 9. The waves retain to the original shape at $t = 0.25$, $t = 0.3$

in a region of high variation. The process ends up with a total number of $N = 155$ adaptively distributed points.

TABLE 3. Comparison of Adaptive and Non-Adaptive methods for the of interaction of two solitons at final time $t = 0.3$

Adaptive/Non-Adaptive RBF	Degrees of Freedom	Max-Error	RMS errors
MQs	201	$9.5E-3$	$4.7E-3$
Gaussian	201	$1.4E-3$	$5.4E-4$
IMQs	201	$3.4E-3$	$5.7E-3$
Adaptive MQ	155	$4.9E-3$	$1.5E-3$

The adaptive method captured the developing features the solution at all times. In Table 3 we presented a comparison with the non-adaptive RBF method [21].

4.3. **Burger's equation.** Burgers equation is a nonlinear advection-diffusion problem [6]. It is defined as

$$\epsilon u_{xx} - uu_x = u_t, \tag{16}$$

where $\epsilon \geq 0$ is a given parameter and will be chosen such that it develops a shock. The solution then exhibits moving fronts that can be made arbitrarily sharp by decreasing the kinematic viscosity ϵ as a coefficient of the second-derivative uu_{xx} term which is the dissipation term.

Burgers equation appears in various fields of applied mathematics. The applications can be seen in modelling of fluid dynamic, shock wave formation, traffic flow, turbulence, boundary layer behaviour. Initially the Burgers equation was introduced to describe turbulence in one space dimension.

We will now consider this moving front problem given by (16) which is a common test problem for adaptive methods; see [9] and references therein. The initial condition is

$$u(\mathbf{x}, 0) = \sin(2\pi x) + \frac{1}{2}\sin(\pi x), \tag{17}$$

with the boundary conditions

$$u(0, t) = u(1, t) = 0. \tag{18}$$

The boundary conditions are taken to be zero which diminish the amplitude of the wave with increasing time. Here $u(x, t)$ is a wave that generates a steep front of width $O(\epsilon)$ and moves towards $x = 1$. This is one of the principal reasons which makes Burger's equation a stringent test problem. The solution $u(x, t)$ steepens with increasing time t and become difficult to resolve spatially.

The computational domain we are using is $(0, 1) \times [0, 1]$ with parameter $\epsilon = 10^-3$. Solution is advanced in time using the MATLAB built-in stiff ode-solver ode15s with $\Delta t = 0.01$. We have given the Jacobian matrix in *odetset* for the stiff solver $ode15s$ to work faster. The thresholds $\theta_{ref} = 10^{-5}$ and $\theta_{crs} = 10^{-8}$. The method starts initially with 13 equally spaced nodes and the adaptation occurs at every time level until the final time arrives. All the residuals becomes less than 10^{-5} with a total number of 155 adaptively selected nodes in the domain. The coarsening percentage for this problem is 20% of the sorted array of best errors. The polyharmonic splines $\phi_{1,2} = r^3$ will be used for the error estimator for Burgers equation.

For the given smooth initial condition (18) the solution is moving towards 1, and generating a steep moving front. Due to the zero boundary conditions solution decays with time. In Figures (10-12) we can observe that our adaptive method is performing well and adaptively refine path of the moving front. The residuals are less than 2×10^{-5} as compared to 5×10^{-4} in [9].

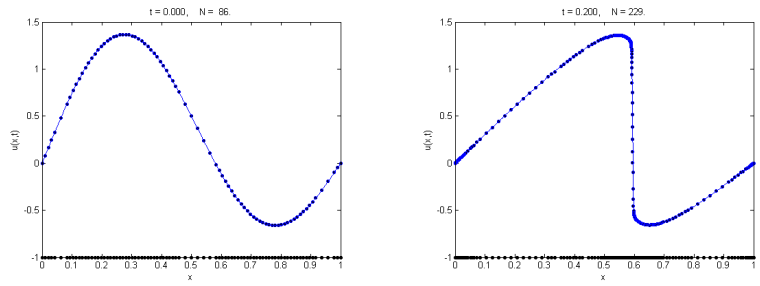

FIGURE 10. Adaptive RBF method for the Burgers equation at time $t = 0.4$ and $t = 0.6$

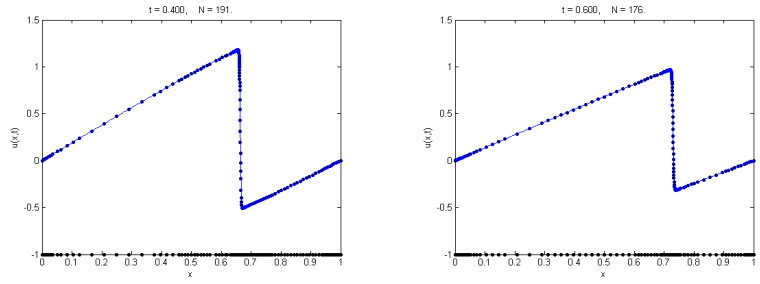

FIGURE 11. Adaptive RBF method for the Burgers equation at time $t = 0.4$ and $t = 0.6$

