The Partition of Unity Finite Element Approach with $hp$-refinement for the Stationary Fokker-Planck Equation

Mrinal Kumar*a, Suman Chakravorty*a,1, Puneet Singla*b,2, John L. Junkins*a,3

*aDepartment of Aerospace Engineering, Texas A&M University, College Station, TX 77843, USA.
bDepartment of Mechanical and Aerospace Engineering, The State University of New York, Buffalo, 14260, USA.

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 the stationary FPE for two, three and four-dimensional 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. The 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 the numerical solution of the FPE.

*Corresponding author. Graduate Research Assistant.

Email addresses: mrinal@neo.tamu.edu (Mrinal Kumar), schakrav@aeromail.tamu.edu (Suman Chakravorty), psingla@buffalo.edu (Puneet Singla), junkins@tamu.edu (John L. Junkins)

URL: http://people.tamu.edu/~mrinal (Mrinal Kumar)

1Assistant Professor
2Assistant Professor
3Distinguished Professor, Holder of the Royce Wisenbaker Chair

Preprint submitted to Journal of Sound and Vibration

December 31, 2008
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 the response of engineering structures like trusses under random excitation (in structural mechanics[3]), propagation of initial condition uncertainty of an asteroid for the 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 the computation of uncertainty in the prediction step in the design of a Bayes filter (in filtering theory[6]). All these applications require the study of the time evolution of the probability density function (pdf) \( \mathcal{W}(t, \mathbf{x}) \), 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 the Monte Carlo methods[7, 8], Gaussian closure (or higher order moment closure)[9, 10, 11, 12], equivalent linearization and stochastic averaging[13, 10]. The Monte Carlo method essentially involves the 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 the desired number of moments from the distribution of the propagated sample 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 the 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 the 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 the FPE contains the complete information about the state pdf, $\mathcal{W}$. With reference to Fig. 1, this implies that the 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, the 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 the 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 the increasing computational capability, more sophisticated algorithms like the global Galerkin method[16, 17, 18], finite differences[19], finite elements[20, 21], and multi-scale finite elements[22] have been developed and used on 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 dimensional systems with success. However, due to the 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[25] (based on the Meshless Petrov-Galerkin method[26] (MLPG)) 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 for the algorithm for such problems 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\cite{27} (PUFEM) is employed to tackle the issue of excessive memory requirement and computation time. The current method retains the extremely attractive meshless nature, thus facilitating the discretization of high dimensional state-space into subdomains. The approach allows for independent selection of basis functions in the individual subdomains, (local $p$-refinement) thus providing flexibility in the construction of the approximation space. Local $p$-refinement is also the key to reducing the size of the discretized problem because it allows for the introduction of higher order polynomials only in selected subdomains where they are required. Such local enrichment of the approximation space directly curbs the curse of dimensionality because the problem size no longer grows purely as an exponential function of the dimensionality of the system. The benefit of local $p$-refinement is clearly demonstrated in this paper by means of an example. The various local approximations are blended together using pasting functions (also known as weight functions) that satisfy a partition of unity over the solution domain, leading to unbiased blending. Tent-functions are typically used for this purpose, providing $C^0$ continuity across local subdomains. In the current paper, the recently developed GLO-MAP functions\cite{28, 29} are used as the weights. These functions provide blending of any desired order of continuity and are of polynomial form, which also simplifies the process of numerical integration. The authors have further developed 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 rest of the current paper is organized as follows: Section 2 introduces the FPE and discusses the key issues surrounding its numerical solution. Section 3 lays out the various aspects of the PUFEM approach in detail and the GLOMAP 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 the FPE and draw conclusions respectively.

