Applied Numerical Mathematics 113 (2017) 168–185

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

An adaptive meshfree spectral graph wavelet method for partial differential equations

Kavita Goyal [a,b,*], Mani Mehra [a]

[a] Department of Mathematics, Indian Institute of Technology, Delhi, Hauz Khas 110016, New Delhi, India
[b] School of Mathematics, Thapar University, Patiala, India

ARTICLE INFO

Article history:
Received 3 November 2015
Received in revised form 19 August 2016
Accepted 28 November 2016
Available online 9 December 2016

Keywords:
Multiresolution analysis (MRA)
Spectral graph wavelet
Adaptive node arrangement
Radial basis function
Meshfree methods

ABSTRACT

This paper proposes an adaptive meshfree spectral graph wavelet method to solve partial differential equations. The method uses radial basis functions for interpolation of functions and for approximation of the differential operators. It uses multiresolution analysis based on spectral graph wavelet for adaptivity. The set of scattered node points is subject to dynamic changes at run time which leads to adaptivity. The beauty of the method lies in the fact that the same operator is used for the approximation of differential operators and for the construction of spectral graph wavelet. Initially, we have applied the method on spherical diffusion equation. After that the problem of pattern formation on the surface of the sphere (using Turing equations) is addressed to test the strength of the method. The numerical results show that the method can accurately capture the emergence of the localized patterns at all the scales and the node arrangement is accordingly adapted. The convergence of the method is verified. For each test problem, the CPU time taken by the proposed method is compared with the CPU time taken by a traditional method (spectral method using radial basis functions). It is observed that the adaptive meshfree spectral graph wavelet method is highly efficient.

1. Introduction

In many real world problems, partial differential equations (PDEs) are needed to be solved. Examples of this exist in the areas of mathematical physics, fluid dynamics, image processing, medical imaging, computer graphics, and pattern formation. Many attractive mathematical properties of wavelets (namely efficient multiscale decompositions, compact support, vanishing moments and the existence of fast wavelet transform etc.) together with the techniques for preconditioning and compression of operators and matrices, motivate their use for numerical solution of PDEs. Wavelet methods have been developed for most kinds of linear PDEs such as Laplace/Poisson equations [7] and advection diffusion problems [35]. For nonlinear PDEs also, there exist a large spectrum of wavelet methods, for example Burgers equation [33,45], reaction–diffusion equations [41] and Stokes equation [10].

While solving a PDE numerically one can either choose to work on a static node arrangement [47] constructed at the beginning of the computation or can opt for an adaptive node arrangement which will keep on modifying itself according to the numerical solution of PDE at different times [1,3,29,37]. A large set of nodes is required to discover all the features of the

[*] Corresponding author.
E-mail addresses: kavita@thapar.edu (K. Goyal), mmehra@maths.iitd.ac.in (M. Mehra).

http://dx.doi.org/10.1016/j.apnum.2016.11.011
0168-9274/© 2016 IMACS. Published by Elsevier B.V. All rights reserved.

solution with a static node arrangement but this will increase the computational as well as storage cost. In some cases the set required for a static node arrangement to capture all the features of the solution may exceed the practical limitations. To deal with these problems we work on an adaptive node arrangement. In case of an adaptive node arrangement, instead of taking a larger set of nodes, more nodes are added only in the areas where the solution of PDE is having sharp features. Computational and storage costs will be saved by using an adaptive node arrangement.

For the generation of adaptive node arrangement, an indicator function is required to guide the refinement. For example in shock capturing schemes, it is the gradient which may serve as an indicator, while in the level set capturing curvature serves the purpose in a better way [26]. In this paper, we use the multiresolution analysis (MRA) based on spectral graph wavelet to guide the node arrangement adaptation.

For more than a decade, wavelets have been used to efficiently solve PDEs that exhibit multiscale solutions [2,6,25,28,33,38,45] on the manifolds with zero curvature. The critical issue is to solve PDEs on general manifolds. Many techniques have been developed to construct wavelets on general manifolds. In [5,8] wavelet bases are constructed on a specific kind of manifolds which can be represented as disjoint union of smooth parametric images of a standard cube. The construction is based solely on smooth parametrization of the unit cube, which has several disadvantages from a practical point of view. This problem is resolved in [9], where finite element based wavelet bases with respect to an arbitrary initial triangularization are constructed. Wavelets are constructed on the sphere in [21].

Despite the existence of vast literature of wavelets on general manifolds, the wavelet theory for numerical solutions of PDEs on general manifold is still in its nascent stage. In [34], second generation wavelets are used to solve PDEs on the sphere. The main difficulty with second generation wavelet is that an initial mesh structure is required to approximate the manifold (e.g. sphere can be approximated using an icosahedron mesh). However, it is difficult to find an initial mesh structure to approximate a general manifold.

This difficulty can be handled with meshfree methods. Meshfree methods are formulated based on a set of scattered nodes and mesh-related difficulties are avoided as no mesh is used. For details of meshfree methods in PDEs, one can see [13,14]. In [13], meshfess approximation methods, such as radial basis function and moving least square method are discussed from a scattered data approximation and PDEs point of views. [14] is a collection of research papers contributing to the development of this field of research. Hitherto, the meshfree methods based on wavelets are very less developed and to best of our knowledge, the developed wavelet-meshfree methods are limited to flat geometry [27,46].

The spectral graph wavelet was introduced in the year 2011 [24]. The main aim of the construction of this wavelet is to accommodate complicated manifolds. Moreover, the construction of this wavelet does not require a mesh to discretize the manifold. These features make the spectral graph wavelet a suitable candidate for meshfree methods on general manifolds. We will use MRA based spectral graph wavelet for the dynamic adaptation of the node arrangement.

The adaptive meshfree spectral graph wavelet method (AMSGWM) will use radial basis functions (RBFs) for interpolation of functions and for approximation of the differential operators. The theory of interpolation of continuous functions by RBFs is well understood [4,18,20,30,36,40]. Their use in numerical solutions of PDEs started in the year 1990 with the work E. J. Kansa [31,32] and research in this direction is progressing till date [12,15–17,19,39].

In order to test the capacity of the proposed method to resolve the local features at different scales, we applied it on the problem of pattern formation on the surface of the sphere (using Turing equations, which exhibit a highly multiscale solution). The numerical results show that the method can accurately capture the emergence of the localized patterns at all the scales and the node arrangement is accordingly adapted.

The paper is organized as follows: section 2 gives a brief description of the spectral graph wavelet. In section 3, different tools required by AMSGWM for numerical solution of PDEs are developed. Section 4 contains the numerical results of AMSGWM with special stress on Turing patterns on the sphere. Section 5 concludes the work done and gives a brief idea of the future direction.

2. A brief description of spectral graph wavelet

In this section, we will give a brief description of the spectral graph wavelet constructed in [24]. The wavelet is constructed on an arbitrary finite weighted graph. A weighted graph $G = \{E, V, w\}$ consists of a set of vertices V, a set of edges E, and a weighted function $w : E \to \mathbb{R}^+$ which assigns a positive weight to each edge. Suppose the graph is finite (i.e., the number of vertices in the graph $< \infty$). The adjacency matrix $A = \{a_{m,n}\}$ for the weighted graph G is the $N \times N$ matrix where

$$a_{m,n} = \begin{cases} w(e) & \text{if } e \in E \text{ connects vertices } m \text{ and } n \\ 0 & \text{otherwise.} \end{cases}$$

It will be assumed that the graph is undirected (which will imply that the matrix A is a symmetric matrix). It should be noted that when we are dealing with PDEs the graphs on which we are working is finite and connected.

For a weighted graph, the degree of each vertex m, written as $d(m)$, is defined as the sum of weights of all the edges incident to it, i.e., $d(m) = \sum_n a_{m,n}$. A matrix D is defined as a diagonal matrix with $d(m)$ as the diagonal entries. A nonnormalized Laplacian for the graph is defined as $\mathcal{L} = D - A$.

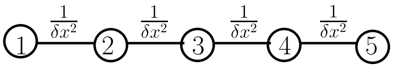


Fig. 1. A finite difference mesh.

For a graph arising from a regular mesh, the graph Laplacian corresponds to the standard stencil approximation of the continuous Laplace Beltrami operator with a difference in sign. For example for the mesh shown in the Fig. 1, the adjacency matrix A and the matrix D are given by

$$A = \begin{bmatrix} 0 & \frac{1}{\delta x^2} & 0 & 0 & 0 \\ \frac{1}{\delta x^2} & 0 & \frac{1}{\delta x^2} & 0 & 0 \\ 0 & \frac{1}{\delta x^2} & 0 & \frac{1}{\delta x^2} & 0 \\ 0 & 0 & \frac{1}{\delta x^2} & 0 & \frac{1}{\delta x^2} \\ 0 & 0 & 0 & \frac{1}{\delta x^2} & 0 \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{\delta x^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{\delta x^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\delta x^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{\delta x^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\delta x^2} \end{bmatrix}$$

Hence

$$\mathcal{L} = \frac{1}{\delta x^2} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

which is the central finite difference approximation of $-\nabla^2$ with Neumann boundary conditions.

