Time-Dependent Generalized Polynomial Chaos

Applied to the Kraichnan-Orszag Three-Mode Problem

Master Thesis

J. van der Steen

December 19, 2008

Supervisor: Dr. Ir. M.I. Gerritsma

Aerodynamics Group, Faculty of Aerospace Engineering
Delft University of Technology

Contents

Chapter 1

Introduction

To describe physical problems we often make use of deterministic mathematical models. Typical constituents of such models – material properties, initial and boundary conditions, interaction and source terms, etc. – are assigned a definite value and we seek a definite solution to the problem. In reality, however, a physical problem will almost always have uncertain components. Material properties, for instance, might be based on experimental data. In other words, the input to a mathematical model of a real-life problem possesses some degree of randomness. We are interested in modelling this uncertainty. To this end we look for methods to quantify the effects of stochastic inputs on the solutions of mathematical models.

The Monte Carlo method is, arguably, both the most popular as well as the least elegant approach to model uncertainty. It is a 'brute-force' method of attack: using a sample of the stochastic inputs we calculate the corresponding realizations of the solution. From the resulting sample of solutions we then determine the desired statistical properties of the solution. In most cases we have to use a large sample size to obtain accurate estimates of these statistical properties. This makes Monte Carlo methods very expensive from a computational point of view. Furthermore, the selection of proper (pseudo-)random number generators needed for a Monte Carlo simulation can be a delicate matter.

Besides the statistical Monte Carlo methods a number of nonstatistical (i.e. deterministic) approaches to modelling uncertainty have been proposed. *Polynomial chaos* is one such nonstatistical method that has been shown to be particularly effective for a number of problems. Polynomial chaos employs orthogonal polynomial functionals to expand the solution spectrally in the random space. The method is based on Wiener's [1] homogeneous chaos theory published in 1938. This publication paved the path for the application of truncated expansions in terms of Hermite polynomials of Gaussianly distributed random variables to model (near-)Gaussian stochastic processes. In the 1960s these Wiener-Hermite expansions were employed in the context of turbulence modelling [2], [3]. Some serious limitations were encountered leading to a decrease of interest in the method in the years that followed.

In 1991 Ghanem and Spanos [4] pioneered the use of Wiener-Hermite expansions in combination with finite element methods and effectively modelled uncertainty for various problems encountered in solid mechanics. At this point

in time the polynomial chaos method was capable of achieving an exponential convergence rate for Gaussian stochastic processes only. In 2002 Xiu and Karniadakis [5] made another big step forward with the introduction of *generalized polynomial chaos* (gPC). It was recognized that the PDF of a number of common random distributions is identical to the weighting function in the orthogonality relationship of certain orthogonal polynomials from the so-called Askey scheme. Xiu and Karniadakis established that, in order to achieve optimal convergence, the type of orthogonal polynomials in the chaos expansion should correspond to the properties of the stochastic process at hand, according to the connecting between PDF and weighting function. This gPC approach has, for instance, been applied to a number of problems in fluid flow [6]. Although the polynomial chaos method was initially generalized to polynomials of the Askey scheme (and the corresponding 'standard' random distributions) only, the extension to arbitrary random distributions (and with that a true generalization of the method) was soon to follow. Just by observing the correspondence between PDF and weighting function in the orthogonality relation we can generate optimal expansion polynomials for an arbitrary random distribution. The resulting expansion polynomials need not necessarily come from the Askey scheme. There exist various ways to calculate these optimal expansion polynomials. Applying a Gram-Schmidt orthogonalization technique is one possibility [7], but there are others options as well [8].

The gPC method has been shown to be effective for a number of problems resulting in exponential convergence of the solution. However, there are also situations in which gPC performs unsatisfactory. A discontinuity of the solution in the random space may, for instance, lead to slow convergence or no convergence at all. In addition, problems may be encountered with long-term integration. The statistical properties of the solution will most likely change with time. This means that the particular orthogonal polynomial basis that leads to exponential convergence for earlier times may loose its effectiveness for later times resulting in a deteriorating convergence behaviour with time. Hence, for larger times unacceptable error levels may develop. These errors may become practically insensitive to an increase of the order of the polynomial expansion. To overcome the aforementioned problems with gPC Wan and Karniadakis [9] have developed a multi-element polynomial chaos method (ME-gPC). The main idea of ME-gPC is to adaptively decompose the space of random inputs into multiple elements and subsequently employ polynomial chaos expansions on element level.

