



# Article Integrating a Stabilized Radial Basis Function Method with Lattice Boltzmann Method

Saleh A. Bawazeer <sup>1,\*</sup>, Saleh S. Baakeem <sup>2</sup> and Abdulmajeed A. Mohamad <sup>2</sup>

- <sup>1</sup> Mechanical Engineering Department, College of Engineering and Islamic Architecture, Umm Al-Qura University, P.O. Box 5555, Makkah 24382, Saudi Arabia
- <sup>2</sup> Department of Mechanical and Manufacturing Engineering, University of Calgary, 2500 University Drive, NW, Calgary, AB T2N 1N4, Canada; saleh.baakeem@ucalgary.ca (S.S.B.); mohamad@ucalgary.ca (A.A.M.)
- \* Correspondence: sambawazeer@uqu.edu.sa

Abstract: The lattice Boltzmann method (LBM) has two key steps: collision and streaming. In a conventional LBM, the streaming is exact, where each distribution function is perfectly shifted to the neighbor node on the uniform mesh arrangement. This advantage may curtail the applicability of the method to problems with complex geometries. To overcome this issue, a high-order meshless interpolation-based approach is proposed to handle the streaming step. Owing to its high accuracy, the radial basis function (RBF) is one of the popular methods used for interpolation. In general, RBF-based approaches suffer from some stability issues, where their stability strongly depends on the shape parameter of the RBF. In the current work, a stabilized RBF approach is used to handle the streaming. The stabilized RBF approach has a weak dependency on the shape parameter, which improves the stability of the method and reduces the dependency of the shape parameter. Both the stabilized RBF method and the streaming of the LBM are used for solving three benchmark problems. The results of the stabilized method and the perfect streaming LBM are compared with analytical solutions or published results. Excellent agreements are observed, with a little advantage for the stabilized approach. Additionally, the computational cost is compared, where a marginal difference is observed in the favor of the streaming of the LBM. In conclusion, one could report that the stabilized method is a viable alternative to the streaming of the LBM in handling both simple and complex geometries.

**Keywords:** lattice Boltzmann method; radial basis function; Hermite expansion; shape parameter; meshless; non-uniform mesh

## 1. Introduction

In recent years, the lattice Boltzmann method (LBM) has been used as a powerful numerical tool to simulate a wide range of fluid flow and heat transfer problems [1–3]. The LBM has competitive advantages over conventional computational fluid dynamic (CFD) methods due to its simplicity, parallel computing nature, and the exactness of advection term (no dissipation error) [2,4]. The solution in the LBM is obtained by two steps: collision (local) and streaming (non-local). The collision step relaxes the distribution functions to their equilibrium values. In the non-local streaming step, the distribution functions at the local points are streamed (advected) to the neighbor points using the discrete lattice speeds. Notably, this process leads to perfect shifting to the neighbor nodes because the time and space increments are equal. The perfect shifting justifies the exactness of the advection of the distribution functions and explains the high accuracy of the LBM. However, it constrains the applicability of the method to uniform meshes, which impedes the ability of the method to handle complex geometries [4].

Many approaches have been followed in the literature to alleviate this constraint. The first approach employs both multi-blocks and local mesh refinement techniques [5–16]. The mesh refinement techniques require a second-order interpolation to achieve the correct



Citation: Bawazeer, S.A.; Baakeem, S.S.; Mohamad, A.A. Integrating a Stabilized Radial Basis Function Method with Lattice Boltzmann Method. *Mathematics* **2022**, *10*, 501. https://doi.org/10.3390/ math10030501

Academic Editor: James M. Buick

Received: 15 January 2022 Accepted: 1 February 2022 Published: 4 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Navier–Stokes equations. In addition, they need complex interfacing between different levels of refinement, which might adversely impact the conservation across the interface. Furthermore, they require recalculations of the collision for each refinement level. To avoid this, a second approach is used, which uncouples the time and space increments and uses conventional computational fluid dynamic (CFD) methods such as the finitevolume method (FVM) [17–22], the finite-difference method (FDM) [23–34], the finiteelement method (FEM) [35–41], or the meshless method [42–45]. This approach suffers from numerical diffusion, which leads to dissipation error in proportion to the mesh distribution. Additionally, complex mesh processing is required, thereby increasing the computational cost. In order to remediate this issue, the radial basis function (RBF) method is used because it has a large number of bases that equals the number of the considered neighbors. Lin et al. [46] used the local semi-Lagrangian RBF for the streaming step in LBM. Musavi et al. [47] handled the streaming of the LBM using the local Petrov–Galerkin RBF method, which is based on the weak form of the partial differential equation (PDE). Unfortunately, the methods depending on weak form solve for the average value over a finite element or volume, which have less accuracy than the perfect shifting of the standard LBM. Therefore, a very fine mesh should be used to achieve accuracy close to that of the perfect shifting (streaming). This mesh refinement comes with the price of high computational and storage costs. To reduce the computational cost, the interpolationbased approach (third approach) is used by researchers [48–52]. In the third approach, the streaming in the LBM is treated as a Lagrangian, and then the shifted distribution functions are interpolated back to their original location. The interpolation-based approach usually generates numerical diffusion in the solution. This diffusion can be reduced by drastically increasing the number of grids or using a higher-order interpolation. Increasing the number of mesh points is fraught with many negative effects such as raising the computational cost. Although the higher-order interpolation is easy and applicable for the structured grid, it is difficult for non-uniform meshes.

The present work proposes using the interpolation-based approach to achieve a viable alternative to the streaming step, where the high-accuracy RBF interpolation is used to interpolate the post-collision distribution functions. The usage of high-order RBF interpolation enhances the accuracy and allows the usage of coarser mesh when compared to linear interpolation. However, the accuracy of RBF is predicated on the shape parameter, which needs to be selected carefully in order to achieve the desired stability and accuracy. Hence, it is necessary to stabilize RBF for a wide range of shape parameter values. The stability of the RBF depends strongly on shape parameter values that should be carefully selected. Using a novel stabilized RBF method proposed in our previous work [53] could beat this issue. The stabilized RBF approach has a weak dependency on the shape parameter, where any value of the shape parameter can be selected. In the current study, the stabilized approach is used to solve the streaming step of the LBM to enhance the accuracy and applicability of the method to problems with non-uniform grids. To evaluate the proposed method, three benchmark problems are solved using the stabilized RBF method and the streaming of the LBM. Moreover, the computational cost is compared for both used methods.

