Journal of Advances in Mathematical & Computational Sciences,
Vol. 7 No. 2, June, 2019

Article Citation Format
Durojaye M O & Odeyemi J.K (2019): Meshless Radial Basis Function Method for Numerical Solution of Shallow Water Equations in Spherical Geometry
Journal of Advances in Mathematical & Computational Sciences
Vol. 7, No. 2. Pp 47-56

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received 10th March. 2019
Review Type: Blind
Final Acceptance: 14th April, 2019
Article DOI:
dx.doi.org/10.22624/AIMS/MATHS/V7N2P5

Meshless Radial Basis Function Method for Numerical Solution of Shallow Water Equations in Spherical Geometry

[1]Durojaye M O & [2]Odeyemi J.K
Department of Mathematics
University of Abuja
Abuja, Nigeria.
Email: [1]mayojaye@yahoo.com, [2]larsody@gmail.com

ABSTRACT

This work presents high-order Radial Basis Function in Finite-Difference mode (RBF-FD) for numerical solution of shallow water equations in spherical geometry. We explore a higher-order form of RBF-FD weights and RBF-FD discretization for each RBF-FD stencil using Hermite interpolation technique. The accuracy of the RBF-FD discretization is increased by considering not only the data sites but also the derivative values on the nodes present in the supporting region. Numerical results and its comparison with other RBF types are presented.

Keywords – Meshless, Radial Basis Function, Numerical Solutions, Shallow Water Equations, Spherical Geometry

1. INTRODUCTION

Finite difference methods are numerical techniques for finding solutions to PDEs which approximate the solution on a mesh of points that are equally spaced across the domain. In situations where the preferred points are not on a mesh, or when the domain does not allow for simple meshes, it is desirable to have mesh-free methods. Methods like, Meshless Petrov–Galerkin method, the partition of unity method, smoothed particle hydrodynamics, element free Galerkin method, reproducing kernel particle method, the hp-clouds method, and finite point method are well- known meshless methods considered for fluid flow problems. Radial Basis Function Finite Difference methods (RBF-FD) are also such mesh-free methods. Recently, a RBF-FD method has been proposed by Flyer and Fornberg [1]. The RBF-FD method generates a local RBF interpolant for expressing the function derivatives at a node as a linear combination of the function values on the present nodes in the neighborhood of the considered node.

Also, this RBF interpolants are used to generate the weights of a Finite Difference (FD) formula [2].

The strength of the RBF-FD method has been leveraged to solve problems on planar domains, e.g., [3, 4, 5, 6, 7], the surface of a sphere [8, 9], and more recently, very general surfaces represented just by point clouds and normal vectors [10]. A major difficulty faced with scattered node FD formulas when the weights are generated by polynomials or RBFs is that symmetries cannot be broken to increase the accuracy of the formulas. Therefore, the number of nodes in the stencils tends to be so large compared to the resulting accuracy. To circumvent this problem, Wright and Fornberg [2] proposed a generalization of compact finite difference (CFD) formulas which will keep the stencil size fixed and to also include in the FD formula as a linear combination of derivatives of u at surrounding nodes. This idea is termed the higher order RBF-FD method based on Hermite RBF interpolation. Higher order RBF-FD methods have been used for the solution of linear and nonlinear Possion problems [7]. In this work, the idea is extended to develop higher-order schemes for the shallow water equations in spherical geometry.

The remaining part of the paper is structured as follows: In Section 2 gives a brief overview the formulation of surface differential operators in Cartesian coordinates. We give the review of Hermite RBF interpolation methods with RBF and compute the higher-order RBF-FD weights from the Hermite interpolant in Section 3. In Section 4, we proceed to compute the higher-order RBF-FD weights from iterated differentiation. In Section 5, an illustration of the higher-order RBF-FD weights with the method-of-lines is given. In Section 6, we present the steady state solution to the non-linear shallow water equations and discuss various implementation details. The concluding remark is given in Section 7.

2. OVERVIEW OF DIFFERENTIAL GEOMETRY

Here we will use the surface differential operators in Cartesian form of the shallow water equations will since such a coordinate system has no singularities.

The full shallow water equations in a 3-D Cartesian coordinate system for a rotating fluid are

$$\frac{\partial \mathbf{u}}{\partial \mathrm{t}} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - \mathrm{f}(\mathbf{x}\times\mathbf{u}) - g\nabla \mathrm{h}, \tag{1}$$

$$\frac{\partial \mathrm{h}}{\partial \mathrm{t}} = -\nabla\cdot(\mathrm{h}\mathbf{u}), \tag{2}$$

where f is the Coriolis force, $\nabla = \partial_x \hat{\mathbf{i}} + \partial_y \hat{\mathbf{j}} + \partial_z \hat{\mathbf{k}}$, $\mathbf{u} = \mathrm{u}\hat{\mathbf{i}} + \mathrm{v}\hat{\mathbf{j}} + \mathrm{w}\hat{\mathbf{k}}$ is the velocity vector, h is the geopotential height, g is gravity and $\mathbf{x} = \{x, y, z\}^T$ represents the is the position vector.

