Generalized moving least squares and moving kriging least squares approximations for solving the transport equation on the sphere

Vahid Mohammadi a, Mehdi Dehghan a, Amirreza Khodadadian b*, Thomas Wick b

aDepartment of Applied Mathematics, Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, No. 424, Hafez Ave., 15914, Tehran, Iran

bInstitute of Applied Mathematics, Leibniz University of Hannover, Welfengarten 1, 30167 Hanover, Germany

April 12, 2019

Abstract. In this work, we apply two meshless methods for the numerical solution of the time-dependent transport equation defined on the sphere in spherical coordinates. The first technique, which was introduced by Mirzaei (BIT Numerical Mathematics, 54 (4) 1041-1063, 2017) in Cartesian coordinates is a generalized moving least squares approximation, and the second one, which is developed here, is moving kriging least squares interpolation on the sphere. These methods do not depend on the background mesh or triangulation, and they can be implemented on the transport equation in spherical coordinates easily using different distribution points. Furthermore, the time variable is approximated by a second-order backward differential formula. The obtained fully discrete scheme is solved via the biconjugate gradient stabilized algorithm with zero-fill incomplete lower-upper (ILU) preconditioner at each time step. Three well-known test problems namely solid body rotation, vortex roll-up, and deformational flow are solved to demonstrate our developments.

Keywords: Transport equation on the sphere, Meshless methods, Generalized moving least squares approximation, Moving kriging least squares interpolation, Biconjugate gradient stabilized method.

AMS subject classifications: 35R01, 74G15.

1 Introduction

In this paper, we consider the following transport equation that is defined on the sphere in spherical coordinates [1 2]

\[
\frac{\partial u}{\partial t} + v \cdot \nabla u = 0, \quad (1.1)
\]

*Corresponding author, E-mail addresses: v.mohammadi@aut.ac.ir, v.mohammadi.aut@gmail.com (V. Mohammadi), mdehghan@aut.ac.ir, mdehghan.aut@gmail.com (M. Dehghan), amirreza.khodadadian@ifam.uni-hannover.de (A. khodadadian), thomas.wick@ifam.uni-hannover.de (T. Wick)
where \( u \) is the scalar quantity being transported and \( \mathbf{v} = \mathbf{v}(\lambda, \theta, t) = (v_1(\lambda, \theta, t), v_2(\lambda, \theta, t))^T \) represents the velocity field. Here \( \lambda \) denotes the longitude and \( \theta \) is the latitude which both are measured from the equator \([1]\). Furthermore, \( \nabla \) is the gradient operator on the surface of a unit sphere in spherical coordinates which is defined as \([1, 2]\)

\[
\nabla := \left( \frac{1}{\cos(\theta)} \frac{\partial}{\partial \lambda}, \frac{\partial}{\partial \theta} \right)^T.
\]

(1.2)

As seen, in the north and south pole, i.e., \( \theta = \pm \frac{\pi}{2} \), this operator is singular.

The transport equation has various applications such as modeling transport in layered magnetic materials \([3]\), spin valves \([4]\), ocean surface modeling \([5]\), numerical weather prediction \([6]\), modeling oil weathering, and transport in sea ice \([7]\). Moreover, we should note that in the atmospheric modeling, transport processes have significant importance \([8]\).

Different standard tests are considered for this problem, which are solid body rotation and deformational flow \([8]\). A benchmark transport test on the sphere is the solid body rotation of a cosine bell along a great circle trajectory was introduced in \([9]\). Another test, which is introduced and studied in \([10]\) is deformational flow (vortex). In the Cartesian geometry, two important deformational tests with the analytic solutions are called the “Smolarkiewicz’s test” \([11]\) and "Doswell vortex" \([12]\), respectively (see also \([13]\)). Besides, Leveque introduced a deformational test, in which flow trajectories are much more complex \([14]\). In \([8]\), the authors followed \([14]\), and constructed different deformational tests in Cartesian and spherical geometries, in which some tests are non-divergent flow and the others are divergent. As said in \([1]\), geophysical fluid motions on all scales are dominated by the advection process. Therefore, computing the numerical solutions play a more important role for solving the transport equation.

In recent years, there are diverse research works based on numerical methods to solve the transport equation on the sphere such as high-order finite volume methods (FVMs) \([15, 16, 17]\), continuous and discontinuous Galerkin (DG) methods \([18, 19, 20]\), radial basis functions (RBFs) in spherical coordinates \([1]\), radial basis functions \([21]\), adaptive mesh refinement technique \([22, 23, 24]\), the conservative semi-Lagrangian multi-tracer transport scheme \([25]\), stabilization of RBF-generated finite difference techniques (RBF-FD) in spherical coordinates (adding an artificial hyperviscosity) \([2]\) as well as global, local and partition of unity RBFs methods combined with the semi-Lagrangian approach in Cartesian coordinates \([26]\), and a higher-order compatible finite element scheme for the nonlinear rotating shallow water equations on the sphere \([27]\).

In this article, we employ two meshless techniques, namely generalized moving least squares (GMLS) and moving kriging least squares (MKLS) approximations on the sphere in spherical coordinates for discretizing the spatial variables of the transport equation \((1.1)\). For the first time, the GMLS technique in subdomains of \( \mathbb{R}^d \) was introduced by Mirzaei and his co-workers \([28]\). The generalized moving least squares reproducing kernel approach is also introduced and analyzed in \( \Omega \subset \mathbb{R}^d \) \([29]\). Recently, the GMLS technique on the sphere was introduced and analyzed by Mirzaei in Cartesian coordinates \([30]\). Here, we approximate \( \nabla u \) defined in Eq. \((1.1)\) via GMLS in spherical coordinates. Besides, we have developed an MKLS interpolation on the sphere, which approximates \( \nabla u \) in spherical coordinates. This technique was first introduced in \([31]\) for subdomains in \( \mathbb{R}^d \), and it is not considered on the sphere. Against GMLS approximation, the MKLS method satisfies the Kronecker the delta property \([31]\). Of course, it depends on a parameter that is similar to the shape parameter of an RBF interpolation. As discussed in \([30, 32]\), these methods can be considered as ”generalized finite differences” in which the differential operators involved in a PDE such as Eq. \((1.1)\), can be approximated at each scattered data point on each local sub-domain. Both presented approximations are
implemented simply on transport equation defined on the sphere since they do not depend on a background mesh or triangulation.

The main advantage of GMLS and MKLS approximations developed here is that since two techniques do not depend on any background mesh or triangulation, there is no difficulty in implementation. In this work, we apply them on a transport equation on the unit sphere via two different set of points. The proposed methods can be simply implemented to solve numerically various model equations defined on the sphere in different scientific problems. The temporal variable of Eq. (1.1) is discretized by a second-order backward differential formula (BDF) [33, 34].