2. The Fokker-Planck Equation

As previously mentioned, the Fokker-Planck equation provides the exact description of the uncertainty propagation problem for dynamical systems driven by white-noise excitation. In essence, the FPE is a scalar partial differential equation of the 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 $N$-dimensional white-noise driven nonlinear dynamic system with uncertain initial conditions, given by the following equation:

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

(1)

where, $\Gamma(t)$ represents a $M$-dimensional Gaussian white-noise process with the correlation function $Q_\delta(t_1 - t_2)$, and $\bar{x}_0$ represents the mean initial state. Vector functions $f(t,x) : [0, \infty) \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ and $g(t,x) : [0, \infty) \times \mathbb{R}^N \rightarrow \mathbb{R}^{N \times M}$ are measurable functions. The initial probability distribution of the state is given by the pdf $\mathcal{W}(t_0, x)$, which captures the state uncertainty at time $t_0$.

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

$$\frac{\partial}{\partial t} \mathcal{W}(t, x) = \mathcal{L}_x \mathcal{W}(t, x)$$

(2)
where,

\[
\mathcal{L}_{FP} = \left[ -\sum_{i=1}^{N} \frac{\partial}{\partial x_i} D_i^{(1)}(\cdot) + \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial^2}{\partial x_i \partial x_j} D_{ij}^{(2)}(\cdot) \right]
\]

\[
D^{(1)}(t, x) = f(t, x) + \frac{1}{2} \frac{\partial g(t, x)}{\partial x} Q g(t, x)
\]

\[
D^{(2)}(t, x) = \frac{1}{2} g(t, x) Q g^T(t, x)
\]

where, \( \mathcal{L}_{FP} \) is the Fokker-Planck operator, \( D^{(1)} \) is known as the drift coefficient vector and \( D^{(2)} \) is the diffusion coefficient matrix. The drift vector captures the drifting apart of the mean of the propagated pdf from the propagated mean of the initial pdf. Generally, this drift increases with the degree of the nonlinearity of the underlying dynamics, i.e., \( f(t, x) \). The diffusion matrix captures the spreading out of the substantial portion of the pdf (e.g., the 3\( \sigma \) region) over the 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., \( g(t, x) = 0 \) and the source of uncertainty lies only in the 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 the initial state uncertainty of an asteroid through the underlying 2/\( N \)-body equations of motion, for the 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 \( D^{(1)} \). In engineering fields however, the Stratonovich form in preferred, to avoid the necessity of Itô calculus which is required to deal with the Itô form. The two forms are identical in the case of state additive noise, i.e., when \( g(t, x) = g(t) \). 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 the steady state solution of Eq.2, i.e. the following equation:

\[ \mathcal{L}_{FP} W(x) = 0 \]  \hspace{1cm} (6)

In other words, we desire to extract the null-space of the Fokker-Planck operator. A methodology for the transient problem using the current approach, coupled with modal analysis has been discussed in another paper[30]. The 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 the time invariance of the system dynamics, i.e. \( f(t, x) = f(x) \) and \( g(t, x) = g(x) \). Since we are currently interested only in state-additive noise, the function \( g(t, x) \) reduces to a constant. Other necessary conditions for the existence of a nontrivial and unique stationary solution of the 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 the FPE for the pdf is a formidable problem because of the following issues:

1. Positivity of the pdf: \( W(t, x) \geq 0, \ \forall t \ & \ x. \)
2. Normalization constraint of the pdf: \( \int_{-\infty}^{\infty} W(t, x)dV = 1. \)
3. No fixed solution domain for numerical implementation: how to impose boundary conditions in a finite region and restrict numerical computation to regions where \( W \gtrsim 10^{-9} \).
4. Curse of Dimensionality: Exponential increase in the size of the discretized problem with the dimensionality of the 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 renormalization of the obtained solution, (1) is a tough proposition. Several researchers have used a log-transformation of the 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. 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 the state-space, and the curse of dimensionality (issue 4) remains the greatest challenge confronting its successful numerical solution. The curse of dimensionality can be roughly defined as an exponential growth in the size of a problem with the dimensionality of the system under consideration. In the current approach, if we use \( n \) nodes to discretize each of the \( N \) dimensions of the state-space and endow every node with a complete set of basis functions using polynomials up to \( p^{th} \) order, the number of coefficients to determine (or, the degrees of freedom of the approximation) is given by the following expression: (depicted by \( N \mathcal{D}_p^n \))

\[
N \mathcal{D}_p^n = n^N \times \sum_{k=0}^{p} \binom{N + k - 1}{N - 1}
\]  