In Flyer and Wright [11], the shallow water equations on a sphere were implemented in Cartesian coordinates as follows:

$$\frac{\partial \mathrm{u}}{\partial \mathrm{t}} = -\mathbf{p}_x \begin{pmatrix} (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{u} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{i}} + g(\mathbf{p}_x\cdot\nabla)\mathrm{h} \\ (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{v} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_y\cdot\nabla)\mathrm{h} \\ (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{w} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{k}} + g(\mathbf{p}_z\cdot\nabla)\mathrm{h} \end{pmatrix} \tag{3}$$

$$\frac{\partial \mathrm{v}}{\partial \mathrm{t}} = -\mathbf{p}_{\boldsymbol{y}}\begin{pmatrix} (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{u} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{i}} + g(\mathbf{p}_{\boldsymbol{x}}\cdot\nabla)\mathrm{h} \\ (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{v} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_{\boldsymbol{y}}\cdot\nabla)\mathrm{h} \\ (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{w} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{k}} + g(\mathbf{p}_{\boldsymbol{z}}\cdot\nabla)\mathrm{h} \end{pmatrix} \quad (4)$$

$$\frac{\partial \mathrm{w}}{\partial \mathrm{t}} = -\mathbf{p}_{\boldsymbol{z}}\begin{pmatrix} (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{u} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{i}} + g(\mathbf{p}_{\boldsymbol{x}}\cdot\nabla)\mathrm{h} \\ (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{v} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_{\boldsymbol{y}}\cdot\nabla)\mathrm{h} \\ (\mathbf{u}\cdot\mathbf{P}\nabla)\mathrm{w} + \mathrm{f}(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{k}} + g(\mathbf{p}_{\boldsymbol{z}}\cdot\nabla)\mathrm{h} \end{pmatrix} \quad (5)$$

$$\frac{\partial \mathrm{h}}{\partial \mathrm{t}} = -(\mathbf{P}\nabla)\cdot(\mathrm{h}\mathbf{u}) \quad (6)$$

The projection operator $\mathbf{P}$ limits the flow to the sphere and is defined as:

$$\mathbf{P} = \mathbf{I} - \mathbf{x}\mathbf{x}^{\boldsymbol{T}} = \begin{bmatrix} (1-x^2) & -xy & -xz \\ -xy & (1-y^2) & -yz \\ -xz & -yz & (1-z^2) \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{\boldsymbol{x}}^{\boldsymbol{T}} \\ \boldsymbol{p}_{\boldsymbol{y}}^{\boldsymbol{T}} \\ \boldsymbol{p}_{\boldsymbol{z}}^{\boldsymbol{T}} \end{bmatrix} \quad (7)$$

where $\mathbf{p}_{\boldsymbol{x}}$ represents the projection operator in the x direction, $\mathbf{p}_{\boldsymbol{y}}$ represents the projection operator in the y direction and $\mathbf{p}_{\boldsymbol{z}}$ represents the projection operator in the z direction.

Thus, the only spatial operator that needs to be discretized is the projected gradient, $\mathbf{P}\nabla$ and its components, $\mathbf{p}_{\boldsymbol{x}}\cdot\nabla$, $\mathbf{p}_{\boldsymbol{y}}\cdot\nabla$, $\mathbf{p}_{\boldsymbol{z}}\cdot\nabla$.

We proceed to to delineate how to formulate the projected gradient operator in RBFs.

Let $\mathbf{x} = \{x, y, z\}$ and $\mathbf{x}_k = \{x_k, y_k, z_k\}_{k=1}^{N}$ be points on the unit sphere, then the Euclidean distance from $\mathbf{x}$ to $\mathbf{x}_k$ is

$$r(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\| = \sqrt{(x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2} = \sqrt{2(1-\mathbf{x}^{\boldsymbol{T}}\mathbf{x}_k)} \quad (8)$$

Let $\phi_k(r(\mathbf{x}))$ be an RBF centered at $\mathbf{x}_k$. Using the chain rule, it was shown in [11] that the projected RBF gradient operator is given as

$$\mathbf{P}\nabla\phi_k(r(\mathbf{x})) = \mathbf{P}(\mathbf{x}-\mathbf{x}_k)\frac{\phi_k'(r(\mathbf{x}))}{r(\mathbf{x})} = -\mathbf{P}\mathbf{x}_k\frac{\phi_k'(r(\mathbf{x}))}{r(\mathbf{x})} = \begin{bmatrix} x\mathbf{x}^{\boldsymbol{T}}\mathbf{x}_k \\ y\mathbf{x}^{\boldsymbol{T}}\mathbf{x}_k \\ z\mathbf{x}^{\boldsymbol{T}}\mathbf{x}_k \end{bmatrix}\frac{\phi_k'(r(\mathbf{x}))}{r(\mathbf{x})} \quad (9)$$