The remainder of this manuscript is as follows. In Section 2, the time variable of Eq. (1.1) is discretized by a second-order backward differential formula. In Section 3, the generalized moving least squares approximation in spherical coordinates, and how it can be applied to approximate the advection operator in Eq. (1.1) are presented. In Section 4, a new approximation namely moving kriging least squares is introduced on the sphere, and we have approximated the advection operator of an unknown solution using this technique. In Section 5, we have obtained the full-discrete scheme of transport equation (1.1) defined on the unit sphere due to the time and spatial discretizations proposed here. Some numerical simulations are reported in Section 6 for three test problems, which were studied in the literature works. Finally, concluding remarks are given in Section 7.

2 The time discretization

In this section, we apply a second-order BDF for discretization the time variable of Eq. (1.1) [33, 34]. For this purpose, the time interval \([0, T]\) is divided uniformly into \(M\) sub-intervals such that \(T = M \Delta t\), where \(\Delta t\) indicates the time step. By defining \(t_n := n \Delta t\) and \(u^n := u(t_n)\), a second-order BDF for transport equation (1.1) can be written as follows

\[
\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} + \frac{v_1^{n+1}}{\cos(\theta)} \frac{\partial u^{n+1}}{\partial \lambda} + \frac{v_2^{n+1}}{\partial \theta} = 0, \quad n = 1, 2, ..., M - 1, \tag{2.1}
\]

where \(v_1^{n+1}\) and \(v_2^{n+1}\) are the components of the velocity field at \(t = t_{n+1}\).

Also, for the first time step, we have used the first step of backward time stepping as follows

\[
\frac{u^1 - u^0}{\Delta t} + \frac{v_1^1}{\cos(\theta)} \frac{\partial u^1}{\partial \lambda} + \frac{v_2^1}{\partial \theta} = 0, \tag{2.2}
\]

where \(v_1^1\) and \(v_2^1\) are the components of the velocity field at \(t = t_1\).

Reformulating Eq. (2.1), we have

\[
3u^{n+1} + 2\Delta t \left( \frac{v_1^{n+1}}{\cos(\theta)} \frac{\partial u^{n+1}}{\partial \lambda} + \frac{v_2^{n+1}}{\partial \theta} \right) = 4u^n - u^{n-1}, \quad n = 1, 2, ..., M - 1. \tag{2.3}
\]

Besides, Eq. (2.2) can be rewritten in the following formula

\[
u^1 + \Delta t \left( \frac{v_1^1}{\cos(\theta)} \frac{\partial u^1}{\partial \lambda} + \frac{v_2^1}{\partial \theta} \right) = u^0. \tag{2.4}
\]

In what follows, we will come back to Eqs. (2.3) and (2.4) for deriving their full-discrete schemes.
3 The GMLS formulation for advection operator

As was mentioned earlier, the GMLS approximation on the sphere was introduced by Mirzaei in his recent work [30]. Here, we have the derivation of advection operator (1.2) of a given function such as \( u \) in spherical coordinates via the GMLS approximation.

Assume that \( u \in C^{m+1}(\mathbb{S}^2) \) is a function defined on the unit sphere \( \mathbb{S}^2 \), where \( \mathbb{S}^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1 \} \). Besides, consider a set of \( N \) points \( X = \{x_1, x_2, ..., x_N\} \) on \( \mathbb{S}^2 \). The approximation of \( u \) by GMLS can be written [30]

\[
u(x) \approx \bar{u}(x) = \sum_{j \in I(x)} a_j(x)u_j, \quad x \in \mathbb{S}^2,
\]

where \( u_j \) is the value \( u \) at point \( x_j \in I(x) \). Here \( I(x) \) is a set of indices for scattered points \( X \) that is defined as [30, 35]

\[
I(x) := \{j \in \{1, 2, ..., N\} : \text{dist}(x, x_j) < \delta \},
\]

of centers contained in the cap of radius \( \delta > 0 \) around \( x \in \mathbb{S}^2 \), and \( \text{dist}(x, x_j) \) represents geodesic distance between \( x \) and \( x_j \). In spherical coordinates, \( x := (x, y, t)^T \in \mathbb{S}^2 \) can be considered as follows, e.g., see [1]

\[
x = \cos(\lambda)\cos(\theta), \quad y = \sin(\lambda)\cos(\theta), \quad z = \sin(\theta).
\]

In Eq. (3.1), \( a_j(x) \), \( j = 1, 2, ..., |I(x)| \) for each point \( x \in \mathbb{S}^2 \), is constructed in the following vector form [30]

\[
a^*(x) = W(x)P(x)\left(P^T(x)W(x)P(x)\right)^{-1}Y(x) := [a_1(x), \cdots, a_{|I(x)|}(x)]^T.
\]

Here \( Y \) is the vector of spherical harmonics of degree at most \( m \) [30, 35], \( P(x) \in \mathbb{R}^{|I(x)| \times N(3,m)} \) is a matrix, and its rows contain the vector of spherical harmonics. \( W(x) \) represents a diagonal matrix with size \( |I(x)| \times |I(x)| \) with elements \( \phi\left(\frac{\text{dist}(x, y)}{\delta}\right) \) on its diagonal, where \( \phi : [0, \infty) \to [0, \infty) \) and with the \( \phi(r) > 0 \) for \( r \in [0, 1/2] \) and \( \phi(r) = 0 \) for \( r \geq 1 \) [30, 35], or

\[
w(x, y) := \phi\left(\frac{\text{dist}(x, y)}{\delta}\right), \quad x, y \in \mathbb{S}^2,
\]

and \( \delta > 0 \), and it is known as the weight function. As mentioned in [30, 35], there are different weight functions as Eq. (3.4), which can be considered in this approximation. In this article, the following weight function has been used in GMLS approximation [37, 38]

\[
\phi(r) = \begin{cases} 
(1 - r)^4(4r + 1), & 0 \leq r \leq 1, \\
0, & r > 1,
\end{cases}
\]

where \( r = \text{dist}(x, y)/\delta \) and \( x, y \in \mathbb{S}^2 \).