(7)

Notice that increasing the number of nodes (\( n \)) or the order of basis functions per node (\( p \)) both cause an explosive growth in the problem size, \( N \mathcal{D}_p^n \), especially for large \( N \). Therefore, in the absence of local \( p \)-refinement, the current approach would also suffer from the curse of dimensionality. Local \( p \)-refinement (local \( p \)-enrichment) allows us to increase the order of basis functions at only selected nodes. In this situation, we count the basis functions for each node individually leading to an expression for the total number of degrees of freedom (DOFs) that is different from Eq. 7:

\[
N \mathcal{D}_{locp}^n = \sum_{i=1}^{N} \sum_{k=0}^{p_i} \binom{N + k - 1}{N - 1}
\]

(8)

The above equation shows that local \( p \)-refinement does not reduce the curse of dimensionality directly since the number of nodes still grows exponentially with \( N \). However, it invariably leads to a significant reduction in the number of nodes required because the approximability of the enriched nodes increases with the addition of high order polynomials. Due to the local nature of the enrichment it is not required to enrich all nodes equally, thus 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. 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 (PUFEM)

The partition of unity (PU) approach to finite elements was developed by Babuška and Melenk [27]. It is one of the several recently developed meshless algorithms [27, 32, 26, 33, 34, 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 \( \Omega \) and a set of overlapping subdomains \( \Omega_s, s = 1, 2, \ldots, P \), which form a cover for \( \Omega \). A “partition of unity” on \( \Omega \) is a mathematical paradigm in which each of the overlapping subdomains \( \Omega_s \) is associated with a compactly supported function \( \varphi_s \) called the PU pasting function, which is strictly zero outside \( \Omega_s \) and has the property that:

\[
\sum_s \varphi_s(x) = 1, \quad \forall x \in \Omega
\]  

(9)

The above paradigm represents the skeleton for a powerful meshless finite element method for solving PDEs on the domain \( \Omega \). By assigning each of the subdomains (\( \Omega_s \)) to individual nodes distributed over the global domain \( \Omega \), we obtain an implicit “discretization” of \( \Omega \), 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 the overlap among neighboring subdomains and the fact that there is minimal role of inter-node or inter-element connectivity in this formulation. 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 six dimensional state-space discretization. Also, the absence of inter-element boundaries is a great convenience in problems with moving boundaries, since no remeshing is required to maintain the inter-element boundary information.
Figure 2 illustrates the implicit PUFEM discretization of the 1-D domain $[-1, 1]$ using 5 nodes. The tent functions shown in Fig.2(a) are typically used as PU functions ($\varphi_s$). The PU functions and their supports for the first two nodes have been highlighted, and the overlap between elements $\Omega_1$ and $\Omega_2$ is clearly visible. In this paper, we employ weight functions used in the GLO-MAP algorithm[28] (see Fig.2(b)), which provide several advantages over the tent functions as PU functions. These functions are further discussed 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 $\hat{W}$ of the function $W$ can be constructed by setting up a conformal space of shape functions in $\Omega$. This is accomplished by introducing basis functions - $\psi_{sj}$ ($j = 1, \ldots, Q_s$), within each $\Omega_s$. 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 methods. In PUFEM, the basis functions $\psi_{sj}$ 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 ba-
sis 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 sub-
domains. 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 ap-
proximation is not ensured. This task is accomplished by the PU pasting
functions, which merge together the various local approximations. Figure 3
illustrates the process of shape function construction in the PUFEM algo-
rithm. In these figures, the basis functions (ψsj) have been drawn using bold
lines and the PU pasting functions (ϕs) 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 do-
main [−1, 1] is discretized using 5 subdomains with tent-functions in Fig.3(a)
and C1 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 high-
lighted Node # 3). Clearly, these basis functions do not form a conformal
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 3(d)):

$$\Psi_{sj} = \phi_s \psi_{sj}, \quad j = 1, \ldots, Q_s$$

(10)

