Journal of Sound and Vibration 327 (2009) 144–162

Contents lists available at ScienceDirect

Journal of Sound and Vibration

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

The partition of unity finite element approach with hp-refinement for the stationary Fokker–Planck equation

Mrinal Kumar [a,*], Suman Chakravorty [a], Puneet Singla [b], John L. Junkins [a]

[a] Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843, USA
[b] Department of Mechanical and Aerospace Engineering, The State University of New York, Buffalo, 14260, USA

ARTICLE INFO

Article history:
Received 31 December 2008
Received in revised form
20 May 2009
Accepted 26 May 2009
Handling Editor: M.P. Cartmell
Available online 9 July 2009

ABSTRACT

In this paper, the stationary Fokker–Planck equation (FPE) is solved for nonlinear dynamical systems using a local numerical technique based on the meshless partition of unity finite element method (PUFEM). The method is applied to stationary FPE for 2-, 3- and 4-D systems and is argued to be an excellent candidate for higher dimensional problems and the transient problem. Local refinement is applied by introducing higher order polynomials in selected subdomains (local *p*-refinement) to keep the problem size small while ensuring high approximation accuracy. Various local approximations are blended using novel pasting functions that provide any desired order of continuity. Results are compared with existing global and local techniques. Local *p*-refinement is touted as an important step towards breaking the curse of dimensionality in numerical solution of FPE.

1. Introduction

Numerous fields of science and engineering present the problem of uncertainty propagation through nonlinear dynamical systems with stochastic excitation and uncertain initial conditions [1,2]. One may be interested in the determination of response of engineering structures like trusses under random excitation (in structural mechanics [3]), propagation of initial condition uncertainty of an asteroid for determination of its probability of collision with a planet (in astrodynamics [4]), motion of particles under the influence of stochastic force fields (in particle physics [5]), or simply computation of uncertainty in the prediction step in the design of a Bayes filter (in filtering theory [6]). All these applications require study of time evolution of the probability density function (pdf) $\mathscr{W}(t,\mathbf{x})$, corresponding to the state $\mathbf{x}$ of the underlying stochastic dynamic system.

While being of such great relevance, the problem of uncertainty propagation is also a very difficult one to solve in the exact sense, i.e. to solve for the exact time varying pdf. Therefore, several techniques have been developed that approximate the actual pdf with a finite number of parameters, e.g. its first N moments. The most popular among these techniques are Monte Carlo methods [7,8], Gaussian closure (or higher order moment closure) [9–12], equivalent linearization and stochastic averaging [10,13]. The Monte Carlo method essentially involves sampling of the underlying probability space to generate a family of test points, which are individually propagated forward through the exact nonlinear system. The pdf at any time step is then approximated by evaluating desired number of moments from the distribution of propagated sample

* Corresponding author. Tel.: +1 979 204 3815.
E-mail addresses: mrinal@neo.tamu.edu (M. Kumar), schakrav@aeromail.tamu.edu (S. Chakravorty), psingla@buffalo.edu (P. Singla), junkins@tamu.edu (J.L. Junkins).
URL: http://people.tamu.edu/~mrinal (M. Kumar).

0022-460X/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2009.05.033

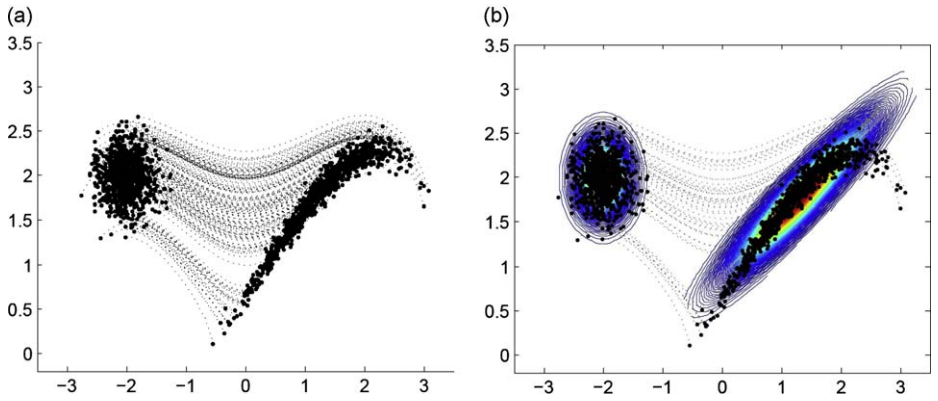

Fig. 1. The problem of uncertainty propagation. The x- and y-axes depict states of the dynamical system (e.g. x, $\dot{x}$). (a) The state-pdf. (b) Approximate characterization of the exact pdf.

points (see Fig. 1). This method generally requires extensive computational resources and effort, and becomes increasingly infeasible for dynamical systems with high-dimensional state-space and for long-term simulations. All the latter methods are similar to one another at some level and involve linearization (or a higher order approximation) of the underlying dynamical system. Consequently, they are suitable only for linear or moderately nonlinear systems, because the effect of neglected higher order terms can lead to significant errors. Furthermore, all these approaches provide only an approximate description of the uncertainty propagation problem by restricting the solution to a small number of parameters, for example the first N moments of the sought pdf.

The exact description of uncertainty propagation under white-noise excitation is provided by the well-known Fokker–Planck–Kolmogorov equation (FPKE), or simply the Fokker–Planck equation (FPE) [2]. The solution to FPE contains the complete information about the state pdf, $\mathscr{W}$. With reference to Fig. 1, this implies that FPE captures the exact shape of the uncertainty cloud rather than providing a fit that is correct to some order of approximation (e.g. Gaussian, as shown in Fig. 1(b)). Unfortunately, FPE is very difficult to solve for the exact pdf. Analytical solutions are known to exist only for linear systems and a particular class of nonlinear systems with a Hamiltonian like function [2]. This class of systems fails to represent a noteworthy percentage of the multifarious nonlinear systems encountered in science and engineering. Therefore, numerical techniques are required to approximately solve FPE for general N-D nonlinear systems. In the pre and early-computing era, several numerical and semi-numerical approaches were developed—e.g. eigenfunction expansion techniques [14] and perturbation methods [15] which produced numerical results for moderately nonlinear systems residing in 1- or 2-D state-space. With increasing computational capability, more sophisticated algorithms like global Galerkin method (GM) [16–18], finite differences [19], finite elements [20,21], and multi-scale finite elements [22] have been developed and used for this problem. Of particular note is the work of Wojtkiewicz et al. [23,24], wherein traditional FEM was utilized to solve the FPE for simple 3- and 4-D systems with success. However, due to inherent difficulties associated with FEM/FD when faced with dimensionality issues (e.g. discretization of a high-dimensional state-space), their application to complicated high-dimensional systems has been restricted. Recently, a multi-resolution meshless method (MRMM) [25] (based on the Meshless Petrov–Galerkin (MLPG) method [26]) was implemented to solve the stationary FPE and shown to provide immense benefits over existing global and local methods. By virtue of its meshless nature, the implementation of this algorithm on higher dimensional problems is straightforward, which is an advancement over the conventional mesh-based FEM. However, the associated time of execution and memory requirement of the algorithm for suchproblems is significantly greater than can be provided by a workstation, and convergence remains a challenge.

In this paper, a variation of a different meshless method, namely the partition of unity finite element method (PUFEM) [27] is employed to tackle the issue of curse of dimensionality in the numerical solution of FPE. The current method retains the extremely attractive meshless nature, thus facilitating discretization of high-dimensional state-space into subdomains. The approach allows for independent selection of basis functions in individual subdomains (local p-refinement), thus providing flexibility in construction of the approximation space. Local p-refinement is also the key to reducing the size of discretized problem because it allows for introduction of higher order polynomials only in selected subdomains where they are required. Such local enrichment of the approximation space curbs the curse of dimensionality because problem size no longer grows purely as an exponential function of dimensionality of the system. The benefit of local p-refinement is clearly demonstrated in this paper by means of several examples. Various local approximations are blended together using pasting functions (also known as weight functions) that satisfy partition of unity (PU) over the solution domain, leading to unbiased blending. Tent-functions have typically been used for this purpose, providing C^0 continuity across local subdomains. In the current paper, recently developed GLO-MAP functions [28,29] are used as weights. These functions provide blending of any desired order of continuity and are of polynomial form, which also simplifies the process of numerical integration. At this point, it is emphasized that the current paper is not an exposition on meshless techniques,

but presents an advanced approach to solve high-dimensional FPEs with relatively small computer resources. The most important practical utility of this work is the development of a technique that curbs the curse of dimensionality in FPE, thus making it solvable for high-dimensional systems on a small computer. The authors have further extended the current method, combining it with modal analysis to develop a near real-time solution to the transient FPE for nonlinear dynamics in another paper [30].

The remainder of the current paper is organized as follows: Section 2 introduces FPE and discusses the key issues surrounding its numerical solution. Section 3 lays out various aspects of the PUFEM approach in detail and the GLO-MAP weight functions mentioned above. In Section 4 numerical examples for systems with state-space in 2, 3 and 4 dimensions are presented. Finally, Sections 5 and 6 compare the current PUFEM technique against other popular methods for numerically solving FPE and draw conclusions, respectively.

2. The Fokker–Planck equation

As previously mentioned, Fokker–Planck equation provides the exact description of the uncertainty propagation problem for dynamical systems driven by white-noise excitation. In essence, FPE is a scalar partial differential equation of parabolic type, which captures the time evolution of the actual state-pdf of a given stochastic system. The exact solution to this equation contains complete information about the probability distribution of the considered system's state-space at all times.

Consider a general N-D white-noise-driven nonlinear dynamic system with uncertain initial conditions, given by the following equation:

$$\dot{\mathbf{x}} = \mathbf{f}(t,\mathbf{x}) + \mathbf{g}(t,\mathbf{x})\Gamma(t), \quad \mathrm{E}[\mathbf{x}(t_0)] = \bar{\mathbf{x}}_0 \tag{1}$$

where $\Gamma(t)$ represents an M-D Gaussian white-noise process with the correlation function $\mathbf{Q}\delta(t_1 - t_2)$, and $\bar{\mathbf{x}}_0$ represents the mean initial state. Vector functions $\mathbf{f}(t,\mathbf{x}) : [0,\infty) \times \Re^N \to \Re^N$ and $\mathbf{g}(t,\mathbf{x}) : [0,\infty) \times \Re^N \to \Re^{N\times M}$ are measurable functions. Initial probability distribution of the state is given by the pdf $\mathscr{W}(t_0,\mathbf{x})$, which captures state uncertainty at time t_0.