For any $f \in \mathbb{R}^N$ defined on the vertices of the graph G, its graph Fourier transform $\hat{f}$ is defined by

$$\hat{f}(l) = \langle \chi_l, f \rangle = \sum_{n=1}^{N} \chi_l^*(n) f(n),$$

where $\{\chi_l, \quad l = 0, 1, 2, \cdots, N-1\}$ are the eigenvectors corresponding to the eigenvalues $0 = \lambda_0 < \lambda_1 \le \lambda_2 \cdots \le \lambda_{N-1}$ of the matrix $\mathcal{L}$. Note that since the graph Laplacian $\mathcal{L}$ is a real symmetric matrix, it has a complete set of orthonormal eigen vectors. The inverse graph Fourier transform is

$$f(n) = \sum_{l=0}^{N-1} \hat{f}(l) \chi_l(n).$$

2.1. Spectral graph wavelet transform (SGWT)

A kernel function $g : \mathbb{R}^+ \to \mathbb{R}^+$ is chosen satisfying $g(0) = 0$ and $\lim_{x \to \infty} g(x) = 0$ (we will refer g as wavelet kernel). For the wavelet kernel g, the wavelet operator $T_g = g(\mathcal{L})$ acts on a given function f by modulating each Fourier mode as

$$\widehat{T_g f}(l) = g(\lambda_l) \hat{f}(l),$$

which implies

$$(T_g f)(m) = \sum_{l=0}^{N-1} g(\lambda_l) \hat{f}(l) \chi_l(m).$$

The wavelet operator at scale t is then defined by $T_g^t = g(t\mathcal{L})$. It should be noted that even though the spatial domain of the graph is discrete, the domain of the wavelet kernel g is continuous and thus the scaling may be defined for any positive real number t. The spectral graph wavelets are defined as

$$\psi_n^t = T_g^t \delta_n,$$

where δ_n represents the Kronecker's delta which takes value 1 at the n^{th} vertex and 0 otherwise. This implies

$$\psi_n^t(m) = T_g^t \delta_n(m) = \sum_{l=0}^{N-1} g(t\lambda_l) \hat{\delta}_n(l) \chi_l(m) = \sum_{l=0}^{N-1} g(t\lambda_l) \chi_l^*(n) \chi_l(m).$$

The wavelet coefficients of a function f are obtained by taking the inner product of that function with these wavelets, as

$$W_f(t,n) = \langle \psi_n^t, f \rangle \text{ (SGWT)}.$$

Using the orthonormality of the $\{\chi_l\}$, the wavelet coefficients can be achieved directly from the wavelet operators as

$$W_f(t,n) = (T_g^t)(n) = \sum_{l=0}^{N-1} g(t\lambda_l)\hat{f}(l)\chi_l(n). \tag{1}$$

2.2. Spectral graph scaling function transform (SGST)

Initially, the spectral graph scaling functions are determined by a single real valued function $h : \mathbb{R}^+ \to \mathbb{R}$ which satisfies $h(0) > 0$ and $\lim_{x\to\infty} h(x) = 0$ (we will refer h as scaling function kernel). Which are given by

$$\phi_n = T_h\delta_n = h(\mathcal{L})\delta_n,$$

and the scaling function coefficients are given by

$$S_f(n) = \langle \phi_n, f \rangle \text{ (SGST)}. \tag{2}$$

Note that the scaling functions defined in this way are present merely to smoothly represent the low frequency content of the function f. They do not generate the wavelets ψ through the two-scale relation as for traditional orthogonal wavelets.

2.3. Fast SGWT and fast SGST

The naive way of computing SGWT and SGST, by directly using equation (1) and equation (2) respectively, requires explicit computation of entire set of eigenvalues and eigenfunctions of the Laplacian operator $\mathcal{L}$. This approach is computationally inefficient for large graphs. A fast transform that avoids the need for computing the complete spectrum of $\mathcal{L}$ is needed for SGWT and SGST to be a useful tool for practical computational problems (see Fig. 2).

In order to achieve this the wavelet kernel g and the scaling function kernel h are approximated by low order polynomials.

The kernels g and h are approximated by their Chebyshev polynomial expansions. The Chebyshev polynomials $T_k(x)$ may be generated by the stable recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. For $x \in [-1, 1]$, they satisfy the trigonometric expression $T_k(x) = \cos(k\cos^{-1}(x))$. The Chebyshev polynomials form an orthogonal basis for $L^2([-1,1], \frac{dx}{\sqrt{1-x^2}})$. Every $f \in L^2([-1,1], \frac{dx}{\sqrt{1-x^2}})$ has a convergent Chebyshev series

$$f(x) = \frac{1}{2}e_0 + \sum_{k=1}^{\infty} e_k T_k(x),$$

with the Chebyshev coefficients

$$e_k = \frac{2}{\pi}\int_{-1}^{1} \frac{T_k(x) f(x)}{\sqrt{1-x^2}} dx = \frac{2}{\pi}\int_{0}^{\pi} \cos(k\theta) f(\cos\theta) d\theta.$$

For a fixed wavelet scale t_j, approximating $g(t_j x)$ for $x \in [0, \lambda_{max}]$ (where λ_{max} is the largest eigenvalue of the operator $\mathcal{L}$) can be done by shifting the domain using the transformation $y = a(x+1)$ with $a = \frac{\lambda_{max}}{2}$. Denote the shifted Chebyshev polynomials $\overline{T}_k(x) = T_k\left(\frac{x-a}{a}\right)$. We may then write

$$g(t_j x) = \frac{1}{2}e_0^j + \sum_{k=1}^{\infty} e_k^j \overline{T}_k(x), \tag{3}$$

valid for $x \in [0, \lambda_{max}]$, with

$$e_k^j = \frac{2}{\pi}\int_{0}^{\pi} \cos(k\theta) g(t_j(a\cos\theta + 1)) d\theta.$$

For each scale t_j, the approximating polynomial p_j is achieved by terminating the Chebyshev expansion given by the equation (3) to M_j terms. Exactly same scheme is used to approximate the scaling function kernel h by the polynomial p_0.

The selection of the values of M_j may be considered a design problem, posing a trade off between accuracy and computational cost. The approximate wavelet and scaling function coefficients are given by

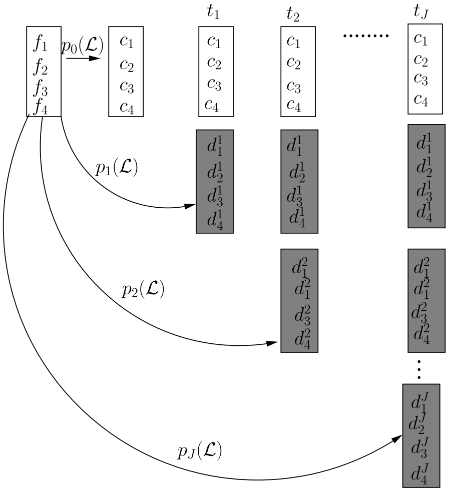


Fig. 2. Fast SGST and SWGT.

$$\widetilde{W}_f(t_j, n) = \frac{1}{2}e_0^j f(n) + \sum_{k=1}^{M_j} e_k^j \overline{T}_k(\mathcal{L}) f(n),$$

$$\widetilde{S}_f(n) = \frac{1}{2}e_0^0 f(n) + \sum_{k=1}^{M_j} e_k^0 \overline{T}_k(\mathcal{L}) f(n).$$

2.4. Spectral graph wavelet frames

The spectral graph wavelets depend on the continuous scale parameter t. For any practical computation, t must be sampled to a finite number of scales. Choosing J scales $\{t_j\}_{j=1}^{J}$ will yield a collection of NJ wavelets $\psi_n^{t_j} = \psi_n^j$, along with N scaling functions ϕ_n. The capacity of this set of vectors for representing functions on the graph is given by the following theorem.

Theorem. *Given a set of scales $\{t_j\}_{j=1}^{J}$, the set $\{\phi_n\}_{n=1}^{N} \cup \{\psi_n^j\}_{j=1,n=1}^{J\ \ N}$ forms a frame with bounds A, B given by*

$$A = \min_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

$$B = \max_{\lambda \in [0, \lambda_{N-1}]} G(\lambda),$$

where $G(\lambda) = h^2(\lambda) + \sum_j (g(t_j\lambda))^2$. Hence any function f defined on the graph can be written as

$$f(n) = \sum_{k=1}^{N} c_k \phi_k(n) + \sum_{j=1}^{J} \sum_{k=1}^{N} d_k^j \psi_k^j(n), \tag{4}$$

where $\{c_k\}_{k=1}^{N}$ are the scaling function coefficients and $\{d_k^j\}_{k=1}^{N}$ are wavelet coefficients at the scale j [24].

3. Adaptive meshfree spectral graph wavelet method (AMSGWM)

In this section we will explain AMSGWM. The RBFs are used for approximating Laplace–Beltrami and gradient operators. The approximated Laplace–Beltrami operator will serve the purpose of the matrix $\mathcal{L}$ in the construction of spectral graph wavelet. The beauty of the method lies in the fact that the same operator is used for the approximation of the differential operator involved in the differential equation and for the construction of the spectral graph wavelet.

3.1. Approximation of the Laplace–Beltrami and gradient operator using RBFs

Let X be the manifold and $X^N = \{x_1, x_2, \cdots, x_N\}$ be its discretization using N points. The RBFs are derived from bizonal kernels $\Phi : X \times X \to \mathbb{R}$ of the form $\Phi(x, y) = \varphi(x \cdot y), \quad x, y \in X$. Here φ is a univariate function defined on $[-1, 1]$ and $x \cdot y$ is Euclidean dot product of the position vectors of the points $x, y \in X$.

For a fixed value of x, the value of $\Phi(x, y)$ depends only on the geodesic distance from x to y, hence the function $\Phi(x, \cdot)$ is radially symmetric function with respect to the point x and is therefore called an RBF [22]. For every point $x_j \in X^N$, an RBF is defined as

$$\Phi_j(x) = \Phi(x, x_j) = \varphi(x \cdot x_j) = \Psi(|x - x_j|). \tag{5}$$

We have used Wendland's RBFs which depend on two parameters d and k, where d is the dimension of the space and $2k$ is the smoothness of the function Ψ. For these RBFs the matrix $\mathcal{A} = [\Phi_j(x_i)]_{i,j=1}^N$ is positive definite and hence invertible for every X^N.