Vos [10] has taken a different approach to attack the long-term integration problems with gPC. Recognizing that the initial polynomial chaos expansion loses its optimal convergence behaviour for later times he developed a *time-dependent* polynomial chaos (TDgPC) method. The main idea of TDgPC is to determine new, optimal polynomials for the chaos expansion at a number of discrete instants in time. These new polynomials are based on the stochastic properties of the solution *at the particular time level*. In this way a (near-)optimal convergence behaviour is retained over the complete time interval. Vos successfully applied his TDgPC approach to a simple stochastic ordinary differential equation. Thus far the extension to more complex problems has not yet been made.

In this work we will apply the TDgPC method to the Kraichnan-Orszag three-mode problem. The Kraichnan-Orszag problem is described by a *system*

of three coupled ordinary differential equation with stochastic initial conditions. Obviously, this problem is more complicated than the single differential equation on which TDgPC has been tested by Vos. Hence, we will see that we need to refine and develop Vos's original TDgPC method a little bit in order to successfully apply it to the Kraichnan-Orszag problem.

This thesis is divided into two parts. Part I, the main part, is devoted to a TDgPC solution of the Kraichnan-Orszag problem, while Part II revisits the model problem used by Vos.

First off, Chapter 2 concisely describes the original gPC method. Chapter 3 introduces the Kraichnan-Orszag problem and some of its properties. In Chapter 4 we present results of the original gPC method applied to the Kraichnan-Orszag problem. We will see that for some cases these gPC results are very unsatisfactory. Chapter 5 introduces Vos's time-dependent approach. In this chapter we also discuss a first attempt to apply TDgPC to the Kraichnan-Orszag problem. We will see that our initial approach is flawed. Some further modifications of Vos's TDgPC method are necessary. These are discussed in Chapter 6 along with the now much improved results of our TDgPC approach to the Kraichnan-Orszag problem. Finally, in Chapter 7 we look at TDgPC results for a three-dimensional instead of a one-dimensional random input to the problem.

We have already mentioned that Vos's original TDgPC method was refined at some points in order to make the method applicable to the three coupled differential equations of the Kraichnan-Orszag problem. In Part II of this thesis we revisit the single differential equation test problem employed by Vos. We solve this problem using the modified TDgPC method. With our refinements we expect to achieve the same accuracy as obtained by Vos, but at a substantially lower computational cost. Chapter 8 introduces Vos's test problem: a simple growth equation. In Chapter 9 we discuss the TDgPC algorithm for this problem. In Chapter 10 results are presented.

Finally, Chapter 11 presents conclusions and a number of suggestions for further research. This work has one appendix in which a collection of additional figures can be found. As a final note, we remark that the work documented in this thesis was presented by the author at the 2008 International Conference on Numerical Analysis and Applied Mathematics (ICNAAM) [11].

Chapter 2

Polynomial Chaos

A stochastic process can in many cases be expanded in terms of orthogonal polynomials of random variables. We say that the stochastic process is represented by a polynomial chaos expansion. These polynomial chaos expansions can be used to solve stochastic problems, i.e. problems involving some kind of randomness. In this chapter we introduce the polynomial chaos expansion and we outline a solution method for stochastic problems based on these expansions.

2.1 The polynomial chaos expansion

Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space. Here Ω is the sample space, $\mathcal{F} \subset 2^{\Omega}$ its σ-algebra of events and $\mathcal{P}$ the associated probability measure. In addition, let $S \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ and $T \subset \mathbb{R}$ be certain spatial and temporal domains, respectively. In a physical context we frequently encounter stochastic processes in the form of a scalar- or vector-valued random function like

$$\boldsymbol{u}(\boldsymbol{x}, t, \omega) : S \times T \times \Omega \to \mathbb{R}^b \tag{2.1}$$

where $\boldsymbol{x}$ denotes position, t stands for time and ω represents an element of the sample space Ω. The probability space can oftentimes be described by a finite number of random variables

$$\xi_1, \xi_2, \ldots, \xi_n : \Omega \to \mathbb{R} \tag{2.2}$$

In this case the stochastic process of Eq. (2.1) can equivalently be written as

$$\boldsymbol{u}(\boldsymbol{x}, t, \boldsymbol{\xi}) : S \times T \times \mathbb{R}^n \to \mathbb{R}^b \tag{2.3}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$ is an n-dimensional vector of random variables. In this work we will exclusively be dealing with stochastic processes of the form of Eq. (2.3), i.e. processes that can be characterized by a finite set of random variables.