The rest of the paper is organized as follows. Section 2 describes the LBM. Section 3 explains the RBF interpolation method and how to use it for solving the streaming step of the LBM. Section 4 shows the results of one- and two-dimensional benchmark problems. Finally, Section 5 presents the conclusions.

#### 2. Lattice Boltzmann Method

The LBM is derived from the Boltzmann transport equation after being discretized in velocity space using a finite velocity set  $e_i$  [54–57].

$$\frac{\partial f_i}{\partial t} + \boldsymbol{e}_i \cdot \frac{\partial f_i}{\partial \boldsymbol{x}} = \frac{1}{\tau} \Big( f_i(t, \boldsymbol{x}) - f_i^M(\rho, \boldsymbol{u}) \Big) + G_i(t, \boldsymbol{x}), \tag{1}$$

where f,  $f_i^M$ , t, x,  $\tau$ , G,  $\rho$ , e, and u denote the distribution function, the Maxwell–Boltzmann distribution function, the time, the position vector, the relaxation time, the force term, the density, the microscopic velocity vector, and the macroscopic velocity, respectively. Many approaches have been followed in the literature to discretize Equation (1). In the original LBM approach, the Lagrangian discretization is selected for the left-hand side [54–57].

$$f_i(t+dt, \mathbf{x}+\mathbf{e}_i dt) - f_i(t, \mathbf{x}) = \frac{dt}{\tau} \left( f_i(t, \mathbf{x}) - f_i^{\mathbf{e}q} \right) + dt G_i(t, \mathbf{x}).$$
(2)

Space and time increments are chosen to be equal to achieve the perfect shifting (streaming), where each distribution function is shifted perfectly to the neighbor node in the upstream direction. This approach is very efficient computationally and has no diffusion error. However, it works only on a structured uniform grid, which is very restrictive and might curb the applicability of the method to complex geometries. To relax this restriction, space and time increments are separated by Lee et al. [58] using the  $\theta$  method to discretize the Boltzmann transport equation (i.e., Equation (1)) along the characteristic, which then leads to the three-step method: pre-collision, streaming, and post-collision.

> (3) $f_i^*(t, \mathbf{x}) = f_i(t, \mathbf{x}) + \frac{dt(1-\theta)}{\tau} \left( f_i^{eq}(t, \mathbf{x}) - f_i(t, \mathbf{x}) \right) + dt(1-\theta)G_i(t, \mathbf{x}),$

Streaming

Pre-collision

Post-collision

$$f_i^*(t+dt, \mathbf{x}+\mathbf{e}_i dt) = f_i^*(t, \mathbf{x}), \text{ and}$$
(4)

$$f_i(t+dt, \mathbf{x}+\mathbf{e}_i dt) = \frac{f_i^*(t+dt, \mathbf{x}+\mathbf{e}_i dt) + dt\theta G_i(t+dt, \mathbf{x}+\mathbf{e}_i dt) + \frac{dt\theta}{\tau} f_i^{\mathbf{e}q}(t+dt, \mathbf{x}+\mathbf{e}_i dt)}{1+\frac{dt\theta}{\tau}}.$$
(5)

Evidently, Equations (3) and (5) are local and denote the pre- and post-collision, respectively. However, Equation (4) is the non-local streaming step. The streaming step can be solved using conventional CFD methods such as FVM, FDM, or FEM. However, they are not recommended due to the complicated and pedantic mesh processing of 2D and 3D problems. Furthermore, the accuracy of those methods is considered low when compared to the RBF method for the same number of points. The RBF-based interpolation can be used to solve the streaming step (i.e., Equation (4)) even for an unstructured mesh. In the following, the RBF method and methodology of implementation into the LBM are presented in detail.

#### 3. Radial Basis Function Method

The RBF method is used to interpolate a function by finding the coefficients of a basis function. The primary difference between the RBF and the polynomial approach is the basis ( $\phi$ ), which is a function of both distances between two points in space  $(r(x,\xi) = \sqrt{(x-\xi)} \cdot (x-\xi))$ and the shape parameter ( $\varepsilon$ ). In the RBF method, the number of bases is the same as the number of interpolation points. For this reason, it generates a square interpolation matrix. To illustrate, the specified data  $f(x_i)$  at  $x_i$  locations can be interpolated by first defining the interpolant,

$$(\mathbf{x}) = \sum_{j} \alpha_{j} \phi(r(\mathbf{x}, \mathbf{x}_{j}), \varepsilon), \qquad (6)$$

where  $\alpha_i$  represent the coefficients associated with each basis. The coefficients can be determined through the imposition of the interpolation constraints on the interpolant. This can be achieved by equating the interpolant (I(x)) at points  $x_i$  to the specified data  $(f(x_i))$ at the same location.

$$I(\mathbf{x}_i) = \sum_j \alpha_j \phi(r(\mathbf{x}_i, \mathbf{x}_j), \varepsilon) = f(\mathbf{x}_i).$$
<sup>(7)</sup>

(4)

Equation (7) is a linear system of equations that can be expressed in a matrix format

$$[\phi][\alpha] = [f]. \tag{8}$$

The matrix  $[\phi]$  is a symmetric and an invertible matrix if no repeated point is used. This property is found to be effective because it lowers the computational cost. Once the coefficients  $\alpha_j$  are calculated, the data can be interpolated to any point *x* using Equation (6). Subsequently, this RBF-based interpolation can be used to solve the streaming step (i.e., Equation (4)) of the LBM.