Then, the time evolution of $\mathscr{W}(t_0,\mathbf{x})$ is described by the following FPE, which is a second order, linear parabolic PDE in $\mathscr{W}(t,\mathbf{x})$:

$$\frac{\partial}{\partial t}\mathscr{W}(t,\mathbf{x}) = \mathscr{L}_{\mathscr{FP}}[\mathscr{W}(t,\mathbf{x})] \tag{2}$$

where

$$\mathscr{L}_{\mathscr{FP}} = \left[-\sum_{i=1}^{N} \frac{\partial}{\partial x_i} \mathrm{D}_i^{(1)}(\cdot) + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2}{\partial x_i \partial x_j} \mathrm{D}_{ij}^{(2)}(\cdot) \right] \tag{3}$$

$$\mathrm{D}^{(1)}(t,\mathbf{x}) = \mathbf{f}(t,\mathbf{x}) + \frac{1}{2}\frac{\partial \mathbf{g}(t,\mathbf{x})}{\partial \mathbf{x}} Q \mathbf{g}(t,\mathbf{x}) \tag{4}$$

$$\mathrm{D}^{(2)}(t,\mathbf{x}) = \tfrac{1}{2}\mathbf{g}(t,\mathbf{x}) Q \mathbf{g}^{\mathrm{T}}(t,\mathbf{x}) \tag{5}$$

where $\mathscr{L}_{\mathscr{FP}}$ is the Fokker–Planck operator, $\mathrm{D}^{(1)}$ is known as the drift coefficient vector and $\mathrm{D}^{(2)}$ is the diffusion coefficient matrix. The drift vector captures drifting apart of the mean of the propagated pdf from the propagated mean of the initial pdf. Generally, this drift increases with degree of nonlinearity of underlying dynamics, i.e. $\mathbf{f}(t,\mathbf{x})$. The diffusion matrix captures spreading out of the substantial portion of the pdf (e.g. the 3σ region) over state-space. In simple terms, it governs how flat (or diffuse) the pdf turns out to be. In case the underlying governing dynamics (Eq. (1)) is deterministic, i.e. $\mathbf{g}(t,\mathbf{x}) = \mathbf{0}$ and the source of uncertainty lies only in initial state, the diffusion matrix is identically zero and the reduced FPE is called the Liouville equation. An example of such a problem is the forward propagation of initial state uncertainty of an asteroid through underlying $2/N$-body equations of motion, for purpose of prediction of its probability of collision with a planet.

We mention here that Eq. (4) represents the Stratonovich form of the drift vector. There exists another form known as the Itô form, which is generally different from the Stratonovich form and is considered by mathematicians to be the rigorously correct expression for $\mathrm{D}^{(1)}$. In engineering fields, however, the Stratonovich form is preferred since it precludes the need for Itô calculus which is required to deal with the Itô form. The two forms are identical in case of state additive noise, i.e. when $\mathbf{g}(t,\mathbf{x}) = \mathbf{g}(t)$. This is typically the case with most real life stochastic systems and we consider only such systems in this paper.

2.1. Stationary FPE

In the current paper, we are concerned only with steady-state solution of Eq. (2), i.e. the following equation:

$$\mathscr{L}_{\mathscr{F}\mathscr{P}}[\mathscr{W}(\mathbf{x})] = 0 \tag{6}$$

In other words, we desire to extract the null-space of the Fokker–Planck operator, which governs the long-term behavior of stochastic dynamic systems. A methodology for the transient problem using the current approach, coupled with modal analysis has been discussed in another paper [30]. Conditions for the existence of a unique null-space of the Fokker–Planck operator are well known and can be found in Fuller [1]. A necessary condition is time invariance of system dynamics, i.e. $\mathbf{f}(t,\mathbf{x}) = \mathbf{f}(\mathbf{x})$ and $\mathbf{g}(t,\mathbf{x}) = \mathbf{g}(\mathbf{x})$. Since we are currently interested only in state-additive noise, the function $\mathbf{g}(t,\mathbf{x})$ reduces to a constant. Other necessary conditions for existence of a nontrivial and unique stationary solution of FPE are the existence of finite intensity noise and at least one attractor in the system. More details can be found in Fuller [1].

2.2. FPE: difficulties

Despite its innocuous appearance, solving FPE for the pdf is a formidable problem because of the following difficult issues:

1. Positivity of the *pdf*: $\mathscr{W}(t,\mathbf{x}) \geq 0,\ \forall t$ and $\mathbf{x}$.
2. Normalization constraint of the *pdf*: $\int_{-\infty}^{\infty} \mathscr{W}(t,\mathbf{x})\,\mathrm{d}V = 1$.
3. No fixed solution domain for numerical implementation: how to impose boundary conditions in a finite region and restrict numerical computation to regions where $\mathscr{W} \gtrsim 10^{-9}$.
4. Curse of dimensionality: Exponential increase in size of the discretized problem with dimensionality of state-space.

(1) and (2) represent additional constraints that the solution obtained for Eq. (2) must satisfy in order to be a valid pdf. Therefore these constraints must be accommodated in the numerical method. While (2) can be enforced by a simple post-processing step of re-normalization of the obtained solution, (1) is a tough proposition and is primarily relevant in the tail regions of the pdf. Several researchers have used a log-transformation of FPE to ensure positivity (the inverse exponential transform of the solution obtained ensures positive values) [16,17]. However, this approach converts the linear PDE (Eq. (2)) into a nonlinear PDE, which is generally not desirable. The issue of solution domain (3) is also difficult to resolve and heuristic methods are typically used to define a conservatively sized domain for numerical implementation. Determining such a domain of appropriate size, shape and orientation is a challenging task because the theoretical domain of an N-state pdf is $\Re^N$. Thus artificial boundary conditions need to be enforced on a "large enough solution domain," e.g. $\mathscr{W}(t,\Gamma) = 10^{-9}$, where Γ is the boundary of the chosen domain. The current authors have utilized homotopic principles to deduce the location and size of the domain for numerical computation [31].

The stationary FPE is a PDE involving partial derivatives of all components of state-space, and the "curse of dimensionality" (issue 4) remains the greatest challenge confronting its successful numerical solution. Curse of dimensionality can be roughly defined as an exponential growth in the size of a problem with dimensionality of the system under consideration. For example, consider a complete set of polynomial basis functions including up to pth-order polynomials used in a global approximation scheme to estimate an unknown function (the probability density function in this case). The size of such a basis set increases as $\sum_{k=0}^{p} \binom{N+k-1}{N-1}$, which exhibits explosive growth with N. Next consider a local approximation scheme like the finite element method, in which a grid is constructed with n nodes along each of N dimensions and every individual node is endowed with a complete set of pth-order polynomials. In this situation, the total number of basis functions (or, "degrees of freedom" (dofs) of the approximation) is given by the following expression (depicted by ${}^{N}\mathscr{D}_{p}^{n}$):

$${}^{N}\mathscr{D}_{p}^{n} = n^N \times \sum_{k=0}^{p} \binom{N+k-1}{N-1} \tag{7}$$

Note that increasing the number of nodes (n), or, the order of basis functions per node (p) both cause an explosive growth in problem size, ${}^{N}\mathscr{D}_{p}^{d}$, especially for large N. This behavior is discussed in greater detail in Section 3.5 (e.g. see Fig. 5(a)). In the local meshless method utilized in the current paper (PUFEM), nodes used for discretization are required to be placed as if they lie on a grid. Therefore, the nodal discretization continues to grow exponentially as per Eq. (7) ($\sim n^N$). Hence, in the absence of local p-refinement, the current approach would also suffer from the curse of dimensionality. However, local p-refinement (local p-enrichment) allows us to increase the order of basis functions at only selected nodes. In this scenario, we count basis functions for each node individually leading to an expression for the total number of degrees of freedom that is different from Eq. (7):

$${}^{N}\mathscr{D}_{\mathrm{loc}_p}^{n} = \sum_{i=1}^{n^N} \sum_{k=0}^{p_i} \binom{N+k-1}{N-1} \tag{8}$$

Eqs. (7) and (8) are exactly the same when each node has same order of polynomial basis, i.e. $p_i = p$. Therefore, local p-refinement does not alleviate the curse of dimensionality directly since the number of nodes still grows exponentially with N and there is no immediate reason for preferring Eq. (8) over Eq. (7). The key, however, lies in that local p-refinement permits only selected nodes to carry high order polynomials, thus significantly cutting down on dofs where they are not necessarily required (e.g. in the tail regions of the pdf where constant basis functions are sufficient). This in turn invariably leads to a significant reduction in the number of nodes actually required because all existing nodes are endowed with polynomials that provide optimal local approximability (ability to approximate locally), in turn leading to a significant reduction in the number of degrees of freedom. Although the curse of dimensionality is not broken rigorously, it is thus indirectly significantly curtailed. These issues are discussed in detail in Section 3.5. In the following section the meshless partition of unity finite element method used in this paper to solve the FPE for N-D dynamic systems is described in detail.

3. The partition of unity finite element method

The partition of unity approach to finite elements was developed by Babuška and Melenk [27]. It is one of the several recently developed meshless algorithms [26,27,32–35] for solving PDEs in complicated high-dimensional domains. In this section, we describe briefly the various aspects of the method—domain discretization, shape function construction and development of the local weak form equations.

3.1. PUFEM discretization

Consider a domain Ω and a set of overlapping subdomains Ω_s, $s = 1, 2, \ldots, P$, which form a cover for Ω. A "partition of unity" on Ω is a mathematical paradigm in which each of the overlapping subdomains Ω_s is associated with a compactly supported function $\varphi_s(\mathbf{x})$ called the PU pasting function (also, PU weight function), which is strictly zero outside Ω_s and has the property that

$$\sum_s \varphi_s(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \Omega \tag{9}$$

The above paradigm represents the skeleton for a powerful meshless finite element method for solving PDEs on the domain Ω. By assigning each of the subdomains (Ω_s) to individual nodes distributed over the global domain Ω, we obtain an implicit "discretization" of Ω (see Fig. 2 for an example in 1 dimension), using which a local variational form of the PDE to be solved can be formulated. The discretization is not to be understood in the usual sense because of two primary reasons: (a) overlap among neighboring subdomains and (b) minimal role of inter-node or inter-element connectivity. By virtue of the latter property, this discretization is "meshless" and has immense advantage in application to high-dimensional PDEs, e.g. the FPE for the 3-D motion of an asteroid, which involves a 6-D state-space discretization. Also, the absence of inter-element boundaries is a great convenience in problems with moving boundaries, since no re-meshing is required to maintain inter-element boundary information.

Fig. 2 illustrates the implicit PUFEM discretization described above for the 1-D domain $[-1, 1]$ using five nodes. The "tent functions" shown in Fig. 2(a) are traditionally used as pasting functions ($\varphi_s(\mathbf{x})$) in the meshless FEM literature. They satisfy the PU property (Eq. (9)) and provide C^0 blending of local approximations. Note that PU weights have value one at their parent node and decay to zero at all neighboring nodes. PU functions and their supports for the first two nodes have been highlighted, and the overlap between elements Ω_1 and Ω_2 is clearly visible. In the current paper, we use weight functions developed recently in the GLO-MAP algorithm [28] (see Fig. 2(b)), which are essentially smoother piecewise polynomial generalizations of the tent-functions and offer several advantages as PU weights. They can be designed to provide any

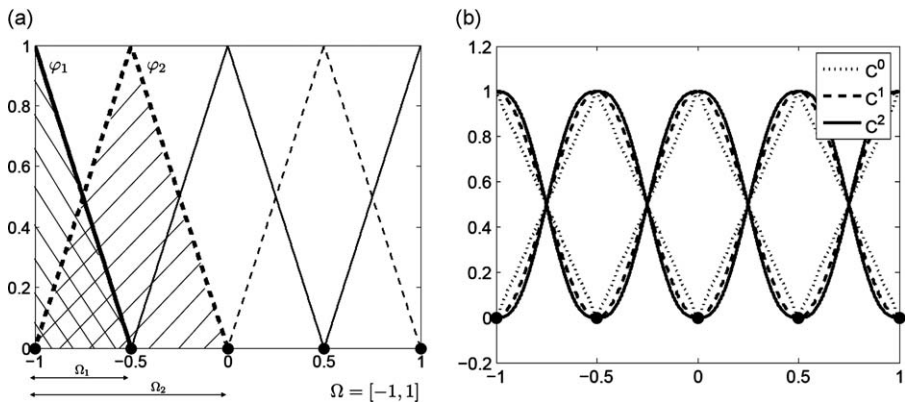

Fig. 2. PUFEM domain discretization. x-axis: x; y-axis: function evaluated at x. (a) The PUFEM domain discretization in 1-D with *tent* functions. (b) Polynomial GLO-MAP weights up to C^2 smoothness in 1-D.

desired order of smoothness in high-dimensional state-space. GLO-MAP functions up to C^2 smoothness in 1 dimension are shown in Fig. 2(b) and a detailed discussion is provided in Section 3.4.