3. HERMITE INTERPOLATION WITH RBFS

We now begin with brief overview of Hermite interpolation method. Let $\mathcal{L}$ be an arbitrary differential operator and let ζ be a vector containing some combination of $m \leq n$ distinct numbers from the set $\{1,2,\ldots,n\}$. The data values $u(\boldsymbol{x}_i)$ are specified at each of the n distinct data points $\{\boldsymbol{x}_i\}_{i=1}^{n}$. The data corresponding to the differential operator operating on the function, $\mathcal{L}u(\boldsymbol{x}_{\zeta_i})$, specified at m points $\{\boldsymbol{x}_{\zeta_i}\}_{i=1}^{m}$,where the point set $\{\boldsymbol{x}_{\zeta_i}\}_{i=1}^{m} \subset \{\boldsymbol{x}_i\}_{i=1}^{n}$. The Hermite RBF interpolation method we consider is to find an interpolant of the form:

$$u(\boldsymbol{x}) \approx s(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\boldsymbol{x}-\boldsymbol{x}_i\|) + \sum_{j=1}^{m} \check{\lambda}_j \mathcal{L}_2 \phi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{\zeta_j}\right\|\right) + \beta, \tag{10}$$

where $\mathcal{L}_2\phi(\|\cdot\|)$ is a basis function derived by the functional $\mathcal{L}$ acting on infinitely smooth radial basis $\phi(\|\cdot\|)$ as a function of the second variable and β is a constant. The unknown coefficients are obtained by enforcing the conditions $s(\boldsymbol{x}_i) = u(\boldsymbol{x}_i)$, $i = 1,2,\ldots,n$; $\mathcal{L}s\left(\boldsymbol{x}_{\zeta_j}\right) = \mathcal{L}u\left(\boldsymbol{x}_{\zeta_j}\right)$, $j = 1,2,\ldots,m$; and $\sum_{i=1}^{n} \lambda_i = 0$. Imposing these interpolation conditions leads to the following block linear system of equations

$$\begin{bmatrix} \Phi & \mathcal{L}_2\Phi & \boldsymbol{e} \\ \mathcal{L}\Phi & \mathcal{L}\mathcal{L}_2\Phi & \boldsymbol{0} \\ \boldsymbol{e}^T & \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \check{\boldsymbol{\lambda}} \\ \beta \end{bmatrix} = \begin{bmatrix} u \\ \mathcal{L}u \\ 0 \end{bmatrix} \tag{11}$$

where

$$\begin{aligned} \Phi_{i,j} &= \phi\left(\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|\right), \quad i,j = 1,2,\ldots,n, \\ \mathcal{L}_2\Phi_{i,j} &= \mathcal{L}_2\phi\left(\left\|\boldsymbol{x}_i-\boldsymbol{x}_{\zeta_j}\right\|\right), \quad i = 1,2,\ldots,n,\ j = 1,2,\ldots,m, \\ \mathcal{L}\Phi_{i,j} &= \mathcal{L}\phi\left(\left\|\boldsymbol{x}_{\zeta_i}-\boldsymbol{x}_j\right\|\right), \quad i = 1,2,\ldots,m,\ j = 1,2,\ldots,n, \\ \mathcal{L}\mathcal{L}_2\Phi_{i,j} &= \mathcal{L}\mathcal{L}_2\phi\left(\left\|\boldsymbol{x}_{\zeta_i}-\boldsymbol{x}_{\zeta_j}\right\|\right), \quad i = 1,2,\ldots,m,\ j = 1,2,\ldots,m, \end{aligned}$$

and $\boldsymbol{e}_i = 1$, $i = 1,2,\ldots,n$. Equation (10) is solved using a backward substitution routine.