If in Eq. (3.1) we consider the spherical gradient, i.e., \( \nabla_{\mathbb{S}^2} := \nabla_0 \), we will have [30]

\[
\nabla_0 u(x) \approx \nabla_0 \bar{u}(x) = \sum_{j \in I(x)} a_j \nabla_0(x) u_j, \quad x \in \mathbb{S}^2,
\]

\[
(3.5)
\]
where

\[
a_{Y_0}(x) = W(x)P(x) \left( P^T(x)W(x)P(x) \right)^{-1} \nabla_0(Y(x)),
\]

in which \( \nabla_0 \) acts only on the vector of spherical harmonics of degree at most \( m \), i.e., \( Y(x), x \in S^2 \). Due to [30, Lemma 3.1] or [30, Definition 3.3], it is easy to compute the surface gradient in spherical coordinates. Therefore, the partial derivatives of \( a_j(x) \) with respect to \( \lambda \) and \( \theta \) can be written as follows

\[
\frac{\partial a_j(x)}{\partial \lambda} = (\nabla_0)_1 a_j(x) \frac{\partial x}{\partial \lambda} + (\nabla_0)_2 a_j(x) \frac{\partial y}{\partial \lambda} + (\nabla_0)_3 a_j(x) \frac{\partial z}{\partial \lambda}
\]

\[
= (\nabla_0)_1 a_j(x) (-\sin(\lambda) \cos(\theta)) + (\nabla_0)_2 a_j(x) (\cos(\lambda) \cos(\theta)),
\]

and

\[
\frac{\partial a_j(x)}{\partial \theta} = (\nabla_0)_1 a_j(x) \frac{\partial x}{\partial \theta} + (\nabla_0)_2 a_j(x) \frac{\partial y}{\partial \theta} + (\nabla_0)_3 a_j(x) \frac{\partial z}{\partial \theta}
\]

\[
= (\nabla_0)_1 a_j(x) (-\cos(\lambda) \sin(\theta)) + (\nabla_0)_2 a_j(x) (-\sin(\lambda) \sin(\theta)) + (\nabla_0)_3 a_j(x) (\cos(\theta)),
\]

where \((\nabla_0)_1, (\nabla_0)_2 \) and \((\nabla_0)_3 \) are the components of \( \nabla_0 \). Inserting Eqs. (3.7) and (3.8) into Eq. (1.2) yields

\[
\nabla a_j(x) = \begin{pmatrix}
\frac{1}{\cos(\theta)} \frac{\partial a_j(x)}{\partial \lambda}, \frac{\partial a_j(x)}{\partial \theta}
\end{pmatrix}^T := (G_\lambda, G_\theta)^T,
\]

where

\[
G_\lambda = (\nabla_0)_1 a_j(x) (-\sin(\lambda)) + (\nabla_0)_2 a_j(x) (\cos(\lambda)),
\]

\[
G_\theta = (\nabla_0)_1 a_j(x) (-\cos(\lambda) \sin(\theta)) + (\nabla_0)_2 a_j(x) (-\sin(\lambda) \sin(\theta)) + (\nabla_0)_3 a_j(x) (\cos(\theta)).
\]

### 4 The MKLS formulation for advection operator

Our goal of this part is to introduce a new approximation namely MKLS on the unit sphere. Previously, this technique had been given in subdomains of \( \mathbb{R}^d \) [31]. Here, we first introduce the methodology of MKLS technique on the sphere, and then we approximate \( \nabla u \) for a given function \( u \) defined on the unit sphere in spherical coordinates using this approach.

We suppose that \( u \in C^{m+1}(S^2) \) is a function defined on \( S^2 \). Also, we consider a set of \( N \) points on the unit sphere. The approximation of the function \( u \) by \( \overline{u} \) can be given by

\[
u(x) \approx \overline{u(x)} = Y^T(x)\mathbf{a}(x) + Z(x), \quad x \in S^2,
\]

where \( Y(x) \) and \( \mathbf{a}(x) \) are the vectors of spherical harmonics of at most \( m \), and the unknown coefficient, respectively. Also, \( Z(x) \) represents the realization of a stochastic process with mean zero, variance \( \sigma^2 \) and non-zero covariance. The matrix form of the covariance can be written as

\[
cov \{ Z(x_i), Z(x_j) \} = \sigma^2 R[R(x_i, x_j)], \quad i, j = 1, 2, ..., N,
\]

\[
\text{Eqn. (4.2)}
\]
where $R[R(x_i, x_j)]$ and $R(x_i, x_j)$ called the correlation matrix and the correlation function between any pair of points located at $x_i$ and $x_j$ on $S^2$, respectively. The following Gaussian function can be chosen as the correlation function

$$R(x_i, x_j) = e^{-c \text{dist}(x_i, x_j)^2}, \quad (4.3)$$

where $c > 0$ denotes the value of the correlation parameter, which can be effected on the approximation solution. In the same manner [31], Eq. (4.1) can be written as below

$$u(x) = Y^T(x)(P^T R^{-1} P)^{-1} P^T R^{-1} u + r^T(x) R^{-1} \left( I - P (P^T R^{-1} P)^{-1} P^T R^{-1} \right) u, \quad x \in S^2, \quad (4.4)$$

where the vector $r^T(x) = [R(x_1, x) R(x_2, x)...R(x_{|I(x_e)|}, x)]$ such that $x_e$ is the evaluation point on $S^2$. Due to Eq. (4.4), the shape functions of the MKLS approximation on the unit sphere are denoted by the following formula

$$a^T(x) := Y^T(x)(P^T R^{-1} P)^{-1} P^T R^{-1} + r^T(x) R^{-1} \left( I - P (P^T R^{-1} P)^{-1} P^T R^{-1} \right)$$

$$= \left[ a_1(x) \ a_2(x) \ ... \ a_{|I(x_e)|}(x) \right]. \quad (4.5)$$

The approximation of a function $u$ can be given as follows

$$\overline{u(x)} = \sum_{j \in I(x)} a_j(x) u_j, \quad x \in S^2, \quad (4.6)$$

where $a_j(x)$, $j = 1, 2, ..., N$ are obtained in (4.5), and $u_j$ is the value $u$ at point $x_j \in I(x)$. As shown in [31], the constructed shape functions of MKLS approximation in $\Omega \subset \mathbb{R}^d$ satisfy Kroncker’s delta property. This feature also remains for the shape functions (4.5) obtained on the unit sphere. Now, we approximate the surface gradient operator $\nabla u$ using MKLS method. The partial derivatives of $a_j(x)$ with respect to $\lambda$ and $\theta$ can be obtained as

$$\frac{\partial a_j(x)}{\partial \lambda} = (\nabla_0)_1 a_j(x) \frac{\partial x}{\partial \lambda} + (\nabla_0)_2 a_j(x) \frac{\partial y}{\partial \lambda} + (\nabla_0)_3 a_j(x) \frac{\partial z}{\partial \lambda}$$

$$= (\nabla_0)_1 a_j(x) (-\sin(\lambda) \cos(\theta)) + (\nabla_0)_2 a_j(x) (\cos(\lambda) \cos(\theta)), \quad (4.7)$$