Given a continuous function f on X, there uniquely exists a sequence of numbers $\{\tilde{f}^j\}_{j=1}^N$ such that the function

$$I_{X^N} f(x) = \sum_{j=1}^{N} \tilde{f}^j \Phi_j(x), \tag{6}$$

satisfies the interpolating condition

$$I_{X^N} f(x_k) = f(x_k), \quad 1 \leq k \leq N.$$

Put $x = x_1, x_2, \cdots, x_N$ in equation (6) to get

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix} = \begin{bmatrix} \Phi_1(x_1) & \Phi_2(x_1) & \cdots & \Phi_N(x_1) \\ \Phi_1(x_2) & \Phi_2(x_2) & \cdots & \Phi_N(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ \Phi_1(x_N) & \Phi_2(x_N) & \cdots & \Phi_N(x_N) \end{bmatrix} \begin{bmatrix} \tilde{f}^1 \\ \tilde{f}^2 \\ \vdots \\ \tilde{f}^N \end{bmatrix},$$

which can be written in matrix form as

$$\mathbf{f} = \mathcal{A}\tilde{\mathbf{f}}.$$

Now since the matrix $\mathcal{A}$ is invertible

$$\tilde{\mathbf{f}} = \mathcal{A}^{-1}\mathbf{f},$$

where $\mathbf{f} = [f(x_1), f(x_2), \cdots, f(x_N)]'$ etc. Hence

$$f(x) \approx I_{X^N} f(x) = \sum_{j=1}^{N} \tilde{f}^j \Phi_j(x). \tag{7}$$

A result in [23] says that $\|f - I_{X^N} f\|_\infty \leq C\|f\|_{H^s(\mathbb{R}^d)} h^{k+\frac{1}{2}}$, therefore $E^{(k)}(f) = \frac{\|f - I_{X^N} f\|_\infty}{\|f\|_{H^{2+k}(\mathbb{R}^d)}} = O(N^{-(k+\frac{1}{2})})$. Fig. 3b shows the variation of $E^{(k)}(f)$ with N for different values of k for $f(x) = \frac{4}{5\nu} e^{-\frac{(a-1)^2+b^2+c^2}{5\nu}}$, $\nu = \frac{1}{2\pi^2}$ (with $(a, b, c) \in X = S^2$, the unit sphere) which is shown in the Fig. 3a.

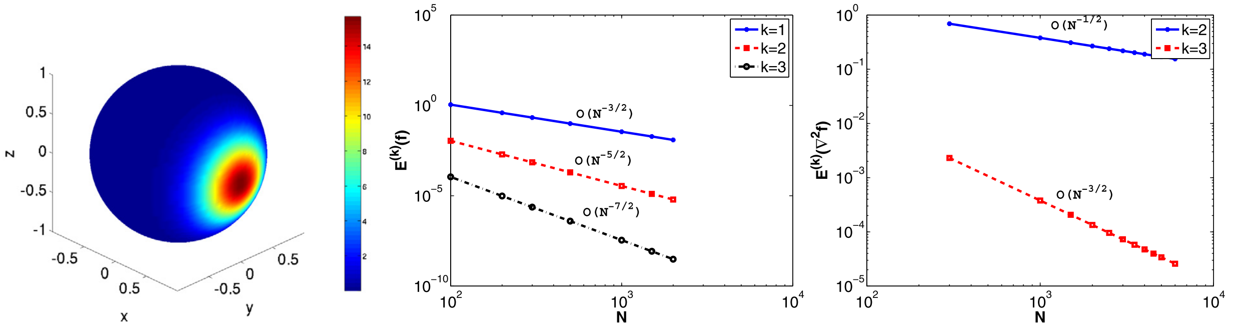
Equation (7) implies

$$\nabla^2 f(x) \approx \sum_{j=1}^{N} \tilde{f}^j \nabla^2 \Phi_j(x) = \mathcal{B}\tilde{\mathbf{f}} = \mathcal{B}\mathcal{A}^{-1}\mathbf{f}, \quad \mathcal{B} = [\nabla^2 \Phi_j(x_i)]_{i,j=1}^N.$$

Therefore the matrix $\mathcal{D}^{(2)} = \mathcal{B}\mathcal{A}^{-1}$ is an approximation of the Laplace–Beltrami operator ∇^2 on X. Applying gradient operator on both sides of the equation (7)

$$\vec{\nabla} f(x) \approx \sum_{j=1}^{N} \tilde{f}^j \vec{\nabla} \Phi_j(x) = \mathcal{C}\tilde{\mathbf{f}} = \mathcal{C}\mathcal{A}^{-1}\mathbf{f}, \quad \mathcal{C} = [\vec{\nabla} \Phi_j(x_i)]_{i,j=1}^N.$$

It is clear that the matrix $\vec{\nabla}_{app} = \mathcal{C}\mathcal{A}^{-1}$ approximates the gradient operator $\vec{\nabla}$ on X. For computation of the entries of the matrices $\mathcal{A}$ and $\mathcal{C}$ one can see [23]. The error in approximation of Laplace Beltrami operator is $E^{(k)}(\nabla^2 f) = \frac{\|\nabla^2 f - \mathcal{D}^{(2)} f\|_\infty}{\|\nabla^2 f\|_{H^{2+k}(\mathbb{R}^d)}} =$



(a) $f(x) = \frac{4}{5\nu} e^{-\frac{(a-1)^2+b^2+c^2}{5\nu}}$.

(b) Interpolation error.

(c) Error while computing Laplace-Beltrami operator.

Fig. 3. $f(x) = \frac{4}{5\nu} e^{-\frac{(a-1)^2+b^2+c^2}{5\nu}}$, $E^{(k)}(f)$ versus N, $E^{(k)}(\nabla^2 f)$ versus N.

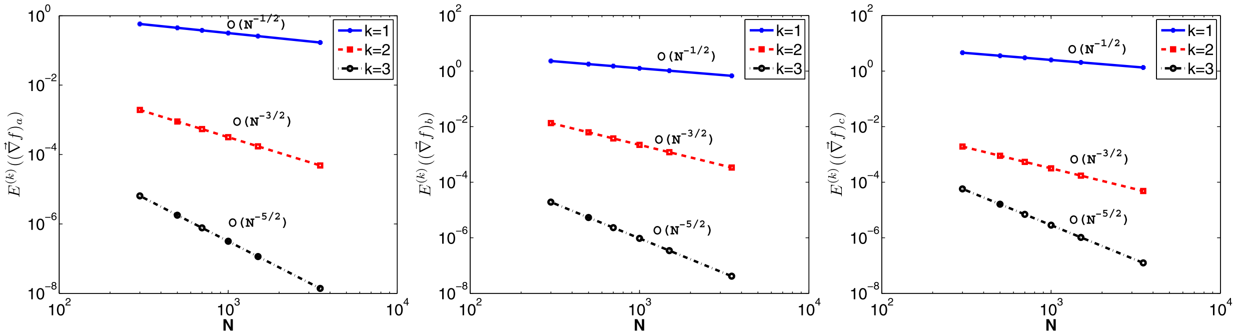


Fig. 4. Error while computing gradient operator of $f(x) = \frac{4}{5\nu} e^{-\frac{(a-1)^2+b^2+c^2}{5\nu}}$.

$O(N^{-(k-\frac{3}{2})})$. Fig. 3c shows the convergence of $\mathcal{D}^{(2)}$ to the Laplace–Beltrami operator ∇^2. The error introduced in approximating the gradient operator is $E^{(k)}((\vec{\nabla} f)_a) = \frac{\|(\vec{\nabla} f)_a - (\mathcal{D}^{(1)} f)_a\|_\infty}{\|(\vec{\nabla} f)_a\|_{H^{2+k}(\mathbb{R}^d)}} = O(N^{-(k-\frac{1}{2})})$. Note that here $(\vec{\nabla} f)_a$ stands for component of the vector $\vec{\nabla} f$ in the direction of the unit vector $\hat{i}$ etc. Fig. 4 shows the convergence for $f(x) = \frac{4}{5\nu} e^{-\frac{(a-1)^2+b^2+c^2}{5\nu}}$. The expected orders of convergence are achieved.

3.2. Interpolation using RBFs

Suppose we have two sets of node points X^c and X^f (X^c is the coarse set and X^f is fine set). Suppose $\{f(x_j)\}_{x_j \in X^c}$ is known. Using RBFs we can compute $\{f(x_j)\}_{x_j \in X^f}$ as follows

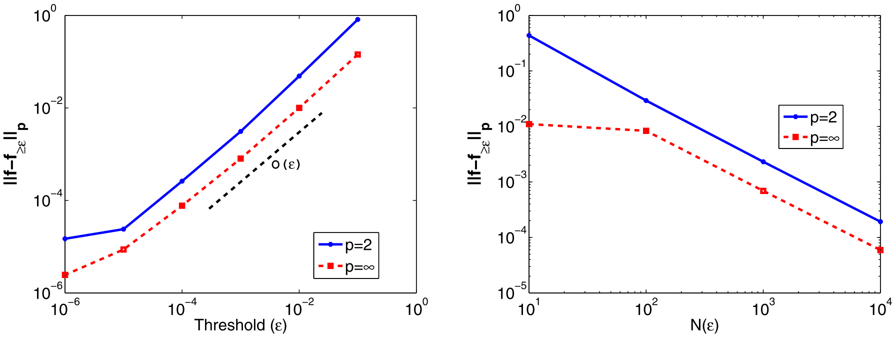
$$\begin{aligned}
\{f(x_j)\}_{x_j \in X^c} &= [\Phi(x_i, x_j)]_{\{x_i, x_j \in X^c\}} \{\tilde{f}^j\}^{j=1,2,\cdots,\#X^c},\\
\{f(x_j)\}_{x_j \in X^f} &= [\Phi(x_i x_j)]_{\{x_i \in X^f, x_j \in X^c\}} \{\tilde{f}^j\}^{j=1,2,\cdots,\#X^c},\\
\{f(x_j)\}_{x_j \in X^f} &= [\Phi(x_i, x_j)]_{\{x_i \in X^f, x_j \in X^c\}} [\Phi(x_i, x_j)]^{-1}_{\{x_i, x_j \in X^c\}} \{f(x_j)\}_{x_j \in X^c}.
\end{aligned}$$