The Hermite interpolant can also be written in Lagrange form as

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} \psi(\|\boldsymbol{x}-\boldsymbol{x}_i\|) u(\boldsymbol{x}_i) + \sum_{j=1}^{m} \check{\psi}\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{\zeta_j}\right\|\right) \mathcal{L}u(\boldsymbol{x}_{\zeta_j}), \tag{12}$$

where $\psi(\|\boldsymbol{x}-\boldsymbol{x}_i\|)$ and $\check{\psi}\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{\zeta_j}\right\|\right)$ are of the form Equation (10) and satisfy the cardinal conditions, i.e.

$$\psi(\|\boldsymbol{x}-\boldsymbol{x}_i\|) = \begin{cases} 1, & \text{if } k = i, \\ 0, & \text{if } k \neq i, \end{cases} \quad k = 1,2,\ldots,n \tag{13}$$

$$\mathcal{L}\psi\left(\left\|\boldsymbol{x}_{\zeta_k}-\boldsymbol{x}_i\right\|\right) = 0,\ k = 1,2,\ldots,m, \tag{14}$$

$$\check{\psi}\left(\left\|\boldsymbol{x}_k-\boldsymbol{x}_{\zeta_j}\right\|\right) = 0, k = 1,2,\ldots,n, \tag{15}$$

$$\mathcal{L}\breve{\psi}\left(\left\|\boldsymbol{x}_{\zeta_k}-\boldsymbol{x}_{\zeta_j}\right\|\right)=\begin{cases}1, & \text{if } k=j,\\ 0, & \text{if } k\neq j,\end{cases}\quad k=1,2,\ldots,m, \tag{16}$$

3.1 Computation of the higher-order RBF-FD weights from the Hermite interpolant

Our goal is to obtain a higher-order RBF-FD discretization of $\mathcal{L}u(\boldsymbol{x}_1)$. The higher-order RBF-FD discretization of $\mathcal{L}u(\boldsymbol{x}_1)$ is given by the Lagrange form of the interpolant,

$$\mathcal{L}u(\boldsymbol{x}_1)\approx\mathcal{L}\ddot{s}(\boldsymbol{x}_1)=\sum_{i=1}^{n}\mathcal{L}\psi(\|\boldsymbol{x}_1-\boldsymbol{x}_i\|)u(\boldsymbol{x}_i)+\sum_{j=1}^{m}\mathcal{L}\breve{\psi}\left(\left\|\boldsymbol{x}_1-\boldsymbol{x}_{\zeta_j}\right\|\right)\mathcal{L}u\left(\boldsymbol{x}_{\zeta_j}\right) \tag{17}$$

Equation (17) can be written as a compact FD formula of the form

$$\mathcal{L}u(\boldsymbol{x}_1)\approx\sum_{i=1}^{n}w_i u(\boldsymbol{x}_i)+\sum_{j=1}^{m}\breve{w}_j\mathcal{L}u\left(\boldsymbol{x}_{\zeta_j}\right) \tag{18}$$

where the weights for the higher-order RBF-FD $\{w_i\}_{i=1}^{n}$ and $\{\breve{w}_j\}_{j=1}^{m}$ are as follows

$$w_i=\mathcal{L}\psi(\|\boldsymbol{x}_1-\boldsymbol{x}_i\|),\qquad \breve{w}_j=\mathcal{L}\breve{\psi}\left(\left\|\boldsymbol{x}_1-\boldsymbol{x}_{\zeta_j}\right\|\right) \tag{19}$$

Now, the weights are computed by solving the linear system

$$\begin{bmatrix}\Phi & \mathcal{L}_2\Phi & \boldsymbol{e}\\ \mathcal{L}\Phi & \mathcal{L}\mathcal{L}_2\Phi & \boldsymbol{0}\\ \boldsymbol{e}^T & \boldsymbol{0}^T & 0\end{bmatrix}^T\begin{bmatrix}w\\ \breve{w}\\ \mu\end{bmatrix}=\begin{bmatrix}\mathcal{L}^*\Phi_1\\ \mathcal{L}^*\widetilde{\Phi}_1\\ 0\end{bmatrix} \tag{20}$$

where $\mathcal{L}^*\Phi_1$ and $\mathcal{L}^*\widetilde{\Phi}_1$ represent the evaluation of the column vectors $\mathcal{L}^*\Phi=[\mathcal{L}\phi(\|\boldsymbol{x}-\boldsymbol{x}_1\|)\mathcal{L}\phi(\|\boldsymbol{x}-\boldsymbol{x}_2\|)\ldots\mathcal{L}\phi(\|\boldsymbol{x}-\boldsymbol{x}_n\|)]^T$ and $\mathcal{L}^*\widetilde{\Phi}=\left[\mathcal{L}\mathcal{L}_2\phi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{\zeta_1}\right\|\right)\mathcal{L}\mathcal{L}_2\phi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{\zeta_2}\right\|\right)\ldots\mathcal{L}\mathcal{L}_2\phi\left(\left\|\boldsymbol{x}-\boldsymbol{x}_{\zeta_n}\right\|\right)\right]^T$ at the node $\boldsymbol{x}_1$. Here, μ is a scalar value related to the constant β in Equation (11) and enforces the condition

$$\sum_{i=1}^{n}w_i=0$$

this ensures that the stencil is exact for all constants. The issue with using Equation (20) for determining the RBF-FD weights is that one has to explicitly compute $\mathcal{L}^*\Phi_1$ and $\mathcal{L}^*\widetilde{\Phi}_1$. Thus, constructing the system (20) analytically will not be possible. However, it is possible to construct an approximation to the entries of this system using the regular RBF-FD method from [10], which is based on iterated differentiation.