The stochastic process of Eq. (2.3) can be represented by the following *polynomial chaos expansion*

$$\boldsymbol{u}(\boldsymbol{x}, t, \boldsymbol{\xi}(\omega)) = \sum_{i=0}^{\infty} \boldsymbol{u}_i(\boldsymbol{x}, t) \Phi_i(\boldsymbol{\xi}(\omega)) \tag{2.4}$$

where the trial basis $\{\Phi_i(\boldsymbol{\xi})\}$ consists of orthogonal polynomials in terms of the random vector $\boldsymbol{\xi}$. So, employing deterministic coefficients, the random function $\boldsymbol{u}(\boldsymbol{x}, t, \boldsymbol{\xi}(\omega))$ is expanded spectrally in the random dimensions. We will see below that the stochastic characteristics of the process at hand determine which *type* of orthogonal polynomials can best be employed as a trial basis.

Historically, Wiener [1] first formulated a polynomial chaos expansion in terms of Hermite polynomials of Gaussianly distributed random variables. It follows from a theorem by Cameron and Martin [12] that this Hermite-chaos expansion converges to any stochastic process $\boldsymbol{u}(\omega) \in L_2(\Omega, \mathcal{F}, \mathcal{P})$ in the L_2 sense. This means that a Hermite-chaos expansion can – in principle – be used to represent any stochastic process with finite variance (a requirement that is met for most physical processes). In practice, however, optimal convergence is limited to processes with Gaussian inputs. This can easily be understood: Gaussian random inputs generally result in a stochastic process that has a large Gaussian part, at least for early times. This Gaussian part is represented by the first order terms in the Hermite-chaos expansion. Higher order terms can be thought of as (relatively small) non-Gaussian corrections. Hence, for Gaussian random inputs we can expect a Hermite-chaos expansion to converge rapidly.

For general, non-Gaussian random inputs, however, the rate of convergence of a Hermite-chaos expansion will most likely be substantially worse. Although convergence is ensured by the Cameron-Martin theorem, we will generally need a large number of higher-order terms in the expansion to account for the now far more dominant non-Gaussian part. To obtain an optimal rate of convergence in case of general random inputs we need to tailor the expansion polynomials to the stochastic properties of the process under consideration. Although Ogura [13] had already employed Charlier-chaos expansions to describe Poisson processes, Xiu and Karniadakis [5] were the first to present a comprehensive framework to determine the optimal trial basis $\{\Phi_i\}$.

The *optimal* set of expansion polynomials forms a complete orthogonal basis in $L_2(\Omega, \mathcal{F}, \mathcal{P})$ with orthogonality relation

$$\langle \Phi_i, \Phi_j \rangle = \left\langle \Phi_i^2 \right\rangle \delta_{ij} \tag{2.5}$$

where δ_{ij} is the Kronecker delta and $\langle \ldots \rangle$ denotes the ensemble average. To be more specific, the optimal set $\{\Phi_i(\boldsymbol{\xi})\}$ is an orthogonal basis in the Hilbert space with associated inner product

$$\begin{aligned} &\langle G\left(\boldsymbol{\xi}(\omega)\right), H\left(\boldsymbol{\xi}(\omega)\right)\rangle = \\ &\qquad \int_\Omega G\left(\boldsymbol{\xi}(\omega)\right) H\left(\boldsymbol{\xi}(\omega)\right) \mathrm{d}\mathcal{P}(\omega) = \int_{\mathrm{supp}(\boldsymbol{\xi})} G(\boldsymbol{\xi}) H(\boldsymbol{\xi}) f_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \end{aligned} \tag{2.6}$$

where $f_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ is the probability density function (PDF) of the random variables that make up the vector $\boldsymbol{\xi}$. Note that the PDF acts as a weighting function in the orthogonality relation for $\{\Phi_i(\boldsymbol{\xi})\}$. So, the *type* of orthogonal expansion polynomials (determined by the weighting function in the orthogonality relation) that can best be used in a polynomial chaos expansion depends on the nature of the stochastic process at hand through the PDF of the random variables that describe the probability space. The fact that the trial basis defined in Eqs. (2.5) and (2.6) is optimal hinges on the presumption that the random function $\boldsymbol{u}(\boldsymbol{x}, t, \boldsymbol{\xi}(\omega))$ represented by the polynomial chaos expansion has roughly