3.3. Reconstruction and compression error

For any $f : V \to \mathbb{R}$ defined on the vertices of the graph and a given threshold ϵ, equation (4) can be written as $f(n) = f_{\geq\epsilon}(n) + f_{<\epsilon}(n)$, where

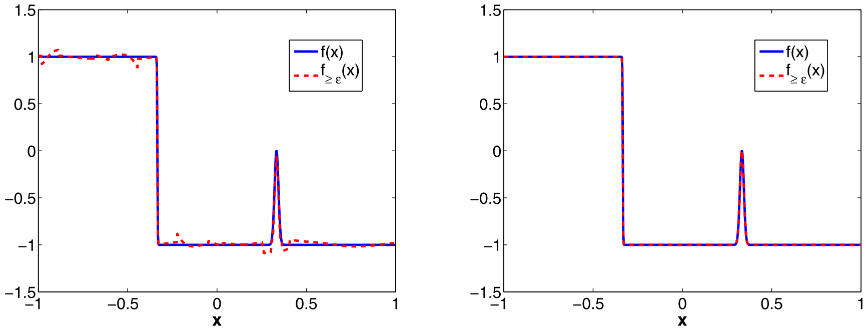
$$f_{\geq\epsilon}(n) = \sum_{k=1}^{N} c_k \phi_k(n) + \sum_{j=1}^{J} \sum_{|d_k^j| \geq \epsilon} d_k^j \psi_k^j(n) \text{ and } f_{<\epsilon}(n) = \sum_{j=1}^{J} \sum_{|d_k^j| < \epsilon} d_k^j \psi_k^j(n).$$

$\|f - f_{\geq\epsilon}\|$ is called the compression error and for $\epsilon = 0$ there is no compression and in that case it is reconstruction error. The number of the node points required for $f_{\geq\epsilon}$ is called number of significant coefficients denoted by $N(\epsilon)$. Donoho in [11] proved that for sufficiently smooth f



(a) $\|f - f_{\geq\epsilon}\|_p$ versus ϵ for test function 1. (b) $\|f - f_{\geq\epsilon}\|_p$ versus $N(\epsilon)$ for test function 1.

Fig. 5. Variation of compression error versus ϵ and $N(\epsilon)$ for test function 1.



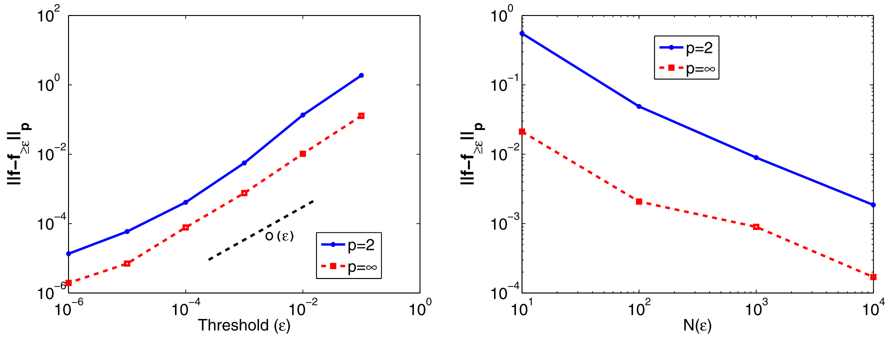
(a) $f(x)$ and $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-1}$. (b) $f(x)$ and $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-4}$.

Fig. 6. Test function 1 and the corresponding reconstructed function for different values of ϵ.

$$\|f - f_{\geq\epsilon}\|_\infty < C\epsilon, \tag{8}$$

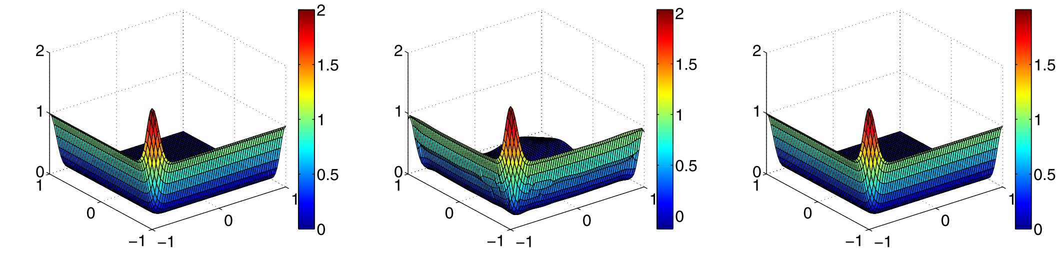
where C is a constant.

- We have taken the test function 1 as; $f(x) = -\tanh\left(\frac{x+x_0}{2\nu}\right) + e^{-64^2(x-x_0)^2}$ with $x_0 = \frac{1}{3}$ and $\nu = 10^{-3}$ on the interval $[-1, 1]$. Following results are obtained
  - The reconstruction is of the order 10^{-8} for $N = 128$.
  - Fig. 5a shows the relation between the compression error $\|f - f_{\geq\epsilon}\|_p$ and the threshold ϵ. It can be observed that the relation (8) is satisfied.
  - Fig. 5b shows the graph between the compression error and $N(\epsilon)$ for $\epsilon = 10^{-1}$. It can be observed that, the compression error decreases with $N(\epsilon)$.
  - Fig. 6a and Fig. 6b show $f(x)$ and the $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-1}$ and $\epsilon = 10^{-4}$ respectively.
- We have taken the test function 2 as; $f(x) = e^{-50(a+1)^2} + e^{-50(b+1)^2}$, $x = (a, b) \in [-1, 1] \times [-1, 1]$ shown in the Fig. 8a. Following results are obtained
  - The reconstruction is of the order 10^{-8} for $N = 128$.
  - Fig. 7a shows the relation between the compression error $\|f - f_{\geq\epsilon}\|_p$ and the threshold ϵ.
  - Fig. 7b shows the graph between the compression error and $N(\epsilon)$ for $\epsilon = 10^{-1}$.
  - Fig. 8b and Fig. 8c show $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-1}$ and $\epsilon = 10^{-4}$ respectively.
- We have taken test function 3 as; $f(x) = \frac{4}{5\nu} e^{\frac{-((a-a_0)^2+(b-b_0)^2+(c-c_0)^2)}{5\nu}}$, $a_0 = 1, b_0 = 0, c_0 = 0$ and $\nu = \frac{1}{2\pi^2}$ on the unit sphere shown in Fig. 10a. Following results are obtained
  - The reconstruction is of the order 10^{-8} for $N = 128$.
  - Fig. 9a shows the relation between the compression error $\|f - f_{\geq\epsilon}\|_p$ and the threshold ϵ.
  - Fig. 9b shows the graph between the compression error and $N(\epsilon)$ for $\epsilon = 10^{-1}$.
  - Fig. 10a and Fig. 10b show $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-1}$ and $\epsilon = 10^{-4}$ respectively.



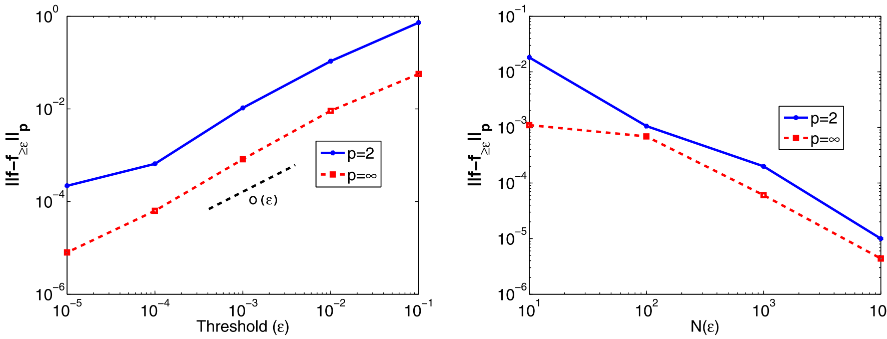
(a) $\|f - f_{\geq\epsilon}\|_p$ versus ϵ for test function 2. (b) $\|f - f_{\geq\epsilon}\|_p$ versus $N(\epsilon)$ for test function 2.

Fig. 7. Variation of compression error versus ϵ and $N(\epsilon)$ for test function 2.



(a) $f(x) = e^{-50(a+1)^2} + e^{-50(b+1)^2}$. (b) $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-1}$. (c) $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-4}$.

Fig. 8. Test function 2 and the corresponding reconstructed function for different values of ϵ.



(a) $\|f - f_{\geq\epsilon}\|_p$ versus ϵ for test function 3. (b) $\|f - f_{\geq\epsilon}\|_p$ versus $N(\epsilon)$ for test function 3.

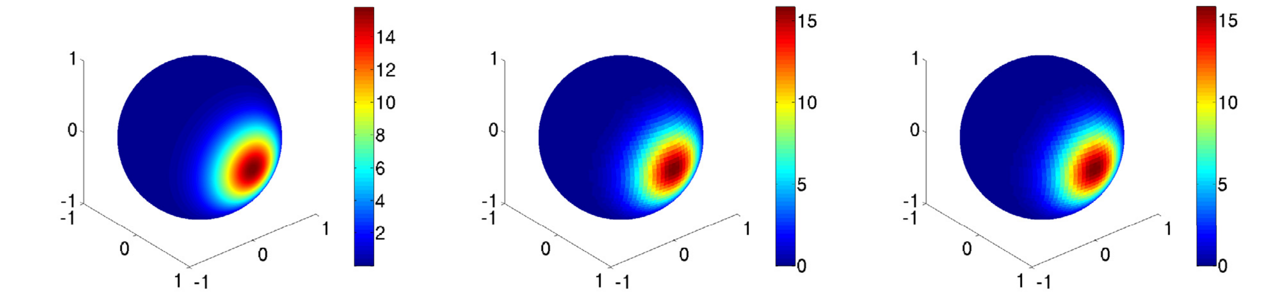
Fig. 9. Variation of compression error versus ϵ and $N(\epsilon)$ for test function 3.