and

$$\frac{\partial a_j(x)}{\partial \theta} = (\nabla_0)_1 a_j(x) \frac{\partial x}{\partial \theta} + (\nabla_0)_2 a_j(x) \frac{\partial y}{\partial \theta} + (\nabla_0)_3 a_j(x) \frac{\partial z}{\partial \theta}$$

$$= (\nabla_0)_1 a_j(x) (-\cos(\lambda) \sin(\theta)) + (\nabla_0)_2 a_j(x) (-\sin(\lambda) \sin(\theta)) + (\nabla_0)_3 a_j(x) (\cos(\theta)), \quad (4.8)$$

where $(\nabla_0)_1$, $(\nabla_0)_2$ and $(\nabla_0)_3$ act on the vector functions $Y^T(x)$ and $r^T(x)$ according to Eq. (4.5). The partial derivatives of $Y^T(x)$ with respect to $\lambda$ and $\theta$ can be obtained similar to GMLS approximation, which is described in the previous section. On the other hand, since $r^T(x)$ is a radial function, its partial derivatives with respect to $\lambda$ and $\theta$ can be computed in the same way that was given in [1], and then $\nabla a_j(x)$ will be computed at each point $x \in S^2$. 

6
5 The full-discrete scheme

In this section, we apply two approximations, which are given in Sections 3 and 4 for discretizing the spatial variables of semi-discretized equations (2.3) and (2.4). We consider \( N \) points such as \( X = \{ x_1, x_2, ..., x_N \} \) on the unit sphere in spherical coordinates, and we assume that the approximation solution of \( u_{n+1} \) at each point \( x \in S^2 \) is

\[
 u^{n+1}(x) \approx \sum_{j \in I(x)} a_j(x)u_j^{n+1}, \tag{5.1}
\]

where \( a_j(x) \) can be chosen from (3.1) or (4.6), and \( n = 0, 1, ..., M - 1 \). The surface gradient, i.e., \( \nabla u^{n+1} \) can be approximated by the following formula

\[
 \nabla u^{n+1}(x) \approx \sum_{j \in I(x)} \nabla a_j(x)u_j^{n+1}, \tag{5.2}
\]

where \( \nabla a_j(x) \) are defined due to GMLS or MKLS approximation. Replacing Eqs. (5.1) and (5.2) into Eq. (2.4) at each point \( x_i \) for \( n = 0 \) gives

\[
 \sum_{j \in I(x_i)} a_j(x_i)u_j^1 + \Delta t v^1. \sum_{j \in I(x_i)} \nabla a_j(x_i)u_j^1 = \sum_{j \in I(x_i)} a_j(x_i)u_j^0, \tag{5.3}
\]

where \( i = 1, 2, ..., N \). Substituting Eqs. (5.1) and (5.2) into Eq. (2.3) yields

\[
 3 \sum_{j \in I(x_i)} a_j(x_i)u_j^{n+1} + 2\Delta t v^{n+1}. \sum_{j \in I(x_i)} \nabla a_j(x_i)u_j^{n+1} = 4 \sum_{j \in I(x_i)} a_j(x_i)u_j^n - \sum_{j \in I(x_i)} a_j(x_i)u_j^{n-1}, \tag{5.4}
\]

for \( i = 1, 2, ..., N \) and \( n = 1, 2, ..., M - 1 \).

The matrix form of Eq. (5.3) can be written as follows

\[
 A_X U_X^1 + \Delta t (v_1 \cdot B_X^1)U_X^1 + (v_1 \cdot B_X^2)U_X^1 = A_XU_X^0, \tag{5.5}
\]

where \( A_X, B_X^1 \) and \( B_X^2 \) are the global matrices as

\[
 A_X = [a_j(x_i)]_{1 \leq i \leq N, 1 \leq j \leq N}, \quad B_X^1 = \left[ \frac{1}{\cos(\theta_i)} \frac{\partial a_j(x_i)}{\partial \lambda} \right]_{1 \leq i \leq N, 1 \leq j \leq N},
\]

\[
 B_X^2 = \left[ \frac{\partial a_j(x_i)}{\partial \theta} \right]_{1 \leq i \leq N, 1 \leq j \leq N}.
\]

\( U_X^0 \) and \( U_X^1 \) are the vectors of approximation at \( t = t_0 \) and \( t = t_1 \), respectively. \( v_1^1 \) and \( v_1^1 \) are vectors of velocity field at \( N \) points at \( t = t_1 \), and for example in MATLAB notation, \( \cdot \) denotes the pointwise product between each row of \( v_1 \) and each row of the matrix \( B_X \). Similarly, Eq. (5.4) can be represented in the following matrix form

\[
 3A_X U_X^{n+1} + 2\Delta t ((v_1^{n+1} \cdot B_X^1)U_X^{n+1} + (v_2^{n+1} \cdot B_X^2)U_X^{n+1}) = 4A_X U_X^n - A_X U_X^{n-1}, \tag{5.6}
\]

where \( U_X^{n-1}, U_X^n \) and \( U_X^{n+1} \) are the vectors of approximation at \( t = t_{n-1}, t = t_n \) and \( t = t_{n+1} \), respectively.
In order to solve the linear system of algebraic equations obtained here, i.e., (5.5) and (5.6), an iterative algorithm namely the biconjugate gradient stabilized (BiCGSTAB) method with zero-fill incomplete lower-upper (ILU) preconditioner is employed. In the literature [39], it has been shown that the method is efficiently solvable for the linear system of algebraic equations generated by the third-order semi-implicit backward differential formula and combined with a meshless technique for the solution of reaction-diffusion equation on the surfaces [39]. It also should be noted that this algorithm can be used only for the linear system of algebraic equations with a large sparse coefficient matrix [39]. According to the approximations presented here, the final coefficient matrices in Eqs. (5.5) and (5.6) are sparse, and thus the BiCGSTAB algorithm could be applied without any difficulty as we will observe in the next section.

Figure 1: Set of distribution points on the sphere for PTS points with an example of spherical cap of radius $\delta$.

6 Numerical results

In this section, in order to investigate the ability of the proposed method we use the three standard tests, which have been proposed in the literature [2, 40, 8, 26]. The first case is the known ”solid-body rotation of a cosine bell” [26], the second one is called ”vortex roll-up” [2, 40], and the latest is ”deformational flow” [8]. The numerical results reported here via quasi-uniformly distributed point sets $X$, which are known as the minimum energy (ME) [1, 41] and phyllotaxis spiral (PTS) [26], and their fill distance $h$ is proportional to $N^{-1/2}$, where $N$ is the number of points distributed on the unit spheres. For the readers convenience, the procedure of the presented numerical methods is discussed in Algorithm 1 and Algorithm 2.