#### 3.1. RBF for Solving the Streaming Step

The RBF interpolation can be used to interpolate the shifted distribution functions  $(f_i^*(t + dt, \mathbf{x} + \mathbf{e}_i dt))$  to their original locations ( $\mathbf{x}$ ), which can be achieved by first defining the basis matrix  $\phi(r(\mathbf{x}_i + \mathbf{e}_k dt, \mathbf{x}_j + \mathbf{e}_k dt), \varepsilon)$ . This matrix can be simplified by noticing that  $r(\mathbf{x}_i + \mathbf{e}_k dt, \mathbf{x}_j + \mathbf{e}_k dt) = r(\mathbf{x}_i, \mathbf{x}_j)$ , i.e.,

$$\phi(r(\mathbf{x}_i + \mathbf{e}_k dt, \mathbf{x}_j + \mathbf{e}_k dt), \varepsilon) = \phi(r(\mathbf{x}_i, \mathbf{x}_j), \varepsilon).$$
(9)

Subsequently, the coefficients  $\alpha_i$  are calculated from Equation (8) as follows:

$$\boldsymbol{\phi}[\boldsymbol{\alpha}] = \left[\boldsymbol{\phi}\big(r\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big), \boldsymbol{\varepsilon}\big)\right]^{-1} [f_k^*(t + dt, \boldsymbol{x}_i + \boldsymbol{e}_k dt)]. \tag{10}$$

The coefficients are then used to evaluate the data at the original location *x*.

$$f_k^*(t+dt,\mathbf{x}) = [\phi(r(\mathbf{x},\mathbf{x}_j+\mathbf{e}_kdt),\varepsilon)][\alpha] = \underbrace{[\phi(r(\mathbf{x},\mathbf{x}_j+\mathbf{e}_kdt),\varepsilon)][\phi(r(\mathbf{x}_i,\mathbf{x}_j),\varepsilon)]^{-1}}_{W^{(k)}(\mathbf{x})} [f_k^*(t+dt,\mathbf{x}_i+\mathbf{e}_kdt)] , \quad (11)$$

where  $W^{(k)}(x)$  is the interpolation matrix associated with  $k^{th}$  distribution function  $(f_k)$ . This calculation needs to be made once before starting the time iteration to improve the computational cost. Then, the interpolation operators  $(W^{(k)}(x))$  can be used at each time step to obtain the solution of the streaming. However, it involves an unavoidable storage cost because there is a need to store huge interpolation operators  $(W^{(k)}(x))$ . This cost can be reduced by using sparse matrix techniques.

To explain the difference between the perfect shifting and the RBF method in handling the streaming step of the LBM, four cases for the D1Q2 model are presented in Figure 1. Case A represents the perfect streaming, where each distribution function is shifted in the original streaming step to its new position by magnitude and direction of  $e_i dt$ . The streaming step in the RBF method consists of two steps, imaginary and interpolation steps, as shown in Cases B, C, and D. The imaginary step is an unhappened step, mentioned here just to demonstrate the procedures of the RBF method. For example, we imagine that each distribution function in Case B is shifted to the imaginary position by magnitude and direction of  $e_i dt$ . Then, each one is repositioned to its correct location using the interpolation relation, Equation (11). Because the imaginary and the correct locations are the same in Case B, Cases C and D are presented. As shown in the imaginary step of Case C, each distribution function is shifted by magnitude and direction of  $e_i dt$  to the new imaginary position. To illustrate the interpolation step, the distribution function labeled 7 is resulted by using the interpolation between the distribution functions labeled 1 and 2. Case D shows that each distribution function generated in the interpolation in the direction  $e_i$  is produced from all neighbor distribution functions that are in the same direction weighted with appropriate weight (i.e., Equation (11)). For example, the distribution function labeled 13 is mainly produced from distribution functions labeled 1, 2, 3, 4, and 5. The weight function is monotonically decreasing. The far distribution functions (labeled 1 and 5) have less weight compared with the near distribution functions (labeled 2, 3, and 4). This



makes the value of the interpolated distribution function labeled 13 closer to the values of distribution functions labeled 2, 3, and 4 than that of distribution functions labeled 1, 5.

**Figure 1.** Streaming step of LBM by perfect shifting and the RBF method. In all presented cases, the arrows with the BC label represent the distribution functions that are calculated based on the boundary conditions.

As mentioned previously, the interpolation is predicated on the evaluation of the basis function  $\phi$ . The inverse of the matrix  $[\phi]$ , which is generated from the basis  $\phi$ , is used to approximate a function via the interpolation operator. The shortcoming of this approach is its poor stability and hence occasionally yields inaccurate results, particularly when the shape parameter ( $\varepsilon$ ) approaches zero. A method to alleviate this problem is explained in the subsequent section.

## 3.2. Basis Function

The accuracy of the RBF-based method is known to be contingent on the shape parameter. In our previous work [53], the Hermite expansion with respect to the shape parameter was proposed to stabilize the RBF method and to achieve the same accuracy as the original RBF with weak or no dependency on the shape parameter. The RBF  $\phi$  can

be expressed in terms of the Hermite polynomial by projecting the RBF on the Hermite polynomial space.

$$\phi(r,\varepsilon) = \sum_{n=0}^{N} a_n(r) H_n(\varepsilon) , \qquad (12)$$

where  $H_n$  and  $a_n$  signify the  $n^{th}$ -order Hermite polynomial and its expansion coefficient, respectively. The proposed expansion of the RBF can be used to weaken the coupling between r and  $\varepsilon$  by truncating the Hermite expansion at a particular order N. Notably, the Hermite polynomial is an orthogonal polynomial that satisfies the following recursion relation

$$H_n(\varepsilon) = 2\varepsilon H_{n-1}(\varepsilon) - 2(n-1)H_{n-2}(\varepsilon).$$
(13)

It is possible to obtain higher-order polynomials from Equation (13) by knowing the first two polynomials ( $H_0(\varepsilon) = 1$  and  $H_1(\varepsilon) = 2\varepsilon$ ).

