TWMS J. App. and Eng. Math. V.10, N.4, 2020, pp. 897-908

ADAPTIVE MOVING MESH FOR SOLVING THE KORTEWEG-DE VRIES-BURGERS EQUATION

ADULGHANI ALHARBI, §

Abstract. This paper presents an adaptive mesh method for solving the Korteweg-de Vries-Burgers (KdVB) equation numerically. The r-adaptive mesh method which employs a monitor function and one of the mesh equations (MMPDEs) presented by [11] which uses to concentrate and move the mesh. This technique correctly resolves the various structures seen in these problems. Furthermore, it decreases the computational effort in comparison to the numerical solution using a fixed uniform mesh scheme. Here, the exact solution of the KdVB equation at a fixed time is utilized to compare with that employing an adaptive moving mesh and a uniform mesh. By comparing the relative error for the obtained solutions of the problem, I will check the accuracy of the adaptive method.

Keywords: KdVB equation; Adaptive mesh; r-adaptive method; Moving mesh PDEs (MMPDEs); monitor function.

AMS Subject Classification: 65N06, 65N40, 65N45, 65N50.

1. Introduction

The Korteweg-de Vries-Burger's (KdVB) equation is of the form:

$$h_t + \epsilon h h_x - \nu h_{xx} + \mu h_{xxx} = 0, \tag{1}$$

where, $h = h(x,t)$, x and t are referred to as the spatial variable and time, respectively. The parameters $\epsilon > 0$, ν, and μ are referred to as the flow velocity coefficient, the diffusion coefficient, and the coefficient of the dispersive term, respectively, with $\epsilon\nu\mu$ is a non-negative number. Such an equation appears from several various physical phenomena as a model of a non-linear problem containing the effects of non-linearity, dispersion, and dissipation. Eq. (1) is derived by Johnson [1] as the controlling equation for waves spreading in a liquid-filled flexible pipe in which the effects of non-linearity, dispersion, and dissipation have appeared. Van Wijngaarden [2] utilised the equation as a model of non-linear in the flow of liquids containing gas bubbles. It is used by [3, 4] as a turbulence model. Moreover, Grua and Hu [5] utilised a steady-state of Eq. (1) to characterise a small

Taibah University, Faculty of Science, Mathematics Department, Universities Road, PO Box: 344, Medina, Kingdom of Saudi Arabia.
e-mail: abdul928@hotmail.com, arharbi@taibahu.edu.sa
ORCID: https://orcid.org/0000-0002-1430-4684.

§ Manuscript received: January 9, 2019; accepted: May 6, 2019.
TWMS Journal of Applied and Engineering Mathematics, Vol.10, No.4;

shock profile in plasmas. Numerous theoretical studies mattering the KdVB equation have been given enormous interest for the past many decades. Primarily, the solution to the KdV equation has significantly obtained. Johnson [1] studied the solution to the KdVB problem in the stage plane utilising a disturbance method in the areas where $\nu \ll \mu$, $\mu \ll \nu$. Grua and Hu [6] examined the same equation applying a method comparable to that employed by Johnson [1] and a related comparison was considered by Jeffrey [7]. Canosa, Gazdag and Uretaev [8, 9] presented numerically the approximation solution of Eq. (1). The uniqueness and existence of the bounded wave solution to Eq. (1) are investigated by Bona and Schonbek [10], which tends to fixed states at $\pm\infty$. The purpose of this paper is to show that a more general exact solution to the KdVB equation can indeed be obtained by the Painlevé analysis. Grad and Hu [5] showed that the dissipation effects dominate over dispersive effect when:

$$4\mu \leq \nu^2. \tag{2}$$

In this case the solution of Eq. (1) is a shock decreasing monotonically from the upstream to the downstream value of $h(x,t)$.

$$\text{If:} \quad \nu^2 \leq 4\mu, \tag{3}$$

the dispersive effects dominate over the dissipative effects and in this case, the shock becomes oscillatory upstream and monotonic downstream. Here, the non-uniform mesh scheme is presented by [11, 12, 13] which uses a monitor function and mesh equations to focus and transfer the mesh points coupled to the problem. This non-uniform mesh scheme is referred to as r-adaptive methods. These methods are entirely appropriate for KdVB equation compared to some different techniques. The r-adaptive schemes improve the mesh points based on a monitor function that directly follows particular solution components (for instance, the solution involves significant variations, large curvatures, and shock waves) and therefore could resolve the regions, where the solution has variations, more accurately. The MMPDEs commonly use the standard form of a non-linear diffusion equation [11, 12] which coupled to the problems. r-adaptive schemes can be effortlessly applied using the finite difference method well appropriate for these problems. On the other hand, h-adaptive techniques are commonly performed utilizing the finite element method.

The r-adaptive methods are a modern improvement and are not applied as often as h or hp- refinements, they happily are employed in many applications, for example convective heat transfer [16], meteorological [17, 18] and computational fluid mechanics [14, 15] problems. The r-adaptive mesh schemes make the basis of common purpose openly free solvers for only one dimensional PDE, such as MOVCOL [19] and TOMS731 [20]. Nevertheless, they have just examined for the first and second order parabolic PDEs (such as the Burger's equation [11, 21, 22, 23, 24, 25, 26]) and to our knowledge, this is the first try to use r-adaptive mesh methods for solving the KdVB equation.

