Engineering Analysis with Boundary Elements 50 (2015) 291–303

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

Numerical solution of stochastic elliptic partial differential equations using the meshless method of radial basis functions

Mehdi Dehghan *, Mohammad Shirzadi

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

ARTICLE INFO

Article history:
Received 22 June 2014
Received in revised form
14 August 2014
Accepted 15 August 2014
Available online 10 October 2014

Keywords:
Meshless method
Radial basis functions
Pseudospectral methods
Stochastic partial differential equations
Stochastic Laplace equation
Stochastic biharmonic equation

ABSTRACT

In this paper, we propose two numerical methods to solve the elliptic stochastic partial differential equations (SPDEs) in two and three dimensions obtained by Gaussian noises using radial basis functions (RBFs) collocation and pseudospectral (PS) collocation methods. For approximating the solution, generalized inverse multiquadrics (GIMQ) RBFs have been used. The Gaussian noises are approximated at the collocation points. The schemes work in a similar fashion as Hermite-based interpolation methods. The methods are tested via several problems. The numerical results show usefulness and accuracy of the new methods.

1. Introduction

Many natural phenomena and physical applications are modeled by partial differential equations (PDEs). Actually many phenomena, both in nature and engineering, which are described by systems of deterministic PDEs, may be more fully modeled by systems of stochastic PDEs (SPDEs) instead, for instance see [1,3]. Hence, the extensive applications of random effects in describing practical sciences show the importance of the theory of SPDEs. The theoretical foundation of SPDEs and their applications are described in [15].

1.1. Radial basis functions

To avoid mesh generation, in recent years meshless techniques have attracted the attention of researchers. In a meshless method a set of scattered nodes is used instead of meshing the domain of the problem. Some meshless methods are the smooth-particle hydrodynamics, the element–free Galerkin method, kernel methods, moving least squares(MLS) method [38,58], meshless local Petrov–Galerkin (MLPG) methods [39,59], partition of unity method, radial point interpolation method (RPIM) [22] and the method of radial basis functions (RBFs) [23-25]. Each technique has its advantages for specific classes of problems. For some research works in these subjects we referee the interested reader to [4,22,38,39,45–48,51,52]. For more descriptions of meshless methods see [42,56].

The technique of RBFs is one of the most recently developed meshless methods that has attracted attention of many researchers in recent years, especially in the area of computational mechanics [34,35]. This method does not require mesh generation which makes them advantageous for three-dimensional (3D) problems as well as problems that require frequent re-meshing such as those arising in nonlinear analysis. Due to its simplicity to implement, it represents an attractive alternative to the finite element and finite difference methods [2,9,10,57], the boundary elements method [12,13,41], the Wiener chaos expansion as a numerical solution method of SPDEs [30]. One of the domain-type meshless methods, the so-called Kansa's method developed by Kansa in 1990 [34,35], is obtained by directly collocating the RBFs, particularly the multiquadric (MQ), for the numerical approximation of the solution. Kansa's method was recently extended to solve various ordinary and partial differential equations, for instance see [18–21,31] and the references therein. Fasshauer [27] later modified Kansa's method to a Hermite type collocation method for the solvability of the resultant collocation matrix. For some research works based on Hermite collocation method we refer the interested reader to [53,54] and the references therein.

Most available RBFs methods involve a parameter, called shape factor (such as multiquadric, inverse multiquadric, and Gaussian)

* Corresponding author.
E-mail addresses: mdehghan.aut@gmail.com, mdehghan@aut.ac.ir (M. Dehghan), m.shirzadi@aut.ac.ir (M. Shirzadi).

http://dx.doi.org/10.1016/j.enganabound.2014.08.013
0955-7997/

which needs to be selected so that the required accuracy of the solution is attained. The traditional RBFs are globally defined functions which result in a full coefficient matrix. This hinders the application of the RBFs to solve large-scale problems due to severe ill-conditioning of the coefficient matrix. For some useful papers on error estimate, condition number of interpolation matrix and investigation on the shape parameter we refer the interested readers to [11,14,40]. Some efforts have been made in order to overcome the issues of the full and ill-conditioning of the coefficient matrix. For example, various methods employing local radial basis functions have been proposed [49,50,26]. Further, the algorithm in [5] allows one to avoid large matrices thanks to a suitable combination of operator splitting and exponential radial basis functions.

