Engineering Analysis with Boundary Elements 64 (2016) 290–294

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Fast method of approximate particular solutions using Chebyshev interpolation

A.R. Lamichhane [a], D.L. Young [b], C.S. Chen [a,c,*]

[a] Department of Mathematics, University of Southern Mississippi, USA
[b] Department of Civil Engineering, National Taiwan University, Taiwan
[c] College of Mathematics, Taiyuan University of Technology, China

ARTICLE INFO

Article history:
Received 30 October 2015
Received in revised form
20 December 2015
Accepted 21 December 2015
Available online 14 January 2016

Keywords:
Radial basis functions
Meshless method
Method of approximate particular solutions
Chebyshev interpolation
Fast summation method
Fast method of approximate particular solutions

ABSTRACT

The fast method of approximate particular solutions (FMAPS) is based on the global version of the method of approximate particular solutions (MAPS). In this method, given partial differential equations are discretized by the usual MAPS and the determination of the unknown coefficients is accelerated using a fast technique. Numerical results confirm the efficiency of the proposed technique for the PDEs with a large number of computational points in both two and three dimensions.

1. Introduction

During the past two decades, the radial basis functions (RBFs) have been widely applied for solving partial differential equations (PDEs). Kansa [13] first proposed the so-called RBF collocation method in 1990 for computationally solving fluid dynamic problems. One of the attractions of the Kansa method is its simplicity for solving problems in high dimensional and irregular domains. Due to the popularity of the Kansa method, several other RBF collocation methods have been proposed. Among them, the method of approximate particular solutions (MAPS) [5,6] is another effective method using the particular solution of the given RBFs as the basis function. The formulation of the MAPS produces a full and dense matrix system which is often very ill-conditioned. Traditionally, this matrix system is solved by using direct or iterative methods. Direct methods such as Gaussian elimination require $O(N^3)$ operations for an $N \times N$ system of equations. For iterative methods, we may obtain the approximate solution in k steps with each step needing a matrix vector multiplication $O(N^2)$.

In recent years, the MAPS has been widely applied to solve physical and engineering problems like Navier–Stokes equations [3], Stokes flow problems [4], Linear elasticity equations [2], Time-fractional diffusion equations [10], Inverse problem of non-homogeneous convection-diffusion equation [12], Diffusion equation with non-classical boundary [1] and Nonhomogeneous cauchy problems of elliptic PDEs [15]. Simulations of these kinds of problems involve a large number of interpolation points. The high computational cost using traditional solvers has become an issue. In this paper we pay special attention on how to develop a fast algorithm to alleviate the issue of high cost for solving large-scale problems using the MAPS. Consequently, we propose to couple the MAPS with a fast summation method to reduce the computational time by multiplying a matrix and a vector in each step inside the iterative method. This fast summation method is based on the Chebyshev interpolation [9]. As we shall see in the section of numerical results, we have successfully solved a 2D problem using 694,541 collocation points with only 77.15 s of computer running time and 343,000 collocation points with 105.15 s for a 3D problem. Moreover, we do not compromise the accuracy for our proposed fast computation.

The structure of the paper is as follows. In Section 2, we give a brief review of the MAPS and the closed-form particular solution of the Gaussian for Laplacian in 2D and 3D. In Section 3, we review the algorithm of fast summation method (FSM). In Section 4, we propose the fast method of approximate particular solutions (FMAPS) by coupling the FSM and the MAPS as a fast algorithm for solving PDEs which require a large number of collocation points. A

* Corresponding author.
E-mail address: cschen.math@gmail.com (C.S. Chen).

http://dx.doi.org/10.1016/j.enganabound.2015.12.015
0955-7997/

specific algorithm has been given. In Section 4.1, to demonstrate the efficiency of the proposed method, two numerical examples in 2D and 3D are given. Finally, a short conclusion is drawn in Section 4.1.1.

2. Method of approximate particular solutions (MAPS)

For simplicity, let us consider the following Poisson equation with Dirichlet boundary condition

$$\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{1}$$

$$u(\boldsymbol{x}) = g(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \tag{2}$$

where Δ is the Laplacian operator, Ω and $\partial\Omega$ are the interior and boundary of the computational domain, respectively. Suppose $\{\boldsymbol{x}_i\}_{i=1}^{n}$ are the interpolation points containing n_i interior points in Ω and n_b boundary points on $\partial\Omega$; i.e., $n = n_i + n_b$. Let ϕ be a given radial basis function. By the MAPS [6], we assume the solution to (1) and (2) can be approximated by

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \Phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|), \tag{3}$$

where $\|\cdot\|$ is the Euclidean norm, $\{\alpha_j\}$ are the undetermined coefficients, and

$$\Delta\Phi = \phi. \tag{4}$$

By the collocation method, from (1) and (2), we have

$$\sum_{j=1}^{n} \alpha_j \phi(\|\boldsymbol{x_i} - \boldsymbol{x_j}\|) = f(\boldsymbol{x_i}), \quad 1 \le i \le n_i, \tag{5}$$

$$\sum_{j=1}^{n} \alpha_j \Phi(\|\boldsymbol{x_i} - \boldsymbol{x}_j\|) = g(\boldsymbol{x_i}), \quad n_i + 1 \le i \le n. \tag{6}$$

From (5) and (6), we can formulate a linear system of equations

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{F} \tag{7}$$

where

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\phi} \\ \mathbf{\Phi} \end{bmatrix}$$