The expansion coefficients  $a_n$  are obtained for the Gaussian RBF ( $\phi(r, \varepsilon) = \exp(-\varepsilon^2 r^2)$ ) in the following manner

$$a_n = \begin{cases} Z_n \frac{r^n}{R^{n+1}} & \text{even } n \\ 0 & \text{odd } n \end{cases}$$
(14)

where  $Z_n = \frac{(-1)^{n/2}}{2^n (\frac{n}{2})!}$  and  $R = \sqrt{r^2 + 1}$ .

In the next section, these bases are used in the context of the RBF method with a view for solving the streaming step of the LBM.

## 4. Results and Discussions

In this work, 1D and 2D problems are used to evaluate the RBF method in handling the streaming step of the LBM. Furthermore, the results of the RBF method are compared with those of the regular streaming of the LBM. The analytical solutions or published results are presented to evaluate the accuracy of the RBF method and the streaming. Two mesh distributions for the RBF method are used: uniform and stretched distributions. In the uniform mesh distribution, the points are distributed uniformly along the x- and y-axis, while in the stretched mesh distribution, the points are stretched at the center and condensed at the edges of the domain to achieve a non-uniform mesh. The stretched mesh is presented to investigate the ability of the proposed method to solve problems associated with the irregular mesh. The following formula is used to map a uniform grid to a stretched grid:

$$x = x_{\min} + \left[\frac{\operatorname{erf}\left(\frac{\pi}{2} \frac{x - x_{\min}}{x_{\max} - x_{\min}} - \frac{\pi}{4}\right)}{2 \operatorname{erf}\left(\frac{\pi}{4}\right)} + \frac{1}{2}\right] (x_{\max} - x_{\min}),$$
(15)

where erf denotes the error function.

### 4.1. One-Dimensional Problems

The solutions of 1D diffusion and 1D advection–diffusion equations are provided to exemplify the efficiency of the RBF combined with the modified bases for solving the streaming step. The Dirichlet boundary condition is used [2]. Three mesh sizes of 100, 200, and 300 are considered for these problems. Notably, the D1Q3 lattice model can solve the diffusion and advection–diffusion equations, where it has the following lattice weights (*Wt*), microscopic speeds (*e*), speed of sound (*C*<sub>s</sub>), and equilibrium distribution ( $f^{eq}$ ):

$$Wt = [4, 1, 1]/6,$$
 (16)

$$e = [0, -1, 1], \tag{17}$$

$$C_s^2 = \frac{1}{3} \text{ , and} \tag{18}$$

The advection velocity (*u*) can be substituted with zero in the case of pure diffusion. The diffusion coefficient ( $\alpha$ ) can be related to the relaxation time ( $\tau = 3\alpha + dt/2 + dt\theta$ ), where  $\theta$  denotes a parameter to control the implicitness of the method (see Equation (3)).

4.1.1. 1D Diffusion

The 1D diffusion equation can be expressed in the following manner.

$$\frac{\partial}{\partial t} = \alpha \frac{\partial^2 \rho}{\partial x^2} \quad . \tag{20}$$

A step function centered at the origin is used as a benchmark for the diffusion equation within the range of (-50, 50). The analytical solution of this step function's diffusion can be written as a function of the complementary error function (erfc) [59]

$$\rho(x,t) = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{4\alpha t}}\right) \quad . \tag{21}$$

The boundary conditions can be deduced from the analytical solution ( $\rho(-50, t)$  and  $\rho(50, t)$ ). The solution is obtained for  $\alpha = 0.25 \text{ m}^2/\text{s}$  at two different time steps (t = 200 and 600 s).

The results of the diffusion equation solved with the LBM and the conventional streaming are depicted in Figure 2. The results illustrate an excellent agreement with the analytical solution. Figure 3 presents the results of using the RBF interpolation to solve the streaming step. The figure compares the uniform and stretched mesh distributions of 100-, 200-, and 300-mesh sizes. The results of the 100-mesh size are shown to be inferior to the conventional streaming step, even for the stretched mesh distribution. This might be due to numerical diffusion, which leads to dissipation error. Increasing the mesh size to 200 or 300 reduces numerical error and raises the computational cost. The results of the uniformly distributed mesh are shown to deviate slightly from the analytical solution for the 200-mesh size, while the results of the 300-mesh size improve the accuracy and achieve accuracy that is comparable to conventional streaming. Previous results are essentially qualitative, which prevents a fair comparison. Hence, it is necessary to make a quantitative comparison.



**Figure 2.** Diffusion problem solved by LBM with regular streaming for 100 uniformly distributed points.



**Figure 3.** Diffusion problem solved by LBM combined with RBF interpolation. (**a**,**c**,**e**) have a uniform mesh with 100, 200, and 300 points, respectively, while (**b**,**d**,**f**) have a stretched mesh with 100, 200, and 300 points, respectively.

Table 1 compares the error and the computational time of different mesh sizes and distributions. The error is evaluated with respect to the analytical solution. It reveals that the difference in the accuracy between the streaming and RBF interpolation decreases with

an increase in the number of mesh points. To illustrate, the error per node at 300-mesh is getting nearer to the error of the conventional streaming for both mesh distributions. However, the computational cost for the RBF interpolation approach is higher than that of the streaming approach. The difference is not very substantial, which demonstrates that the used approach serves as a viable alternative to the streaming step, especially when it is impossible to apply the conventional streaming as in the case of non-uniform meshes.

Model –	Mesh		1	Time ner Iteration (c)	
	Distribution	Size	Norm	Norm per Node	- Time per iteration (s)
Streaming	Uniform	100	0.004746	$4.75  imes 10^{-5}$	$1.58 imes 10^{-4}$
Interpolation	Uniform	100	0.734430	$7.34 imes10^{-3}$	$2.26 imes10^{-4}$
Interpolation	Stretched	100	2.186103	$2.19 imes10^{-2}$	$2.28 imes10^{-4}$
Interpolation	Uniform	200	0.197118	$9.81 imes10^{-4}$	$2.70 imes10^{-4}$
Interpolation	Stretched	200	0.074056	$3.68 imes10^{-4}$	$2.50 imes10^{-4}$
Interpolation	Uniform	300	0.108697	$3.56 imes10^{-4}$	$2.92  imes 10^{-4}$
Interpolation	Stretched	300	0.046173	$1.53 imes10^{-4}$	$2.79 imes10^{-4}$

Table 1. Error and computational cost of the diffusion problem.