1.1. **Exact Solution of the KdVB Equation.** The exact solution of the KdvB equation appeared for the first time for the two dimensional KdVB equation at [27]. It is modified to take the simplified form:

$$h(x,t) = \frac{12\nu^2}{25\epsilon\mu}\left(1 - \frac{exp\left(\frac{2\nu}{5\mu}(x-\omega t)\right)}{\left[exp\left(\frac{\nu}{5\mu}(x-\omega t)\right)+c\right]^2}\right), \tag{4}$$

where c is a positive real number, $\omega = \frac{12\nu^2}{25\mu}$, ϵ, ν and μ are positive parameters.

2. Numerical results on an Adaptive Mesh

The numerical solutions of Eq. (1) are sought with the following boundary conditions

$$h_x(0,t) = h_x(M,t) = 0, \quad h_{xx}(0,t) = h_{xx} = 0, \tag{5}$$

and the initial condition Eq. (4) with taking $t = 0$ using finite difference methods. The numerical method is performed on an adaptive moving mesh.

Here, the exact solution Eq. (4) is used at a fixed time $t = 2$ to study the accuracy of applying non-uniform and uniform meshes. The adaptive mesh method requires to transform from the computational domain to the physical domain utilising the following transformation

$$x = x(\xi,t)\colon \Omega_c \equiv [0,M] \to \Omega_p \equiv [0,M],\ t > 0.$$

Ω_c is called the computational domain, Ω_p is the physical domain, ξ is the computational coordinate, and M is a length of the physical domain. Thus the solution is formed as

$$h(x,t) = h(x(\xi,t),t).$$

A moving mesh associated with the solutions h is described as

$$\mathcal{J}_h(ts)\colon x_j(\xi) = x(\xi_j,t),\ j = 1,\cdots,N+1 \tag{6}$$

where the boundary points are provided by

$$x_1 = \xi_1, \quad x_{N+1} = \xi_{N+1}. \tag{7}$$

A uniform mesh on Ω_c is given by

$$\mathcal{J}_h^c(t)\colon \xi_j = \frac{(j-1)\,M}{N},\ j = 1,\cdots,N+1. \tag{8}$$

Applying the chain rule yields

$$h_x = \frac{h_\xi}{x_\xi},\ h_t = h_t - \frac{h_\xi}{x_\xi}\,x_t. \tag{9}$$

Using the above, Eq. (1) can be written as

$$h_t - \frac{h_\xi}{x_\xi}x_t = -\frac{Q_\xi}{x_\xi}, \tag{10}$$

$$Q = \mu\frac{1}{x_\xi}\left(\frac{h_\xi}{x_\xi}\right)_\xi - \nu\frac{h_\xi}{x_\xi} + \epsilon\frac{h^2}{2}, \tag{11}$$

Notice that the third order derivative is approximated requiring a standard 5-point stencil for h and a 3-point stencil for x. Thus, evaluating Eq. (10) at $j = 2$, and $j = N$ requires fictitious points h_0 and h_{N+2}. Since h is symmetric in x then for $j = 1, N+1$,

$$h_0 = h_1, \qquad h_{N+2} = h_{N+1}$$

The boundary conditions of the solution $h(0,t) = \beta_1$, $h(L,t) = \beta_2$ are taken by their ODE form:

$$h_{t,1} = 0, \quad h_{t,N+1} = 0, \tag{12}$$

where β_1, β_2 are constants as the problem required. The transformation $x = x(\xi,t)$ gained via solving the equation mesh (MMPDEs). Four different MMPDEs are employed, the so-called MMPDE 4, 5 and 6 and modified MMPDE 5, presented in [11]. The finite centred differences are used to discretise the derivatives of the spatial variable. The semi-discretisation of MMPDEs is as follows: (the reader can also refer to [28, 29, 30])

$$\text{MMPDE 4}: \quad (\hat{\rho}x_{t\xi})_\xi = -\frac{1}{\tau}(\hat{\rho}x_\xi)_\xi, \tag{13}$$

$$\text{MMPDE5}: \quad x_t = \frac{1}{\tau}(\hat{\rho}x_\xi)_\xi, \tag{14}$$

$$\text{MMPDE6}: \quad x_{t,\xi\xi} = -\frac{1}{\tau}(\hat{\rho}x_\xi)_\xi, \tag{15}$$

$$\text{Modified MMPDE5}: \quad x_t = \frac{1}{\hat{\rho}\tau}(\hat{\rho}x_\xi)_\xi. \tag{16}$$

Where, $\hat{\rho}(x,t)$ is a monitor function and $\tau > 0$ is a relaxation parameter. τ is the timescale over which the mesh responds to variations in $\hat{\rho}(x,t)$. Experts observed that taking small enough value of the relaxation parameter τ, makes the mesh responds to variations in $\hat{\rho}(x,t)$ [21]. MMPDE 5 and modified MMPDE 5 and their discretization provided in Eqs. (14,16) are usually extremely stiff and, in practice, it is suggested to employ a more normal system,

$$\text{Regularised MMPDE5}: \quad x_t - \gamma_1 x_{t,\xi\xi} = \frac{1}{\tau}(\hat{\rho}x_\xi)_\xi, \tag{17}$$

$$\text{Regularised modified MMPDE5}: \quad x_t - \gamma_1 x_{t,\xi\xi} = \frac{1}{\hat{\rho}\tau}(\hat{\rho}x_\xi)_\xi. \tag{18}$$