3.4. Adaptive node arrangement

Node adaptation occurs quite naturally in wavelet methods [25,33]. One of the important property of wavelet is that the wavelet coefficients d_k^j decreases rapidly for smooth functions. Moreover, if a function has a discontinuity in one of its derivatives then the wavelet coefficients will decrease slowly only near the point of discontinuity and maintain fast decay where the function is smooth. This property of wavelet makes it suitable to detect where in the numerical solution of a PDE the shocks are located and hence an adaptive node arrangement can be generated.

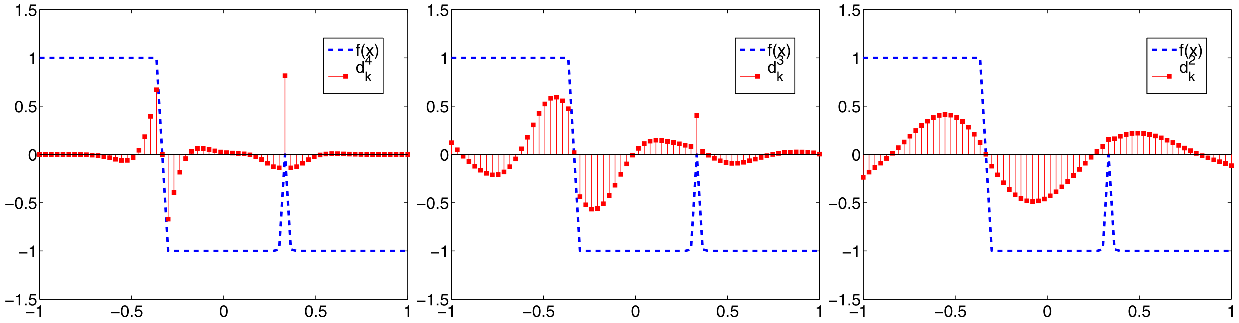
In case of spectral graph wavelet, if $\{t_j\}_{j=1}^{J}$ is the set of scales, then d_k^j follows the above said fact strictly when j is near J. Fig. 11 and Fig. 12 show d_k^js for two different functions for different values of j (note that the $J = 4$). Hence we will consider d_k^Js (wavelet coefficients at scale t_J) for node adaptation.

To illustrate the algorithm for node adaptation, let us consider a function $f(x)$ defined on a set of node points $\mathcal{G}^{old}$ and a pre-decided threshold parameter ϵ. Perform SGST and SGWT to obtain the set of coefficients $\{c_k\}_{k=1}^{|\mathcal{G}^{old}|} \cup \{d_k^j\}_{j=1,k=1}^{J \quad |\mathcal{G}^{old}|}$. We will construct $\mathcal{G}^{new}$ (fine node arrangement) from $\mathcal{G}^{old}$ (coarse node arrangement). Since each $x_k \in \mathcal{G}^{old}$ is associated with a



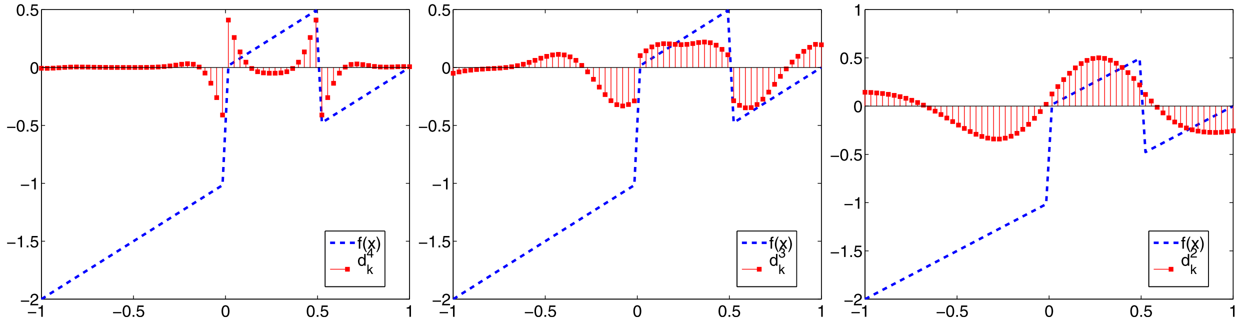
(a) $f(x) = \frac{4}{5\nu} e^{\frac{-((a-a_0)^2+(b-b_0)^2+(c-c_0)^2)}{5\nu}}$. (b) $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-1}$. (c) $f_{\geq\epsilon}(x)$ for $\epsilon = 10^{-4}$.

Fig. 10. Test function 3 and the corresponding reconstructed function for different values of ϵ.



(a) d_k^4 (b) d_k^3 (c) d_k^2

Fig. 11. d_k^j for different value of j for $f(x) = -\tanh\left(\frac{x+x_0}{2\nu}\right) + e^{-64^2(x-x_0)^2}$, $x_0 = \frac{1}{3}$, $\nu = 10^{-3}$.



(a) d_k^4 (b) d_k^3 (c) d_k^2

Fig. 12. d_k^j for different value of j for $f(x) = x$, if $0 < x < 0.5$ and $x - 1$ otherwise.

spectral scaling function ϕ_k, we will keep all the points of $\mathcal{G}^{old}$ in $\mathcal{G}^{new}$. We recall that every spectral wavelet function at the scale t_J, i.e. ψ_k^J, is uniquely associated with an $x_k \in \mathcal{G}^{old}$. Analyze wavelet coefficients d_k^J. If $|d_k^J| \geq \epsilon$, then the corresponding node point $x_k \in \mathcal{G}^{old}$ is called an active node point. For each active node point x_k, an adjacent zone will be added in $\mathcal{G}^{new}$. Adjacent zone will serve the following two purposes:

1. It will make $\mathcal{G}^{new}$ finer in the area of sharp features.
2. In solving evolution PDEs, additional criteria for node adaptation should be added. In particular, the computational set of node points should consist of node points associated with wavelets whose coefficients are or can possibly become significant during the period of the time when the set of node points remain unchanged [33].

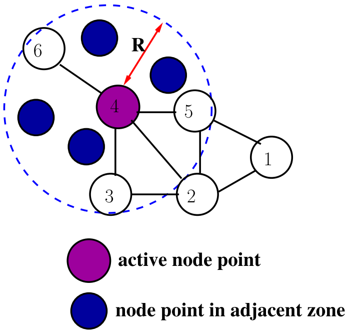
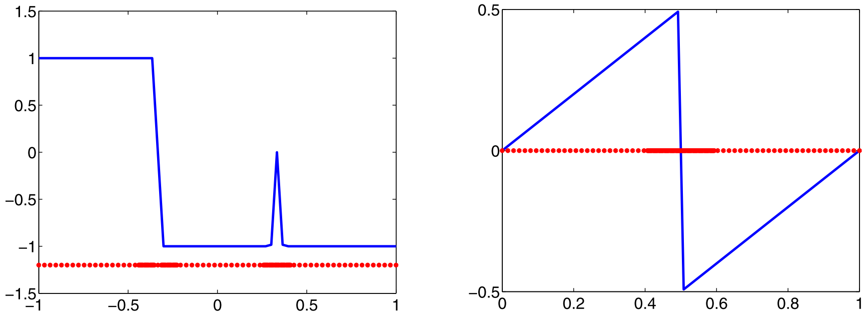


Fig. 13. Adjacent zone.



(a) $f(x) = -\tanh\left(\frac{x+x_0}{2\nu}\right) + e^{-64^2(x-x_0)^2}$, $x_0 = \frac{1}{3}, \nu = 10^{-3}$ and corresponding adaptive node arrangement.

(b) Sawtoooth function with discontinuity at $x = 0.5$ and corresponding node arrangement.

Fig. 14. Functions and the corresponding adaptive node arrangements for different functions in one dimensional setting for $R = 0.1$ and $M = 4$.

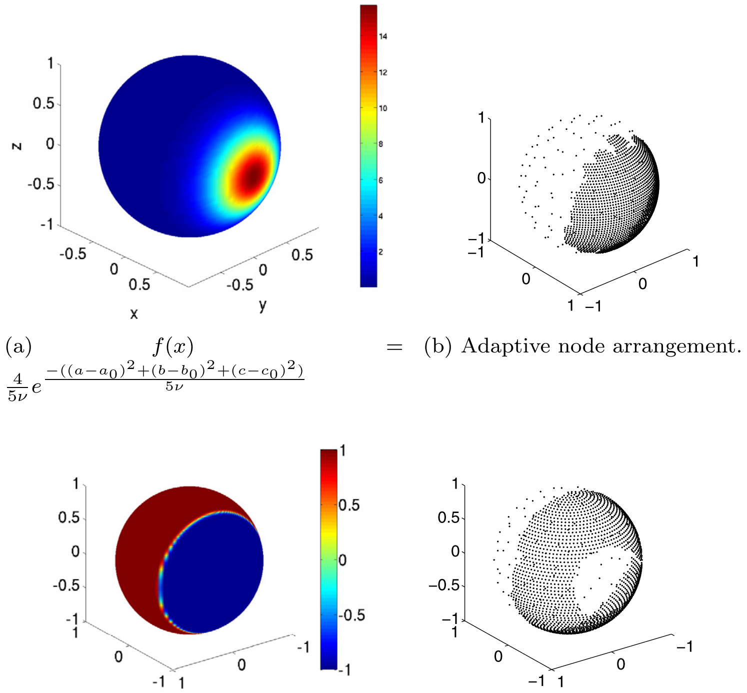
For an active node point x_k, we fix two positive numbers R and M (called adjacent zone constants). These parameters are problem dependent. In the region $geodis(x - x_k) \leq R$ (*geodis* stands for the geodesic distance) we obtain M points simulating the surface of the manifold. This will be called the adjacent zone of the point x_k. For example as shown in the Fig. 13, if node point 4 is an active node point then for $M = 4$ we have added 4 points in the region $geodis(x - x_4) \leq R$. The process of spectral graph wavelet based adaptive node arrangement generation is as follows

```
$\mathcal{G}^{new}$ = **AdaptiveNodeArrangement**($f, \mathcal{G}^{old}$)
  • Choose a threshold parameter $\epsilon$ and positive adjacent zone constants $R$ and $M$.
  • $m = 0$
  • $\mathcal{G}^m = \mathcal{G}^{old}$
  • $f^m = f$
  • **do while** $m = 0$ or $\mathcal{G}^m \neq \mathcal{G}^{m-1}$
      – Perform SGWT on $f^m$ to get $\{d_k^J\}_{k=1}^{|\mathcal{G}^m|}$.
      – $\mathcal{G}^{m+1} = \mathcal{G}^m$.
      – Analyze wavelet coefficients $\{d_k^J\}_{k=1}^{|\mathcal{G}^m|}$ and collect all the active node points.
      – Corresponding to each active node point, add an adjacent zone in $\mathcal{G}^{m+1}$.
      – Interpolate $f^m$ onto new grid $\mathcal{G}^{m+1}$ using RBFs as explained in section 3.2 and call it $f^{m+1}$.
      – $m = m + 1$
  • $\mathcal{G}^{new} = \mathcal{G}^m$
```