In essence, the above approach delegates the burden of enforcing inter-
element continuity to the PU functions so that the user is free to select basis
functions purely on the criteria of local approximability. In comparison, in
the conventional FEM the basis functions are the same as the shape func-
tions. 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 non-polynomials. We mention that the order of continuity of the

12
shape functions across the local subdomains is inherited from the continuity of the PU functions [27]. Thus, the 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 the GLO-MAP weight with \( C^1 \) continuity leads to \( C^0 \) derivatives (Fig.3(f)), which is an advantage (see Sec.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:

\[
\tilde{\mathcal{W}}(x) = \sum_{s,j} a_{sj} \psi_{sj} = \sum_{s=1}^{P} \varphi_s(x) \sum_{j=1}^{Q_s} a_{sj} \psi_{sj}(x)
\]  

(11)

where, \( a_{sj} \) are the Fourier coefficients 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 \( \tilde{\mathcal{W}} \) given in Eq.11, we obtain the following local variational form for the N-D stationary FPE over the subdomain \( \Omega_s \) with boundary \( \Gamma_s \):

\[
\int_{\Omega_s} \mathcal{L}_{\text{FP}}(\tilde{\mathcal{W}})vd\Omega + \alpha \int_{\Gamma_s \cap \Gamma} \tilde{\mathcal{W}}vd\Gamma = 0
\]  

(12)

where \( v \) is a test function belonging to the space of functions \( \mathcal{V} \), onto which the residual is projected for error minimization, and \( \alpha \) is a penalty parameter used to enforce the boundary condition \( \mathcal{W}_T = 0 \). The test functions in the above variational equation have compact support on \( \Omega_s \). Notice that the integral is carried out over the local domain and the boundary condition is enforced over the part of the local boundary that coincides with the global
Figure 3: Shape function construction in PUFEM.
boundary. Substituting Eq.11 in Eq.12, we obtain the following relation:

\[
\sum_{s=1}^{P} \sum_{j=1}^{Q_s} \left[ \int_{\Omega_s} \mathcal{L}_{FP}(\varphi_s(x)a_{sj}\psi_{sj}(x))vd\Omega + \alpha \int_{\Gamma_s} \varphi_s(x)a_{sj}\psi_{sj}(x)v\Gamma \right] = 0
\]  

(13)

The coefficients \(a_{sj}\) in Eq.13 are the unknowns, which can be found by using sufficient number of distinct test functions in order to set up a determined system of equations. From the above equation, a total of \(\sum_{s=1}^{P} Q_s\) distinct test functions is required for this purpose. Following the Galerkin approach, we choose the test functions to be the same as the shape functions, i.e., \(\mathcal{V} = \{\varphi_s, \psi_{sj}\}\). (no summation of repeated index implied) This leads to the following system of linear algebraic equations involving the stiffness matrix \(K\) and load vector \(f\):

\[
Ka = f
\]  

(14)

\[
K_{ij} = \int_{\Omega_s} \mathcal{L}_{FP}(\varphi_k(x)\psi_{kl}(x))\varphi_p\psi_{pq}(x)d\Omega + \alpha \int_{\Gamma_s} \varphi_k(x)\psi_{kl}(x)\varphi_p\psi_{pq}(x)v\Gamma
\]

(15)

\[
f_j = 0
\]  