Here, the parameter $\gamma_1 > 0$ may be related to $\hat{\rho}$ (read Budd *et al.* [12] and references therein). The boundary conditions of the mesh are taken by

$$x_{t,1} = 0, \;\; x_{t,N+1} = 0. \tag{19}$$

The initial condition

$$x(\xi, 0) = \frac{(j-1)\,M}{N}, \quad j = 1, 2, \ldots, N+1, \tag{20}$$

which represents a uniform mesh as an initial condition on the physical domain $\Omega_p \equiv [0, M]$.

Finally, $\hat{\rho}(x,t)$ is given by

$$\text{Arc-length monitor function}: \quad \hat{\rho}(x,t) = \sqrt{1 + \beta|\hat{h}_x|^2}. \tag{21}$$

$$\text{Curvature monitor function}: \quad \hat{\rho}(x,t) = \sqrt{1 + \beta|\hat{h}_{xx}|^2}. \tag{22}$$

Here, β is user-defined parameters. If $\hat{h}$ is not smooth, the discretised monitor function computed as above can change abruptly and slow down the computation. It is usually to work in the context of adaptive mesh methods to smooth the monitor function to have a smoother mesh and get the MMPDEs gently to integrate. Huang [11, 31] proposed an effective smoothing scheme, which is given by

$$\hat{\rho}_j := \sqrt{\frac{\sum_{k=j-p}^{j+p} \hat{\rho}_k^2 \gamma^{|k-j|}}{\sum_{k=j-p}^{j+p} \gamma^{|k-j|}}}, \quad j = 1, \ldots, N+1 \tag{23}$$

where $\gamma \in (0,1)$ is a positive smoothing parameter, and p is a positive integer number referred to as the smoothing index. Many sweeps of the scheme may be implemented for each integration step.

Eqs. (10,11) and MMPDE, shape a system of ODEs for $h(x,t)$, and $x(\xi,t)$ with the initial and boundary conditions provided later. Note that the stiff solver DASPK [32] employed which utilizes an iterative scheme based on some numerical methods for the solution of the linearized system including preconditioning using Incomplete LU decomposition of the Jacobian matrix.

2.1. **Numerical results.** The non-uniform mesh scheme Eqs. (10,11) are used for discretising Eq. (1). This problem is derived from the boundary conditions given by Eq. (5) and the initial condition is chosen as

$$h(x,t=0)=\frac{12\nu^2}{25\epsilon\mu}\left(1-\frac{exp\left(\frac{2\nu}{5\mu}x\right)}{\left[exp\left(\frac{\nu}{5\mu}x\right)+c\right]^2}\right),\ x\in[0,M]. \tag{24}$$

The numerical solution of this problem is sought to determine the evolution of the film thickness $h(x,t)$ using the Finite Difference technique and the Method of Lines. in all the solutions shown here, the value of the parameters are $\nu=1\times10^{-2}$, $\mu=\nu^2/4$, $\epsilon=6$, $\gamma=0.5$, $c=1$, $M=5$; the parameter values of Eq.(23) are $p=1$, $\gamma=5\times10^{-1}$. It is then utilised to examine the accuracy and the convergence of the obtained results both a non-uniform and a uniform meshes.

Figure $1(a,b)$ illustrate the time development of $h(x,t)$ to the travelling wave solution and the corresponding trajectories $x(\xi,t)$, respectively, obtained applying the non-uniform mesh scheme with $N=500$ (so, initial $\Delta x=10^{-2}$). In this simulation, MMPDE 2 (using $\tau=10^{-2}$) with the curvature monitor function (using $\beta=10^3$) are used. The results are presented for t ranging among 1 to 3.

Figure $1(a)$ shows that the adaptive moving mesh scheme captures the essential features of the solution, including the steep front, with less number of points ($N=500$ here compared to $N=5000$ for the uniform mesh computation (not shown here)). Figure $1(b)$ presents that the monitor function redistributes the mesh points such that the area close to the steep front takes more points than elsewhere ($\Delta x\approx10^{-4}$ in the steep front area and $\Delta x\approx2\times10^{-1}$ elsewhere; the initial $\Delta x=10^{-2}$ uniform everywhere).

Figure $2(a)$ presents the solution $h(x,t)$ at $t=2$. Observe that the insets display the steep front area takes more numbers of the mesh points than elsewhere. Figure $2(b)$ presents the monitor function compared with the solution shown in figure $2(a)$. Observe that the massive values of the curvature region result in the risen number of the mesh points redistributed in the steep front area of the solution (see also figure $2(a)$). Figure $2(c)$ illustrates the equidistributing coordinate transformation $x(\xi,t)$ at $t=2$. Remark that a massive number of the mesh points in the region of the steep front (located nearby $x=2$) compared to other places. Consequently, the adaptive mesh method gives a massive number of the mesh points where the solution characterised has a quick change by significant differences in its variation.