3.2. PUFEM shape functions

Using the above described collection of node-centered, compactly supported subdomains forming a cover for the solution domain, a finite element approximation $\widehat{\mathscr{W}}$ of the function $\mathscr{W}$ can be constructed by setting up a conformal space of shape functions in Ω. This is accomplished by introducing basis functions—$\psi_{sj}(\mathbf{x})$ $(j=1,\ldots,Q_s)$, within each Ω_s. Note that in this paper, we differentiate between the terms "basis functions" and "shape functions". Shape functions are the final form of the functions used in the approximation space and are constructed out of the basis functions in different ways, depending on the particular meshless method. In PUFEM, the basis functions $\psi_{sj}(\mathbf{x})$ may be chosen from the space of polynomials or they may be "special functions", based on prior knowledge about the problem. For example, if the solution of the PDE in question is known to be highly oscillatory, harmonic functions with frequencies close to those of the system can be used in the basis set. Such special functions may be introduced either by themselves or to supplement a previously existing polynomial basis. This aspect, called "basis enrichment", is one of the greatest advantages of PUFEM because it allows the use of local functions of varying form and number in the individual subdomains. While such freedom provides great flexibility and can immensely improve the approximability, it generally prevents the basis functions from constituting a conformal space; i.e. the inter-element continuity of the approximation is not ensured. This task is accomplished by the PU pasting functions, which merge together the various local approximations. Fig. 3 illustrates the process of shape function construction in the PUFEM algorithm. In these figures, the basis functions (ψ_{sj}) have been drawn using bold lines and the PU pasting functions (φ_s) using light lines. Also, all functions corresponding to the odd-numbered nodes are drawn with solid lines and those corresponding to even-numbered nodes with dashed lines. The 1-D domain $[-1,1]$ is discretized using five subdomains with tent-functions in Fig. 3(a) and C^1 GLO-MAP weights in Fig. 3(b). The use of quadratic polynomials as basis functions has been shown in all the subdomains. Additionally, a sinusoidal function (which enriches the existing polynomial basis) has been introduced locally only in the third subdomain (corresponding to the highlighted node #3). Clearly, these basis functions do not form aconformal space on their own. However, when these functions are multiplied with the PU pasting functions of the corresponding nodes, the resulting functions satisfy inter-element continuity and we refer to the product as pasted basis functions, or shape functions (see Figs. 3(c) and (d)):

$$\Psi_{sj}(\mathbf{x}) = \varphi_s(\mathbf{x})\psi_{sj}(\mathbf{x}), \quad j=1,\ldots,Q_s \tag{10}$$

In essence, the above approach delegates the burden of enforcing inter-element continuity to the pasting functions so that the user is free to select basis functions purely on the criteria of local approximability. In comparison, in the conventional FEM, basis functions are the same as the shape functions. Therefore they need to form a conformal space on their own, which limits their range of selection. In most other meshless methods like SPH, MLPG, etc., shape functions are constructed using data fitting algorithms like the moving least squares (MLS), using a pool of basis functions which may be nonpolynomials. The order of continuity of shape functions across local subdomains is inherited from the continuity of PU pasting functions [27]. Thus, tent-functions, which are C^0 continuous, lead to shape functions whose derivatives are not continuous (Fig. 3(e)). On the other hand, the use of GLO-MAP weight with C^1 continuity leads to C^0 derivatives (Fig. 3(f)), which is an advantage (see Section 3.4).

3.3. Local variational formulation of the stationary FPE

Given the shape functions described above, the global approximation is written by putting together the various local approximations as follows:

$$\widehat{\mathscr{W}}(\mathbf{x}) = \sum_{s,j} a_{sj}\Psi_{sj}(\mathbf{x}) = \sum_{s=1}^{P}\varphi_s(\mathbf{x})\sum_{j=1}^{Q_s} a_{sj}\psi_{sj}(\mathbf{x}) \tag{11}$$

where a_{sj} are Fourier coefficients (to be determined) corresponding to the various basis functions denoted by $\psi_{sj}(.)$. The role of the PU functions is further clarified by the above equation. Besides providing compact support and enforcing inter-element continuity, they act as averaging weights, merging together the approximations from neighboring subdomains. By virtue of the PU property (Eq. (9)), these weights add to 1 over the entire domain, hence leading to an unbiased average.

Using the PUFEM approximation $\widehat{\mathscr{W}}$ given in Eq. (11), we obtain the following local variational form for the N-D stationary FPE over the subdomain Ω_s with boundary Γ_s:

$$\int_{\Omega_s} \mathscr{L}_{FP}[\widehat{\mathscr{W}}(\mathbf{x})]v(\mathbf{x})\,\mathrm{d}\Omega + \alpha\int_{\Gamma_s\cap\Gamma}\widehat{\mathscr{W}}(\mathbf{x})v(\mathbf{x})\,\mathrm{d}\Gamma = 0 \tag{12}$$

where $v(\mathbf{x})$ is a test function belonging to the space of functions $\mathscr{V}$, onto which the residual is projected for error minimization, and α is a penalty parameter used to enforce the boundary condition $\mathscr{W}_\Gamma = 0$. The test functions in the above variational equation have compact support on Ω_s (note that the domain of integration in Eq. (12)). Substituting

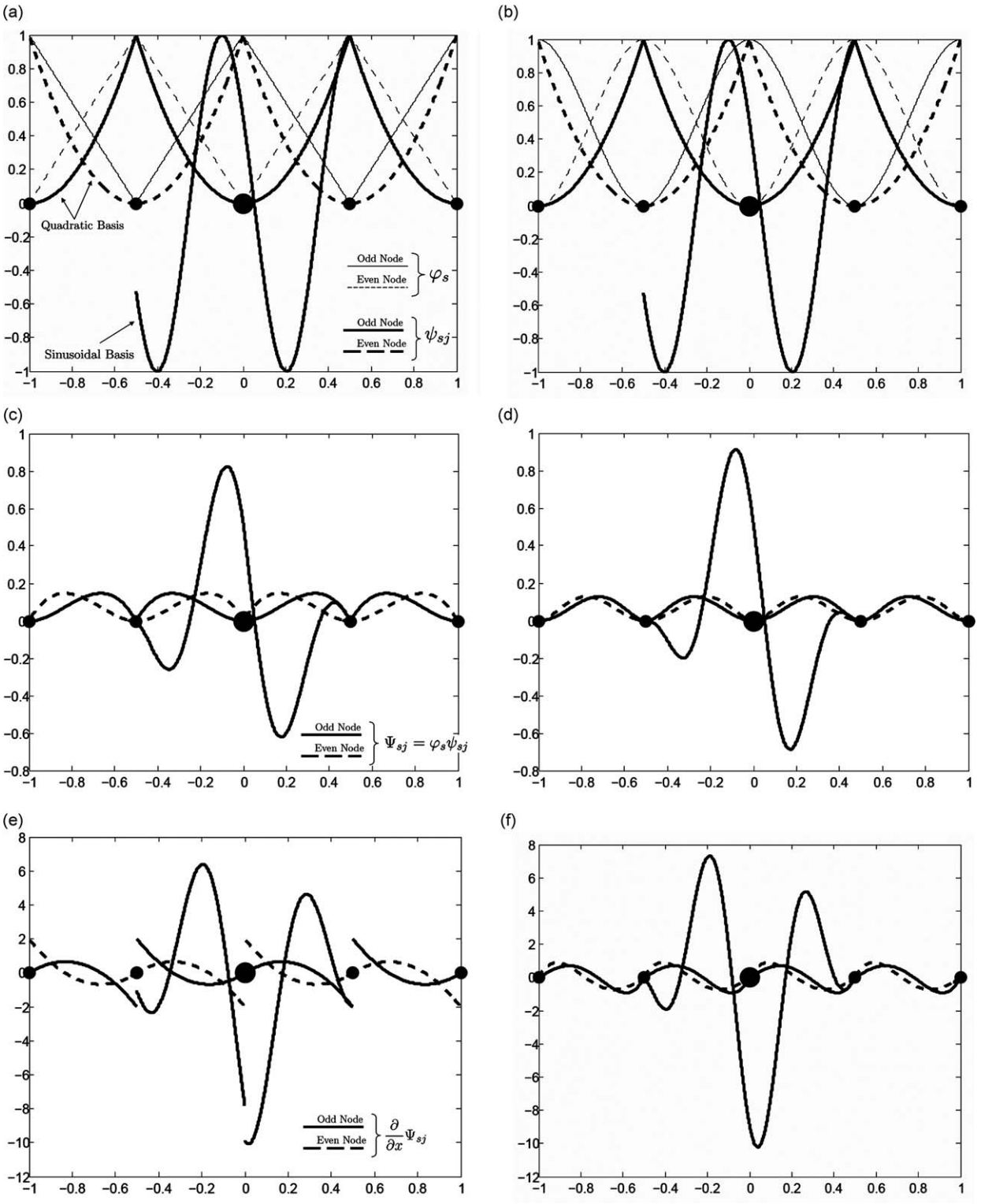

Fig. 3. Shape function construction in PUFEM. x-axis: x, y-axis: function evaluated at x. (a) PUFEM basis functions and the C^0 PU function. (b) PUFEM basis functions and the C^1 PU function. (c) PUFEM shape functions with C^0 pasting. (d) PUFEM shape functions with C^1 pasting. (e) Derivative of the shape functions—C^0 pasting. (f) Derivative of the shape functions—C^1 pasting.