#### 4.1.2. 1D Advection–Diffusion

To investigate the proposed method furthermore, the 1D advection–diffusion equation is used to compare the RBF interpolation and the streaming. The 1D advection–diffusion equation can be expressed as follows

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} = \alpha \frac{\partial^2 \rho}{\partial x^2},\tag{22}$$

where *u* represents the advection velocity. The advection velocity represents a challenging numerical issue, where it plays the main role in the numerical stability of the method used. The same step function is used as a benchmark of the advection–diffusion equation. The analytical solution of the advection–diffusion of the step function can be written in the following manner.

$$\rho(x,t) = \frac{1}{2} \operatorname{erfc}\left(\frac{x-tu}{\sqrt{4\alpha t}}\right).$$
(23)

The step function is centered at the origin within the range (-50, 50). The boundary conditions can be deduced from the analytical solution. The solution is obtained for  $\alpha = 0.25 \text{ m}^2/\text{s}$ , u = 0.1 m/s, and different time steps up to t = 200 s.

The LBM is used to solve the advection–diffusion equation that utilizes the conventional streaming, as illustrated in Figure 4. An outstanding agreement with the analytical solution is observed. Figure 5 illustrates the results of using the RBF interpolation as a feasible alternative for the streaming approach to solve the advection–diffusion equation. The results of the interpolation are shown to be inferior to those of the conventional streaming approach. However, this gap can be reduced by increasing the number of mesh points. For example, the accuracy of the results of the 300-mesh size is comparable to that of conventional streaming with the 100-mesh size.

Comparing the obtained results of the diffusion and advection–diffusion problems for the 100-mesh size, the diffusion results yielded by the RBF are more accurate than that of the advection–diffusion problem in both cases of the uniform and stretched mesh distributions. The authors believe that this huge error in advection–diffusion results for the 100-mesh size is due to the effect of the advection velocity, which causes dispersion error. However, both error sources (dispersion and dissipation) are reduced significantly by increasing the mesh size with a small computational overhead, as shown in Table 2. As a result, the proposed method is a powerful method to handle the streaming of the LBM with a comparable computational cost to that of the streaming.

Model —	Mesh		Ι	Time per Iteration (s)	
	Distribution	Size	Norm	Norm per Node	Time per iteration (5)
Streaming	Uniform	100	0.018873	$1.89 imes10^{-4}$	$1.66  imes 10^{-4}$
Interpolation	Uniform	100	0.567023	$5.67 imes10^{-3}$	$2.31  imes 10^{-4}$
Interpolation	Stretched	100	1.317779	$1.32 imes10^{-2}$	$2.43 imes10^{-4}$
Interpolation	Uniform	200	0.179421	$8.93 imes10^{-4}$	$2.54 imes10^{-4}$
Interpolation	Stretched	200	0.087822	$4.37 imes10^{-4}$	$2.59 imes10^{-4}$
Interpolation	Uniform	300	0.117243	$3.88  imes 10^{-4}$	$2.79 imes10^{-4}$
Interpolation	Stretched	300	0.071195	$2.36 imes10^{-4}$	$2.92 imes10^{-4}$

Table 2. Error and computational cost of the advection-diffusion problem.

Table 2 compares the error and the computational time of the different mesh sizes and distributions. It demonstrates that the difference in the accuracy between the streaming and interpolation approaches can be reduced by increasing the number of mesh points. Moreover, a small difference is found in the computational cost between the interpolation approach and the streaming approach. As a result, it can be inferred that there is a low computational overhead in the utilization of the interpolation approach as an alternative to the streaming step.

In conclusion, one can say that though the present approach consumes a little bit more time compared to the perfect streaming of the LBM, it is still useful and effective in complex geometries.

## 4.2. 2D Lid-Driven Cavity

The 2D lid-driven cavity problem is studied to show the applicability of the proposed method to higher dimensions. Notably, the D2Q9 lattice model can solve the Navier–Stokes equations, where it has the following lattice weights (Wt), speeds (e), and equilibrium distribution ( $f^{eq}$ )

$$Wt = [16, 4, 4, 4, 4, 1, 1, 1, 1]/36, (24)$$

$$e = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{bmatrix},$$
 (25)

$$C_s^2 = \frac{1}{3}, \text{ and}$$
(26)

$$f_i^{eq} = \rho W t_i \left( 1 + \frac{\boldsymbol{e}_i \cdot \boldsymbol{u}}{C_s^2} + \frac{(\boldsymbol{e}_i \cdot \boldsymbol{u})^2}{2C_s^4} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2C_s^2} \right).$$
(27)



**Figure 4.** Advection–diffusion problem solved by LBM with regular streaming for 100 uniformly distributed points.



**Figure 5.** Advection–diffusion problem solved by LBM and RBF interpolation. (**a**,**c**,**e**) have a uniform mesh with 100, 200, and 300 points, respectively, while (**b**,**d**,**f**) have a stretched mesh with 100, 200, and 300 points, respectively.

The bounce-back boundary condition is used for the left, right, and bottom boundaries, while the ZouHe boundary condition is used for the upper boundary (for more details, see [2]). In the present study, the Reynolds number (*Re*) and lid velocity ( $u_o$ ) are set to be 100 and 0.1, respectively. In the current example, one uniform mesh size of 40 is used for the conventional streaming of the LBM, while 40-, 60-, and 80-mesh uniform or stretched distributions are used for the RBF approach. The obtained results are compared with that of Ghia et al. [60] for the horizontal velocity ( $u_x$ ) at the mid-x plane and vertical velocity ( $u_y$ ) at the mid-y plane.