4. COMPUTATION OF THE HIGHER-ORDER RBF-FD WEIGHTS FROM ITERATED DIFFERENTIATION

Let a geophysical quantity $f(x) = f(x, y, z)$ be at the node locations $\{x_i\}_{i=1}^{N}$on the surface of a unit sphere, we call $f(x)$ as a RBF expansion

$$f(x) = \sum_{k=1}^{N} c_k \phi_k\big(r(x)\big), \tag{21}$$

where the coefficients c_k are determined by collocation.

First, we are to compute approximation of the entries in $\mathcal{L}\Phi^T$ in Equation (20) and the entries of the first vector block in the right hand side of this equation.

$$\mathcal{L}\Phi^T = \begin{bmatrix} \mathcal{L}^*\phi\big(\|\boldsymbol{x}_{\zeta_1} - \boldsymbol{x}_1\|\big) & \cdots & \mathcal{L}^*\phi\big(\|\boldsymbol{x}_{\zeta_1} - \boldsymbol{x}_n\|\big) \\ \vdots & \ddots & \vdots \\ \mathcal{L}^*\phi\big(\|\boldsymbol{x}_{\zeta_m} - \boldsymbol{x}_1\|\big) & \cdots & \mathcal{L}^*\phi\big(\|\boldsymbol{x}_{\zeta_m} - \boldsymbol{x}_n\|\big) \end{bmatrix} \tag{22}$$

We compute these approximations by constructing an approximation $\mathcal{L}$ to discrete approximations to $\mathbf{p}_{\boldsymbol{x}} \cdot \nabla$, $\mathbf{p}_{\boldsymbol{y}} \cdot \nabla$, $\mathbf{p}_{\boldsymbol{z}} \cdot \nabla$ in Equation (7) computed from the standard RBF interpolant (21) over the candidate stencil nodes $\{x_i\}_{i=1}^{N}$ gives

$$[\mathbf{p}_{\boldsymbol{x}} \cdot \nabla f(\mathbf{x})]|_{\mathbf{x}=\mathbf{x}_i} = \sum_{k=1}^{N} c_k [\mathbf{p}_{\boldsymbol{x}} \cdot \nabla f(\mathbf{x})]|_{\mathbf{x}=\mathbf{x}_i}, \quad i = 1, \dots, N$$

$$= \sum_{k=1}^{N} \underbrace{c_k [x_i \mathbf{x}^{\boldsymbol{T}} \mathbf{x}_k - x_k] \frac{\phi_k'\big(r(\mathbf{x})\big)}{r(\mathbf{x})}}_{b_{j,k}^{x}}, \quad i = 1, \dots, N$$

$$= b^x c = (b^x A_R^{-1}) f = D_N^x f \tag{23}$$

where A_R^{-1} is the inverse of the matrix defined and D_N^x are the RBF weights that will form one row of the differentiation matrix, for this operator. Similarly, one gets D_N^y and D_N^z, the discrete RBF approximations to the y and z components, respectively, of the projected gradient operator as follows

$$\big[\mathbf{p}_{\boldsymbol{y}} \cdot \nabla f(\mathbf{x})\big]\big|_{\mathbf{x}=\mathbf{x}_i} = \sum_{k=1}^{N} c_k \big[\mathbf{p}_{\boldsymbol{y}} \cdot \nabla f(\mathbf{x})\big]\big|_{\mathbf{x}=\mathbf{x}_i}, \quad i = 1, \dots, N$$

$$= \sum_{k=1}^{N} \underbrace{c_k [y_i \mathbf{x}^{\boldsymbol{T}} \mathbf{x}_k - y_k] \frac{\phi_k'\big(r(\mathbf{x})\big)}{r(\mathbf{x})}}_{b_{j,k}^{y}}, \quad i = 1, \dots, N$$

$$= b^y c = (b^y A_R^{-1}) f = D_N^y f \tag{24}$$

$$[\mathbf{p}_z \cdot \nabla f(\mathbf{x})]|_{\mathbf{x}=\mathbf{x}_i} = \sum_{k=1}^{N} c_k [\mathbf{p}_z \cdot \nabla f(\mathbf{x})]|_{\mathbf{x}=\mathbf{x}_i}, \quad i = 1, \dots, N$$

$$= \sum_{k=1}^{N} \underbrace{c_k [z_i \mathbf{x}^T \mathbf{x}_k - z_k] \frac{\phi_k'(r(\mathbf{x}))}{r(\mathbf{x})}}_{b_{j,k}^z}, \quad i = 1, \dots, N$$

$$= b^z c = (b^z A_R^{-1}) f = D_N^z f \tag{25}$$

Second, we are to compute approximation of the entries in $\mathcal{L}\mathcal{L}_2\Phi$ in Equation (20) and the entries of the second vector block in the right hand side of this equation.