$\boldsymbol{\phi} = [\phi(\|\boldsymbol{x_i} - \boldsymbol{x_j}\|)]_{ij}, \quad \mathbf{\Phi} = [\Phi(\|\boldsymbol{x_k} - \boldsymbol{x_j}\|)]_{kj}, \quad 1 \le i \le n_i,\ 1 \le j \le n,$
$n_i + 1 \le k \le n,$
$\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \cdots\ \alpha_n]^T, \quad \mathbf{F} = [f(\boldsymbol{x}_1) \cdots f(\boldsymbol{x}_{n_i})\ g(\boldsymbol{x}_{n_i+1}) \cdots g(\boldsymbol{x}_n)]^T.$

For a more general form of the PDEs such as the following convection-diffusion-reaction equation in 2D,

$$k\Delta u + f(\boldsymbol{x})\frac{\partial u}{\partial x} + g(\boldsymbol{x})\frac{\partial u}{\partial y} + h(\boldsymbol{x})u = l(\boldsymbol{x}), \ \boldsymbol{x} = (x, y) \in \Omega, \tag{8}$$

we have [5]

$$\sum_{j=1}^{N} \alpha_j (k\phi(\|\boldsymbol{x} - \boldsymbol{x}_{\|}) + f(\boldsymbol{x})\Phi_x(\|\boldsymbol{x} - \boldsymbol{x}_j\|) + g(\boldsymbol{x})\Phi_y(\|\boldsymbol{x} - \boldsymbol{x}_j\|)$$
$$+ h(\boldsymbol{x})\Phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|)) = l(\boldsymbol{x}). \tag{9}$$

One of the key procedures in the implementation of the MAPS is to obtain the closed-form expression for the particular solutions of the corresponding RBFs. The derivation of the particular solutions for the well-known RBFs has already been known [7,16,18–20]. Recently, the closed-form particular solutions of the Gaussian have been derived in [14], which are stated as follows:

Theorem 1. *Let* $\phi(r) = \exp(-cr^2)$, $c > 0$, *and* $\Delta\Phi(r) = \phi(r)$ *in 2D. Then,*

$$\Phi(r) = \begin{cases} \dfrac{1}{4c}\mathrm{Ei}(cr^2) + \dfrac{1}{2c}\log(r), & r \ne 0, \\ \dfrac{-1}{4c}(\gamma + \log(c)), & r = 0, \end{cases} \tag{10}$$

where

$$\mathrm{Ei}(x) = \int_x^{\infty} \frac{e^{-u}}{u} du, \tag{11}$$

and $\gamma \simeq 0.5772156649015328$ *is the Euler–Mascheroni constant* [11]. *Note that* Ei(x) *is the special function known as the exponential integral function* [11].

Theorem 2. *Let* $\phi(r) = \exp(-cr^2)$, *and* $\Delta\Phi(r) = \phi(r)$ *in 3D. Then,*

$$\Phi(r) = \begin{cases} \dfrac{-\sqrt{\pi}}{4c^{3/2}r}\ \mathrm{erf}(\sqrt{c}r), & r \ne 0, \\ \dfrac{-1}{2c}, & r = 0, \end{cases} \tag{12}$$

where erf(x) *is the special function defined as follows* [11]:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2}\, du. \tag{13}$$

In this paper, we adopt the Gaussian, ϕ, as the RBF basis function and the corresponding particular solutions, Φ, as the basis functions for the approximation of the partial differential equation. The particular solutions of the Gaussian in (10) and (12) contain the special functions, Ei(x) and erf(x), which is costly in terms of numerical evaluation. The efficiency can be significantly improved using compiled MATLAB MEX functions.

In the next section, we briefly introduce a fast summation method for the matrix vector multiplication.

3. Fast summation method (FSM)

Consider the evaluation of the sum of the form

$$s(\boldsymbol{x}) = \sum_{j=1}^{n} b_j \kappa(||\boldsymbol{x} - \boldsymbol{x}_j||), \tag{14}$$

where κ is either the RBFs or the particular solution of the corresponding RBFs. We can evaluate the sum (14) in an efficient way by using the Chebyshev interpolation technique as described in [9].

From [9], let

$$P_M(\xi, \eta) = \frac{1}{M} + \frac{2}{M} \sum_{i=1}^{M-1} T_i(\xi) T_i(\eta), \tag{15}$$

where ξ, $\eta \in [-1, 1]$, T_i is the first kind Chebyshev polynomial of order i.

Let H be a hypercube in D dimension which contains all the given collocation points $\{\boldsymbol{x}_i\}_{i=1}^{n}$. Consider $\{\boldsymbol{\xi}_l\}_{l=1}^{M^D}$, $\{\boldsymbol{\eta}_l\}_{l=1}^{M^D}$ be two identical sets of Chebyshev nodes in $[-1, 1]^D$. Then by using linear transformations, we can map $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ into $\boldsymbol{\xi}_i$, $\boldsymbol{\eta}_j$, and $\boldsymbol{\xi}_l$, $\boldsymbol{\eta}_m$ in $[-1, 1]^D$ into $\boldsymbol{x}_l$, $\boldsymbol{x}_m$ in H, respectively.

Instead of directly evaluating (14), we approximate it by using the following functional approximation

$$\kappa(||\boldsymbol{x} - \boldsymbol{y}||) = \sum_l \sum_m k(||\boldsymbol{x}_l - \boldsymbol{y}_m||) Q_M(\boldsymbol{\xi}_l, \boldsymbol{\xi}) Q_M(\boldsymbol{\eta}_m, \boldsymbol{\eta}), \tag{16}$$

where $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ are the linear transformations of $\boldsymbol{x}, \boldsymbol{y}$, respectively, and