The results of the 40-mesh size for the streaming and the RBF interpolation are shown in Figure 6. The results indicate that the RBF interpolation with a uniform mesh is better than the streaming, where the results are closer to the reference data. The error for horizontal velocity  $(u_x)$  with respect to Ghia et al.'s result is 0.0410 for the streaming, 0.0353 for the RBF with a uniform mesh, and 0.1529 for the RBF with a stretched mesh, refer to Table 3. Moreover, the error for the vertical velocity  $(u_y)$  is 0.0367 for the streaming, 0.03368 for the RBF with a uniform mesh, and 0.1028 for the RBF with a stretched mesh. Overall, the results show a marginal difference. However, the RBF consumes more time for convergence than the streaming. One may think that increasing the number of grid points could enhance the results; however, this does not occur in our case. For instance, increasing the number of grid points from 60 to 80 yields an error in  $u_x$  of 0.1214 and 0.0372, respectively, for the RBF interpolation with a uniform mesh, as shown in Figure 7 and Table 3. Moreover, the error in the stretched mesh in  $u_y$  is 0.1028 for the 40-mesh, 0.0448 for the 60-mesh, and 0.0532 for the 80-mesh sizes. This is due to the effect of the shape parameter on the accuracy of RBF interpolation, which does not appear for the mildly stretched mesh. In a very fine mesh, the shape parameter would require fine-tuning and updating of the matrix  $W^{(k)}(x)$  at each time step to achieve the desired accuracy, which would increase the computational cost drastically. Hence, the accuracy of the stabilized RBF [53] should be further investigated and improved in the future for problems with very stretched mesh distribution to be less sensitive for changes in the shape parameter. This might be achieved using the local stabilized RBF approach, which uses a local support domain on each node instead of selecting all nodes. This results in a sparse matrix, which can be solved faster than a dense matrix. Hence, the computational cost would be improved.



**Figure 6.** Lid-driven cavity solved using streaming with 40-mesh and RBF interpolation with 40-mesh uniform and stretched.

Model	Mesh		No. of Steps	Time		Error w.r.t Ghai et al.'s Results	
	Distribution	Size		Total (s)	Per Iteration (s)	$u_x$	u <sub>y</sub>
Streaming	Uniform	40	7227	173.9970	0.0241	0.0410	0.0367
Interpolation	Uniform	40	8332	256.7160	0.0308	0.0353	0.0368
Interpolation	Stretched	40	7620	237.4643	0.0312	0.1529	0.1028
Interpolation	Uniform	60	8395	564.0292	0.0672	0.1214	0.0806
Interpolation	Stretched	60	8629	616.4658	0.0714	0.0453	0.0448
Interpolation	Uniform	80	9000	1285.9117	0.1429	0.0372	0.0315
Interpolation	Stretched	80	8906	1302.6666	0.1463	0.0668	0.0532

Table 3. Error and computational cost of the 2D lid-driven cavity problem.



**Figure 7.** Lid-driven cavity solved using streaming with 40-mesh and RBF interpolation with 60- and 80-mesh uniform and stretched.

As shown in Table 3, the convergence time for the RBF interpolation with the 40-mesh size is greater than that for the streaming for the same size. Increasing the number of grid points to 80 leads to an increase in the convergence time of the RBF interpolation by five times. Nevertheless, the increased computational cost is the price paid for simulating problems with a non-uniform mesh.

## 5. Conclusions

Handling the streaming step in complex geometries is one of the main issues of the LBM. Therefore, the meshless methods appear as a savior to the LBM for solving that problem. However, those methods suffer from some issues such as conservation across the interface, recalculations of the collision, numerical diffusion, computational cost, etc. In the current work, the novel stabilized RBF [53] interpolation is used as an alternative to the conventional streaming of the LBM. The stabilized RBF [53] method is examined and compared with the conventional LBM streaming by solving three benchmark problems, namely the 1D diffusion, 1D advection–diffusion, and 2D lid-driven cavity. Excellent agreements with analytical solutions and published results are realized. Regarding the computational cost, the stabilized RBF method is a viable alternative to the conventional streaming approach to handling complex geometries, even though it needs a little more computational cost. The proposed method can also be enhanced by using the local RBF approach in order to improve accuracy. Moreover, the local RBF approach facilitates the utilization of the sparse matrix's technique, which should reduce the computational cost. This will be investigated in another study.

**Author Contributions:** Conceptualization, S.A.B.; methodology, S.A.B. and S.S.B.; software, S.A.B.; validation, S.A.B.; formal analysis, S.A.B.; investigation, S.A.B., S.S.B. and A.A.M.; resources, A.A.M.; data curation, S.A.B. and S.S.B.; writing—original draft preparation, S.A.B.; writing—review and editing, S.S.B. and A.A.M.; visualization, S.S.B., S.A.B. and A.A.M.; supervision, A.A.M.; project administration, A.A.M.; funding acquisition, A.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

**Data Availability Statement:** The data that support the findings of this study are available on request from the corresponding author.

Acknowledgments: The authors would like to extend their appreciation to the Ministry of Education of Saudi Arabia and the National Science Engineering Research Council of Canada (NSERC) for their support.

Conflicts of Interest: The authors declare no conflict of interest.