(16)

where, \(i = \left(\sum_{s=1}^{k-1} Q_s + l\right)\) and \(j = \left(\sum_{s=1}^{p-1} Q_s + q\right)\). Notice 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 the FPE is unique and globally asymptotically stable. However, this may not hold true for the numerical implementation shown above. We mention that if the parameter \(\alpha\) is chosen to be too large, it may cause a rank deficiency in \(K\) of greater than 1. In such event, one may study the equation error to determine the best solution. Alternatively, the penalty parameter \(\alpha\) can be tuned to obtain a 1-D null-space. Another approach is to implement the boundary conditions not as \(\mathcal{W}_T = 0\), but a very small value, e.g. \(\mathcal{W}_T = \epsilon (\approx 10^{-9})\), so that the RHS of Eq.14 is not zero. This approach gives highly acceptable results even with a very coarse tuning of \(\alpha\). The rank deficiency of \(K\) may also be caused due to other factors besides \(\alpha\), like the failure to incorporate the constraints (1) and (2) mentioned in Sec.2.2 in the numerical method. Of course, the matrix \(K\) will always be ill-conditioned if \(\alpha\) is not
chosen judiciously.
Notice that the solution to Eq.14 results in a functional approximation of the 
$\text{pdf} \; W$. 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 the PU pasting functions are 
of central importance in the PUFEM algorithm. They bring about the implicit domain discretization, merge together the various local approximations 
by performing an unbiased average, and determine their order of continuity 
across the local boundaries\cite{27}. Because of the requirement of the PU constraint, it is generally a difficult task to construct pasting functions that 
ensure 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: $\varphi_i(.) = \frac{\varphi_i(.)}{\sum_j \varphi_j(.)}$ (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 depending on the specified order of 
continuity of the approximation has not been explored to a great extent in 
the PUFEM literature hence far. In this section, we briefly discuss the weight 
functions derived in the recently developed GLO-MAP algorithm \cite{28, 29} as 
PU pasting functions. These functions are of polynomial form, satisfy Eq.9, 
and have compact support - thus satisfying the 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 the GLO-MAP weights is surprisingly simple - given a node 
belonging to a discretized domain, the polynomial function of the lowest degree, 
which assumes the value unity at the concerned node and decays to zero 
at all its neighboring nodes with the specified degree of smoothness satisfies 
the property of partition of unity on the global domain $\Omega$. It is possible to 
write a general expression for such polynomials to any order of continuity as
follows[28]:

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

(17)

where, \( m \) is the order of continuity. The tent-functions shown in Fig.2(a) are a special case with \( m = 0 \). 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 very 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 co-ordinates centered at the corresponding nodes and scaled with the inter-nodal distance along each dimension, \( h^{(i)} \). This implies that in the local co-ordinates, the central node is at the centroid of a \( N \)-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 (tensor 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., \( w_{(2)}(x,y) = w_{(2)}(x)w_{(2)}(y) \), 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 PUC 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 a \( N \)-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 2.2.2, 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 \( p = 8 \)) while vertical curves represent contours of fixed number of nodes used for discretization along each dimension.\((n = 5 \text{ to } n = 50)\) A grid is therefore formed and it is only possible to jump from one grid point to another. The \( y \)-axis shows the number of degrees of freedom associated with each \((n,p)\) case. Three points are highlighted: \( 4D_2^{15} = 759375 \), \( 4D_2^{16} = 983040 \) and \( 4D_3^{15} = 1771875 \).
Figure 5: Growth of DOF in Standard PUFEM Alongside PUFEM with Local $p$-Refinement.

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 800000 degrees of freedom because it is 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. $^{4}\mathcal{D}_{loc} \approx 800000$.

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 Two Dimensions

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

$$\ddot{x} + \eta \dot{x} + \alpha x + \beta x^3 = g(t)\mathcal{G}(t)$$  \hspace{1cm} (18)

Eq.18 represents a stochastic Duffing oscillator with a soft spring ($\alpha, \beta < 0, \eta > 0$) and included damping (to ensure the presence of a stationary solution, which is a bimodal pdf, shown in Fig. 6(a)). The expression for the
true solution to the stationary FPE for this system is as follows:

$$W_s(x, \dot{x}) |_{\text{true}} = C \exp \left( -\frac{2 \cdot \eta}{g^2 Q} \left( \frac{\alpha x^2}{2} + \frac{\beta x^4}{4} + \frac{\dot{x}^2}{2} \right) \right), \quad (19)$$