Eq. (11) in Eq. (12), we obtain the following relation:

$$\sum_{s=1}^{P}\sum_{j=1}^{Q_s}\left[\int_{\Omega_s}\mathscr{L}_{\mathscr{F}\mathscr{P}}[\varphi_s(\mathbf{x})a_{sj}\psi_{sj}(\mathbf{x})]\nu\,\mathrm{d}\Omega+\alpha\int_{\Gamma_s\cap\Gamma}\varphi_s(\mathbf{x})a_{sj}\psi_{sj}(\mathbf{x})\nu\,\mathrm{d}\Gamma\right]=0 \tag{13}$$

The coefficients a_{sj} in Eq. (13) are unknowns, and can be found by using sufficient number of linearly independent test functions in order to set up a determined system of equations. It is clear from the above equation that a total of $\sum_{s=1}^{P}Q_s$ distinct test functions are required for this purpose. Following the Galerkin approach, we choose test functions to be same as the shape functions, i.e., $\mathscr{V}=\{\varphi_p\psi_{pq}\}$ (no summation of repeated index implied). Thus replacing $\nu(\mathbf{x})$ in Eq. (13) by $\varphi_p(\mathbf{x})\psi_{pq}(\mathbf{x})$ we obtain the following system of linear algebraic equations in the unknown coefficients involving the "stiffness matrix" **K** and "load vector" **f**:

$$\mathbf{Ka}=\mathbf{f} \tag{14}$$

In other words, Eq. (14) is a compact version of Eq. (13) with $\nu(\mathbf{x})=\varphi_p(\mathbf{x})\psi_{pq}(\mathbf{x})$. This is similar to the projection equations in the least-squares framework. The individual elements of the stiffness matrix and load vector can be identified as

$$K_{ij}=\int_{\Omega_s}\mathscr{L}_{FP}[\varphi_k(\mathbf{x})\psi_{kl}(\mathbf{x})]\varphi_p(\mathbf{x})\psi_{pq}(\mathbf{x})\,\mathrm{d}\Omega+\alpha\int_{\Gamma_s\cap\Gamma}\varphi_k(\mathbf{x})\psi_{kl}(\mathbf{x})\varphi_p\psi_{pq}(\mathbf{x})\,\mathrm{d}\Gamma \tag{15}$$

$$f_j=0 \tag{16}$$

where $i=(\sum_{s=1}^{k-1}Q_s+l)$ and $j=(\sum_{s=1}^{p-1}Q_s+q)$. Note that **K** is a square matrix of dimension $\sum_{s=1}^{P}Q_s$. Clearly for this problem, the solution lies in the span of the null-space of the matrix **K**. Theoretically, if it exists, the null-space of **K** is unique and has unit dimensionality, because the stationary solution of FPE is unique and globally asymptotically stable. However, this may not hold true for the numerical implementation shown above. If the parameter α is chosen to be too large, it may cause a rank deficiency in **K** of greater than 1. In such event, one can study the equation error to determine the best solution. Alternatively, the penalty parameter α can be tuned to obtain a single dimensional (hence unique) null-space. Another approach is to implement the boundary conditions not as $\mathscr{W}_\Gamma=0$, but a very small value, e.g. $\mathscr{W}_\Gamma=\varepsilon\,(\approx 10^{-9})$, so that the RHS of Eq. (14) is not zero. This approach gives highly acceptable results even with a very coarse tuning of α. The rank deficiency of **K** may also be caused due to other factors besides α, like the failure to incorporate the constraints (1) and (2) mentioned in Section 2.2 in the numerical method. Of course, the matrix **K** will always be ill-conditioned if α is not chosen judiciously.

Note that the solution to Eq. (14) results in a functional approximation of the pdf $\mathscr{W}$. This is highly desirable because there is no additional error incurred (interpolation error) in finding the solution at points other than the nodal points. In the next section, we discuss the use of GLO-MAP weight functions as PU functions to improve the PUFEM approximation obtained above.

3.4. Partition of unity pasting functions: GLO-MAP weights

It was stated in foregoing sections that PU pasting functions are of central importance in the PUFEM algorithm. They bring about the implicit domain discretization, merge together various local approximations by performing an unbiased average, and determine their order of continuity across local boundaries [27]. Because of the requirement of PU constraint (Eq. (9)), it is generally a difficult task to construct pasting functions that enforce continuity of any desired order. As previously mentioned, simple "tent-functions" are typically used, which provide C^0 continuity. More sophisticated positive functions have been used after re-normalization to enforce the PU constraint in the following manner: $\varphi_i(.)=\varphi_i(.)/\sum_j\varphi_j(.)$ (Shepard's functions). However, these functions are generally very difficult to integrate due to their rational-function form. The use of higher order polynomials as PU functions which could be automatically generated given a specified order of continuity of approximation has not been explored to a great extent in the PUFEM literature hence far. In this section, we briefly discuss the role of weight functions developed in the recently proposed GLO-MAP algorithm [28,29] as PU pasting functions. These functions are of polynomial form, satisfy Eq. (9), and have compact support—thus satisfying all requirements for a PU. Figs. 2(b) and 4 illustrate GLO-MAP weights upto C^2 continuity in 1 and 2 dimensions, respectively.

The idea behind GLO-MAP weights is surprisingly simple—given a node belonging to a discretized domain, the polynomial function of lowest degree which assumes the value unity at its parent node and decays to zero at all its neighboring nodes with specified degree of smoothness satisfies the property of partition of unity on the global domain Ω. These conditions can be easily used to determine the coefficients of such a polynomial in one variable and assumes the following general expression form (more details can be found in Ref. [28]):

$$w_{(m)}(x)=1-y^{m+1}\left\{\frac{(-1)^m(2m+1)!}{(m!)^2}\sum_{k=0}^{m}\frac{(-1)^k}{2m-k+1}\binom{m}{k}y^{m-k}\right\},\quad y=\frac{|x-x_i|}{2h} \tag{17}$$

where m is the desired order of smoothness at the boundary nodes. In the above equation, the function is defined in terms of a normalized variable y, which has value unity at its parent node and zero at all neighboring nodes. The tent-functions

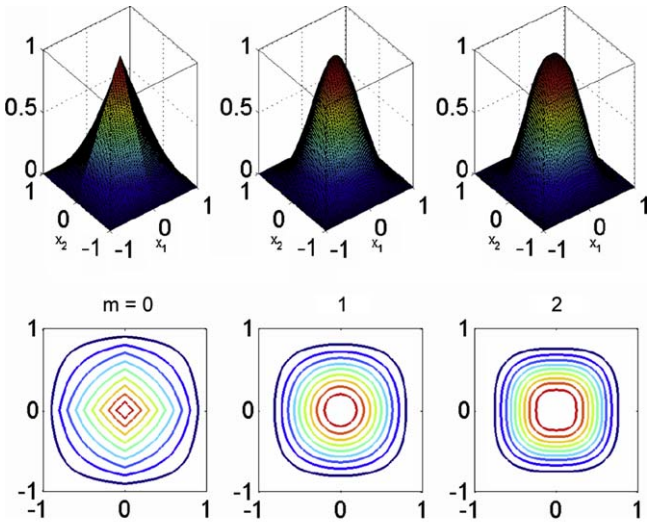

Fig. 4. GLO-MAP weight functions in 2-D, up to C^2 smoothness.

shown in Fig. 2(a) are a special case with $m = 0$. Functions with higher degree of smoothness ($m = 1$ and 2) have also been shown, and their benefits illustrated in Fig. 3. In the PUFEM framework, these weights come as an invaluable construct because of their several relevant properties [28,29]:

1. Polynomial form: By virtue of their polynomial form, GLO-MAP weights are easy to integrate. Additionally, if polynomial bases are used, the resulting weak form integrals can be evaluated analytically.
2. They satisfy the PU property. It is very easy to prove the fulfillment of this constraint when the GLO-MAP weights are written in local coordinates centered at the corresponding nodes and scaled with the inter-nodal distance along each dimension, $h^{(i)}$. This implies that in the local coordinates, the central node is at the centroid of an N-hypercube and all its neighboring nodes are at the various 2^N vertices. The value of the GLO-MAP weights are 1 and 0, respectively, at these locations.
3. They can provide any desired order of continuity across subdomain boundaries. This is very useful in applications which require the solution derivatives to satisfy error bounds.
4. Easy extension to higher dimensions: It is surprisingly easy to construct GLO-MAP weights in higher dimensions. A simple continued product of 1-D weights written along the various dimensions gives the weight function in the higher dimensional space which satisfies all the properties mentioned above. E.g. $w_{(2)}(x_1, x_2) = w_{(2)}(x_1)w_{(2)}(x_2)$, i.e. a GLO-MAP weight in 2-D providing C^2 continuity is simply the continued product of two 1-D weights providing the same level of smoothness.

In summary, the generality provided by GLO-MAP weight functions and their easy extension to N-dimensions opens the path for implementation of the PUFEM algorithm to solve high-dimensional PDEs (including the FPE). Furthermore, if basis functions orthogonal to these weight functions are used, we obtain an improvement in the condition number of the stiffness matrix, **K** [29]. A limitation of these functions is that in order to satisfy the PU constraint, the nodes must be aligned as if on a rectangular grid. This restriction implies that the PUFEM algorithm can be applied directly only to PDEs defined on N-hypercuboids. Domains of all other shapes would require a transformation into a hypercuboid. In the current application (FPE), however, this is not a problem, because the domain of solution can be chosen to be an N-hypercube.

3.5. *Local p-refinement*

The partition of unity functions described above have the capability of generating conformal approximation spaces out of independently chosen local basis functions for individual nodes. The selection of local basis functions can be made on the basis of a priori knowledge about the dynamical system. As mentioned in Section 2.2, local p-refinement curbs the curse of dimensionality indirectly by enriching only selective nodes, as demonstrated by Fig. 5. Fig. 5(a) shows the growth in degrees of freedom of the approximation in a standard PUFEM method without local p-refinement. The dimensionality of the underlying system is assumed to be 4. Horizontal curves are contours of constant polynomial order per node (varied from $p = 0$ to 8) while vertical curves represent contours of fixed number of nodes used for discretization along each dimension ($n = 5$–50). A grid is therefore formed and it is only possible to jump from one grid point to another. The y-axis

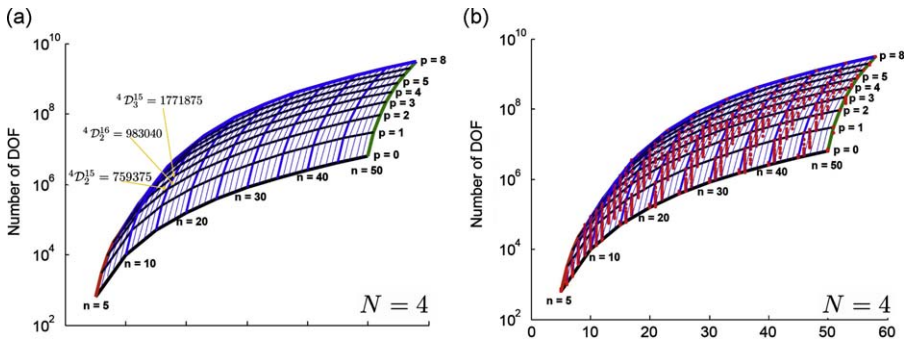

Fig. 5. Growth of dof in standard PUFEM alongside PUFEM with local p-refinement.

shows the number of degrees of freedom associated with each (n,p) pair. Three points are highlighted: $^4\mathscr{D}_2^{15} = 759,375$, $^4\mathscr{D}_2^{16} = 983,040$ and $^4\mathscr{D}_3^{15} = 1,771,875$. Clearly, the jump in degrees of freedom per added node for each dimension or per polynomial order for each node are enormous, which greatly restricts flexibility—for example, it is not possible to build an approximation with 800,000 degrees of freedom because it does not lie on the grid. On the other hand, having the option of enriching only selected nodes helps control this growth and balance between enrichment of the approximation space and growth in problem size. It makes the points in-between the grid-points in Fig. 5(a) accessible (see Fig. 5(b)), e.g. $^4\mathscr{D}_{\mathrm{loc}_p}^{15} \approx 800,000$.

4. Results

In this section, we present results for various dynamic systems residing in 2, 3 and 4 dimensions using the above outlined algorithm.