Distribution of ξ	Expansion polynomials $\{\Phi_i(\xi)\}$
Uniform	Legendre
Gaussian	Hermite
Beta	Jacobi
Gamma	Laguerre

Table 2.1: Orthogonal polynomials from the Askey scheme constitute an optimal trial basis for a number of well-known random distributions

the same stochastic characteristics as the random variables in $\boldsymbol{\xi}$, at least for early times. Hence, the higher order terms in the expansion are expected to be (negligibly?) small, reducing the dimensionality of the problem and resulting in rapid convergence. As a generalization of the Cameron-Martin theorem, we also expect this generalized polynomial chaos expansion (with $\{\Phi_i(\boldsymbol{\xi})\}$ being a complete basis) to converge to any stochastic process $\boldsymbol{u}(\omega) \in L_2(\Omega, \mathcal{F}, \mathcal{P})$ in the L_2 sense.

Xiu et al. [5] recognized that the weighting functions associated with a number of orthogonal polynomials from the so-called Askey scheme are identical to the PDFs of certain 'standard' random distributions. Table 2.1 gives some examples. Xiu et al. studied a simple test problem subject to different random inputs with 'standard' distributions like the ones in Table 2.1. Exponential error convergence was obtained for a polynomial chaos expansion with an optimal trial basis (i.e. in accordance with Table 2.1 or, equivalently, Eqs. (2.5) and (2.6)). Furthermore, it was shown that exponential convergence is generally not retained when the optimal trial basis is *not* used (for example, employing Hermite chaos instead of Jacobi chaos when the random input has a beta distribution).

Xiu et al. focused on orthogonal polynomials from the Askey scheme and corresponding 'standard' random distributions. However, there is no reason to limit the members of possible trial bases to polynomials from the Askey scheme. With Eqs. (2.5) and (2.6) we can determine an optimal trial basis for arbitrary, 'nonstandard' distributions of $\boldsymbol{\xi}$ as well. When the PDF of $\boldsymbol{\xi}$ is known we can use various orthogonalization techniques to calculate the corresponding optimal trial basis $\{\Phi_i(\boldsymbol{\xi})\}$. In this work we will use Gram-Schmidt orthogonalization. We will come back to this technique in more detail in Section 6.3.

Sometimes the probability space can be characterized by a single random variable, i.e. $n = 1$ in Eq. (2.2) and the vector $\boldsymbol{\xi}$ is reduced to the scalar ξ. In this case the index i in $\{\Phi_i(\xi)\}$ directly corresponds with the degree of the particular expansion polynomial. For example, $\Phi_3(\xi)$ is a third degree polynomial in ξ.

In the more general situation of a multidimensional probability space ($n > 1$ in Eq. (2.2)) the correspondence between i and polynomial degree does not exist and i reduces to merely a counter. To construct the multidimensional expansion polynomials $\{\Phi_i(\boldsymbol{\xi})\}$ we first calculate the one-dimensional polynomials $\phi_p^{(\xi_j)}(\xi_j)$ for $j = 1, \ldots, n$ and $p = 0, 1, 2, \ldots$ using a Gram-Schmidt algorithm

with orthogonality relation

$$\left\langle \phi_p^{(\xi_j)}, \phi_q^{(\xi_j)} \right\rangle = \int_{\mathrm{supp}(\xi_j)} \phi_p^{(\xi_j)}(\xi_j)\, \phi_q^{(\xi_j)}(\xi_j)\, f_{\xi_j}(\xi_j) \mathrm{d}\xi_j = \left\langle {\phi_p^{(\xi_j)}}^2 \right\rangle \delta_{pq} \tag{2.7}$$