Figure 1 illustrates 2500 PTS points with a spherical cap. The $\ell_2$ norm is computed to show the accuracy
**Algorithm 1** Computational algorithm of GMLS (or MKLS) approximation

**Input:** data points on $S^2$, $X = [x_1, x_2, ..., x_N]^T \in \mathbb{R}^{N \times 3}$, $\delta > 0$ as a radius of local spherical cap;

**Output:** Matrices $A_X$, $B^1_X$, $B^2_X$;

**for** $i = 1, 2, ..., N$ **do**

- Construct $I(x_i)$ due to $X$;
- Compute local matrix $P^T W P$ due to points in $I(x_i)$;
- Compute the vector $Y(x_i)$, $\nabla_0 Y(x_i)$ defined in Eqs. (3.3) and (3.6), respectively at each point $x_i$;
- Compute the vectors $a^*(x_i)$ and $a^*_0(x_i)$ at each point $x_i$ due to Eqs. (3.3) and (3.6) (or similarly compute Eq. (4.5) and components of $a^*_0(x_i)$) for MKLS approximation);
- $A_X(i,:) \leftarrow a^*(x_i)$;
- $B^1_X(i,:) \leftarrow a^*_0(x_i)^T$;
- $B^2_X(i,:) \leftarrow a^*_0(x_i)$;

**end for**

**Algorithm 2** Computational algorithm for solving transport equation on $S^2$

**Input:** data points on $S^2$, $X = [x_1, x_2, ..., x_N]^T \in \mathbb{R}^{N \times 3}$, $\delta > 0$ as a radius of local spherical cap; $T$ as final time, time step $\Delta t$;

**Output:** Approximation solution $U_X$ at final time;

**Calling** $A_X, B^1_X, B^2_X$ from Algorithm 1

**Set** the initial condition $U^0_X = \{u(x_i, 0)\}_{i=1}^N, t = 0, m = 0$;

**Set** $m = 1$, $t = m\Delta t$ and compute the velocity vector i.e., $v$ at this time;

Use zero-fill incomplete lower-upper (ILU) preconditioner for Eq. (5.5);

Find $U^1_X$ by solving the linear system (5.5);

**while** $t \leq T$ **do**

- Set $m = m + 1$, $t = m\Delta t$ and compute the velocity vector i.e., $v$ at this time;
- Use zero-fill incomplete lower-upper (ILU) preconditioner for Eq. (5.6);
- Find $U^m_X$ by solving the linear system (5.6);

**end while**
of the proposed two meshless methods that is defined as follows

\[
\left( \int_{S^2} [f(x)]^2 \, dx \right)^{1/2} \approx \left( \frac{4\pi}{N} \sum_{j=1}^{N} [f(\eta_j)]^2 \right)^{1/2} := \|f\|_{L^2},
\]

where \(\{\eta_1, \eta_2, ..., \eta_N\}\) is a set of \(N\) spherical \(t\)-design points on the unit sphere \([36, 41]\). All simulations presented here are run on a 2.2 GHz Intel Core i7-2670QM CPU and 8 GB of RAM and all self-developed codes are written in MATLAB (version 2017a) in standard double precision.

6.1 Solid-body rotation of a cosine bell test

As the first standard test, we considered the transport equation (1.1) on the unit sphere with the following vector velocity field \([9, 26]\)

\[
v_1(\lambda, \theta) = \sin(\theta) \sin(\lambda) \sin(\alpha) - \cos(\theta) \cos(\alpha), \quad v_2(\lambda, \theta) = \cos(\lambda) \sin(\alpha),
\]
where $-\pi \leq \lambda \leq \pi$ and $-\pi/2 \leq \theta \leq \pi/2$. Here, we have chosen $\alpha = \pi/2$, which shows the advection of the initial condition over the north and south poles directly \[26\]. The initial condition for this test is considered
### Table 1: The used CPU time with different values $N$ for the BiCGSTAB method for the first test.

| Method | $N$ | CPU time (s) |
|--------|-----|--------------|
| GMLS   | 1600| 3.21         |
|        | 6400| 10.66        |
|        | 19600| 28.73       |
| MKLS   | 1600| 3.26         |
|        | 6400| 10.96        |
|        | 19600| 26.50       |

Table 2: The $\ell_2$-error for different values $N$ in GMLS approximation for the first test problem.

| $N$  | PTS $\ell_2$ | ME $\ell_2$ |
|------|--------------|-------------|
| 400  | 2.59e−1      | 2.53e−1     |
| 1600 | 1.72e−1      | 1.71e−1     |
| 6400 | 4.66e−2      | 4.64e−2     |
| 16641| 2.05e−2      | 2.06e−2     |

as follows \[26\]

\[
u(\lambda, \theta, t = 0) = \begin{cases} 
\frac{1}{2} \left( 1 + \cos \left( \frac{\pi r}{R_b} \right) \right), & r < R_b, \\
0, & r \geq R_b.
\end{cases} \tag{6.2}
\]

Here $r = \arccos(\cos(\theta)\cos(\lambda))$ and $R_b = \frac{1}{2}$. This example illustrates one full revolution of the bell over the sphere at $T = 2\pi$. To simulate this process via the meshless methods presented here, we have fixed $N = 19600$ PTS points, $\delta = 12h$ with $h = N^{-1/2}$ and $\Delta t = T/1000$. Also, the constant parameter in MKLS method is considered experimentally $c = 20/h$ \[31\]. Figures 2 and 3 show the numerical solutions of $u$ at different time levels $t = T/8, T/4, T/2$ and $t = T$ using GMLS and MKLS approximations. The results obtained via two methods are in a good agreement with those reported in the literature \[9, 26\]. In Figure 4, the used CPU time for constructing all required matrices in Algorithm 1 for both approximations are given using different values of $N$. Table 1 shows the used CPU time in both techniques for different values $N$ during the above simulations. In Tables 2 and 3, $\ell_2$ errors are computed for different values $N$ via both techniques, respectively. As can be observed in results, GMLS and MKLS approximation have almost the same accuracy in solving this example.

### 6.2 Vortex roll-up test

As the second standard test for transport equation on the unit sphere, we have considered vortex roll-up test case, which is known as deformational flow, and it models idealized cyclogenesis \[2, 40\]. We have solved
Table 3: The $\ell_2$–error for different values $N$ in MKLS approximation for the first test.

| $N$  | PTS $\ell_2$ | ME $\ell_2$ |
|------|--------------|-------------|
| 400  | 2.68e–1     | 2.55e–1    |
| 1600 | 2.47e–1     | 2.43e–1    |
| 6400 | 9.53e–2     | 8.73e–2    |
| 16641| 2.36e–2     | 1.97e–2    |

Figure 5: The 3D simulation of the vortex roll-up via GMLS approximation for the second test (given in Subsection 6.2) at $t = 3$ (top left), $t = 6$ (top right), and $t = 9$ (bottom).