4.1. Dynamic system 1: example in 2 dimensions

We first consider the following 2-D nonlinear dynamic system:

$$\ddot{x} + \eta\dot{x} + \alpha x + \beta x^3 = g(t)\mathscr{G}(t) \tag{18}$$

Eq. (18) represents a stochastic Duffing oscillator with damping ($\eta > 0$) and is used widely for modeling of nonlinear vibrations. The expression for the true solution to the stationary FPE for this system is as follows:

$$\mathscr{W}_s(x,\dot{x})|_{\mathrm{true}} = \mathscr{C}\exp\left(-2\frac{\eta}{g^2Q}\left(\frac{\alpha x^2}{2} + \frac{\beta x^4}{4} + \frac{\dot{x}^2}{2}\right)\right) \tag{19}$$

where $\mathscr{C}$ is a normalization constant. Notice that the stationary pdf is an exponential function of the steady-state system energy (a Hamiltonian-like function), scaled by the parameter $-2\eta/g^2Q$ [1]. For simulation purposes, we use $\alpha = -15$, $\beta = 30$, $\eta = 10$ along with $g = 1$. The stationary pdf corresponding to these parameter values is bimodal, shown in Fig. 6(a). Following the discussion of rank deficiency of $\mathbf{K}$ in Section 3.3, the boundary condition was implemented as $W_\Gamma = \varepsilon$ $(= 10^{-9})$, resulting in a nonzero load vector. Fig. 6 shows the solution and error surfaces obtained using the PUFEM algorithm on an 18×18 rectangular grid with quadratic basis functions allocated to each node, i.e. $n = 18$ and $p = 2$ in Eq. (7). This is equivalent to a stiffness matrix of size 1944×1944. In other words, the size of the discretized problem (or the number of coefficients to determine) is $^2\mathscr{D}_2^{18} = 1944$. The results of this discretization are shown in Figs. 6(b) and (c). For comparison, we solved the same problem using the global-Galerkin approach [16] with scaled Hermite polynomials as basis functions. It was found that although the global approximation is able to provide similar accuracy for this problem, it is not a suitable approach because it is extremely sensitive to certain tuning parameters, like a reference pdf which is critical to the global approximation. The reference pdf is used to determine the finite domain of solution, and attaches relative weights to different regions of the domain. A slight perturbation in the reference leads to a significant rise in the error, and the degree of tuning achieved in this study case may not be possible for general nonlinear systems. An example of such sensitivity is shown in Fig. 6(e), in which the mean of the reference pdf was perturbed towards one of the modes, resulting in unbalanced weighting of the domain leading to significant errors. On the other hand, the PUFEM is not subject to such tuning issues. Furthermore, there is no scope for local solution refinement in the globalmethod.

The power of local refinement can be illustrated by considering the size of the discretized problem versus approximation accuracy. It is reasonable to assume that using quartic polynomials (instead of quadratic) would lead to better accuracy of the approximation. Increasing polynomial order for all nodes while maintaining the spatial discretization, i.e. keeping number of nodes fixed is called p-refinement. The corresponding problem size using quartic

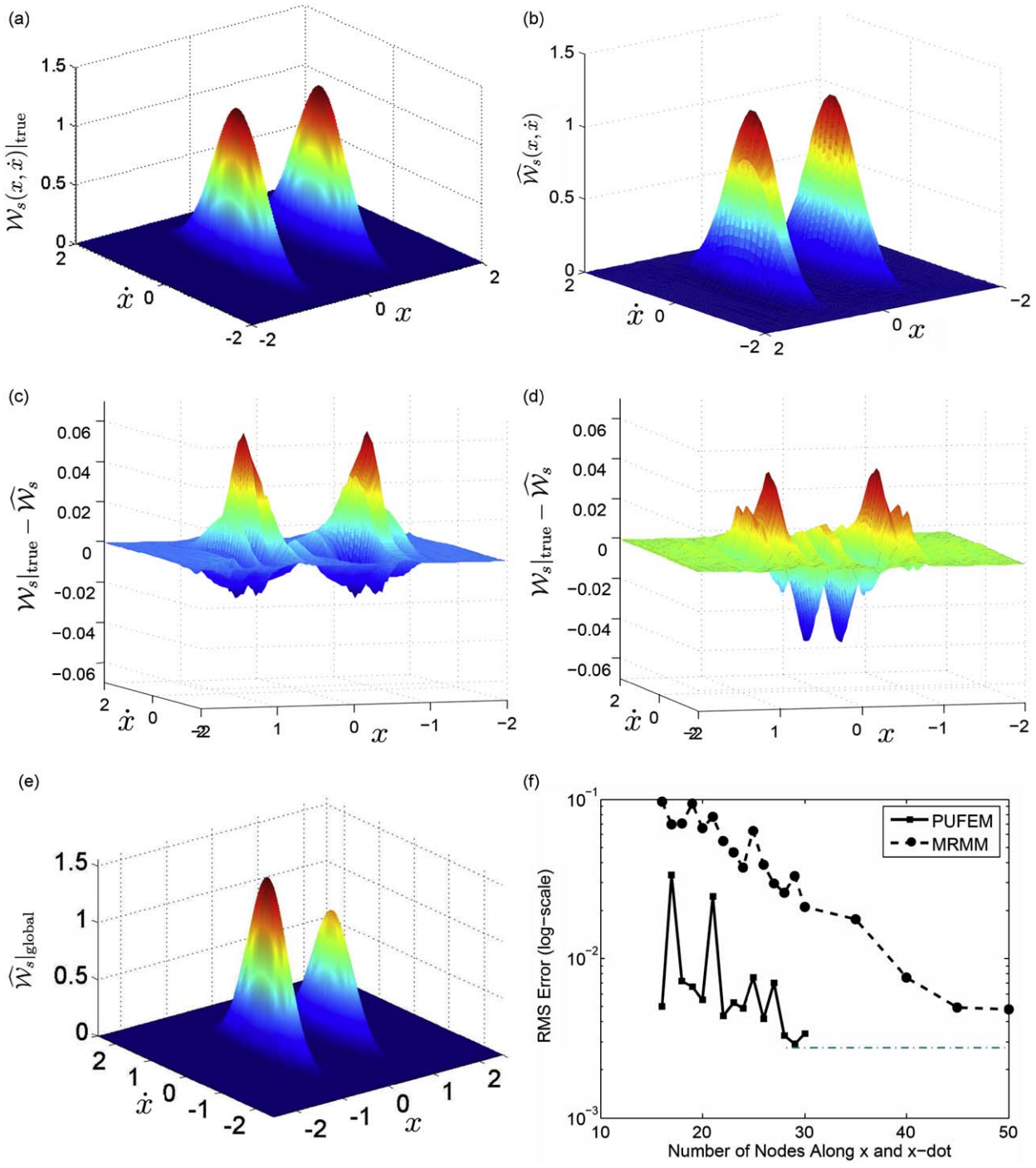

Fig. 6. Numerical results using the PUFEM algorithm and global Galerkin approach. (a) True stationary pdf for the duffing oscillator. (b) Computed solution: PUFEM algorithm, 18×18 grid with quadratic basis. dofs $= 1944$. (c) Error surface: PUFEM algorithm, 18×18 grid with quadratic basis. (d) Error surface: PUFEM algorithm, 9×9 grid with linear and quartic basis. dofs $= 831$. (e) Computed solution (global method, mean of the reference $pdf = (0, 0.0075)$). (f) Comparative convergence characteristics of PUFEM and MRMM for the damped duffing oscillator.

polynomials on an 18×18 grid is ${}^{N}\mathscr{D}_p^d = 4860$. On the other hand, we know that the pdf is expected to be almost flat near the boundary of the global domain, and linear basis functions would likely be sufficient to capture its behavior in these regions. Following this reasoning, it is possible with the current approach to supply the nodes lying on the boundary with linear basis functions and the interior nodes with quartic basis functions (local p-refinement). The resulting discretized FPE contains 4044 dofs, which is a sizeable reduction. Depending on the extent and nature of a priori information available

Table 1
Numerical results using PUFEM with local *p*-refinement: *two-state duffing oscillator.*

Node discretization	Polynomial orders (p)	Problem size (dof)	rms error (e_2)	Max error (e_∞)	Computation time (t_{comp}) [1.86 GHz Pentium M] (s)
18×18	Boundary nodes: $p=2$ Interior nodes: $p=2$	${}^2\mathscr{D}_2^{18} = 1944$	6.856×10^{-3}	5.905×10^{-2}	30.1
9×9	Boundary nodes: $p=1$ Interior nodes: $p=4$	${}^2\mathscr{D}_{\text{loc}_p}^{9} = 831$	5.998×10^{-3}	4.539×10^{-2}	10.7

about the particular problem at hand, it is possible to decide on the best polynomial assignment for each node such that an acceptable accuracy is obtained with a small number of dofs. Table 1 shows one such example for the soft-spring duffing oscillator. Both discretizations shown in Table 1 result in approximations with comparable RMS error (defined as $e_2 \triangleq \sqrt{(1/(r-1))\sum_{i=1}^{r}(\mathscr{W}(\mathbf{x}_i) - \widehat{\mathscr{W}}(\mathbf{x}_i))^2}$) and maximum error ($e_\infty$) (see Figs. 6(c) and (d)). However, the problem size for the second discretization (with local *p*-refinement) is less than half of the first, in addition to a reduction of about a third in the time of computation. This is an extremely important result, because it illustrates the fact that local *p*-refinement can provide same/better accuracy with a much smaller number of degrees of freedom, which augurs extremely well for higher dimensional problems.

In Kumar et al. [25], the above problem was solved using a multi-resolution meshless method based on the meshless Petrov–Galerkin approach. The convergence characteristics of PUFEM is found to be significantly better than MRMM, as seen in Fig. 6(f). Although the convergence rate of the latter algorithm is faster, the RMS-error values are higher. The fast rate of convergence of MRMM is most likely due to the decrease in interpolation errors as the density of nodes is increased. Furthermore, the PUFEM algorithm is considerably more computationally efficient, i.e. the time of execution of the PUFEM algorithm is much less than the computation time for MRMM. This is primarily due to the fact that MRMM requires the solution to several MLS problems (see Section 5) in the process of evaluating the weak form integrals. Thus for this particular problem, PUFEM provides improvement in accuracy and efficiency over other meshless methods based on MLS.

4.2. Dynamic system 2: example in 2 dimensions

Consider now the following 2-D quintic oscillator:

$$\ddot{x} + \eta\dot{x} + x(\varepsilon_1 + \varepsilon_2 x^2 + \varepsilon_3 x^4) = g(t)\mathscr{G}(t) \tag{20}$$

The stationary pdf for this system is given by the following expression:

$$\mathscr{W}_s(x,\dot{x})|_{\text{true}} = \mathscr{C}\exp\left(-2\frac{\eta}{g^2 Q}\left(\frac{\varepsilon_1 x^2}{2} + \frac{\varepsilon_2 x^4}{4} + \frac{\varepsilon_3 x^6}{6} + \frac{\dot{x}^2}{2}\right)\right) \tag{21}$$

The values of the various parameters used in this simulation are: $\varepsilon_1 = 1$, $\varepsilon_2 = -3.065$, $\varepsilon_2 = 1.825$, $\eta = 1.5$, $g = 1$. The stiffness matrix and load vector for this system are constructed exactly in the same manner as for system 1. From Fig. 7 it is clear that the method is able to handle systems with high order nonlinearity with ease. The comparative convergence curves for this system, using PUFEM and MRMM show a similar trend as for system 1. Table 2 shows the results for reduction in problem size for this system. Clearly, the results are quite similar to the ones obtained for the duffing oscillator. This example further reaffirms the ability of the current technique to handle highly nonlinear systems and multi-modal behavior accurately.