$$\mathcal{L}\mathcal{L}_2\Phi = \begin{bmatrix} \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\zeta_1} - \boldsymbol{x}_{\zeta_1}\|) & \cdots & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\zeta_1} - \boldsymbol{x}_{\zeta_m}\|) \\ \vdots & \ddots & \vdots \\ \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\zeta_m} - \boldsymbol{x}_1\|) & \cdots & \mathcal{L}\mathcal{L}_2\phi(\|\boldsymbol{x}_{\zeta_m} - \boldsymbol{x}_{\zeta_m}\|) \end{bmatrix} \tag{26}$$

To approximate the operator $\mathcal{L}\mathcal{L}_2$ we again use iterated differentiation by constructing an approximation to discrete approximations to $\mathbf{p}_x \cdot \nabla$, $\mathbf{p}_y \cdot \nabla$, $\mathbf{p}_z \cdot \nabla$. in Equation (7) computed from the standard RBF interpolant (21) over the candidate stencil nodes $\{x_i\}_{i=1}^{N}$.

Upon finding approximations to the entries of $\mathcal{L}\Phi$ and $\mathcal{L}\mathcal{L}_2\Phi$ and the vector in the right hand of Equation (20), we substitute these into the system (20) and solve it to obtain iterated higher-order RBF-FD weights $\{w_i\}_{i=1}^{n}$ and $\{\breve{w}_j\}_{j=1}^{m}$to be used in (18).

5. HIGHER-ORDER RBF-FD WEIGHTS WITH THE METHOD-OF-LINES

These projected gradient operators are then used to construct a discrete approximation to RHS in (3), (4), (5), (6) respectively

$$\frac{\partial \mathrm{u}}{\partial \mathrm{t}} = -\mathbf{p}_x \begin{pmatrix} \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{u}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{u}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{u}} \\ \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{v}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{v}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{v}} \\ \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{w}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{w}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{w}} \end{pmatrix} + f \begin{pmatrix} \underline{\mathrm{y}} \circ \underline{\mathrm{w}} - \underline{\mathrm{z}} \circ \underline{\mathrm{v}} \\ \underline{\mathrm{z}} \circ \underline{\mathrm{u}} - \underline{\mathrm{x}} \circ \underline{\mathrm{w}} \\ \underline{\mathrm{x}} \circ \underline{\mathrm{v}} - \underline{\mathrm{y}} \circ \underline{\mathrm{u}} \end{pmatrix} + g \begin{pmatrix} D_N^x \\ D_N^y \\ D_N^z \end{pmatrix} \underline{\mathrm{h}} \tag{27}$$

$$\frac{\partial \mathrm{v}}{\partial \mathrm{t}} = -\mathbf{p}_y \begin{pmatrix} \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{u}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{u}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{u}} \\ \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{v}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{v}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{v}} \\ \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{w}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{w}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{w}} \end{pmatrix} + f \begin{pmatrix} \underline{\mathrm{y}} \circ \underline{\mathrm{w}} - \underline{\mathrm{z}} \circ \underline{\mathrm{v}} \\ \underline{\mathrm{z}} \circ \underline{\mathrm{u}} - \underline{\mathrm{x}} \circ \underline{\mathrm{w}} \\ \underline{\mathrm{x}} \circ \underline{\mathrm{v}} - \underline{\mathrm{y}} \circ \underline{\mathrm{u}} \end{pmatrix} + g \begin{pmatrix} D_N^x \\ D_N^y \\ D_N^z \end{pmatrix} \underline{\mathrm{h}} \tag{28}$$

$$\frac{\partial \mathrm{w}}{\partial \mathrm{t}} = -\mathbf{p}_z \begin{pmatrix} \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{u}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{u}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{u}} \\ \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{v}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{v}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{v}} \\ \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{w}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{w}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{w}} \end{pmatrix} + f \begin{pmatrix} \underline{\mathrm{y}} \circ \underline{\mathrm{w}} - \underline{\mathrm{z}} \circ \underline{\mathrm{v}} \\ \underline{\mathrm{z}} \circ \underline{\mathrm{u}} - \underline{\mathrm{x}} \circ \underline{\mathrm{w}} \\ \underline{\mathrm{x}} \circ \underline{\mathrm{v}} - \underline{\mathrm{y}} \circ \underline{\mathrm{u}} \end{pmatrix} + g \begin{pmatrix} D_N^x \\ D_N^y \\ D_N^z \end{pmatrix} \underline{\mathrm{h}} \tag{29}$$