Fig. 14a and Fig. 14b show the adaptive node arrangement for two different functions in one-dimensional setting using the above defined algorithm. Fig. 15 shows two different functions on the sphere and the corresponding adaptive node arrangement.

3.5. Calculation of differential operators on an adaptive node arrangement

While solving PDEs numerically, it is necessary to approximate differential operators of a function from its values at collocation point. A procedure to approximate differential operators, which takes advantage of multiresolution wavelet de-



(a) $f(x) = \frac{4}{5\nu} e^{\frac{-((a-a_0)^2+(b-b_0)^2+(c-c_0)^2)}{5\nu}}$

(b) Adaptive node arrangement.

(c) $f = tanh\left(\frac{a+sin(\frac{\pi}{6})}{2\tau}\right), \tau = 10^{-2}$.

(d) Adaptive node arrangement.

Fig. 15. Functions and the corresponding adaptive node arrangements for different functions on the sphere for $R = 0.1$ and $M = 4$.

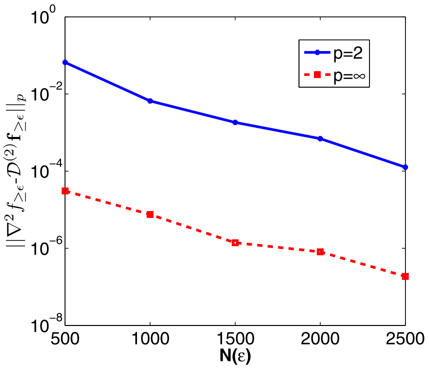


Fig. 16. $\|\nabla^2 f_{\geq\epsilon} - \mathcal{D}^2\mathbf{f}_{\geq\epsilon}\|$ versus $N(\epsilon)$.

composition, fast wavelet transform, and finite difference differentiation is discussed in [45] for one dimensional case, and in [44] for multiple dimensions, but restricted to rectangular domains and flat geometry.

We will use RBFs to obtain the approximation of differential operators on an adaptive node arrangement. We have already computed Laplace–Beltrami and gradient operators using RBFs in section 3.1. For the sake of completeness, we describe here the computation of a general differential operator $\mathcal{T}$, using RBFs on a node arrangement $\mathcal{G}$. Given $\{f(x_j)\}_{x_j\in\mathcal{G}}$, $\{\mathcal{T}f(x_j)\}_{x_j\in\mathcal{G}}$ can be computed as follows

$$\{f(x_j)\}_{x_j\in\mathcal{G}} = [\Phi(x_i, x_j)]_{\{x_i,x_j\in\mathcal{G}\}}\{\tilde{f}^j\}^{j=1,2,\cdots,\#\mathcal{G}},$$
$$\{\mathcal{T}f(x_j)\}_{x_j\in\mathcal{G}} = [\mathcal{T}\Phi(x_i\, x_j)]_{\{x_i,x_j\in\mathcal{G}\}}\{\tilde{f}^j\}^{j=1,2,\cdots,\#\mathcal{G}}.$$

Hence

$$\{\mathcal{T}f(x_j)\}_{x_j\in\mathcal{G}} = [\mathcal{T}\Phi(x_i\, x_j)]_{\{x_i,x_j\in\mathcal{G}\}}[\Phi(x_i, x_j)]^{-1}_{\{x_i,x_j\in\mathcal{G}\}}\{f(x_j)\}_{x_j\in\mathcal{G}}.$$

Fig. 16 verifies the convergence of the approximate Laplace–Beltrami operator on an adaptive node arrangement.

3.6. *Numerical algorithm for PDEs*

We now have all the ingredients necessary to construct an adaptive meshfree spectral graph wavelet method for the solution of PDEs on general manifold. The basic steps of the algorithm are the following

1. Knowing the solution $\mathbf{u}_{\geq}(t)$ on the node arrangement $\mathcal{G}^t$, we construct $\mathcal{G}^{t+\Delta t}$ using $\mathcal{G}^{t+\Delta t} =$ **AdaptiveNodeArrangement** $(\mathbf{u}_{\geq}(t), \mathcal{G}^t)$.
2. If there is no change between the node arrangements $\mathcal{G}^t$ and $\mathcal{G}^{t+\Delta t}$, we go directly to next step. Otherwise we interpolate the solution onto the new node arrangement $\mathcal{G}^{t+\Delta t}$ using RBFs as explained in section 3.2.
3. Compute the differential operators involved in the PDE on $\mathcal{G}^{t+\Delta t}$ as explained in section 3.5.
4. We integrate the resulting system of ordinary differential equations in time (i.e. using Runge–Kutta) to obtain the solution $\mathbf{u}_{\geq}(t+\Delta t)$ at time $t+\Delta t$.
5. Perform several time steps on the same node arrangement. (Note that how many steps to take between node arrangement evaluations depends on the problem at hand. If the problem is such that its solution is rapidly changing, then the node arrangement is needed to be evaluated frequently. If the solution is slowly changing, then the same node arrangement can be used for many time steps.)

The order of accuracy of the proposed method before introducing the adaptivity is $O(N^{-3/2} + \Delta t)$ ($k = 3$ in numerical experiments). For a threshold ϵ, the error introduced during adaptivity is $O(\epsilon)$. Therefore the order of accuracy of the final scheme is $O((N(\epsilon))^{-3/2} + \Delta t + \epsilon)$.

4. Numerical results

In order to illustrate the accuracy and efficiency of the proposed numerical method, initially we apply it to a simple problem of spherical diffusion. Then the method is applied to a more challenging problem of pattern formation on the unit sphere (using Turing equations).

4.1. *Spherical diffusion equation*

For the first test problem, we consider diffusion equation

$$\frac{\partial u(x,t)}{\partial t} = \nu \nabla^2 u(x,t) + f, x \in X = S^2, \tag{9}$$

where f is a localized source chosen in such a way that the solution of the diffusion equation is given by

$$u(x,t) = e^{-\frac{(\theta-\theta_0)^2 + (\phi-\phi_0)^2}{\nu(t+1)}},$$

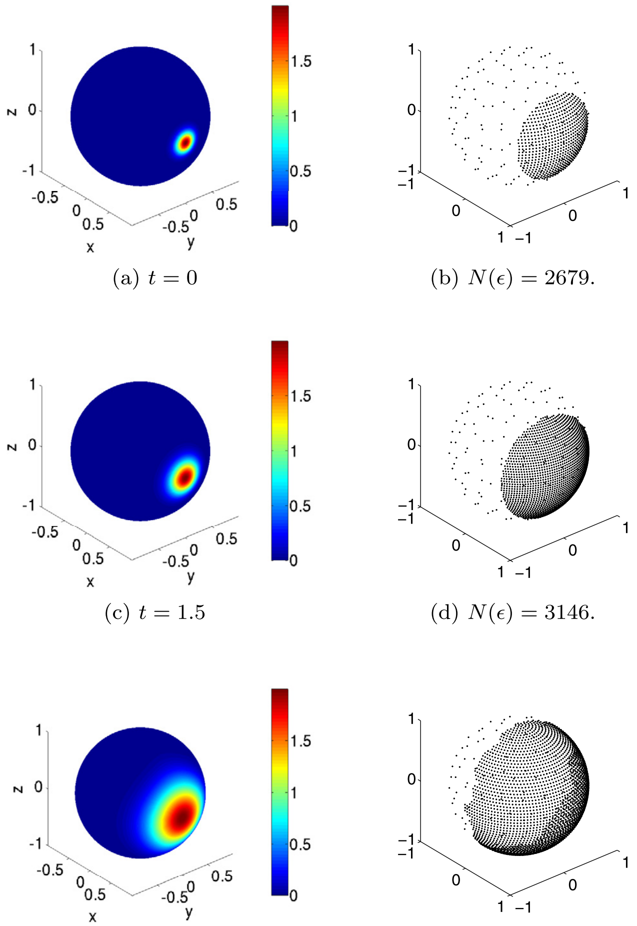
(r, θ, ϕ) is the spherical polar representation of the point $x \in X = S^2$. The initial condition is obtained from the analytic solution, and the problem is solved for the parameters $\nu = 1/(4\pi)^2$, $\theta_0 = 0, \phi_0 = 0$.