Study of the above two systems indicates that *p*-refinement (enriching the basis) typically provides superior error-reduction than *h*-refinement (adding more nodes). Moreover, *p*-refinement, especially local *p*-refinement greatly reduces problem size for the same accuracy level by reducing the number of nodes required along each discretized dimension. This fact is illustrated effectively in Fig. 8, which shows that *p*-refinement is clearly superior for error reduction. Note that it is possible to move towards the darker regions of low error on this graph by either *h*- or *p*-refinement. However, the latter approach clearly requires a very few number of nodes per dimension for achieving low approximation error. On the other hand, if one is constrained to work with fixed order polynomials (e.g. constant or linear polynomials), a very large number of nodes per dimension is required before the dark zone is reached, implying slower convergence. Such *h*-refinement is typically employed in standard FEM.

Overlaid on the error contours are contours of problem size, i.e. number of dofs for a given n and p. As expected, problem size increases monotonically upon increasing n and/or p (see Eq. (7)). Therefore, as long as one remains underneath the dof contour of a particular value, say the $\mathscr{D}^\star$-contour, the size of the discretized problem remains less than $\mathscr{D}^\star$. This is very useful information, because now looking at the composite contour-map in Fig. 8, we see that *p*-refinement provides not

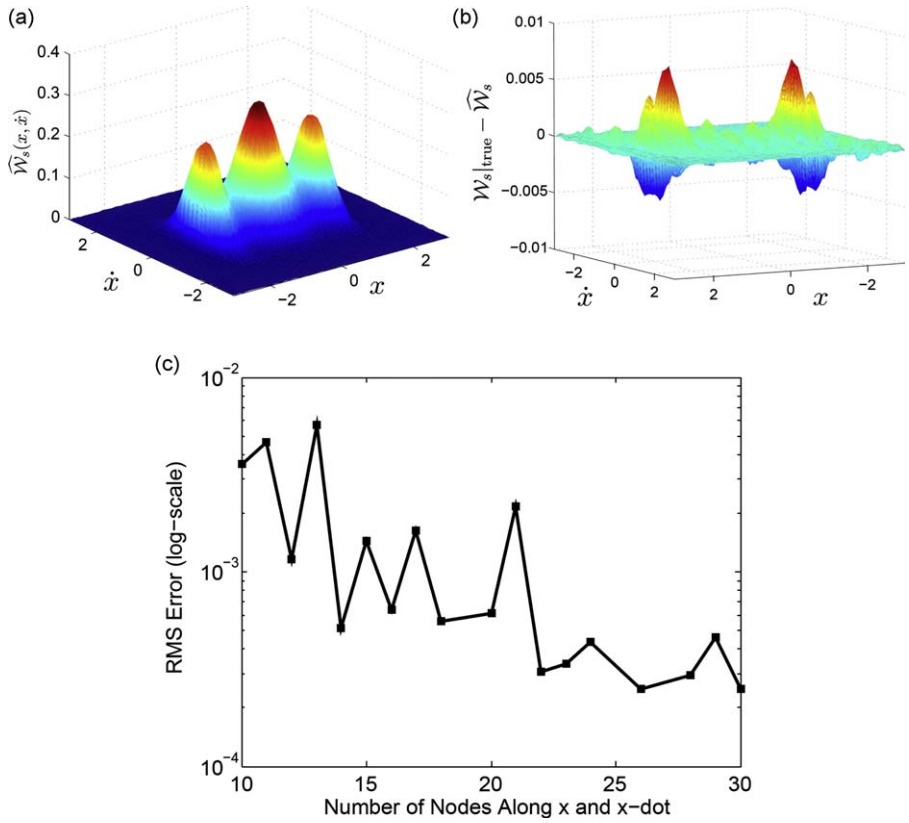

Fig. 7. Numerical results using the PUFEM algorithm for the quintic oscillator. (a) Computed solution: PUFEM algorithm, 18×18 grid with quadratic basis. dofs = 1944. (b) Error surface: PUFEM algorithm, 18×18 grid with quadratic basis. (c) Convergence characteristics for the quintic oscillator using PUFEM.

Table 2
Numerical results using PUFEM with local p-refinement: *two-state quintic oscillator.*

Node discretization	Polynomial orders (p)	Problem size (dof)	rms error (e_2)	Max error (e_∞)	Computation time (t_{comp}) [1.86 GHz Pentium M] (s)
18×18	Boundary nodes: $p = 2$ Interior nodes: $p = 2$	${}^2\mathscr{D}_2^{18} = 1944$	6.121×10^{-4}	7.750×10^{-3}	33.2
9×9	Boundary nodes: $p = 1$ Interior nodes: $p = 4$	${}^2\mathscr{D}_{\text{loc}_p}^9 = 831$	7.469×10^{-4}	6.127×10^{-3}	12.9

only lower error, it also helps keep the problem size small. Combined together, it leads to high accuracy with less computational effort. This is the ultimate criteria for being able to attack problems in higher dimensions. Fig. 8 has been referred to as an "accuracy-feasibility contour map" because it provides complete information about the number of nodes and order of polynomials required for desired accuracy, while keeping in mind available computational resources (i.e. size of the discretized problem to be solved). All results shown in this paper were obtained on a small workstation equipped with a 1.86 GHz Pentium M processor and 1 GB RAM. The unfeasible domain for this machine (i.e. too many dofs) are shown in the top-left section of the contour map.

At this point, we would like to state that the current approach is not claimed to be a remedy for the curse of dimensionality. Instead, it helps significantly ameliorate the curse, rather than cure it. Especially with the added flexibility of selective, local basis enrichment, it is possible to keep the growth of problem size under tight check as system dimensionality increases. Besides the systems considered in this section, the PUFEM algorithm was applied to several other nonlinear oscillators in 2-D state-space. In all systems studied, local p-refinement was seen to offer significant reduction in problem size for the same approximation accuracy. This aspect of the current method gives it advantage over both the global approach as well as traditional FEM.

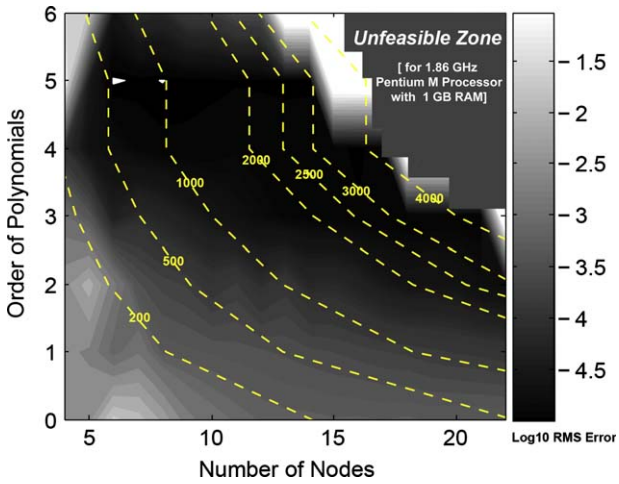

Fig. 8. A accuracy-feasibility contour map for a 2-D system.

Table 3
Comparative results using PUFEM with local *p*-refinement: *three-state linear system*.

Discretization method	Problem size (dof)	rms error (e_2)	Max error (e_∞)	Computation time (t_{comp})
$50 \times 50 \times 50$ Brick elements (FEM)	125,000	1.133×10^{-4}	Not available	Not available
$7 \times 7 \times 7$ nodes Boundary nodes: $p = 1$ Interior nodes: $p = 4$ (PUFEM, local *p*-refinement)	${}^3\mathscr{D}^7_{\text{loc}_p} = 5247$	2.823×10^{-4}	4.037×10^{-3}	18 min, 42.3 s [1.86 GHz Pentium M]

4.3. Dynamic system 3: example in 3 dimensions

Consider now the following 3-D linear system studied by Wojtkiewicz et al. [23]:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -\omega_0 & -2\zeta\omega_0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \tag{22}$$

The constants appearing in the above equation have the following values [23]: $\alpha = \omega_0 = 1,\ \zeta = 0.2$. The reason for studying a linear system is that its stationary distribution can be obtained easily by solving the corresponding algebraic Riccati equation. The stationary pdf for the above system was approximated by Wojtkiewicz et al. [23] using traditional FEM with "brick" elements in 3-D state-space. Comparative results have been shown in Table 3. Approximation accuracy for both methods are approximately the same, but the current method holds a significant advantage in computational load. Fig. 9 compares the computed $x_1 - x_2$ marginal pdf to the true marginal. Note that if we were constrained to use quartic polynomials on all nodes, the resulting problem size would be ${}^3\mathscr{D}^7_4 = 12,005$. This would most likely provide better accuracy, but at a much higher computational cost, likely beyond the capability of a small computer. Therefore, local *p*-refinement provides an attractive balance between approximation accuracy and computational cost. In this case, we obtain the same order of accuracy as traditional FEM with three orders of magnitude less dofs. This is indeed a massive advantage.

4.4. Dynamic system 4: nonlinear example in 3 dimensions

Consider now the following nonlinear system with 3-D state-space:

$$\dot{x} = \sigma(y - x) + \zeta_1(t)$$

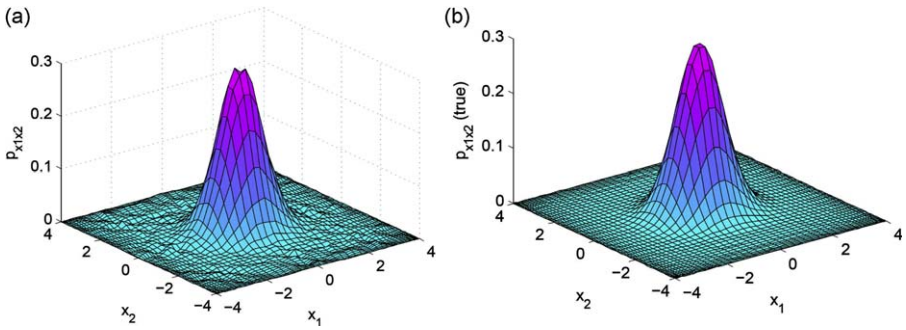

Fig. 9. Computed and true x_1-x_2 marginal distributions for the linear 3-D stochastic dynamics of Eq. (22).