$$\frac{\partial \mathrm{h}}{\partial \mathrm{t}} = \underline{\mathrm{u}} \circ D_N^x \underline{\mathrm{h}} + \underline{\mathrm{v}} \circ D_N^y \underline{\mathrm{h}} + \underline{\mathrm{w}} \circ D_N^z \underline{\mathrm{h}} + \underline{\mathrm{h}} \circ \left(D_N^x \underline{\mathrm{u}} + D_N^y \underline{\mathrm{v}} + D_N^z \underline{\mathrm{w}}\right) \tag{30}$$

The vectors $\mathbf{p}_x$, $\mathbf{p}_y$, $\mathbf{p}_z$ will be evaluated at the nodes $\{x_j\}_{j=1}^{N}$. The method-of-lines (MOL) technique will be used to advance the system in time.

6. STEADY STATE SOLUTION TO THE NON-LINEAR SHALLOW WATER EQUATIONS (NUMERICAL TEST CASE)

We now present numerical results obtained for the higher-order RBF-FD method for the Zonal Flow over an Isolated Mountain. This case is a steady state solution to the non-linear shallow water equations. It consists of solid body rotation or zonal flow with the corresponding geostrophic height field. The Coriolis parameter is a function of latitude and longitude, so the flow can be specified with the spherical coordinate poles not necessarily coincident with earth's rotation axis (for complete details on the test case see [12]).

The surface or mountain height is given by

$$h_s = h_{s_0}\left(1 - \frac{r}{R}\right) \qquad (31)$$

where $h_{s_0} = 200m, R = \frac{\pi}{9}$, and
$r^2 = \min[R^2,\ (\lambda - \lambda_c)^2 + (\theta - \theta_c)^2]$.

The center is taken as $\lambda_c = \frac{3\pi}{2}$and $\lambda_c = \frac{\pi}{6}$. Equation (31) is sum up to the right hand side of the PDE for the geopotential height h given in (30) as a forcing term. The initial conditions are given by,

$$h = h_0 - \frac{1}{g}\left(a\Gamma u_0 + \frac{u_0^2}{2}\right) z^2, \quad \boldsymbol{u} = u_0\{-y, x, 0\}(31) \qquad (32)$$

where $h_0 = 5400m,\ g = 9.8m,\ u_0 = 20m/s,\ a = 6400km$ (average radius of the earth), $\Gamma = 7.29 \times 10^{-5}/s$ (rotation of the earth). A time-series of the solution with $N = 3600$ nodes and stencil size $n = 31$ is given in Figure 1. For $N = 6400$ nodes and stencil size $n = 31$, the solution is given in Figure 2 for the case of the conical mountain.

7. CONCLUSION

The RBF-FD method is presented for solving steady state solution to the non-linear shallow water equations. This method approximates the function derivatives at a node in terms of the data values on a scattered set of points in neighborhood region of the node. A higher-order RBF-FD method is obtained by using Hermite RBF interpolation method to construct the function approximation at each node in the domain. The accuracy of the higher-order RBF-FD method is investigated by solving for a model Zonal Flow over an Isolated Mountain which is a steady state solution to the non-linear shallow water equations. Numerical results obtained indicate that this method is with a higher capability of spatial resolution with respect to RBF-FD method.

REFERENCES

[1] N. Flyer and B. Fornberg, (2011). Radial basis functions: Developments and applications to planetary scale flows, *Computers and Fluids*, 26, 23-32.

[2] Fornberg, B. and G. Wright (2004). Stable computation of multiquadric interpolants for all values of the shape parameter. Comp. Math. with Applications 48, 853-867.

[3] T. Cecil, J. Qian, and S. Osher, (2004). Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions, J. Comput. Phys., 196, 327-347.

[4] G. Chandhini and Y. Sanyasiraju, (2007). Local RBF-FD solutions for steady convection-diffusion problems, Int. J. Numer. Meth., 72, 352-378.

[5] C. Shu, H. Ding, and K. S. Yeo, (2003). Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng., 192, 941-954.

[6] D. Stevens, H. Power, M. Lees, and H. Morvan, (2009). The use of PDE centers in the local RBF Hermitean method for 3D convective-diffusion problems, J. Comput. Phys., 228, 4606-4624.

[7] G. B. Wright and B. Fornberg, (2006). Scattered node compact finite difference-type formulas generated from radial basis functions, J. Comput. Phys., 212, 99- 123.

[8] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr, (2012). A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere, J. Comput. Phys., 231, 4078-4095.