where $C$ 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 $-2 \cdot \frac{\eta}{g^2 Q} \ [1]$. For simulation purposes, we use $\alpha = -15$, $\beta = 30$, $\eta = 10$ along with $g = 1$. Following the discussion of rank deficiency of $K$ in Sec.3.3.3, the boundary condition was implemented as $W_f = c (= 10^{-5})$, resulting in a non-zero load vector. Fig.6 shows the solution and error surfaces obtained using the PUFEM algorithm on a $18 \times 18$ 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 $1944 \times 1944$. In other words, the size of the discretized problem (or the number of coefficients to determine) is $2D_2^{18} = 1944$. The results of this discretization are shown in Figs.6(b)-6(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 global method.

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 polynomials on a $18 \times 18$ grid is $2D_4^{18} = 4860$. On the other hand,
Table 1: Numerical Results Using PUFEM with Local \( p \)-Refinement: Two-State Duffing Oscillator

<table>
<thead>
<tr>
<th>Node Discretization</th>
<th>Polynomial Orders ((p))</th>
<th>Problem Size ((\text{DOF}))</th>
<th>RMS Error ((\epsilon_2))</th>
<th>Max Error ((\epsilon_{\infty}))</th>
<th>Computation Time ((t_{\text{comp}})) [1.86 GHz Pentium M]</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 × 18</td>
<td>Boundary Nodes: ( p = 2 ) \nInterior Nodes: ( p = 2 )</td>
<td>( ^2\mathcal{D}_2^{18} = 1944 )</td>
<td>(6.856 \times 10^{-3} )</td>
<td>(5.905 \times 10^{-2} )</td>
<td>30.1 s</td>
</tr>
<tr>
<td>9 × 9</td>
<td>Boundary Nodes: ( p = 1 ) \nInterior Nodes: ( p = 4 )</td>
<td>( ^2\mathcal{D}_{locp}^{9} = 831 )</td>
<td>(5.998 \times 10^{-3} )</td>
<td>(4.539 \times 10^{-2} )</td>
<td>10.7 s</td>
</tr>
</tbody>
</table>

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 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 \( \epsilon_2 = \sqrt{\frac{1}{r-1} \sum_{i=1}^{r} (W(x_i) - \tilde{W}(x_i))^2} \)) and maximum error \((\epsilon_{\infty})\). (see Figs. 6(c)&6(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 to 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 (MRMM) based on the Meshless Petrov Galerkin approach (MLPG). 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 Sec.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 Two Dimensions

Consider now the following 2-D quintic oscillator:

\[ \ddot{x} + \eta \dot{x} + x(\varepsilon_1 + \varepsilon_2 x^2 + \varepsilon_3 x^4) = g(t) G(t) \]  \hspace{1cm} (20)

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

\[ W_s(x, \dot{x}) \big|_{\text{true}} = C \exp \left( -2 \frac{\eta}{g^2 Q} \left( \frac{\varepsilon_1 x^2}{2} + \frac{\varepsilon_2 x^4}{4} + \frac{\varepsilon_3 x^6}{6} + \frac{\dot{x}^2}{2} \right) \right), \]  \hspace{1cm} (21)

The values of the various parameters used in this simulation are: \( \varepsilon_1 = 1, \varepsilon_2 = -3.065, \varepsilon_3 = 1.825, \eta = 1.5, g = 1 \). 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.

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 the 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 both \( h \)- and \( p \)-refinement reduce approximation error, but \( p \)-refinement is clearly superior for this purpose. The darker regions on this plot are the desirable regions of low error. Note that it is possible to move towards the darker zones by either of \( h \)- or \( p \)-refinement. However, the latter approach clearly requires a very few number of nodes per
(a) True Stationary pdf for the Duffing Oscillator with a Soft Spring

(b) Computed Solution: PUFEM Algorithm, $18 \times 18$ Grid with Quadratic Basis. DOFs = 1944

(c) Error Surface: PUFEM Algorithm, $18 \times 18$ Grid with Quadratic Basis.