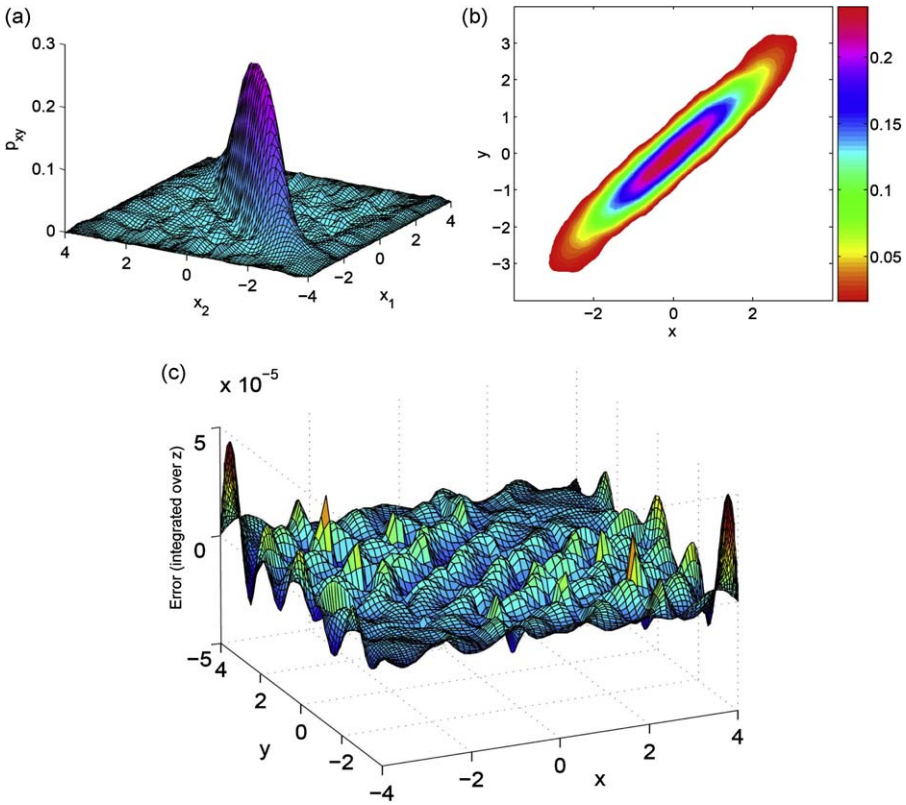

Fig. 10. Computed $x-y$ marginal distribution for the noise-driven Lorenz attractor of Eq. (23). (a) Computed $x-y$ marginal for the Lorenz attractor. (b) Contour plot of the computed $x-y$ marginal surface. (c) Equation error surface integrated along the z-axis.

$$\dot{y} = x(\rho - z) - y + \zeta_2(t)$$

$$\dot{z} = xy - \beta z + \zeta_3(t) \tag{23}$$

The above equations represent a noise-driven Lorenz attractor. The Lorenz attractor is a chaotic system that was originally used to model climatic changes and has been used to study numerous physical systems, e.g. laser systems (Maxwell–Bloch model). Numerical values for various parameters appearing above are: $\sigma = 10$, $\rho = 1$; $\beta = \frac{8}{3}$ and Q (noise intensity) $= 2$. Figs. 10(a) and (b) show the $x-y$ marginal computed from the stationary pdf obtained by solving the static FPE corresponding to Eq. (23). The discretization utilized for this solution was a $6 \times 6 \times 6$ nodal grid with boundary nodes endowed with quadratic polynomials and interior nodes with quartic polynomials (corresponding problem size $= 3760$ dofs). An analytical result is not available for this system, therefore error was quantified in terms of equation-error and is shown in Fig. 10(c). In this figure, equation error has been integrated along the z-direction to obtain a 2-D surface. The shown error surface has an RMS value of 6.668×10^{-6}.

4.5. Dynamic system 5: example in 4 dimensions

Here we look at the following linear dynamical system with a 4-D state-space studied by Wojtkiewicz et al. [24]:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -(k_1+k_2) & -c_2 & k_2 & 0 \\ 0 & 0 & 0 & 1 \\ k_2 & 0 & -(k_2+k_3) & -c_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \zeta(t) \tag{24}$$

The Fokker–Planck equation of concern is

$$\begin{aligned} \frac{\partial p}{\partial t} = & -x_2 \frac{\partial p}{\partial x_1} - \frac{\partial}{\partial x_2}[(-(k_1+k_2)x_1 - c_1 x_2 + k_2 x_3)p] - x_4 \frac{\partial p}{\partial x_3} \\ & - \frac{\partial}{\partial x_3}[(k_2 x_1 - (k_2+k_3)x_3 - c_2 x_4)p] + D_1 \frac{\partial^2 p}{\partial x_2^2} + D_2 \frac{\partial^2 p}{\partial x_4^2} \end{aligned} \tag{25}$$

Constants appearing above have the following values: $k_1 = k_2 = k_3 = 1$, $c_1 = c_2 = 0.4$, $D_1 = D_2 = 0.2$. Fig. 11 shows the $(x_1 - x_2)$ marginal distribution computed with the meshless PUFEM method alongside the true marginal surface for the above linear system. The available computing resources allowed the use of five nodes along each of the 4 dimensions, with all interior nodes carrying cubic basis functions and boundary nodes linear polynomials. This leads to a total of 5555 dofs, which is in sharp contrast to approximately 2.56 million dofs used in the standard FEM approach. Table 4 shows that the current method provides an accuracy of one order of magnitude less than FEM. We note that this is solely due to the limitation of computing resources currently utilized, with which it is not possible to deal with problems of size greater than about 5500. In the absence of local p-refinement, the problem size with cubic polynomials on a $5 \times 5 \times 5 \times 5$ grid would be ${}^4\mathscr{D}_3^5 = 21,875$, which is well beyond the capability of a small computer. It is important to note therefore that the current approach provides highly acceptable accuracy (e.g. for use in the design of control laws via policy iteration) with better than three orders of magnitude reduction in problem size so that it can be solved on a small computer. Given the ability to deal with moderately larger matrices, it is reasonable to believe that excellent approximations can be obtained for high-dimensional problems.

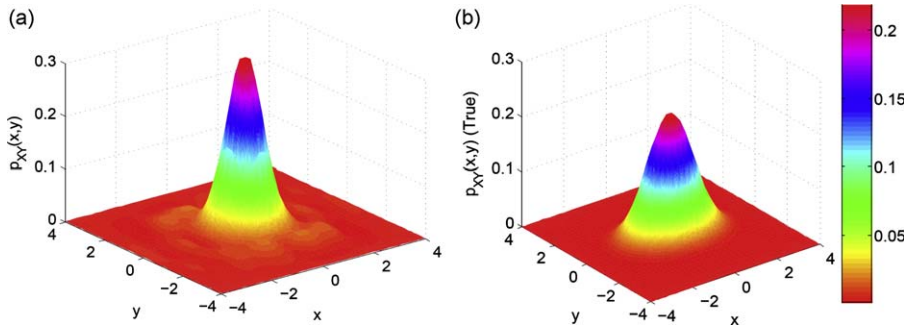

Fig. 11. Comparison of the computed $(x_1 - x_2)$ marginal for the 4-D linear system with the truth. (a) Computed X–Y marginal surface. (b) True X–Y marginal surface.

Table 4
Comparative results using PUFEM with local p-refinement: *four-state linear system*.

Discretization method	Problem size (dof)	rms error (e_2)	Max error (e_∞)	Computation time (t_{comp})
$40 \times 40 \times 40 \times 40$ 4D "Brick" elements (FEM)	256,000	5.237×10^{-5}	2.911×10^{-3}	Not available
$5 \times 5 \times 5 \times 5$ nodes Boundary nodes: $p = 1$ Interior nodes: $p = 3$ (PUFEM, local p-refinement)	${}^4\mathscr{D}_{\text{loc}_p}^5 = 5555$	9.769×10^{-4}	8.870×10^{-2}	23 h 36.4 min [1.86 GHz Pentium M]

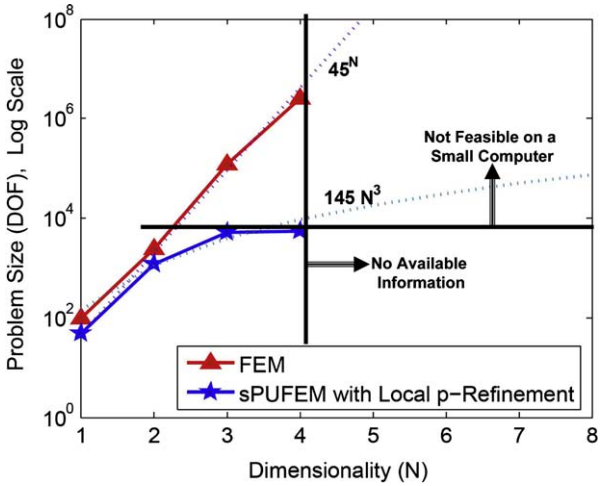

Fig. 12. Ameliorating the *curse of dimensionality*.

Table 5
Growth in problem size with system dimensionality: FEM and PUFEM.

N	Problem size (dof)		Feasible on a small computer?	
	FEM	PUFEM	FEM	PUFEM
1	100	50	Yes	Yes
2	2500	1200	Yes	Yes
3	1,25,000	5200	No	Yes
4	2,560,000	5555	No	Yes

4.6. Remark on the curse of dimensionality

It has been demonstrated above through several examples that the current method, coupled with local *p*-refinement has the ability to minimize the effect of the curse of dimensionality in FPE. Since the number of nodes required for discretization remains an exponential function of the system dimensionality, a claim to breaking the curse cannot be made. At the same time, strong evidence towards curtailment of the curse has been presented. To further support this evidence, we look at Fig. 12 and Table 5. In Fig. 12, several problems residing in dimensions 1–4 have been solved with fixed accuracy ($\mathcal{O}(e_2) \approx 10^{-4}$) using the current meshless algorithm and the required dofs is compared with the finite element technique. Note that dof requirement for FEM grows almost linearly in log-scale indicating the curse of dimensionality. On the other hand, the accuracy curve for PUFEM tends to flatten out as the dimensionality is increased. This trend is extremely encouraging for further progress to problems residing in 5, 6 and even higher dimensions. Approximate fits for the available data have also been shown in Fig. 12. We point out that no rigorous conclusions can be drawn from these fits because very little information (four data points) is available. For FEM, it is well known (also seen in Fig. 12) that an exponential fit best describes the problem size growth. On the other hand, a cubic fit seems to capture the growth in the current approach. If indeed true, it would mean breaking of the curse of dimensionality (polynomial growth), but no formal conclusion is possible. Moreover, it is dangerous to extrapolate and no conclusion can be drawn for dimensions 5, 6 and higher. With greater computational resources, it would be possible to confirm such extension. Table 5 shows the numbers appearing in Fig. 12 along with feasibility on a small computer. Since all results for the current approach were obtained on a small computer, it is easy to imagine that given greater computational ability, the current method would handle problems residing in much higher dimensions.

5. Evaluation of PUFEM against other existing techniques

In this section, we make a pointwise comparison of various finite element-based numerical techniques for solving the FPE for nonlinear dynamic systems on hypercuboids. We shall compare the PUFEM against the global Galerkin method, the conventional FEM, and the other meshless methods like the one used in Kumar et al. [25] (MRMM).

- *Shape function selection*: As seen in the examples presented, PUFEM offers great flexibility in the selection of local approximation spaces because it is possible to introduce different number of independently chosen basis functions in

the different subdomains (local p-refinement). The shape functions are constructed by simply multiplying the basis functions with the PU pasting functions. In most other meshless methods, the shape functions are constructed using data fitting algorithms like the MLS. Consequently, while it is possible to use nonpolynomial functions in the approximation space, it is a relatively difficult task to use different basis functions in different regions of the solution domain. Conventional FEM typically uses only polynomial shape functions in the approximation space, the order of which is determined by the shape of the finite element. Finally, there is no scope for local error improvement of the solution in the global methods because of the nature of the formulation.