#### References

- 1. Bawazeer, S.; Mohamad, A.; Oclon, P. Natural convection in a differentially heated enclosure filled with low prandtl number fluids with modified lattice boltzmann method. *Int. J. Heat Mass Transf.* **2019**, *143*, 118562. [CrossRef]
- 2. Mohamad, A.A. Lattice Boltzmann Method: Fundamentals and Engineering Applications with Computer Codes, 2nd ed.; Springer: New York, NY, USA, 2019.
- Mohamad, A.; Tao, Q.; He, Y.; Bawazeer, S. Treatment of transport at the interface between multilayers via the lattice boltzmann method. *Numer. Heat Transf. Part B Fundam.* 2015, 67, 124–134. [CrossRef]
- 4. Bawazeer, S. Lattice Boltzmann Method with Improved Radial Basis Function Method; University of Calgary: Calgary, AB, Canada, 2019.
- 5. Filippova, O.; Hänel, D. Grid refinement for lattice-bgk models. J. Comput. Phys. 1998, 147, 219–228. [CrossRef]
- Kandhai, D.; Soll, W.; Chen, S.; Hoekstra, A.; Sloot, P. Finite-difference lattice-bgk methods on nested grids. *Comput. Phys. Commun.* 2000, 129, 100–109. [CrossRef]
- Yu, D.; Mei, R.; Shyy, W. A multi-block lattice boltzmann method for viscous fluid flows. *Int. J. Numer. Methods Fluids* 2002, 39, 99–120. [CrossRef]
- 8. Yu, Z.; Fan, L.-S. An interaction potential based lattice boltzmann method with adaptive mesh refinement (amr) for two-phase flow simulation. *J. Comput. Phys.* 2009, 228, 6456–6478. [CrossRef]
- 9. Crouse, B.; Rank, E.; Krafczyk, M.; Tölke, J. A lb-based approach for adaptive flow simulations. *Int. J. Mod. Phys. B* 2003, 17, 109–112. [CrossRef]
- Wu, J.; Shu, C. A solution-adaptive lattice boltzmann method for two-dimensional incompressible viscous flows. J. Comput. Phys. 2011, 230, 2246–2269. [CrossRef]
- 11. Chen, Y.; Kang, Q.; Cai, Q.; Zhang, D. Lattice boltzmann method on quadtree grids. Phys. Rev. E 2011, 83, 026707. [CrossRef]
- 12. Lagrava, D.; Malaspinas, O.; Latt, J.; Chopard, B. Advances in multi-domain lattice boltzmann grid refinement. *J. Comput. Phys.* **2012**, 231, 4808–4822. [CrossRef]
- 13. Eitel-Amor, G.; Meinke, M.; Schröder, W. A lattice-boltzmann method with hierarchically refined meshes. *Comput. Fluids* **2013**, 75, 127–139. [CrossRef]

- 14. Fakhari, A.; Lee, T. Numerics of the lattice boltzmann method on nonuniform grids: Standard lbm and finite-difference lbm. *Comput. Fluids* **2015**, *107*, 205–213. [CrossRef]
- 15. Fakhari, A.; Lee, T. Finite-difference lattice boltzmann method with a block-structured adaptive-mesh-refinement technique. *Phys. Rev. E* 2014, *89*, 033310. [CrossRef] [PubMed]
- 16. Guzik, S.M.; Weisgraber, T.H.; Colella, P.; Alder, B.J. Interpolation methods and the accuracy of lattice-boltzmann mesh refinement. *J. Comput. Phys.* **2014**, 259, 461–487. [CrossRef]
- 17. Succi, S.; Amati, G.; Benzi, R. Challenges in lattice boltzmann computing. J. Stat. Phys. 1995, 81, 5–16. [CrossRef]
- 18. Nannelli, F.; Succi, S. The lattice boltzmann equation on irregular lattices. J. Stat. Phys. 1992, 68, 401–407. [CrossRef]
- 19. Xi, H.; Peng, G.; Chou, S.-H. Finite-volume lattice boltzmann schemes in two and three dimensions. *Phys. Rev. E* **1999**, *60*, 3380. [CrossRef]
- 20. Peng, G.; Xi, H.; Duncan, C.; Chou, S.-H. Finite volume scheme for the lattice boltzmann method on unstructured meshes. *Phys. Rev. E* 1999, *59*, 4675. [CrossRef]
- 21. Shrestha, K.; Mompean, G.; Calzavarini, E. Finite-volume versus streaming-based lattice boltzmann algorithm for fluid-dynamics simulations: A one-to-one accuracy and performance study. *Phys. Rev. E* 2016, *93*, 023306. [CrossRef]
- 22. Cevik, F.; Albayrak, K. A fully implicit finite volume lattice boltzmann method for turbulent flows. *Commun. Comput. Phys.* 2017, 22, 393–421. [CrossRef]
- Cao, N.; Chen, S.; Jin, S.; Martinez, D. Physical symmetry and lattice symmetry in the lattice boltzmann method. *Phys. Rev. E* 1997, 55, R21. [CrossRef]
- 24. Mei, R.; Shyy, W. On the finite difference-based lattice boltzmann method in curvilinear coordinates. *J. Comput. Phys.* **1998**, 143, 426–448. [CrossRef]
- 25. Guo, Z.; Zhao, T.-S. Explicit finite-difference lattice boltzmann method for curvilinear coordinates. *Phys. Rev. E* 2003, 67, 066709. [CrossRef] [PubMed]
- 26. Sofonea, V.; Sekerka, R.F. Viscosity of finite difference lattice boltzmann models. J. Comput. Phys. 2003, 184, 422–434. [CrossRef]
- Sofonea, V.; Lamura, A.; Gonnella, G.; Cristea, A. Finite-difference lattice boltzmann model with flux limiters for liquid-vapor systems. *Phys. Rev. E* 2004, 70, 046702. [CrossRef]
- El-Amin, M.; Sun, S.; Salama, A. On the stability of the finite difference based lattice boltzmann method. *Procedia Comput. Sci.* 2013, 18, 2101–2108. [CrossRef]
- 29. Hejranfar, K.; Ezzatneshan, E. Implementation of a high-order compact finite-difference lattice boltzmann method in generalized curvilinear coordinates. *J. Comput. Phys.* **2014**, *267*, 28–49. [CrossRef]
- 30. Polasanapalli, S.R.G.; Anupindi, K. A high-order compact finite-difference lattice boltzmann method for simulation of natural convection. *Comput. Fluids* **2019**, *181*, 259–282. [CrossRef]
- Yoshida, H.; Nagaoka, M. Lattice boltzmann method for the convection–diffusion equation in curvilinear coordinate systems. J. Comput. Phys. 2014, 257, 884–900. [CrossRef]
- 32. Rao, P.R.; Schaefer, L.A. Numerical stability of explicit off-lattice boltzmann schemes: A comparative study. J. Comput. Phys. 2015, 285, 251–264. [CrossRef]
- 33. Hejranfar, K.; Saadat, M.H. Preconditioned weno finite-difference lattice boltzmann method for simulation of incompressible turbulent flows. *Comput. Math. Appl.* **2018**, *76*, 1427–1446. [CrossRef]
- Krivovichev, G.V.; Mikheev, S.A. On the stability of multi-step finite-difference-based lattice boltzmann schemes. Int. J. Comput. Methods 2019, 16, 1850087. [CrossRef]
- 35. Lee, T.; Lin, C.-L. A characteristic galerkin method for discrete boltzmann equation. J. Comput. Phys. 2001, 171, 336–356. [CrossRef]
- 36. Li, Y.; LeBoeuf, E.J.; Basu, P. Least-squares finite-element scheme for the lattice boltzmann method on an unstructured mesh. *Phys. Rev. E* 2005, *72*, 046711. [CrossRef] [PubMed]
- 37. Li, Y.; LeBoeuf, E.J.; Basu, P. Least-squares finite-element lattice boltzmann method. *Phys. Rev. E* 2004, 69, 065701. [CrossRef] [PubMed]
- Bardow, A.; Karlin, I.V.; Gusev, A.A. General characteristic-based algorithm for off-lattice boltzmann simulations. *EPL* 2006, 75, 434. [CrossRef]
- 39. Jo, J.C.; Roh, K.W.; Kwon, Y.W. Finite element based formulation of the lattice boltzmann equation. Methods 2008, 6, 7. [CrossRef]
- 40. Krivovichev, G. On the finite-element-based lattice boltzmann scheme. *Appl. Math. Sci.* **2014**, *8*, 1605–1620. [CrossRef]
- 41. Patel, S.; Lee, T. A new splitting scheme to the discrete boltzmann equation for non-ideal gases on non-uniform meshes. *J. Comput. Phys.* **2016**, 327, 799–809. [CrossRef]
- 42. Shu, C.; Peng, Y.; Chew, Y. Simulation of natural convection in a square cavity by taylor series expansion-and least squares-based lattice boltzmann method. *Int. J. Mod. Phys. C* 2002, *13*, 1399–1414. [CrossRef]
- 43. Shu, C.; Niu, X.; Chew, Y. Taylor series expansion and least squares-based lattice boltzmann method: Three-dimensional formulation and its applications. *Int. J. Mod. Phys. C* 2003, *14*, 925–944. [CrossRef]
- 44. Shu, C.; Niu, X.; Chew, Y. Taylor-series expansion and least-squares-based lattice boltzmann method: Two-dimensional formulation and its applications. *Phys. Rev. E* 2002, *65*, 036708. [CrossRef]
- 45. Shu, C.; Chew, Y.; Niu, X. Least-squares-based lattice boltzmann method: A meshless approach for simulation of flows with complex geometry. *Phys. Rev. E* 2001, *64*, 045701. [CrossRef]