(d) Error Surface: PUFEM Algorithm, $9 \times 9$ Grid with Linear and Quartic Basis. DOFs = 831

(e) Computed Solution (Global Method, Comparative Convergence Characteristic of the Reference pdf = $(0,0.0075)$) of PUFEM and MRMM for the Damped Duffing Oscillator

Figure 6: Numerical Results using the PUFEM Algorithm and Global Galerkin Approach.
(a) Computed Solution: PUFEM Algorithm, 18 x 18 Grid with Quadratic Basis.
(b) Error Surface: PUFEM Algorithm, 18 x 18 Grid with Quadratic Basis.
DOFs = 1944

(c) Convergence Characteristics for the Quintic Oscillator using PUFEM

Figure 7: Numerical Results using the PUFEM Algorithm for the Quintic Oscillator.
Table 2: Numerical Results Using PUFEM with Local p-Refinement: **Two-State Quintic Oscillator**

<table>
<thead>
<tr>
<th>Node Discretization</th>
<th>Polynomial Orders ((p))</th>
<th>Problem Size ((\text{DOF}))</th>
<th>RMS Error ((e_2))</th>
<th>Max Error ((e_\infty))</th>
<th>Computation Time ((t_{\text{comp}})) [1.86 GHz Pentium M]</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 × 18</td>
<td>Boundary Nodes: (p = 2) Interior Nodes: (p = 2)</td>
<td>(2D_2^{18} = 1944)</td>
<td>(6.121 \times 10^{-4})</td>
<td>(7.750 \times 10^{-3})</td>
<td>33.2 s</td>
</tr>
<tr>
<td>9 × 9</td>
<td>Boundary Nodes: (p = 1) Interior Nodes: (p = 4)</td>
<td>(2D_4^{9,\text{loc}} = 831)</td>
<td>(7.469 \times 10^{-4})</td>
<td>(6.127 \times 10^{-3})</td>
<td>12.9 s</td>
</tr>
</tbody>
</table>

dimension for achieving low approximation error. On the other hand, if one is constrained to work with 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 \(D^*\)-contour, the size of the discretized problem remains less than \(D^*\). This is very useful information, because now looking at the composite contour-map in Fig.8, we see that \(p\)-refinement provides not only lower error, it also helps keep the problem size down. 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. Note that 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. the 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, the current method 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 with dimensionality under tight check. Besides the systems discussed in this section, the discussed 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 the 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.

4.3. Dynamic System 3: Example in Three Dimensions

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

\[
\dot{x} = \begin{bmatrix}
0 & 1 & 0 \\
-\omega_0 & -2\zeta_0 & 1 \\
0 & 0 & -\alpha
\end{bmatrix} x + \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} w(t) 
\tag{22}
\]
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 the stationary distribution for the above system 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. The approximation accuracy for both methods are approximately the same, while the current method gives 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 \( 3D^7_4 = 12005 \). 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 Three Dimensions

Consider now the following nonlinear system with three dimensional state space:

\[
\begin{align*}
\dot{x} &= \sigma(y - x) + \zeta_1(t) \\
\dot{y} &= x(\rho - z) - y + \zeta_2(t) \\
\dot{z} &= xy - \beta z + \zeta_3(t)
\end{align*}
\]

(23)

The above represents a noise-driven Lorenz attractor. Numerical values for the various parameters appearing above are: \(\sigma = 10, \rho = 1; \beta = 8/3\) and \(Q\) (noise intensity) = 2. Figures 10(a) and 10(b) show the \(xy\)-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 the boundary nodes endowed with quadratic polynomials and interior nodes with quartic polynomials. (corresponding problem size = 3760 DOFs) Analytical result is not available for this system, therefore the error was measured in terms of equation-error and is shown in Fig. 10(c). In this figure, the equation error has been integrated along the \(z\)-direction to obtain a 2D surface. The shown error surface has an RMS value of \(6.668 \times 10^{-6}\).
(a) Computed \( x-y \) Marginal for the Lorenz

(b) Contour Plot of the Computed \( x-y \) Marginal Surface.