Figure 3 shows three different solutions the exact solution (solid red line), the obtained solution using a non-uniform mesh (solid blue line), and the numerical solution obtained on a uniform mesh (solid black line) at a fixed time $t=2$. $N=50$ points for both schemes (initial $\Delta x=10^{-1}$) is used to perform a comparison among the adaptive moving mesh and the uniform mesh schemes. Remark that the obtained solution applying the adaptive mesh scheme is quite similar to the exact solution (the blue line). Even though oscillations become visible in the obtained solution using the uniform mesh scheme, denoting that the region of the steep front has not been sufficiently resolved on the uniform mesh (see the black line). Observe that the adaptive mesh scheme uses the risen number of the mesh

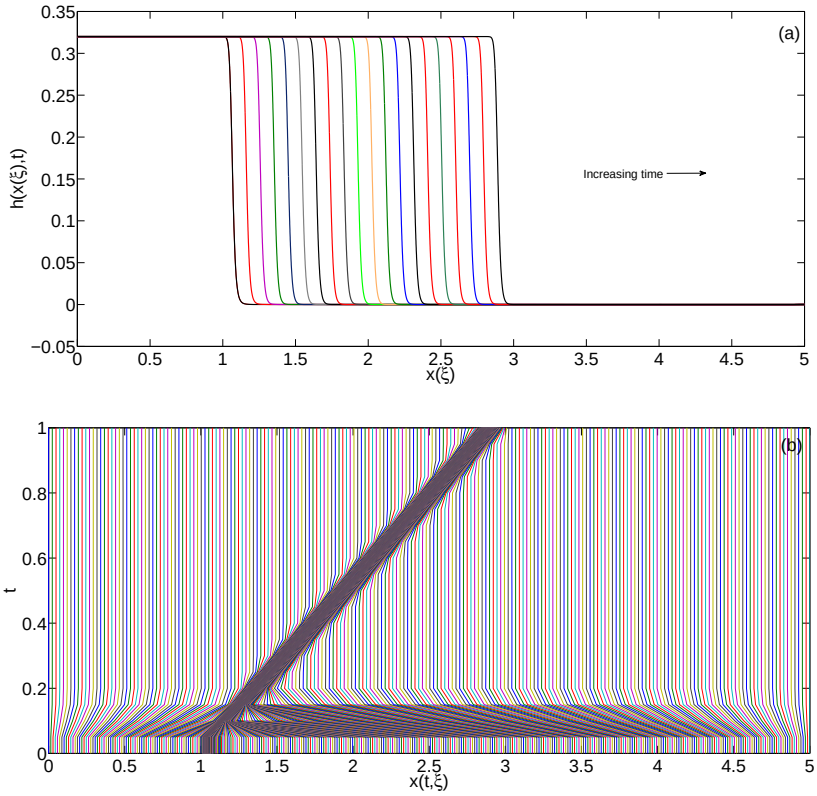

FIGURE 1. (a) time evolution of $h(x,t)$ to the travelling wave solution and (b) the corresponding mesh trajectories $x(\xi,t)$ obtained using the moving adaptive scheme with $N = 500$ (initial $\Delta x = 0.01$), MMPDE2 and curvature monitor function. The time t ranges between 1 to 3. The parameter values are:$\nu = 1 \times 10^{-2}$, $\mu = \nu^2/4$, $\epsilon = 6$, $\gamma = 0.5$, $c = 1$, $M = 5$, $p = 1$, and $\tau = 10^{-2}$.

points in the steep front region, indicating that the step size of the mesh points takes about $\Delta x = 2 \times 10^{-3}$. Nevertheless, the uniform mesh scheme demands about $\Delta x \leq 10^{-3}$ ($N = 5000$ points) for the steep front region to be successfully resolved (not displayed here). The adaptive moving mesh scheme requires less number of points (around $N = 100$ or initial $\Delta x = 2 \times 10^{-2}$) to be well-resolved this area.

2.1.1. *Error analysis and convergence.* Consider the measuring error and convergence of the adaptive mesh scheme for changing MMPDEs and the monitor functions. Table 1 gives a brief statement of the measuring error and the CPU time taken to arrive $t = 2$ utilising MMPDEs $4, 7, 6$ and modified MMPDE 5 for varying τ with fixing $N = 500$. The curvature monitor function with $\beta = 10^3$ is applied. The obtained solution used in measuring the error and the CPU time is gained at $t = 2$.

Table 1 appears MMPDE4 and MMPDE6 provide a more accurate solution and demand less CPU time compared to others. Observe that the mesh equations take a long time and become much stiffer as the value of τ reduces. Therefore, it can deduce that a value of τ ranging between 10^{-3} to 1 is optimal concerning the accuracy and taken time. Observe that also MMPDE 4 and MMPDE6 are the best concerning the accuracy and taken time (see Table 1).