The transport equation (1.1) with the following velocity field [2, 40]

\[ v_1(\lambda, \theta) = \omega(\theta) \cos(\theta), \quad v_2(\lambda, \theta) = 0, \]

where

\[
\omega(\theta) = \begin{cases} 
3\sqrt{3} & \text{if } \rho(\theta) \neq 0, \\
2\rho(\theta) & \text{if } \rho(\theta) = 0,
\end{cases}
\]

with $\sec h^2(\rho(\theta)) \tanh(\rho(\theta))$.
where \( \rho(\theta) = \rho_0 \cos(\theta) \), and \( \rho_0 \) controls the radial extent of the vortex \([2, 40]\). The analytical solution of this test is given as follows \([2, 40]\):

\[
u(\lambda, \theta, t) = 1 - \tanh \left( \frac{\rho(\theta)}{\zeta} \sin(\lambda - \omega(\theta)t) \right), \quad t \geq 0.
\]

For the simulations reported here, we have chosen \( \rho_0 = 3 \) and \( \zeta = 5 \), which were considered previously in \([2]\). All required parameters in two approximations are chosen as the previous test, and \( \Delta t = 1/1000 \). In Figures 5 and 6, we showed the numerical solutions of Eq. (1.1) on the unit sphere via \( N = 19600 \) PTS points at different time levels \( t = 3, 6 \) and \( t = 9 \) by GMLS and MKLS approximations. The results obtained in this test are in agreement with reported results in \([2]\). In Table 4, the used CPU time in both techniques for different values of \( N \) during simulations are given. In Tables 5 and 6, \( \ell_2 \) errors are reported at \( T = 3 \) for different values of \( N \) and \( \Delta t = T/1000 \) via GMLS and MKLS approximations, respectively. The results given here are in good agreement with those reported in \([2]\), and we can see almost the same accuracy using both approximations.
| Method | \( N \)  | CPU time (s) |
|--------|--------|---------------|
| GMLS   | 1600   | 28.50         |
|        | 6400   | 108.86        |
|        | 19600  | 256.68        |
| MKLS   | 1600   | 20.94         |
|        | 6400   | 68.64         |
|        | 19600  | 178.37        |

Table 4: The used CPU used time with different values \( N \) for the BiCGSTAB method for the second test problem.

| \( N \)  | PTS | ME  |
|----------|-----|-----|
|          | \( \ell_2 \) | \( \ell_2 \) |
| 400      | 2.25e − 2 | 2.09e − 2 |
| 1600     | 3.51e − 3 | 3.46e − 3 |
| 6400     | 5.22e − 4 | 7.78e − 4 |
| 16641    | 1.97e − 4 | 1.92e − 4 |

Table 5: The \( \ell_2 \)—error for different values \( N \) in GMLS approximation for the second test.

| \( N \)  | PTS | ME  |
|----------|-----|-----|
|          | \( \ell_2 \) | \( \ell_2 \) |
| 400      | 4.05e − 2 | 4.17e − 2 |
| 1600     | 1.41e − 2 | 1.31e − 2 |
| 6400     | 3.59e − 3 | 1.75e − 3 |
| 16641    | 7.48e − 4 | 2.10e − 4 |

Table 6: The \( \ell_2 \)—error for different values \( N \) in MKLS approximation for the second test.
6.3 Deformational flow test

In this part, we consider the following test, which is known as deformational flow [8]. We have considered the transport equation (1.1) with the following velocity field

\[
v_1(\lambda, \theta, t) = 2 \sin^2(\lambda) \sin(2\theta) \cos(\pi t/T), \quad v_2(\lambda, \theta, t) = 2 \sin(2\lambda) \cos(\theta) \cos(\pi t/T),
\]

which is non-divergent flow [8]. The initial condition for this test is given as follows [8]

\[
u(\lambda, \theta, t = 0) = \begin{cases} 
0.1 + 0.9u_1(\lambda, \theta), & r_1(\lambda, \theta) < r, \\
0.1 + 0.9u_2(\lambda, \theta), & r_2(\lambda, \theta) < r, \\
0.1, & \text{o.w},
\end{cases}
\]

where

\[
u_1(\lambda, \theta) = \frac{1}{2} \left(1 + \cos\left(\frac{\pi r_1(\lambda, \theta)}{r}\right)\right), \quad \nu_2(\lambda, \theta) = \frac{1}{2} \left(1 + \cos\left(\frac{\pi r_2(\lambda, \theta)}{r}\right)\right),
\]

and

\[
r_1(\lambda, \theta) = \arccos \left(\sin(\theta_1) \sin(\theta) + \cos(\theta_1) \cos(\theta) \cos(\lambda - \lambda_1)\right),
\]

\[
r_2(\lambda, \theta) = \arccos \left(\sin(\theta_2) \sin(\theta) + \cos(\theta_2) \cos(\theta) \cos(\lambda - \lambda_2)\right).
\]

In the above formulations, \((\lambda_1, \theta_1) = (5\pi/6, 0)\) and \((\lambda_2, \theta_2) = (7\pi/6, 0)\) are the centers of two cosine bells [8]. This test shows that the flow field will be deformed at \(t = 2.5\), and then with return to its initial position (see Figure 7) at \(T = 5\) [8]. In Figure 8, the numerical solution of \(u\) at different time levels \(t = T/4, T/2, 9T/2\) and \(t = T\) via GMLS approximation, where \(T = 5\) and \(\Delta t = 1/400\) and \(N = 6400\) ME points is shown. Furthermore, the same simulations are obtained in Figure 9 with MKLS approximation.

As observed and expected in these figures, two cosine bells considered at the initial (Figure 7) are deformed at \(t = T/2\), and they are returned to their initial positions at \(t = T\). In Table 7, the used CPU time in GMLS and MKLS approximations for different values of \(N\) during simulations are shown. In Table 8, \(\ell_2\) errors obtained from the implementation of MKLS technique with set of points considered are reported and different values time step such that for \(N = 400\), \(\Delta t = 1/100\), for \(N = 1600\), \(\Delta t = 1/200\), for \(N = 6400\), \(\Delta t = 1/400\), for \(N = 16641\), \(\Delta t = 1/800\). Also in these tables, the accuracy in both approximations is
Figure 8: The 3D simulation of the deformational flow via GMLS approximation for the third test (given in Subsection 6.3) at $t = T/4$ (top left), $t = T/2$ (top right), $t = 9T/2$ (bottom left), and $t = T$ (bottom right).

almost same.