(c) Equation Error Surface Integrated Along the \( z \)-Axis.

Figure 10: Computed \( x_1 - x_2 \) Marginal Distribution for the Noise-Driven Lorenz Attractor of Eq. 23.
4.5. Dynamic System 5: Example in Four Dimensions

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

\[
\dot{x} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-(k_1 + k_2) & -c_2 & k_2 & 0 \\
0 & 0 & 0 & 1 \\
k_2 & 0 & -(k_2 + k_3) & -c_2 \\
\end{bmatrix} \times + \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\end{bmatrix} \zeta(t) \quad (24)
\]

The Fokker Planck equation of concern is:

\[
\frac{\partial p}{\partial t} = -x_2 \frac{\partial p}{\partial x_1} - \frac{\partial}{\partial x_2} [(-(k_1 + k_2)x_1 - c_1x_2 + k_2x_3)p] - x_4 \frac{\partial p}{\partial x_3} \\
- \frac{\partial}{\partial x_3} [(k_2x_1 - (k_2 + k_3)x_3 - c_2x_4)p] + D_1 \frac{\partial^2 p}{\partial x_2^2} + D_2 \frac{\partial^2 p}{\partial x_4^2} \quad (25)
\]

The constants 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 computed \((x_1 - x_2)\) marginal distribution with the meshless PUFEM method alongside the true marginal surface for the above linear system. The available computing resources allowed the use of 5 nodes along each of the 4 dimensions, with all interior nodes carrying cubic basis functions and boundary nodes linear functions. This leads to a total of 5555 DOFs, which is in sharp contrast to the 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 \(4D_3^5 = 21875\), 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.
Table 4: Comparative Results Using PUFEM with Local $p$-Refinement: **Four-State Linear System**

<table>
<thead>
<tr>
<th>Discretization Method</th>
<th>Problem Size (DOF)</th>
<th>RMS Error ($e_2$)</th>
<th>Max Error ($e_\infty$)</th>
<th>Computation Time ($t_{comp}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4D$ Brick elements (FEM)</td>
<td>256000</td>
<td>$5.237 \times 10^{-5}$</td>
<td>$2.911 \times 10^{-3}$</td>
<td>not available</td>
</tr>
<tr>
<td>$5 \times 5 \times 5 \times 5$ Nodes</td>
<td>$4^D \mathcal{D}_{loc,p}^p = 5555$</td>
<td>$9.769 \times 10^{-4}$</td>
<td>$8.870 \times 10^{-2}$</td>
<td>23 hr 36.4 min [1.86 GHz Pentium M]</td>
</tr>
</tbody>
</table>

Figure 11: Comparison of the Computed ($x_1 - x_2$) Marginal for the Four Dimensional Linear System with the Truth.
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 the 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 the curtailment of the curse has been presented. To further support this evidence, we look at Fig. 12 and Table 4.6. In Fig. 12, several problems residing in dimensions 1 – 4 have been solved with fixed accuracy ($O(e_2) \approx 10^{-4}$) using the current meshless algorithm and the required DOFs is compared with the finite element technique. Note that the 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 (4 data points) is available. For FEM, it is well known (also seen in Fig.12) that an exponential fit best describes the problem 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 4.6 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 point-wise 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 (GM), the conventional FEM, and the other meshless methods like the one used in used in Kumar et al.[25] (MRMM).  

32
Table 5: Growth in Problem Size with System Dimensionality: FEM and PUFEM

<table>
<thead>
<tr>
<th>$N$</th>
<th>Problem Size (DOF)</th>
<th>Feasible on a small computer?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FEM</td>
<td>PUFEM</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>2500</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>1,250,000</td>
<td>5200</td>
</tr>
<tr>
<td>4</td>
<td>2,560,000</td>
<td>5555</td>
</tr>
</tbody>
</table>

Figure 12: Ameliorating the *Curse of Dimensionality*
• **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 non-polynomial 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 a 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 its 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


[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.