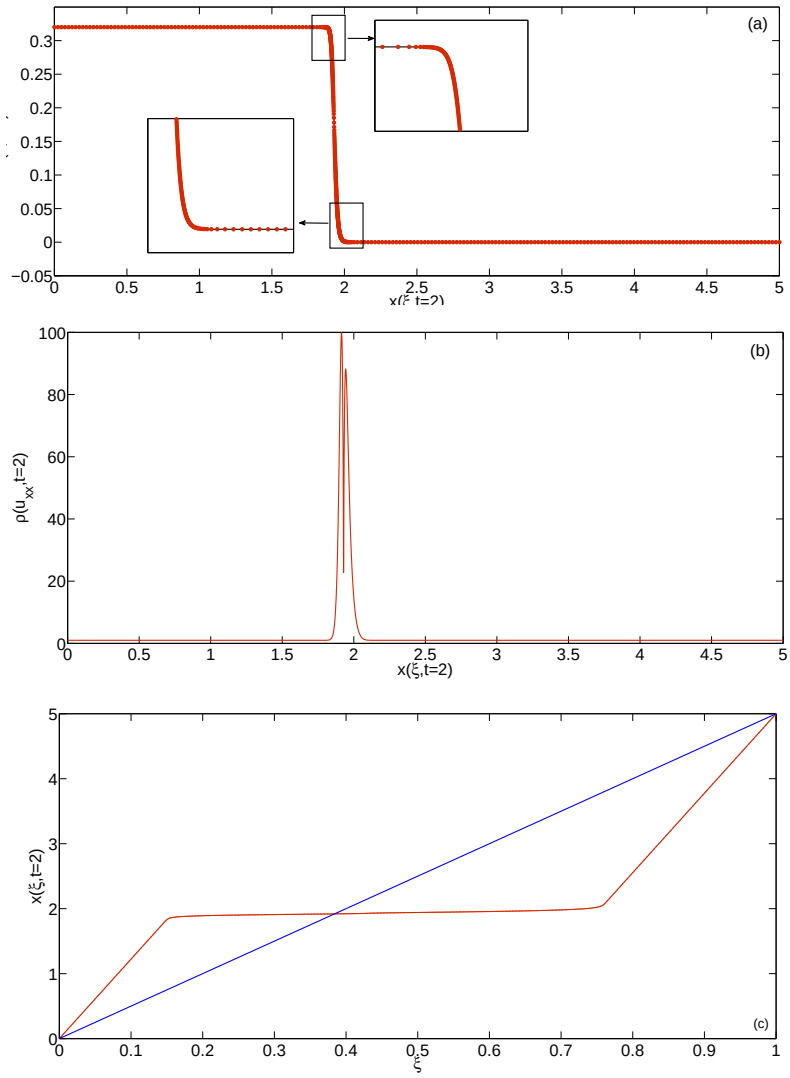

FIGURE 2. (a) $h(x, t = 2)$, (b) curvature monitor function at $t = 2$ and (c) $x(\xi, t = 2)$ obtained using the moving adaptive scheme with $N = 500$ (initial $\Delta x = 10^{-2}$) and MMPDE4. The parameter values are:$\nu = 1\times10^{-2}$, $\mu = \nu^2/4$, $\epsilon = 6$, $\gamma = 0.5$, $c = 1$, $M = 5$, $p = 1$, and $\tau = 10^{-2}$. Insets in (a) show the zoomed-in the steep front. The solid blue line in (c) represents a uniform mesh.

Figure 4 shows a comparison of the error measured using the L_2 norm for the solutions computed on an adaptive mesh scheme with different monitor functions the arc-length (solid blue line) and the curvature (solid red line) and the uniform mesh scheme (solid black line). The numerical solution for both schemes utilised in estimating the error got at a fixed time $t = 2$, and the mesh equation MMPDE 4 is applied to focus and move the mesh. Remark that the numerical solution obtained using the adaptive mesh scheme is higher accurate concerning the error compared to that obtained using the uniform mesh scheme, i.e., it achieves a higher accuracy for the same number of mesh points. For instance, the lowest error measured is 1×10^{-8} for $N = 500$ points (related to $\Delta x = 10^{-2}$) for

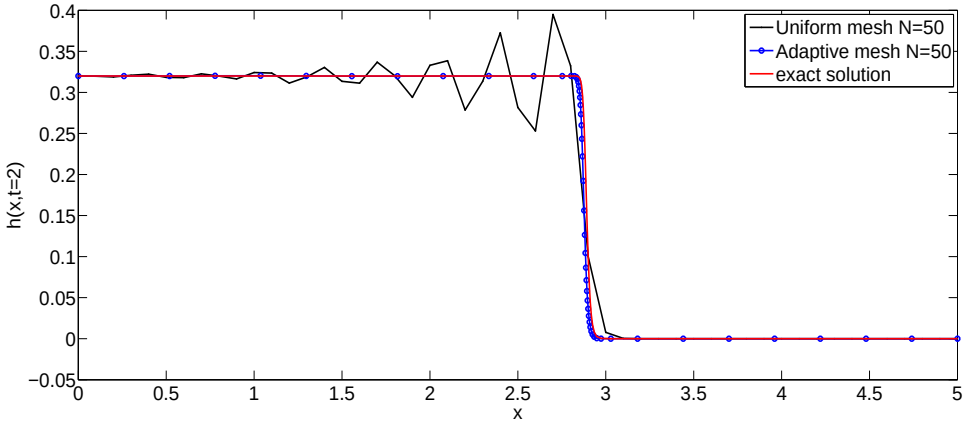

FIGURE 3. This figure presents the exact solution (solid red line) and corresponding numerical results for both uniform (solid black line) and adaptive moving mesh (solid blue line) schemes. The adaptive moving mesh solution is obtained using MMPDE 2 and the curvature monitor function with $\beta = 10^3$. The parameter values are: $N = 50$, $\nu = 1 \times 10^{-2}$, $\mu = \nu^2/4$, $\epsilon = 6$, $\gamma = 0.5$, $c = 1$, $M = 5$, $p = 1$, and $\tau = 10^{-2}$.