For these one-dimensional polynomials p again corresponds to the polynomial degree and the superscript (ξ_j) indicates that the polynomial is orthogonal with respect to f_{ξ_j}. The *multidimensional* expansion polynomials can now be constructed from the simple tensor product

$$\Phi_i(\boldsymbol{\xi}) = \phi_{p_1}^{(\xi_1)}(\xi_1)\, \phi_{p_2}^{(\xi_2)}(\xi_2) \dots \phi_{p_n}^{(\xi_n)}(\xi_n) \tag{2.8}$$

with some mapping $(p_1, p_2, \dots, p_n) \to i$.

The procedure above assumes that $\xi_1, \dots, \xi_n$ are statistically independent which implies that

$$f_{\boldsymbol{\xi}}(\boldsymbol{\xi}) = f_{\xi_1}(\xi_1) f_{\xi_2}(\xi_2) \dots f_{\xi_n}(\xi_n) \tag{2.9}$$

It can now easily be verified that the multidimensional expansion polynomials $\{\Phi_i(\boldsymbol{\xi})\}$ constructed according to Eq. (2.8) form an optimal orthogonal trial basis in agreement with Eqs. (2.5) and (2.6).

2.2 A gPC solution approach

In this section we outline a solution procedure for stochastic problems based on the polynomial chaos expansion defined in Eq. (2.4). Consider the abstract problem

$$\mathcal{L}(\boldsymbol{x}, t, \boldsymbol{\xi}(\omega); \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}, t, \boldsymbol{\xi}(\omega)) \tag{2.10}$$

where $\mathcal{L}$ is a (not necessarily linear) differential operator and $\boldsymbol{f}$ some source function. The randomness, represented by the random vector $\boldsymbol{\xi}$, can enter the problem either through $\mathcal{L}$ (e.g. random coefficients) or $\boldsymbol{f}$, but also through the boundary or initial conditions or some combination of these possibilities.

We approximate the stochastic solution function $\boldsymbol{u}(\boldsymbol{x}, t, \boldsymbol{\xi}(\omega))$ by a truncated polynomial chaos expansion similar to Eq. (2.4), naturally using the optimal expansion basis for the particular problem at hand. The truncation of the infinite series is of course necessary to keep the problem computationally feasible. In this work we will truncate the series in such a way that all expansion polynomials up to a certain maximum degree, denoted by P, are included. The number of terms $(N+1)$ in the expansion now follows from this maximum degree P and the dimensionality n of the random vector $\boldsymbol{\xi}$ according to

$$N + 1 = \binom{P+n}{P} = \frac{(P+n)!}{P!n!} \tag{2.11}$$

Note that the total number of expansion coefficients equals the number of possible combinations for P picks out of a set with $n+1$ elements when each of these elements can be chosen more than once, i.e. the number of *combinations with repetitions* (see, for instance, [14] for details).

We continue by substituting the polynomial chaos expansion for $\boldsymbol{u}$ into the problem equation and execute a Galerkin projection. This means that we subsequently multiply Eq. (2.10) by every polynomial of the expansion basis $\{\Phi_i\}$

and take the ensemble average to obtain

$$\left\langle \mathcal{L}\left(\boldsymbol{x}, t, \boldsymbol{\xi}; \sum_{i=0}^{N} \boldsymbol{u}_i(\boldsymbol{x}, t)\Phi_i(\boldsymbol{\xi})\right), \Phi_j(\boldsymbol{\xi})\right\rangle = \langle f(\boldsymbol{x}, t, \boldsymbol{\xi}), \Phi_j(\boldsymbol{\xi})\rangle \quad , \quad j = 0, 1, \ldots, N \tag{2.12}$$

The Galerkin projection above ensures that the error we make by representing $\boldsymbol{u}$ by a polynomial chaos expansion is orthogonal to the function space spanned by the expansion basis $\{\Phi_i\}$. As a result of the orthogonality of the expansion polynomials, Eq. (2.12) can be reduced to a set of $N + 1$ coupled, *deterministic* equations for the $N + 1$ expansion coefficients $\boldsymbol{u}_i(\boldsymbol{x}, t)$. So, the remaining problem is stripped of all stochastic quantities by the Galerkin projection. The remaining equations can now be solved by a choice of conventional techniques, e.g. a spectral/hp element method in space and Runge-Kutta integration in time.