- 46. Lin, X.; Wu, J.; Zhang, T. A mesh-free radial basis function–based semi-Lagrangian lattice Boltzmann method for incompressible flows. *Int. J. Numer. Methods Fluids* **2019**, *91*, 198–211. [CrossRef]
- 47. Musavi, S.H.; Ashrafizaadeh, M. Meshless lattice boltzmann method for the simulation of fluid flows. *Phys. Rev. E* 2015, 91, 023310. [CrossRef]
- He, X.; Luo, L.-S.; Dembo, M. Some progress in lattice boltzmann method. Part i. Nonuniform mesh grids. J. Comput. Phys. 1996, 129, 357–363. [CrossRef]
- 49. He, X.; Doolen, G.D. Lattice boltzmann method on a curvilinear coordinate system: Vortex shedding behind a circular cylinder. *Phys. Rev. E* **1997**, *56*, 434. [CrossRef]
- He, X.; Doolen, G. Lattice boltzmann method on curvilinear coordinates system: Flow around a circular cylinder. J. Comput. Phys. 1997, 134, 306–315. [CrossRef]
- 51. Chen, H. Volumetric formulation of the lattice boltzmann method for fluid dynamics: Basic concept. *Phys. Rev. E* 1998, 58, 3955. [CrossRef]
- 52. Guo, P.; Qian, F.; Zhang, W.; Yan, H.; Wang, Q.; Zhao, C. Radial basis function interpolation supplemented lattice Boltzmann method for electroosmotic flows in microchannel. *Electrophoresis* **2021**, *42*, 2171–2181. [CrossRef]
- 53. Bawazeer, S.A.; Baakeem, S.S.; Mohamad, A. A new radial basis function approach based on hermite expansion with respect to the shape parameter. *Mathematics* **2019**, *7*, 979. [CrossRef]
- 54. Bawazeer, S. Stability and Accuracy of Lattice Boltzmann Method. Master's Thesis, University of Calgary, Calgary, AB, Canada, 2013.
- Lee, H.; Bawazeer, S.; Mohamad, A. Boundary conditions for lattice boltzmann method with multispeed lattices. *Comput. Fluids* 2018, 162, 152–159. [CrossRef]
- 56. Baakeem, S.S.; Bawazeer, S.A.; Mohamad, A. Comparison and evaluation of shan-chen model and most commonly used equations of state in multiphase lattice boltzmann method. *Int. J. Multiph. Flow* **2020**, *128*, 103290. [CrossRef]
- 57. Bawazeer, S.A.; Baakeem, S.S.; Mohamad, A. A critical review of forcing schemes in lattice boltzmann method: 1993–2019. *Arch. Comput. Methods Eng.* **2021**, *28*, 4405–4423. [CrossRef]
- 58. Lee, T.; Lin, C.-L. An eulerian description of the streaming process in the lattice boltzmann equation. *J. Comput. Phys.* 2003, 185, 445–471. [CrossRef]
- 59. Carslaw, H.S.; Jaeger, J.C. Conduction of Heat in Solids; Clarendon Press: Oxford, UK, 1992.
- 60. Ghia, U.; Ghia, K.N.; Shin, C. High resolutions for incompressible flow using the navier-stokes equations and a multigrid method. *J. Comput. Phys.* **1982**, *48*, 387–411. [CrossRef]