MMPDE	N	τ	CPU	Error
4	500	1	1.07s	1.7×10^{-5}
	500	10^{-1}	1.25s	1×10^{-6}
	500	10^{-2}	4.9s	1×10^{-8}
	500	10^{-3}	10s	1×10^{-8}
7	500	1	22.5s	5.8×10^{-6}
	500	10^{-1}	24.2s	8.8×10^{-8}
	500	10^{-2}	31.2s	3×10^{-8}
	500	10^{-3}	328.4s	3.6×10^{-8}
6	500	1	4.2s	1.5×10^{-6}
	500	10^{-1}	6.5s	5.7×10^{-8}
	500	10^{-2}	38.7s	4.4×10^{-8}
Modified 5	500	1	9.8s	1.3×10^{-7}
	500	10^{-1}	31s	3.5×10^{-8}
	500	10^{-2}	271s	3×10^{-8}
	500	10^{-3}	4250s	2.8×10^{-8}

TABLE 1. Error and CPU time taken to reach $t = 2$ for MMPDEs 4, 7 and 6, and modified MMPDE 5 varying the relaxation parameter τ. The numerical solution for h used is obtained at $t = 2$ and the curvature monitor function is used with $\beta = 10^3$. The parameter values are: $N = 500$, $\nu = 10^{-2}$, $\mu = \nu^2/4$, $\epsilon = 6$, $\gamma = 0.5$, $c = 1$, $M = 5$, and $p = 1$.

the solution obtained adaptive moving mesh solution with the curvature monitor function (solid red line). But, the error measured for the solution obtained the uniform mesh for the same $N = 500$ is 1×10^{-4}. Furthermore, it is useful in the number of the mesh

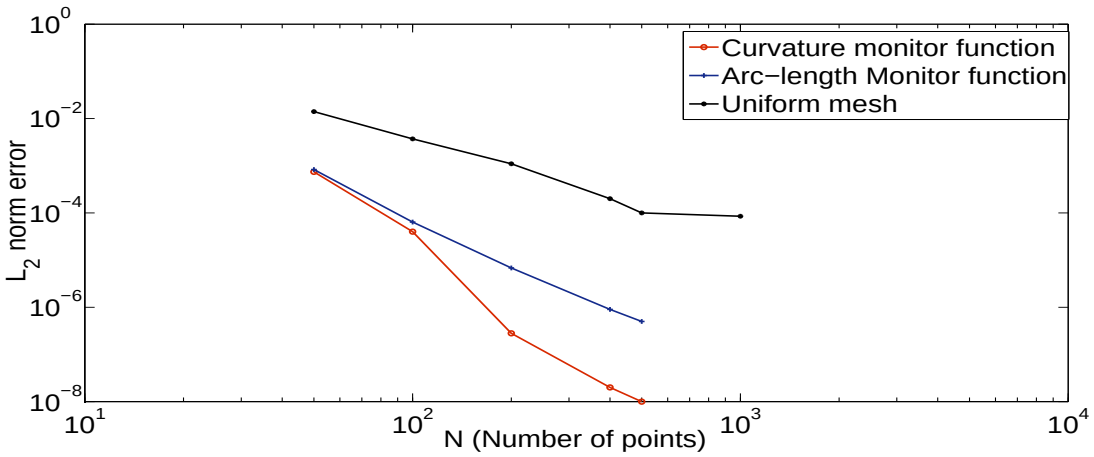

FIGURE 4. The error measured, using L_2 norm, for the obtained solutions using both a uniform mesh (solid black line) and an adaptive mesh with the arc-length (solid blue line) and the curvature (solid red line) monitor functions. Estimated slopes presented for comparison among several solutions. The numerical solution is got at $t = 2$ using MMPDE 4. The parameter values are: $\nu = 1 \times 10^{-2}$, $\mu = \nu^2/4$, $\epsilon = 6$, $\gamma = 0.5$, $c = 1$, $M = 5$, $p = 1$, and $\tau = 10^{-2}$.

points implemented to reach a required level of convergence; i.e., it requires only fewer number of points to reach similar error as the fixed mesh scheme. Observe that the lowest value of the measured error achieved for the solution obtained using the uniform mesh is 8×10^{-5} for $N = 1000$ (corresponding to $\Delta x = 2 \times 10^{-3}$). In contrast, the adaptive mesh needs about $N = 100$ points (corresponding to $\Delta x = 0.05$) to reach almost the same error. Additionally, remark that the adaptive mesh scheme applying the curvature monitor function (solid red line) is more convergent and accurate in varying the number of points N taken compared to that utilising the arc-length (solid blue line). Therefore, it can deduce that the adaptive mesh scheme applying the curvature monitor function is optimal for the problem considered here.