RBFs [23-25] are widely used for solving problems arising in financial mathematic, for instance see [5-7]. However, it is only since rather recently that the meshless method of RBFs has been used to approximate solutions for SPDEs and therefore this area is still relatively unexplored. RBFs have been applied for the numerical solution of time-dependent SPDEs [16,44] and for time-independent SPDE [29]. The theoretical framework of meshfree approximation method for the numerical solution of SPDEs with

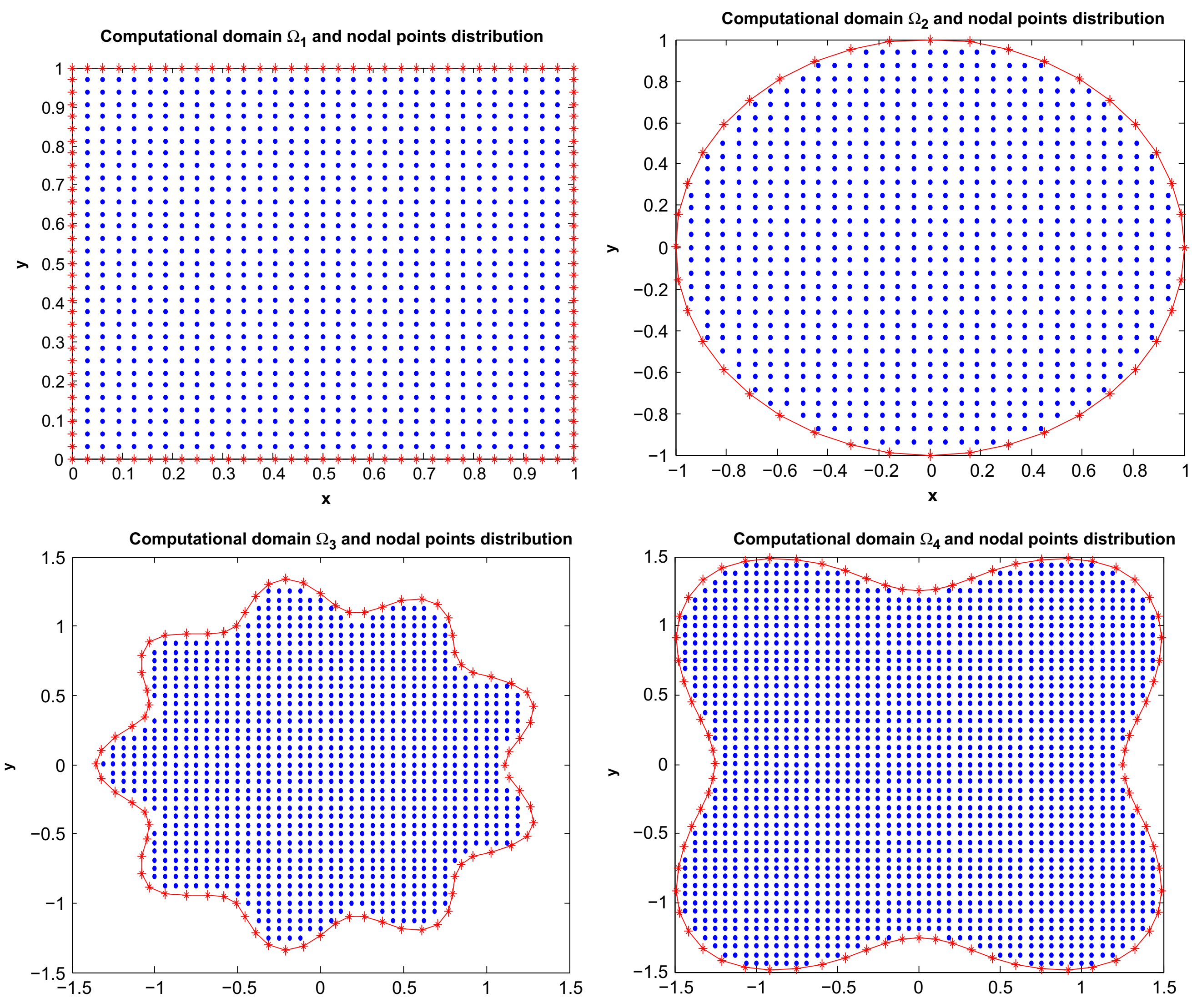

Fig. 1. Graphs of computational domains Ω_1, Ω_2, Ω_3, Ω_4 and related nodal distributions.

Table 1
Errors estimate values using two methods for Example 1 on Ω_1.

s	$h(N+M)$	$RBF-collocation$		$PS-collocation$	
		L_∞	RMS	L_∞	RMS
50	0.25(25)	9.5652×10^{-1}	6.3723×10^{-1}	9.7252×10^{-1}	6.4814×10^{-1}
200	0.125(81)	6.7839×10^{-1}	3.9176×10^{-1}	8.4399×10^{-1}	5.3925×10^{-1}
800	0.0625(289)	8.4950×10^{-3}	6.2441×10^{-3}	2.9831×10^{-2}	1.9824×10^{-2}
3200	0.03125(1089)	5.5562×10^{-4}	3.9759×10^{-4}	1.4685×10^{-3}	8.4216×10^{-4}