[9] B. Fornberg and E. Lehto, (2011). Stabilization of RBF-generated finite difference methods for convective PDEs, J. Comput. Phys., 230, 2270-2285.

[10] V. Shankar, G. B. Wright, R. M. Kirby, and A. L. Fogelson, (2014). A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction-diffusion equations on surfaces, J. Sci. Comput., 63, 745-768.

[11] N. Flyer and G. B. Wright, (2007).Transport schemes on a sphere using radial basis functions*, J. Comput. Phys,* 226, 1059-1084.

[12] W.D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber,(1992). A standard test set for numerical approximations to the shallow water equations in spherical geometry, *J. Comput. Phys*. 102, 211-224.

APPENDIX

FIGURES

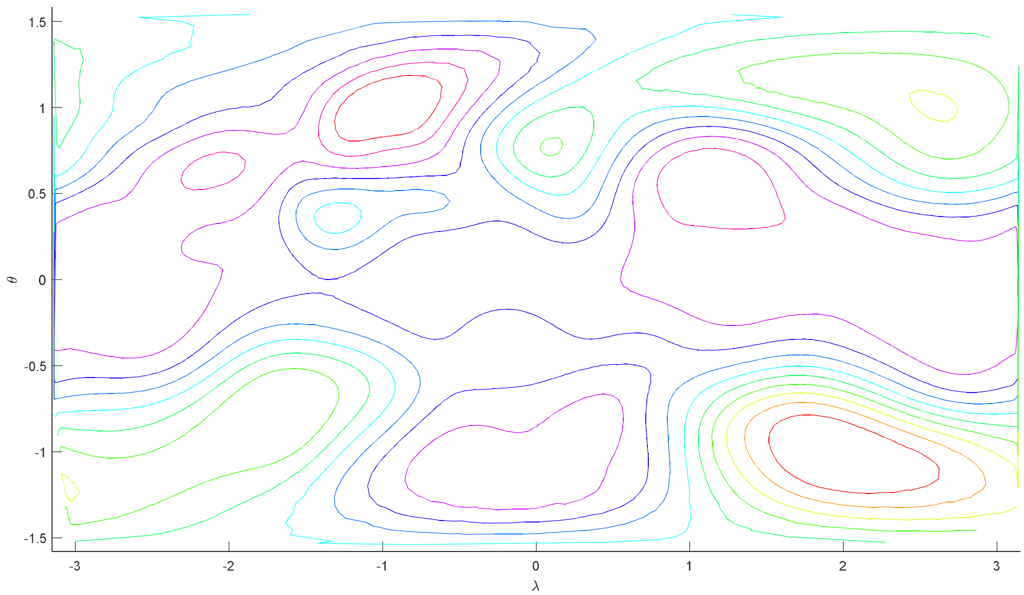

Figure 7: The height field h for flow over a conical mountain with 3600 nodes and 31 FD stencil size.

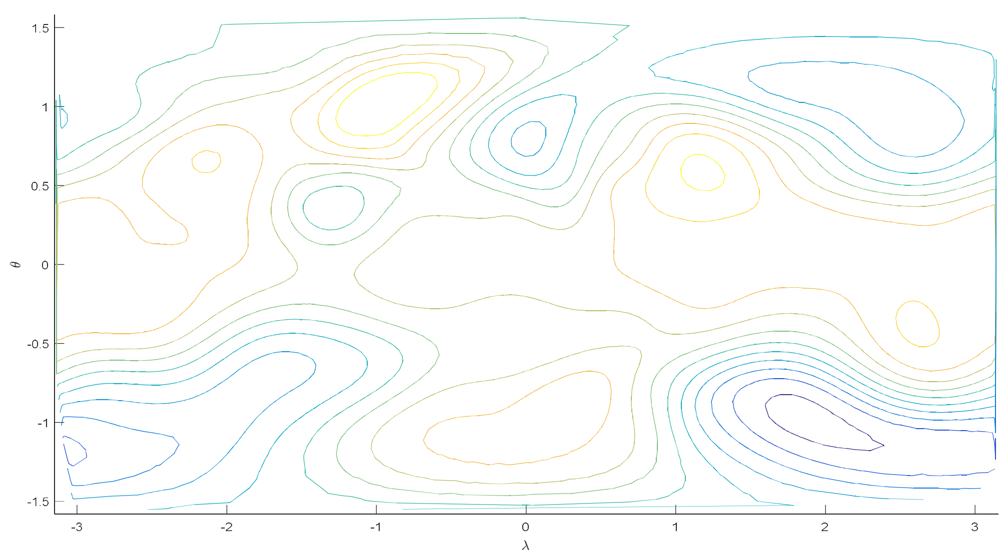

Figure 8: The height field h for flow over a conical mountain with 6400 nodes and 31 FD stencil size.