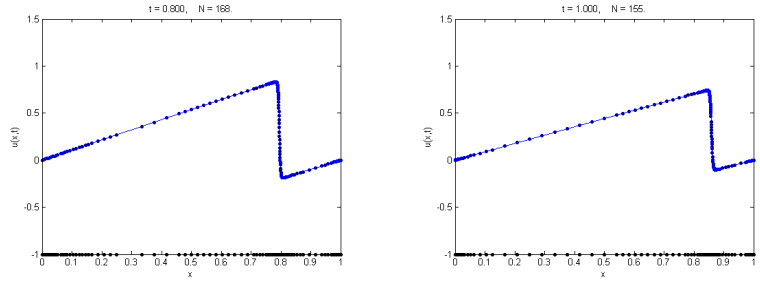

FIGURE 12. Adaptive RBF method for the Burgers equation at time $t = 0.6$ and $t = 0.8$

4.4. **Allen-Chan Equation.** The Allen-Chan equation is defined as:

$$u_t = u(1 - u^2) = \nu u_{xx} \tag{19}$$

where the parameter $\nu = 10^{-}6$ and the computational domain $[-1, 1] \times [0, 8.25]$, the initial condition is:

$$u(x, 0) = 0.6x + 0.4 \sin[\frac{\pi}{2}(x^2 - 3x - 1)] \tag{20}$$

and the boundary conditions are:

$$u(-1, t) = -1, u(1, t) = 1 \tag{21}$$

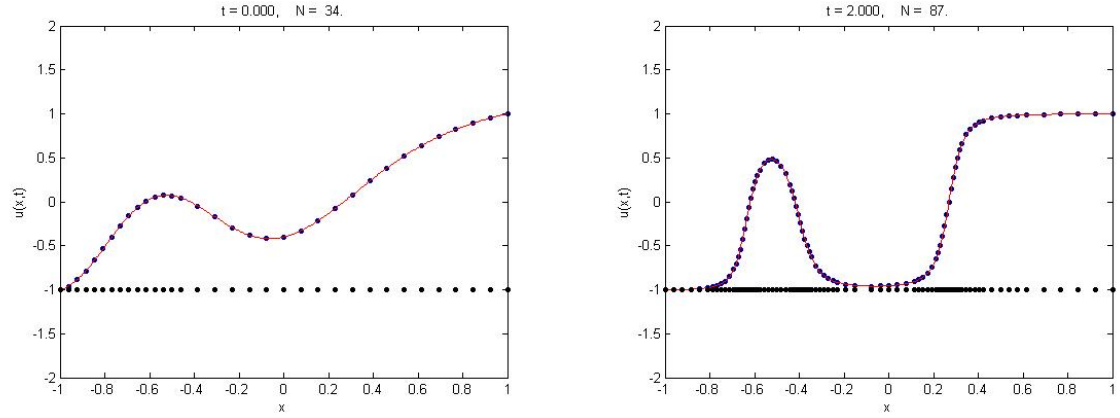

FIGURE 13. Adaptive solution for the Allen-Cahn equation at at time $t = 0$, and $t = 2$