The solution of the problem and the corresponding adaptive node arrangement is shown in the Fig. 17 for $\epsilon = 10^{-4}$. In order to demonstrate the efficiency of the proposed method, we compared the CPU time taken by AMSGWM (denoted by CPU(ϵ)) with the CPU time taken by the spectral method using RBFs (denoted by CPU($\epsilon = 0$), note that regular spectral+RBF method uses the number of nodes equal to the maximum number of nodes used by AMSGWM). For this comparison we define a parameter $\Theta = \frac{\text{CPU}(\epsilon=0)}{\text{CPU}(\epsilon)}$ and call it efficiency coefficient. The larger the efficiency coefficient, the more efficient the adaptive algorithm. $\Theta = 1$ indicates that there is no compression and the node arrangement is non-adaptive. Fig. 18a shows the variation of Θ with ϵ. It is evident from the graph that as we increase ϵ, Θ will increase. Hence the method will be more efficient if we choose large value of ϵ. But for large value of ϵ, the error $\|u_{num} - u_{ana}\|_p$ (u_{num} stands for the numerical solution of the problem and u_{ana} stands for the analytic solution) is also large. So we have to choose an optimal value of ϵ which makes a balance between the efficiency and accuracy.

Next, we test the convergence of the method. Fig. 18b shows the graph between the error ($\|u_{num} - u_{ana}\|_p$) and the number of significant node points $N(\epsilon)$. The expected order of convergence is achieved.

4.2. *Turing patterns on the sphere*

In 1952, A. M. Turing [42] settled the basis for explaining biological patterns using two interacting chemicals that, under certain conditions, can generate stable patterns from an initial near-homogeneity. This phenomenon has now been shown to occur in chemistry and biology. The Turing patterns are governed by a system of nonlinear reaction–diffusion equations. We solve the following system [43]



(a) $t = 0$

(b) $N(\epsilon) = 2679$.

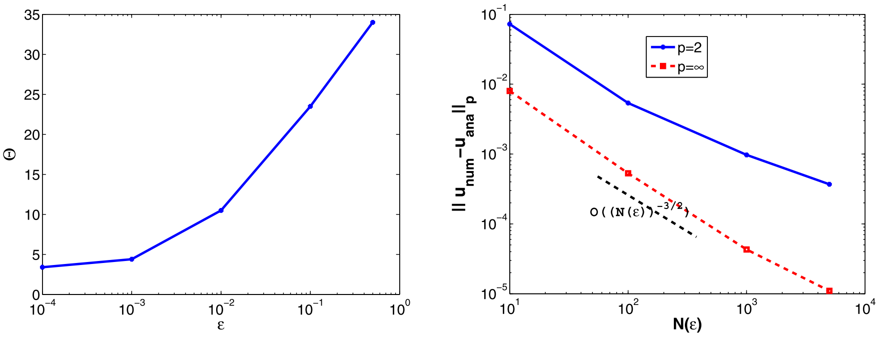
(c) $t = 1.5$

(d) $N(\epsilon) = 3146$.

(e) $t = 10$.

(f) $N(\epsilon) = 5361$.

Fig. 17. Evolution of the solution and dynamically adapted node arrangement for the spherical diffusion equation for $\epsilon = 10^{-4}$, $R = 0.1$ and $M = 4$.



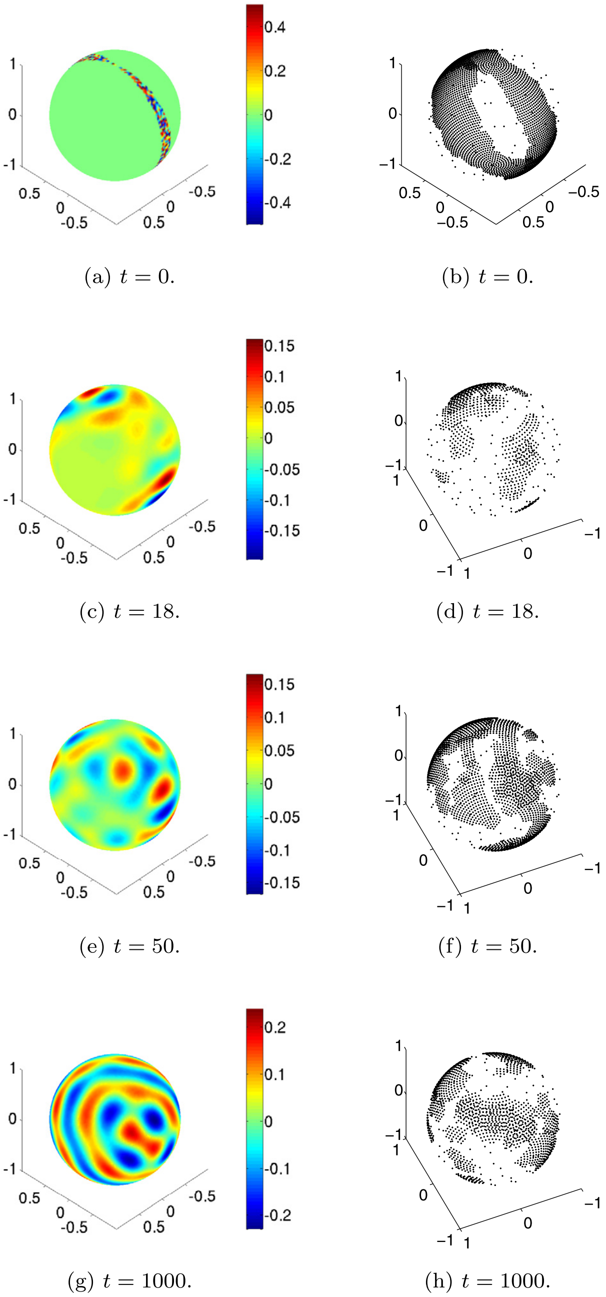
(a) Θ versus ϵ for spherical diffusion equation at $t = 0.5$.

(b) $\|u_{num} - u_{ana}\|_p$ versus $N(\epsilon)$ for spherical diffusion equation.

Fig. 18. Results for spherical diffusion equation.

$$\frac{\partial u}{\partial t} = D\delta\nabla^2 u + \alpha u(1 - r_1 v^2) + v(1 - r_2 u),$$

$$\frac{\partial v}{\partial t} = \delta\nabla^2 v + \beta v\left(1 + \frac{\alpha r_1}{\beta} u v\right) + u(\gamma + r_2 v). \tag{10}$$



(a) $t = 0$.

(b) $t = 0$.

(c) $t = 18$.

(d) $t = 18$.

(e) $t = 50$.

(f) $t = 50$.

(g) $t = 1000$.

(h) $t = 1000$.

Fig. 19. Evolution of the solution and dynamically adapted node arrangement for the Turing problem for $r_1 = 3.5$, $r_2 = 0$, $\epsilon = 10^{-4}$, $R = 0.1$ and $M = 4$ (only one component u is shown).

At initial state, i.e., at $t = 0$ we consider $u = v = 0$, except on a narrow band as shown in Fig. 19a. The stable patterns can be either stripes or spots, depending on the parameters r_1 and r_2. The parameter r_1 favors stripes while r_2 favors spots. We fix the parameters $D = 0.516$, $\delta = 0.0045$, $\alpha = 0.899$, $\beta = -0.91$ and $\gamma = -\alpha$.

As **case** 1, we take $r_1 = 3.5$, $r_2 = 0$. The evolution of the striped pattern and the corresponding adaptive node arrangement is shown in the Fig. 19. It is evident that AMSGWM is able to track the emergence of banded structures over the entire sphere. This demonstrates that the proposed method is able to efficiently and accurately capture the emergence of the localized structures at multiscales which characterize the solution of nonlinear PDEs. Fig. 20 shows the variation of Θ with respect to the threshold ϵ and this graph also gives a glimpse of the efficiency of the proposed method. It can be observed from the graph that, at $\epsilon = 10^{-1}$, $\Theta = 20$ which means there is a 95% reduction in the CPU time. With the computational facility available with us, we can not calculate CPU($\epsilon = 0$) at higher times and this is the reason we have shown the variation of Θ only at $t = 18$. But the efficiency of AMSGWM will increase with time.

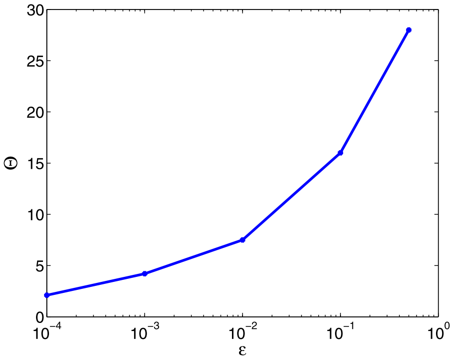
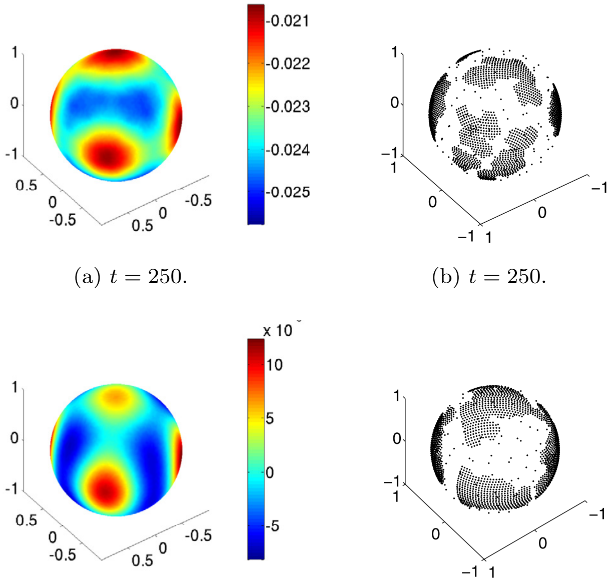


Fig. 20. Θ versus ϵ for Turing problem for case 1 at $t = 18$.