6.4 Comparison between two proposed approximations with other methods

In this section, we compare GMLS and MKLS approximations with other numerical methods, which were applied to solve the transport equation on the sphere in the literature, we have considered CSLAM method [25], DG method [8], RBF-FD technique [2], local and global RBF approaches [26] and RBF-PU method [26]. We have used the deformational flow test for the cosine bell, which is given in Subsection 6.3. A comparison between all mentioned methods is done in Table 10 due to degrees of freedom (DOF) (the unknown coefficients related to each method), and time steps ($\Delta t$) by computing relative $\ell_2$ error. The errors reported here for GMLS and MKLS approximations are computing via PTS points on $S^2$ due to formula (6.1), which approximates $\ell_2$ error. It also should be noted that, in [26], for computing the surface integral, the sixth–order kernel–based meshfree quadrature method has been used [42]. Besides, the results reported here for CSLAM, DG, RBF-FD, local RBF, global RBF, and RBF-PU methods are taken from [26] Table 2, Subsection 4.4].
| Method | N    | CPU time (s) |
|--------|------|--------------|
| GMLS   | 1600 | 5.98         |
|        | 6400 | 19.28        |
|        | 10000| 28.11        |
| MKLS   | 1600 | 5.58         |
|        | 6400 | 19.25        |
|        | 10000| 26.67        |

Table 7: The used CPU time with different values N for the BiCGSTAB method for the third test.

| N  | PTS | ME   |
|----|-----|------|
|    | $\ell_2$       | $\ell_2$       |
| 400| $3.34e-3$ | $3.43e-3$ |
| 1600| $1.11e-3$ | $1.15e-3$ |
| 6400| $2.76e-4$ | $3.96e-4$ |
| 16641| $8.36e-5$ | $1.11e-4$ |

Table 8: The $\ell_2$—error for different values N in GMLS approximation for the third test.

| N  | PTS | ME   |
|----|-----|------|
|    | $\ell_2$       | $\ell_2$       |
| 400| $1.45e-3$ | $1.46e-3$ |
| 1600| $8.75e-4$ | $8.81e-4$ |
| 6400| $2.53e-4$ | $2.55e-4$ |
| 16641| $2.00e-4$ | $1.51e-4$ |

Table 9: The $\ell_2$—error for different values N in MKLS approximation for the third test.
7 Concluding remarks

In this paper, two new techniques, i.e., GMLS and MKLS have been applied to approximate the spatial variables of a transport equation on the sphere in spherical coordinates. The time variable of the model is discretized by a second-order backward differential formula. The resulting of fully discrete scheme is a linear system of algebraic equations per time step, which is solved efficiently by a BiCGSTAB method with zero-fill ILU preconditioner. To ensure the ability of the proposed approaches, we have solved three important test cases namely solid body rotation, vortex roll-up, and deformational flow, which are important examples in the numerical climate modeling community. Both developed techniques do not depend on a background mesh or triangulation, which yields an easy implementation in solving of the transport equation on the sphere. Due to this feature, we have obtained the numerical results using two different distribution points, i.e., PTS and ME on the sphere. Furthermore, pole singularities appeared in this equation have been omitted in differentiation matrices due to the applied approximations. As formulated in our Algorithms 1 and 2 the procedure of implementation in both approximations consists of two main parts, which are the construction of required matrices due to Eqs. (5.5) and (5.6) (Algorithm 1) and solving the obtained full-
| Method             | $\Delta t$   | DOF   | Relative $\ell_2$ error |
|--------------------|--------------|-------|------------------------|
| GMLS               | 5/2400       | 6400  | 2.84e−4                |
| MKLS               | 5/2400       | 6400  | 2.81e−4                |
| CSLAM [25]         | 5/2400       | 86400 | 6.00e−3                |
| DG, $p = 3$ [8]    | 5/2400       | 38400 | 1.39e−2                |
| RBF-FD, $n = 84$   | 5/900        | 23042 | 1.17e−2                |
| Local RBF, $n = 84$| 5/35         | 23042 | 3.45e−3                |
| RBF-PU, $n = 84$   | 5/35         | 23042 | 3.63e−3                |
| Global RBF         | 5/45         | 15129 | 5.10e−3                |

Table 10: A comparison between the presented approximations and other numerical methods for deformational flow test (cosine bell) due to DOF and time steps. The compared methods with GMLS and MKLS techniques are CSLAM [25], which is based on a cubed sphere grid. DG is the discontinuous Galerkin scheme [8] with $p = 3$ degree polynomials (fourth-order accurate) and the cubed sphere grid. RBF-FD is the mesh-free Eulerian scheme [2] via $n = 84$ points in each local domain. Semi-Lagrangian Local RBF, RBF-PU and global RBF methods, which are applied in [26].

discrete scheme (Algorithm 2). As mentioned before, all implementations are done in MATLAB software by writing routines due to the presented algorithms. The results and simulations reported here show that both methods have the same accuracy, but the MKLS approximation depends on a constant parameter, which should be controlled experimentally. Besides, as shown in the first test, the GMLS approximation uses less CPU time than MKLS approximation for constructing all matrices in Algorithm 1. We also have compared GMLS and MKLS approximations with other methods in the literature, i.e., CSLAM, DG, RBF-FD, local RBF, global RBF, and RBF-PU for deformational flow test due to time steps and DOF. From these comparisons, we can observe that the accuracy of GMLS and MKLS approximations with less number of points (DOF) is better than other methods. According to the results and discussions in this paper, the GMLS and MKLS approaches can be applied easily to solve mathematical models in spherical geometries.

References

[1] N. Flyer, G. B. Wright, Transport schemes on a sphere using radial basis functions, Journal of Computational Physics 226 (1) (2007) 1059–1084.

[2] B. Fornberg, E. Lehto, Stabilization of RBF-generated finite difference methods for convective PDEs, Journal of Computational Physics 230 (6) (2011) 2270–2285.

[3] J. MacLaren, L. Malkinski, J. Wang, First principles based solution to the Boltzmann transport equation for Co/Cu/Co spin valves, MRS Online Proceedings Library Archive 614. doi:10.1557/PROC-614-F6.4.1

[4] M. Krems, The Boltzmann transport equation: theory and applications URL http://www.mattkrems.com/projects/completedprojects/boltzmann.pdf

[5] L. C. Bender, Modification of the physics and numerics in a third-generation ocean wave model, Journal of Atmospheric and Oceanic Technology 13 (3) (1996) 726–750.
[6] C. J. Cotter, J. Shipton, Mixed finite elements for numerical weather prediction, Journal of Computational Physics 231 (21) (2012) 7076–7091.