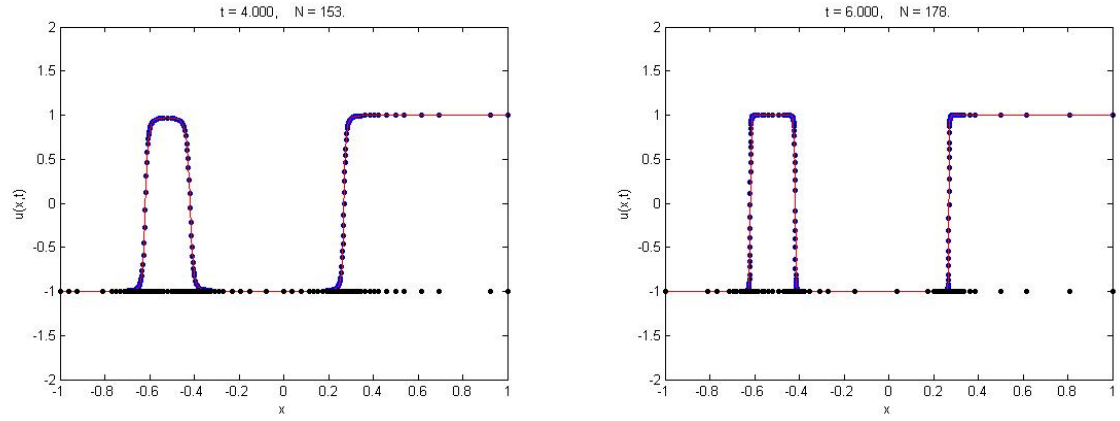

FIGURE 14. Adaptive solution for the Allen-Cahn at time $t = 4$ and $t = 6$

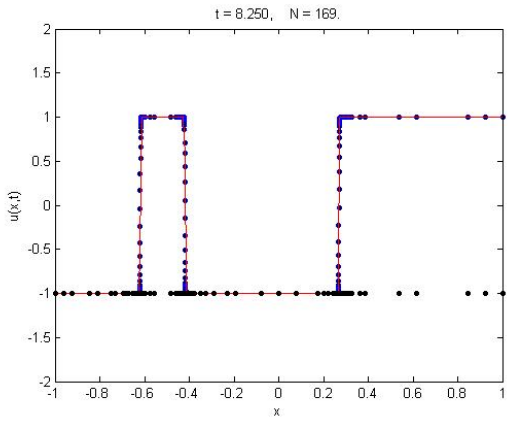

FIGURE 15. Adaptive solution at the final time $t = 5$

In this experiment we have used parameter $\nu = 10^{-6}$ and the computational domain $[-1, 1] \times [0, 8.25]$. The adaptation occurs every 15 time steps since we observed that the solution is not changing its profile rapidly. We started the algorithm with 30 initially equally distributed points in the domain. The thresholds are $\theta_{crs} = 10^{-5}$ and $\theta_{ref} = 10^{-3}$. In this experiment we coarsened 30 percent of the residuals. Figures (13-15) are depicting that adaptive algorithm efficiently place a finer grid in the region of high activity and coarsen where the solution lies in smooth region. The fourth order Runge-Kutta method is used for time stepping and the step size we used is 0.01. All the residuals are below 10^{-3} at the final time and we ended up to a total number of 169 adaptively selected nodes as compared to 178 nodes in [9].

5. CONCLUSIONS

The adaptive spectral methods are preferred for problems with steep gradients, sharp corners, moving fronts and with singularities. We have developed the adaptive scheme for a single soliton and interaction of two solitons solution of the third order nonlinear KdV equation. A large number of nonlinear dispersive systems can be described by the KdV equation or one of its modified form. Some interesting numerical results are found during this journey, but keeping in mind that the RBF is relatively new subject much of its behaviour still requires rigourous justification. In this context the theoretical analysis is far from complete. Our method uses MQ-RBFs to approximate the solution, making use of its spectral orders of convergence. For indicating regions of high error, we reconstruct the solutions in a local neighbourhood around

every node using polyharmonic splines, which is a popular choice for adaptive schemes [3]. For instance, $\eta(\mathbf{x}) = |u(\mathbf{x}) - s_{\mathcal{N}_\mathbf{x}}(\mathbf{x})|$ for all $\mathbf{x} \in \mathbf{X}$, where $u(\mathbf{x})$ is the MQ approximation at the point and $s_{\mathcal{N}_\mathbf{x}}(\mathbf{x})$, is the local reconstruction around $\mathbf{x}$ but not at $\mathbf{x}$. The polyharmonic splines for the 1-D case becomes the natural splines, for the KdV equation we have used $\phi_{1,3} = |x - x_i|^5$.

Our goal was to obtain a numerical solution with as few degrees of freedom as possible, by using an adaptive approach such that the errors are below a prescribed threshold. Solving the problem with fewer DoF can be efficient in saving the computer storage and using adaptation at selected intermediate levels can be computationally less expensive.

The implementation of the schemes results in problem solutions with almost half the number of the grid points than the non-adaptive scheme. The comparison can be seen in Table 2 for a single soliton and Table 3 for interaction of two solitons. The results show that our method is able to track the developing features in the solution profile. The method has also shown promising results for the second order nonlinear difficult problems, i.e., the Burgers equation with the moving front solution and the Allen-Cahn equation. The method is able to add more points in the region of high variation.

For future work we intend to solve higher dimensional problems since the real power of the RBF method is to utilise its robustness to point position (unlike FEM methods) to solve such problems.

References

[1] M.J.H. Anthonissen. *Local defect correction techniques: analysis and application to combustion*. Eindhoven University of Technology, Eindhoven, 2001. Dissertation, Technische Universiteit Eindhoven, Eindhoven, 2001.