Part I

The Kraichnan-Orszag three-mode problem

Chapter 3

The Kraichnan-Orszag three-mode problem

As an introduction to more complicated numerical turbulence models, in 1963 Kraichnan [2] studied a system of three interacting simple shear waves. By simplifying the Fourier representation of the inviscid Navier-Stokes equations for an infinite, incompressible fluid he obtained a set of coupled, non-linear differential equations. Kraichnan solved this system numerically for large Gaussian samples of initial conditions.

In 1967, also in the context of turbulence modelling, Orszag [3] studied this same system of differential equations with Gaussianly distributed initial conditions as a simple model problem. He concluded that a truncated Wiener-Hermite expansion with a given fixed random basis is unsuitable to represent the time evolution of the stochastic solution of the system.

The system of differential equations is now generally referred to as the Kraichnan-Orszag three-mode problem. We give its definition in the next section. We continue with the discussion of some general properties of the problem in Section 3.2. Thereafter, in Section 3.3, we try to find an analytic solution to the Kraichnan-Orszag problem. Finally, we use the result from Section 3.3 to make some predictions about the effectiveness of a generalized polynomial chaos solution approach to the Kraichnan-Orszag problem in Section 3.4.

3.1 Problem Definition

The Kraichnan-Orszag problem is defined by the following system of nonlinear ordinary differential equations

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_2 x_3 \tag{3.1a}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = x_3 x_1 \tag{3.1b}$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = -2x_1 x_2 \tag{3.1c}$$

In this work we will consider this problem subject to stochastic initial conditions. We will mainly be studying initial conditions of the form

$$x_1(0) = \alpha + 0.01\xi\ , \quad x_2(0) = 1.0\ , \quad x_3(0) = 1.0 \tag{3.2}$$

where α is a constant and ξ a *uniformly* distributed random variable with unit variance (i.e. ξ is uniformly distributed on the interval $[-1, 1]$). In Chapter 7 we will look at more complex initial conditions defined on a three-dimensional random space.

3.2 Properties

Multiplying Eq. (3.1a) by x_1, Eq. (3.1b) by x_2 and Eq. (3.1c) by x_3 and adding the three resulting equations together we get

$$\sum_{i=1}^{3} x_i \frac{\mathrm{d}x_i}{\mathrm{d}t} = x_1x_2x_3 + x_1x_2x_3 - 2x_1x_2x_3 = 0 \tag{3.3}$$

which means that

$$\sum_{i=1}^{3} \frac{\mathrm{d}x_i^2}{\mathrm{d}t} = 0 \tag{3.4}$$

When dealing with stochastic initial conditions it follows that

$$\sum_{i=1}^{3} \frac{\mathrm{d}\left\langle x_i \right\rangle^2}{\mathrm{d}t} = 0 \tag{3.5}$$

must also be valid. Here $\langle \ldots \rangle$ denotes the ensemble average. So, we have found that the system (3.1) conserves $\sum_{i=1}^{3} \left\langle x_i \right\rangle^2$.

For *deterministic* initial conditions the solution to Eqs. (3.1) is generally periodic. Based on the initial conditions the solutions can be divided into four different branches separated by the planes $x_1 = x_2$ and $x_1 = -x_2$. In Fig. 3.1 we have plotted a particular solution curve for each of these four branches.

Only for initial conditions lying on either one of the planes $x_1 = x_2$ and $x_1 = -x_2$ the solution is aperiodic. For in that case Eqs. (3.1) become

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \pm x_1x_3 \tag{3.6a}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = x_1x_3 \tag{3.6b}$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t} = \mp 2x_1^2 \tag{3.6c}$$

From these equations we see that for initial conditions on the plane $x_1 = \pm x_2$ the solution stays on this same plane. Making use of the conservation property (Eq. (3.4)) we find from Eqs. (3.6) that solutions with initial conditions on the plane $x_1 = x_2$ are attracted to the fixed point $\left(0, 0, -\sqrt{2x_1^2(0) + x_3^2(0)}\right)$ while solutions with initial conditions on the plane $x_1 = -x_2$ are attracted to the fixed point $\left(0, 0, \sqrt{2x_1^2(0) + x_3^2(0)}\right)$.