[7] M. Afenyo, F. Khan, B. Veitch, M. Yang, Modeling oil weathering and transport in sea ice, Marine pollution bulletin 107 (1) (2016) 206–215.

[8] R. D. Nair, P. H. Lauritzen, A class of deformational flow test cases for linear transport problems on the sphere, Journal of Computational Physics 229 (23) (2010) 8868–8887.

[9] D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, P. N. Swarztrauber, A standard test set for numerical approximations to the shallow water equations in spherical geometry, Journal of Computational Physics 102 (1) (1992) 211–224.

[10] R. D. Nair, C. Jablonowski, Moving vortices on the sphere: A test case for horizontal advection problems, Monthly Weather Review 136 (2) (2008) 699–711.

[11] P. K. Smolarkiewicz, The multi-dimensional Crowley advection scheme, Monthly Weather Review 110 (12) (1982) 1968–1983.

[12] C. A. Doswell III, A kinematic analysis of frontogenesis associated with a nondivergent vortex, Journal of the Atmospheric Sciences 41 (7) (1984) 1242–1248.

[13] A. Staniforth, J. Côté, J. Pudykiewicz, Comments on Smolarkiewicz’s deformational flow, Monthly Weather Review 115 (4) (1987) 894–900.

[14] R. J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM Journal on Numerical Analysis 33 (2) (1996) 627–665.

[15] V. Cheruvu, R. D. Nair, H. M. Tufo, A spectral finite volume transport scheme on the cubed-sphere, Applied Numerical Mathematics 57 (9) (2007) 1021–1032.

[16] C. Chen, F. Xiao, Shallow water model on cubed-sphere by multi-moment finite volume method, Journal of Computational Physics 227 (10) (2008) 5019–5044.

[17] M. Zerroukat, N. Wood, A. Staniforth, SLICE-S: A Semi-Lagrangian Inherently Conserving and Efficient scheme for transport problems on the Sphere, Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric sciences, applied meteorology and physical oceanography 130 (602) (2004) 2649–2664.

[18] F. X. Giraldo, Lagrange–Galerkin methods on spherical geodesic grids: the shallow water equations, Journal of Computational Physics 160 (1) (2000) 336–368.

[19] R. D. Nair, S. J. Thomas, R. D. Loft, A discontinuous Galerkin global shallow water model, Monthly Weather Review 133 (4) (2005) 876–888.

[20] M. Taylor, J. Edwards, S. Thomas, R. Nair, A mass and energy conserving spectral element atmospheric dynamical core on the cubed-sphere grid, in: Journal of Physics: Conference Series, Vol. 78, IOP Publishing, 2007, p. 012074.
[21] N. Flyer, G. B. Wright, A radial basis function method for the shallow water equations on a sphere, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 465 (2106) (2009) 1949–1976.

[22] C. Jablonowski, M. Herzog, J. E. Penner, R. C. Oehmke, Q. F. Stout, B. Van Leer, K. G. Powell, Block-structured adaptive grids on the sphere: Advection experiments, Monthly Weather Review 134 (12) (2006) 3691–3713.

[23] M. Läuter, D. Handorf, N. Rakowsky, J. Behrens, S. Frickenhaus, M. Best, K. Dethloff, W. Hiller, A parallel adaptive barotropic model of the atmosphere, Journal of Computational Physics 223 (2) (2007) 609–628.

[24] A. St-Cyr, C. Jablonowski, J. M. Dennis, H. M. Tufo, S. J. Thomas, A comparison of two shallow-water models with nonconforming adaptive grids, Monthly Weather Review 136 (6) (2008) 1898–1922.

[25] P. H. Lauritzen, W. C. Skamarock, M. Prather, M. Taylor, A standard test case suite for two-dimensional linear transport on the sphere, Geosci. Model Dev 5 (2012) 887–901. doi:https://doi.org/10.5194/gmd-5-887-2012

[26] V. Shankar, G. B. Wright, Mesh-free semi-Lagrangian methods for transport on a sphere using radial basis functions, Journal of Computational Physics 366 (2018) 170–190.

[27] J. Shipton, T. Gibson, C. Cotter, Higher-order compatible finite element schemes for the nonlinear rotating shallow water equations on the sphere, Journal of Computational Physics 375 (2018) 1121–1137.

[28] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, IMA Journal of Numerical Analysis 32 (3) (2012) 983–1000.

[29] R. Salehi, M. Dehghan, A generalized moving least square reproducing kernel method, Journal of Computational and Applied Mathematics 249 (2013) 120–132.

[30] D. Mirzaei, Direct approximation on spheres using generalized moving least squares, BIT Numerical Mathematics 57 (4) (2017) 1041–1063.

[31] L. Gu, Moving kriging interpolation and element-free galerkin method, International Journal for Numerical Methods in Engineering 56 (1) (2003) 1–11.

[32] R. Schaback, Error analysis of nodal meshless methods, in: Meshfree Methods for Partial Differential Equations VIII, Springer, 2017, pp. 117–143.

[33] Q. Li, L. Mei, B. You, A second-order, uniquely solvable, energy stable bdf numerical scheme for the phase field crystal model, Applied Numerical Mathematics 134 (2018) 46–65.

[34] D. Li, Z. Qiao, On second order semi-implicit fourier spectral methods for 2D Cahn–Hilliard equations, Journal of Scientific Computing 70 (1) (2017) 301–341.

[35] H. Wendland, Moving least squares approximation on the sphere, Mathematical Methods for Curves and Surfaces, Vanderbilt Univ. Press, Nashville, TN (2001) 517–526.
[36] K. Atkinson, W. Han, Spherical harmonics and approximations on the unit sphere: an introduction, Vol. 2044, Springer Science & Business Media, 2012.

[37] G. E. Fasshauer, Meshfree approximation methods with MATLAB, Vol. 6, World Scientific, 2007.

[38] H. Wendland, Scattered data approximation, Vol. 17, Cambridge University Press, 2004.

[39] E. Lehto, V. Shankar, G. B. Wright, A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces, SIAM Journal on Scientific Computing 39 (5) (2017) A2129–A2151.

[40] R. Nair, J. Côté, A. Staniforth, Cascade interpolation for semi-lagrangian advection over the sphere, Quarterly Journal of the Royal Meteorological Society 125 (556) (1999) 1445–1468.

[41] R. Womersley, I. Sloan, Interpolation and cubature on the sphere. URL http://web.maths.unsw.edu.au/~rsw/Sphere

[42] E. Fuselier, T. Hangelbroek, F. J. Narcowich, J. D. Ward, G. B. Wright, Kernel based quadrature on spheres and other homogeneous spaces, Numerische Mathematik 127 (1) (2014) 57–92.