[2] J. Behrens and A. Iske. Grid-free adaptive semi-Lagrangian advection using radial basis functions. *Comput. Math. Appl.*, 43(3-5):319–327, 2002. Radial basis functions and partial differential equations.

[3] J. Behrens, A. Iske, and M. Käser. Adaptive meshfree method of backward characteristics for nonlinear transport equations. In *Meshfree methods for partial differential equations (Bonn, 2001)*, volume 26 of *Lect. Notes Comput. Sci. Eng.*, pages 21–36. Springer, Berlin, 2003.

[4] J. Behrens, A. Iske, and Käserm M. Adaptive meshfree method of backward characteristics for nonlinear transport equations. In *Meshfree methods for partial differential equations (Bonn, 2001)*, volume 26 of *Lect. Notes Comput. Sci. Eng.*, pages 21–36. Springer, Berlin, 2003.

[5] A.L. Bertozzi, M.P. Brenner, T. F. Dupont, and L. P. Kadanoff. Singularities and similarities in interface flows. In *Trends and perspectives in applied mathematics*, volume 100 of *Appl. Math. Sci.*, pages 155–208. Springer, New York, 1994.

[6] J. M. Burgers. A mathematical model illustrating the theory of turbulence. In *Advances in Applied Mechanics*, pages 171–199. Academic Press Inc., New York, N. Y., 1948. edited by Richard von Mises and Theodore von Kármán,.

[7] P. Cruza, M. Alvesb, F.D. Magalhesa, and A. Mendes. Solution of hyperbolic pdes using a stable adaptive multiresolution method. *Chemical Enginerring Science*, 58:1777–1792, 2003.

[8] O. Davydov and D.T. Oanh. Adaptive meshless centres and RBF stencils for Poisson equation. *J. Comput. Phys.*, 230(2):287–304, 2011.

[9] T.A. Driscoll and A.R.H. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Comput. Math. Appl.*, 53(6):927–939, 2007.

[10] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive theory of functions of several variables (Proc. Conf., Math. Res. Inst., Oberwolfach, 1976)*, pages 85–100. Lecture Notes in Math., Vol. 571. Springer, Berlin, 1977.

[11] Y. C. Hon, R. Schaback, and X. Zhou. An adaptive greedy algorithm for solving large RBF collocation problems. *Numer. Algorithms*, 32(1):13–25, 2003.

[12] C. Hong, K. Li, and L. Wen-Jun. Numerical solution of pdes via integrated radial basis function networks with adaptive training algorithm. *Applied Soft Computing.*, 11:855–860, 2010.

[13] A. Iske. Particle flow simulation by using polyharmonic splines. In *Algorithms for approximation*, pages 83–102. Springer, Berlin, 2007.

[14] N.A. Libre, A. Emdadi, Shekarchi M. Kansa, E.J., and M. Rahimian. A fast adaptive wavelet scheme in RBF collocation for nearly singular potential PDEs. *CMES Comput. Model. Eng. Sci.*, 38(3):263–284, 2008.

[15] W. R. Madych and S. A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *J. Approx. Theory*, 70(1):94–114, 1992.

[16] J. Meinguet. Basic mathematical aspects of surface spline interpolation. In *Numerische Integration (Tagung, Math. Forschungsinst., Oberwolfach, 1978)*, volume 45 of *Internat. Ser. Numer. Math.*, pages 211–220. Birkhäuser, Basel, 1979.

[17] J. Meinguet. An intrinsic approach to multivariate spline interpolation at arbitrary points. In *Polynomial and spline approximation (Proc. NATO Adv. Study Inst., Univ. Calgary, Calgary, Alta., 1978)*, volume 49 of *NATO Adv. Study Inst. Ser. C: Math. Phys. Sci.*, pages 163–190. Reidel, Dordrecht, 1979.

[18] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.*, 188(2):493–523, 2003.

[19] S.A. Sarra. Adaptive radial basis function methods for time dependent partial differential equations. *Appl. Numer. Math.*, 54(1):79–94, 2005.

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

[21] Q. Shen. A meshless method of lines for the numerical solution of KdV equation using radial basis functions. *Eng. Anal. Bound. Elem.*, 33(10):1171–1180, 2009.

[22] L. Vrankar, E. J. Kansa, L. Ling, F. Runovc, and G. Turk. Moving-boundary problems solved by adaptive radial basis functions. *Comput. & Fluids*, 39(9):1480–1490, 2010.

[23] H. Wendland. *Scattered data approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2005.

[24] J. Xue and G. Liao. Least-squares finite element method on adaptive grid for PDEs with shocks. *Numer. Methods Partial Differential Equations*, 22(1):114–127, 2006.