N	Error		CPU time taken to $t = 2$	
	Uniform mesh	Adaptive moving mesh	Uniform mesh	Adaptive moving mesh
50	1.4×10^{-2}	7.4×10^{-4}	0.1s	0.15s
100	3.7×10^{-3}	4×10^{-5}	0.21s	1.37s
200	1.1×10^{-3}	2.8×10^{-7}	0.54s	2.4s
400	2×10^{-4}	2×10^{-8}	1s	6.6s
500	1×10^{-4}	1×10^{-8}	1.10s	11s
1000	8×10^{-5}	–	3.7s	–

TABLE 2. Comparing the error and CPU time taken to reach $t = 2$ for the uniform mesh and adaptive moving mesh (using the curvature monitor function and MMPDE 4) schemes. The numerical solution is obtained at $t = 2$. The parameter values are: $N = 500$, $\nu = 10^{-2}$, $\mu = \nu^2/4$, $\epsilon = 6$, $\gamma = 0.5$, $c = 1$, $M = 5$, $p = 1$ and $\tau = 10^{-2}$.

Table 2 presents a comparison between the error and taken time computed on both the uniform mesh and the adaptive mesh (applying the curvature monitor function and the mesh equation MMPDE4) schemes. The error columns give a brief description of figure 4. Observe that the lowest value of the error estimated applying the adaptive moving mesh with the curvature monitor function, indicating that the adaptive moving mesh scheme with the curvature monitor function is the optimal scheme regarding the accuracy, and obtained with a fewer number of N points compared to the uniform mesh scheme. However, concerning the CPU time taken, the adaptive mesh scheme requires a long time to get the solution at $t = 2$ compared to the uniform mesh scheme for the same N. This is due to the MMPDE that requires to be simultaneously solved along with the physical PDE. Therefore, it would require to check the accuracy and the CPU time to decide the performance of the adaptive mesh scheme compared to the uniform mesh scheme. For instance, at $N = 500$ points there is a threefold rise in the time taken by the adaptive mesh scheme to obtain the solution at $t = 2$. However, the error records a reduction by about less than six orders of magnitude. Eventually, I can surely deduce that the adaptive mesh scheme is more computationally effective than the uniform mesh scheme. Remark that I cannot compute the solution for $N \geq 1000$. Because the minimum Δx, becomes quite small, so it results in round-off errors. Hence, the numerical solution missing stability and being unstable.

3. Conclusions

I have favourably employed the adaptive mesh method based on several the mesh equations MMPDEs Eqs. (13-15) with the monitor functions Eqs. (21,22) to find numerical solutions to the KdVB equation Eq. (1). The significant highlight of the results, presented above, is clearly illustrated in figures 4, 3 and Table 2 which allows straight comparison with the results of the uniform mesh scheme. Notice that the lowest value of the error estimated applying the adaptive moving mesh with the curvature monitor function, indicating that the adaptive moving mesh scheme with the curvature monitor function is the optimal scheme regarding the accuracy, and obtained with a fewer number of N points compared to the uniform mesh scheme. Alternatively, for a specified error, the adaptive mesh scheme requires a fewer number of points to resolve the steep front regions that appear in the solution compared to the uniform mesh scheme. Nevertheless, the adaptive mesh scheme demands enormously longer time than the uniform mesh scheme due to the additional mesh equation MMPDE that requires to be simultaneously solved along with the physical PDEs.

Note that a suitable selection of the monitor function assists in reaching the highest accuracy. From the excellent results shown above, I can deduce that the curvature monitor function is the optimal choice for the KdVB problem (see figure 4). A meaningful result is related to the adaptation of this monitor function to carefully resolve the solution at the region that its solution changed. The main disadvantage of the adaptive mesh methods is that it redistributes a fixed number of points, in contrast to hp-adaptive methods which authorise for dynamic allocation of mesh points. Some problems would require to begin with a massive number of mesh points if the solution includes many areas. I have also developed the curvature monitor function to contain many solution components (the reader can refer to Huang & Russell [11]). This adaptation allows resolving the many intricate structures in both variations of a solution precisely compared to the uniform mesh scheme (see Figure 2).

In conclusion, the results obtained above imply significant promise in terms of simplicity in its performance and accuracy(in comparison to the results obtained by El-Danaf

[33]) for the adaptive mesh methods based on MMPDEs to be used regularly in KdVB problem. Although I have only examined a particular form of the physical PDEs, the general framework shown here can be utilized to other PDEs such as a thin-film equation coupled (see Alharbi [28, 29, 30].

References

[1] Johnson, R. S., (1970), A nonlinear equation incorporating damping and dispersion, J Fluid Mech., 42, pp. 49-60.

[2] Van, W. L., (1982), On the motion of gas bubbles in a perfect fluid, Arch. Mech., 34, pp. 369-373.

[3] Gao, G., (1985), A theory of interaction between dissipation and dispersion of turbulence, Sci. Sinica. (Ser A), 28, 616-627.

[4] Liu, S. D. and Liu, S. K., (1992), KdV-–Burger's equation modeling of turbulence, Sci. Sinica. (Ser A), 35, pp. 576-586.

[5] Hu, N. P., (1972), Collisional theory of shock and nonlinear waves in a plasma, J. Phys. Fluids, 15, pp. 845-64.

[6] Grua, H. and Hu, N. P., (1967), Unified shock profile in plasma, J. Phys. Fluids, 10, pp. 2596-2602.