- *Error characteristics*: The convergence of PUFEM is expected to be superior to that of FEM, especially with the use of special functions in the approximation space which directly improves its approximability. From our experience in the current application, we conclude that the convergence characteristics of PUFEM is better than that of MRMM, using the same basis function set (Figs. 6(f), 7(c)). This could be partially due to the absence interpolation errors in PUFEM, whereas in MRMM additional errors are introduced due to interpolation required to find the solution at points other than the solution nodes.
- *Computational load*: If we divide the computational load of the current application into three stages, pre-processing (grid generation), integration (evaluation of the weak form integrals) and post-processing (finding solution at given points in the domain), we get the following relative ordering: PUFEM and MRMM rank above FEM in the pre-processing stage because the required grid information is minimal in the former two. PUFEM and FEM are faster than most other meshless methods in the integration stage because the latter require the solution to an MLS problem for every quadrature point used for the numerical evaluation of the integrals. PUFEM ranks above both FEM and other meshless methods in the post-processing stage because it provides a functional form of the approximation; i.e. no interpolation is required to construct the solution at any given point in the domain as with the other methods. At the same time, a functional characterization of the approximation requires greater number of parameters per node (as opposed to the case where only the approximation's value is stored for each node). This would tend to imply greater memory requirement for PUFEM, but the fact that p-refinement is possible in the current algorithm, the number of nodes is reduced drastically. The resulting collective effect is that the number of degrees of freedom (hence memory requirement) is actually very small for PUFEM, as seen in the examples presented in 3 and 4 dimensions.
- *Application to high-dimensional problems*: PUFEM and MRMM stand out from the conventional FEM in the implementation to higher dimensional problems because mesh generation in 3 and higher dimensions is still not practical. Comparing MRMM and PUFEM in this respect, PUFEM has definite advantage because of its simpler algorithm structure and much smaller time of execution. The easy implementation of local p-refinement makes the current approach extremely attractive for use in higher dimensional nonlinear problems. As shown in the examples, the current approach creates a discretized problem that is three orders of magnitudes smaller in size than traditional FEM (for comparable accuracy). It is therefore reasonable to claim that with the current method it would be possible to solve problems in much higher dimensions than FEM before the current computing limit is reached.

Summary: PUFEM emerges as the algorithm that is the easiest to implement in the current application because of its simplicity. The discussed approach involves only the numerical evaluation of integrals in the weak form followed by the inversion of the stiffness matrix, while most other meshless methods require the solution to several MLS problems in addition to these tasks. Furthermore, if polynomial bases are used, the PUFEM integrals can be found analytically. On the other hand, most other meshless methods require numerical integration even with polynomial bases because the shape functions resulting from the MLS procedure are not polynomials. PUFEM scores over traditional FEM in its ability to allow for local p-refinement, leading to smaller sized discretized problems, which is the key for higher dimensional systems.

6. Conclusions

We conclude that the PUFEM is a robust and promising approach for attacking the FPE for high-dimensional systems due to its multiple advantages. The use of GLO-MAP weights as PU pasting functions further improves the quality of approximation by improving its continuity characteristics. Local p-refinement has been utilized effectively to reduce problem size for fixed accuracy. It has been demonstrated through numerous examples in 2–4 dimensions that the curse of dimensionality can be curtailed effectively with local p-enrichment. At the same time, much remains to be done in for the formalization of the technique of local p-refinement and identifying its niche in breaking the curse of dimensionality. An extension of this method coupled with modal analysis has been developed by the authors to develop an algorithm to solve the transient FPE for high-dimensional systems. Development of efficient particle-PUFEM methods is currently under progress for the Fokker–Planck equation to break the exponential growth of the number of nodes for spatial discretization.

References

[1] A.T. Fuller, Analysis of nonlinear stochastic systems by means of the Fokker–Planck equation, *International Journal of Control* 9 (6) (1969) 603–655.
[2] H. Risken, *The Fokker Planck Equation: Methods of Solution and Applications*, *Springer Series in Synergetics*, Springer, Berlin.
[3] D.C. Polidori, J.L. Beck, Approximate solutions for nonlinear vibration problems, *Probabilistic Engineering Mechanics* 11 (1996) 179–185.
[4] P.W. Chodas, D.K. Yeomans, Orbit determination and estimation of impact probability for near earth objects, *Proceedings of the Guidance and Control, Advances in the Astronautical Sciences*, Vol. 101, 1999, pp. 21–40.

[5] M.C. Wang, G. Uhlenbeck, On the theory of Brownian motion II, *Reviews of Modern Physics* 17 (2–3) (1945) 323–342.
[6] S. Chakravorty, M. Kumar, P. Singla, A quasi-Gaussian Kalman filter, *American Control Conference*, Minneapolis, MN, 14–16 June 2006.
[7] E.A. Johnson, S.F. Wojtkiewicz, L.A. Bergman, B.F. Spencer, Observations with regard to massively parallel computation for Monte Carlo simulation of stochastic dynamical systems, *International Journal of Nonlinear Mechanics* 32 (4) (1997) 721–734.
[8] H.P.N. Harnpornchai, G. Schnëller, Stochastic analysis of dynamical systems by phase-space-controlled Monte Carlo simulation, *Computer Methods in Applied Mechanics and Engineering* 168 (1999) 273–283.
[9] W.F. Wu, Y.K. Lin, Cumulant-neglect closure for nonlinear oscillators under parametric and external excitations, *International Journal of Nonlinear Mechanics* 19 (4) (1984) 349–362.
[10] J.B. Roberts, P.D. Spanos, *Random Vibration and Statistical Linearization*, Dover Publications, New York, 2003.
[11] H.J. Pradlwarter, M. Vasta, Numerical solution of the Fokker–Planck equation via Gaussian superposition representation, in: N. Shairaishi, M. Shinozuka, Y.K. Wen (Eds.), *Structural Safety and Reliability*, *ICOSSAR*'97—7*th International Conference on Structural Safety and Reliability*, Vol. 2, Kyoto, 1997, pp. 917–923.
[12] R.S. Park, D. Scheeres, Nonlinear mapping of Gaussian state uncertainties: theory and applications to spacecraft control and navigation, *AAS/AIAA Astrodynamics Specialist Conference*, Lake Tahoe, CA, USA, AAS Paper No. 05-404, 7–11 August 2005.
[13] H.J. Pradlwarter, Nonlinear stochastic response distributions by local statistical linearization, *International Journal of Nonlinear Mechanics* 36 (7) (2001) 1135–1151.
[14] J.P. Johnson, R.A. Scott, Extention of eigenfunction-expansion solutions of a Fokker Planck equation—I. First order system, *International Journal of Nonlinear Mechanics* 14 (1979) 315–324.
[15] S.H. Crandall, Perturbation techniques for random vibration of nonlinear systems, *Journal of Acoustical Society of America* 35 (11) (1963) 1700–1705.
[16] G. Muscolino, G. Ricciardi, M. Vasta, Stationary and non stationary probability density function for non-linear oscillators, *International Journal of Nonlinear Mechanics* 32 (6) (1997) 1051–1064.
[17] M.D. Paola, A. Sofi, Approximate solution of the Fokker–Planck–Kolmogorov equation, *Probabilistic Engineering Mechanics* 17 (2002) 369–384.
[18] A. Kunert, Efficient numerical solution of multidimensional Fokker–Planck equation with chaotic and nonlinear random vibration, in: T.C. Huang, C.S. Hsu, W.Q. Feng, S.C. Sinha, R.A. Ibrahim, R.L. Engelstad (Eds.), *Vibration Analysis—Analytical and Computational*, 1991, pp. 51–60.
[19] S.F. Wojtkiewicz, L.A. Bergman, B.F. Spencer Jr., High fidelity numerical solutions of the Fokker–Planck equation, in: A. Bazzani, J. Ellison, H. Mais, G. Turchetti (Eds.), *Proceedings of ICOSSAR*'97: *The* 7*th International Conference on Structural Safety and Reliability*, Kyoto, Japan, 24–28 November 1997.
[20] R.S. Langley, A finite element method for the statistics of random nonlinear vibration, *Journal of Sound and Vibration* 101 (1) (1985) 41–54.
[21] E.A. Johnson, S.F. Wojtkiewicz, L.A. Bergman, B.F. Spencer Jr., Finite element and finite difference solutions to the transient Fokker–Planck equation, in: A. Bazzani, J. Ellison, H. Mais, G. Turchetti (Eds.), *Proceedings of a Workshop*: *Nonlinear and Stochastic Beam Dynamics in Accelerators—A Challenge to Theoretical and Numerical Physics*, Lüneburg, Germany, 1997.
[22] A. Masud, L.A. Bergman, Application of multi-scale finite element methods to the solution of the Fokker–Planck equation, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 1513–1526.
[23] S.F. Wojtkiewicz, L. Bergman, B.F. Spencer Jr., Numerical solution of some three state random vibration problems, in: L.A. Bergman, B.F. Spencer Jr. (Eds.), *Vibration of Nonlinear, Random and Time-varying Systems*, Boston, MA, USA, 1995.
[24] S.F. Wojtkiewicz, L.A. Bergman, Numerical solution of high dimensional Fokker Planck equations, 8*th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, Notre Dame, IN, USA, 2000.
[25] M. Kumar, P. Singla, J. Junkins, S. Chakravorty, A multi-resolution meshless approach to steady state uncertainty determination in nonlinear dynamical systems, 38*th IEEE Southeastern Symposium on Systems Theory*, Cookeville, TN, 5–7 March 2006.
[26] S.N. Atluri, T. Zhu, A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics, *Computational Mechanics* 22 (2) (1998) 117–127.
[27] I. Babuška, J. Melenk, The partition of unity finite element method, *International Journal of Numerical Methods* 40 (1997) 727–758.
[28] J.L. Junkins, P. Singla, Orthogonal global/local approximation in *N*-dimensions: applications to input/output approximation, *Invited paper, 6th Conference on Dynamics and Control of Systems and Structures in Space*, Cinque-Terre, Italy, July 2004.
[29] P. Singla, Multi-resolution Methods for High Fidelity Modeling and Control Allocation in Large-scale Dynamical Systems, Ph.D. Dissertation, Department of Aerospace Engineering, Texas A&M University, College Station, USA, 2006.
[30] M. Kumar, S. Chakravorty, J.L. Junkins, A semianalytic meshless approach to the transient Fokker–Planck equation, in preparation.
[31] M. Kumar, S. Chakravorty, J. Junkins, A homotopic approach to domain determination and solution refinement for the stationary Fokker–Planck equation, *Probabilistic Engineering Mechanics* 24 (3) (2009) 265–277.
[32] S.N. Atluri, S. Shen, *The Meshless Local Petrov–Galerkin* (*MLPG*) *Method*, Tech Science Press, 2002.
[33] T. Belytschko, Y.Y. Lu, J. Gu, Element free Galerkin method, *International Journal of Numerical Methods in Engineering* 37 (1994) 229–256.
[34] W. Liu, S. Jun, Y. Zhang, Reproducing kernel particle methods, *International Journal for Numerical Methods in Fluids* 20 (8–9) (1995) 1081–1106.
[35] A. Duarte, J.T. Oden, Hp clouds—an h-p meshless method, *Numerical Methods for Partial Differential Equations* 12 (1996) 673–705.