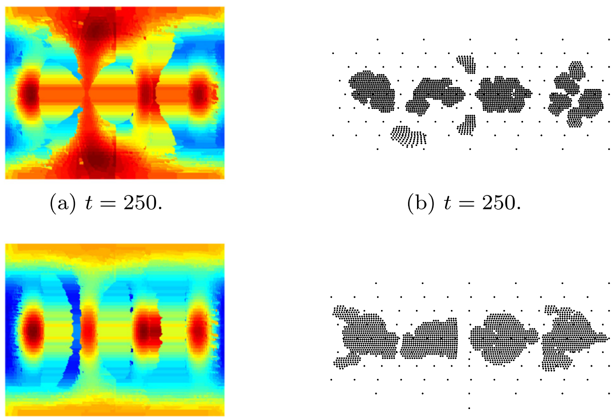
(a) $t = 250$.

(b) $t = 250$.

(c) $t = 1000$.

(d) $t = 1000$.

Fig. 21. Evolution of the solution and the corresponding dynamically adapted node arrangement for the Turing problem for $r_1 = 0.02$, $r_2 = 0.2$, $\epsilon = 10^{-4}$, $R = 0.1$ and $M = 4$ (only one component u is shown).



(a) $t = 250$.

(b) $t = 250$.

(c) $t = 1000$.

(d) $t = 1000$.

Fig. 22. $x - y$ maps of the solution and the corresponding dynamically adapted node arrangement for case 2 of Turing problem.

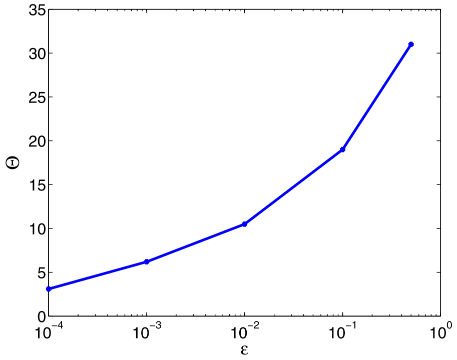


Fig. 23. Θ versus ϵ for Turing problem for case 2 at $t = 25$.

As **case** 2, we take $r_1 = 0.02$, $r_2 = 0.2$. The evolution of the spot pattern and the corresponding adaptive node arrangement is shown in the Fig. 21. To have a better view of the configuration of the spots, $x - y$ maps of the solutions and the corresponding adaptive node arrangement is shown in the Fig. 22. Fig. 23 shows the variation of Θ with respect to the threshold ϵ.

5. Conclusion and future work

In this paper an adaptive meshfree spectral graph wavelet method is developed to solve PDEs. The method uses radial basis functions for interpolation of functions and for approximation of the differential operators. It uses MRA based spectral graph wavelet for the adaptation of the node arrangement. The efficiency and the convergence of the method is verified. The CPU time taken by AMSGWM and by the spectral method using RBFs has been compared and it turns out that AMSGWM performs much better. To best of our knowledge the useful properties of spectral graph wavelet are for the first time exploited to solve PDEs on an adaptive node arrangement. In future we will use AMSGWM to solve real life problems on complex manifolds and for turbulence modeling in particular.

Matlab codes

The 1D codes are provided on the following webpage:
https://sites.google.com/site/drkavitagoyalmathematics/matlab-codes/adaptive-grid-using-spectral-graph-wavelet

Acknowledgement

The first author would like to thank Council of Scientific and Industrial Research for providing PhD scholarship.

References

[1] J.M. Alam, N.K.-R. Kevlahan, O.V. Vasilyev, Simultaneous space–time adaptive wavelet solution of nonlinear parabolic differential equations, J. Comput. Phys. 214 (2006) 829–857.

[2] M. Bergdorf, P. Koumoutsakos, A Lagrangian particle-wavelet method, Multiscale Model. Simul. 5 (2006) 980–995.

[3] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.

[4] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, 2003.

[5] C. Canuto, A. Tabacco, K. Urban, The wavelet element method. Part I: construction and analysis, Appl. Comput. Harmon. Anal. 6 (1999) 1–52.

[6] A. Cohen, W. Dahmen, R. DeVore, Adaptive wavelet methods for elliptic operator equations, Math. Comput. 70 (2001) 27–75.

[7] W. Dahmen, Wavelet and multiscale methods for operator equations, Acta Numer. 6 (1997) 55–228.

[8] W. Dahmen, R. Schneider, Wavelets on manifolds I: construction and domain decomposition, SIAM J. Math. Anal. 31 (1999) 184–230.

[9] W. Dahmen, R. Stevenson, Element-by-element contruction of wavelets satisfying stability and moment conditions, SIAM J. Numer. Anal. 37 (1999) 319–352.

[10] W. Dahmen, K. Urban, J. Vorloeper, Adaptive Wavelet Methods: Basic Concepts and Applications to the Stokes Problem, World Scientific, Singapore, 2002, pp. 39–80.

[11] D.L. Donoho, Interpolating wavelet transforms, Tech. Rep. 408, Department of Statistics, Stanford University, 1992.

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

[13] G.E. Fasshuer, Meshfree Approximation Methods with Matlab, World Scientific, 2007.

[14] A. Ferreira, E.J. Kansa, G.E. Fasshuer, V.M.A. Leitao, Progress in Meshless Methods, Springer, 2009.

[15] N. Flyer, G.B. Wright, Transport schemes on a sphere using radial basis functions, J. Comput. Phys. 226 (1) (2007) 1059–1084.

[16] N. Flyer, G.B. Wright, A radial basis function method for the shallow water equations on a sphere, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 465 (2106) (2009) 1949–1976.

[17] B. Fornberg, E. Lehto, Stabilization of RBF-generated finite difference methods for convective PDEs, J. Comput. Phys. 230 (6) (2011) 2270–2285.

[18] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM J. Sci. Comput. 30 (1) (2007) 60–80.

[19] B. Fornberg, C. Piret, On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, J. Comput. Phys. 227 (5) (2008) 2758–2780.

[20] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, SIAM J. Sci. Comput. 33 (2) (2011) 869–892.

[21] W. Freeden, M. Schreiner, Orthogonal and non-orthogonal multiresolution analysis, scale discrete and exact fully discrete wavelet transform on the sphere, Constr. Approx. 14 (1997) 493–515.

[22] Q.T.L. Gia, Approximation of linear partial differential equations on sphere, Ph.D. thesis, Texax A & M University, 2008.

[23] K. Goyal, M. Mehra, An adaptive meshfree diffusion wavelet method for partial differential equations on the sphere, J. Comput. Phys. 272 (2014) 747–771.

[24] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2011) 129–150.

[25] A. Harten, Adaptive multiresolution schemes for shock computations, J. Comput. Phys. 115 (2) (1994) 319–338.

[26] S.E. Hieber, P. Koumoutsakos, A Lagrangian particle level set method, J. Comput. Phys. 210 (2005) 342–367.

[27] S.L. Ho, S. Yang, H.C. Wong, G. Ni, A meshless collocation method based on radial basis functions and wavelet, IEEE Trans. Magn. 40 (2004) 1021–1024.

[28] M. Holmstrom, J. Walden, Adaptive wavelets methods for hyperbolic PDEs, J. Sci. Comput. 13 (1998) 19–49.

[29] L. Jameson, A wavelet-optimized, very high order adaptive grid and order numerical method, SIAM J. Sci. Comput. 19 (1998) 1980–2013.

[30] K. Jetter, J. Stockler, J.D. Ward, Error estimates for scattered data interpolation on sphere, Math. Comput. 68 (1999) 733–747.

[31] E.J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics–I surface approximations and partial derivative estimates, Comput. Math. Appl. 19 (8) (1990) 127–145.

[32] E.J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput. Math. Appl. 19 (8) (1990) 147–161.

[33] J. Liandrat, P. Tchamitchian, Resolution of the 1D regularized Burgers equation using a spatial wavelet approximation, NASA Contactor Report 187480: ICASE report No. 90-83, 1990.

[34] M. Mehra, N.K.-R. Kevlahan, An adaptive wavelet collocation method for the solution of partial differential equations on the sphere, J. Comput. Phys. 227 (2008) 5610–5632.

[35] M. Mehra, B.V.R. Kumar, Time-accurate solution of advection-diffusion problems by wavelet-Taylor–Galerkin method, Commun. Numer. Methods Eng. 21 (2005) 313–326.

[36] F. Narcowich, J.D. Ward, Scattered-data interpolation on spheres: error estimates and locally supported basis functions, SIAM J. Math. Anal. 33 (2002) 1393–1410.

[37] T. Plewa, T. Linde, V.G. Weirs, Adaptive Mesh Refinement–Theory and Applications, Springer, 2005.

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

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

[40] R. Schaback, H. Wendland, Characterization and construction of radial basis functions, Multivar. Approx. Appl. (2001) 1–24.

[41] K. Schneider, F. Chemie, T. Chemie, J. Fröhlich, J. Frohlich, An adaptive wavelet-vaguelette algorithm for the solution of PDEs, J. Comput. Phys. 130 (1997) 90–174.

[42] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. 237 (1952) 37–72.

[43] C. Varea, J.L. aragon, R. Barrio, Turing pattern on sphere, Phys. Rev. E 60 (1999) 4588–4592.

[44] O.V. Vasilyev, Solving multi-dimensional evolution problems with localized structures using second generation wavelets, Int. J. Comput. Fluid Dyn. 17 (2003) 151–168.

[45] O.V. Vasilyev, C. Bowman, Second generation wavelet collocation method for the solution of partial differential equations, Chin. J. Comput. Phys. 165 (2000) 660–693.

[46] S. Yang, G. Ni, J.R. Cardoso, S.L. Ho, J.M. Machado, A combined wavelet-element free Galerkin method for numerical calculations of electromagnetic fields, IEEE Trans. Magn. 39 (2003) 1413–1416.

[47] L. Zhang, J. Ouyang, X. Wang, X. Zhang, Variational multiscale element-free Galerkin's method for 2D Burger's equation, J. Comput. Phys. 229 (2010) 7147–7161.