[7] Jeffrey, A., (1979), Some aspects of the mathematical modeling of long nonlinear waves, Arch. Mech., 31, pp. 559-74.

[8] Canosa, J. and Gazdag, J., (1977), The Korteweg-de Vries–Burger's equation, J. Comput. Phys., 23, pp. 393-403.

[9] Turetaev, I. D., (1991), Investigation of projection-difference schemes for the Bona-Smith and the Burgers-Korteweg-de Vries equations, Zh. Vychisl. Mat. Mat. Fiz., vo. 31, no. 2, pp. 272–285.

[10] Bona, J. L. and Schonbek M. E., (1985), Traveling wave solutions to Korteweg-de Vries–Burgers equation, Proc. R. Soc. Edin. A. Maths., 101, pp. 207-26.

[11] Huang, W. and Russell R. D., (2011), The Adaptive Moving Mesh Methods, Springer.

[12] Budd, C. J., Huang, W. and Russell R. D., (2009), Adaptivity with moving grids, Acta Num., 18 pp. 111-241.

[13] Liseikin, V. D., (2009), Grid Generation Methods, Springer: New York, NY, USA.

[14] Tang, T. (2005), Moving mesh methods for computational fluid dynamics, Contemp. Maths., 383, pp. 141-173.

[15] Yibao, L., Darae, J. and Kim, J., (2014), Adaptive mesh refinement for simulation of thin film flows, Meccanica, 49, pp. 239-252.

[16] Ceniceros, H. D. and Hou, T. Y., (2001), An efficient dynamically adaptive mesh for potentially singular solutions, J. Comput. Phys., vo. 172, is. 2, pp. 609-639.

[17] Walsh, E., (2010), Moving mesh methods for problems in meteorology, Univ. of Bath (PhD thesis).

[18] Budd, C. J., Cullen, M. J. P. and Walsh, E., (2013), Monge-Ampere based moving mesh methods for numerical weather prediction, with applications to the Eady problem, J. Comput. Phys., 236, pp. 247-270.

[19] Huang, W. and Russell, R. D, (1996), A Moving Collocation Method for Solving Time Dependent Partial Differential Equations, Appl. Numer. Math., 20, pp. 101-116.

[20] Blom, J. G. and Zegeling, P. A., (1994), Algorithm 731: A moving-grid interface for systems of one-dimensional partial differential equations, ACM Trans. Math. Softw., 20, pp. 194-214.

[21] Russell, R. D., Huang, W. and Ren, Y., (1994), Moving Mesh Partial Differential Equations (MMPDES) Based on the Equidistribution Principle, Siam. J. Numer. Anal., 31, pp. 709-730.

[22] Soheili, A. R., Kerayechian, A. and Davoodi, N., (2012), Adaptive numerical method for Burgers-type nonlinear equations, Appl. Math. Comput., 219, pp. 3486–3495.

[23] Uzunca, M., Karasozen, B. and Kucukseyhan, T., (2017), Moving mesh discontinuous galerkin methods for PDEs with traveling waves, Appl. Math. Comput., 292, pp. 9-18.

[24] Wu, Z. M., (2005), Dynamical knot and shape parameter setting for simulating shock wave by using multi-quadric quasi-interpolation, Eng. Anal. Bound. Elem., 29, pp. 354-358.

[25] Zhu, C. G. and Wang, R. H., (2009), Numerical solution of Burgers' equation by cubic B-spline quasi-interpolation, Appl. Math. Comput., pp. 260-272.

[26] Gao, Q. and Zhang, S., (2016), Moving mesh strategies of adaptive methods for solving nonlinear partial differential equations, Algorithms, vo. 9, no. 4, 86.

[27] Ma, W., (1993), An exact solution to two-dimensional Korteweg-de Vries-Burgers equation, J. Phys. A. Math. Gen., 26, L17-L20.
[28] Alharbi, A. and Shailesh, N., (2017), An adaptive moving mesh method for thin film flow equations with surface tension, J. Comput. Appl. Math., 319,(4), pp. 365-384.
[29] Alharbi, A. and Shailesh, N., (2019), An adaptive moving mesh method for two-dimensional thin film flow equations with surface tension, J. Comput. Appl. Math., 356, pp. 219-230.
[30] Alharbi, A., (2016), Numerical solution of thin-film flow equations using adaptive moving mesh methods, Keele University, (Ph.D. thesis).
[31] Huang, W., Ren, Y. and Russell R. D., (1994), Moving mesh methods based on moving mesh partial differential equations, J. Comput. Phys., 113, (2), pp. 279-290.
[32] Brown, P. N., Hindmarsh, A. C. and Petzold, L. R., (1994), Using Krylov methods in the solution of large-Scale differential- algebraic systems, SIAM J. Sci. Comput., 15, pp. 1467-1488.
[33] El-Danaf, T., (2008), Septic B-spline method of the Korteweg-de Vries–Burger's equation, Communications in Nonlinear Science and Numerical Simulation, 13, pp. 554-566.

Dr Abdulghani Ragaa Alharbi completed his PhD in field of numerical analysis, interested in using numerical methods for solving PDEs. His PhD is from Keele University in the UK in 2015. Currently, he is as an assistant professor of numerical analysis in the department of mathematics, Taibah University, Al-Madinah, Saudi Arabia.