A final case are initial conditions on the line $x_1 = x_2 = 0$. We see from Eqs. (3.1) that the solution remains at the point $(0, 0, x_3(0))$ for this case.

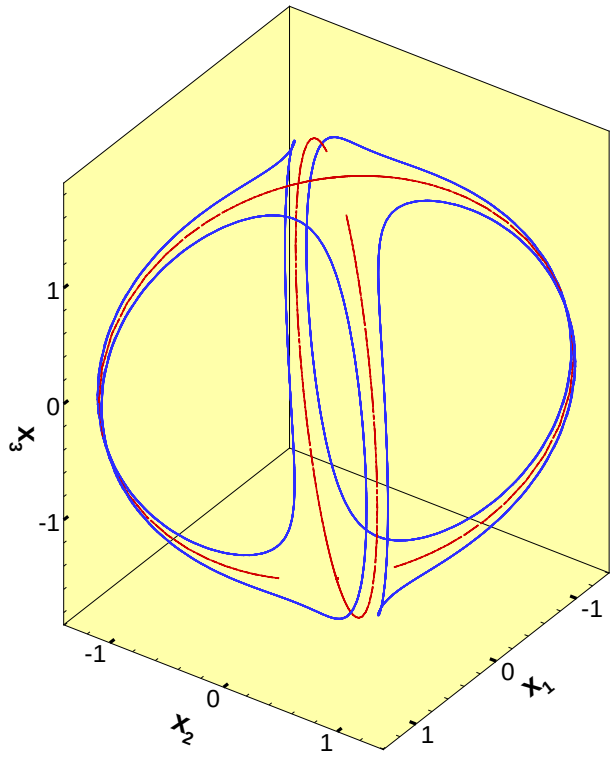

Figure 3.1: Four branches of deterministic periodic solutions to the Kraichnan-Orszag problem

3.3 Exact solution

Finding an analytic solution to Eqs. (3.1) is generally not straightforward since it involves the theory of elliptic functions [15]. The analytic solution can be obtained in terms of Jacobi's elliptic functions. These functions are defined as infinite summations and are therefore not easily evaluated. Use is often made of tables. Wan [9] derives the analytic form of one of the four branches of solution curves, namely those curves lying in the region

$$\{(x_1, x_2, x_3) : \; x_2 > 0; \; -x_2 < x_1 < x_2\} \tag{3.7}$$

Since this analytic solution helps to understand the performance of a gPC solution approach to the Kraichnan-Orszag problem, we will follow the argument of Wan and derive this analytic form below.

Let us start off by discussing some properties of Jacobi's elliptic functions $\operatorname{sn} u$, $\operatorname{cn} u$ and $\operatorname{dn} u$. Jacobi's elliptic functions are periodic. All three functions depend on the parameter k, the modulus of elliptic functions, and are therefore sometimes written as $\operatorname{sn}(u,k)$, $\operatorname{cn}(u,k)$ and $\operatorname{dn}(u,k)$. The periods of Jacobi's elliptic functions are a function of k. The real period of $\operatorname{dn} u$ is one half of the real periods of $\operatorname{sn} u$ and $\operatorname{cn} u$. Also, $\operatorname{sn} u$ and $\operatorname{cn} u$ oscillate around zero while $\operatorname{dn} u$ does not.

The derivatives of Jacobi's elliptic functions are given by

$$\frac{\mathrm{d}}{\mathrm{d}u} \operatorname{sn} u = \operatorname{cn} u \operatorname{dn} u \tag{3.8a}$$

$$\frac{\mathrm{d}}{\mathrm{d}u} \operatorname{cn} u = -\operatorname{sn} u \operatorname{dn} u \tag{3.8b}$$

$$\frac{\mathrm{d}}{\mathrm{d}u} \operatorname{dn} u = -k^2 \operatorname{sn} u \operatorname{cn} u \tag{3.8c}$$

Looking in Fig. 3.1 at the particular branch of solution curves under consideration we see that $x_2 > 0$ for all t. Also, the period of x_2 is one half of the periods of x_1 and x_3. Hence, we look for a solution to Eqs. (3.1) in the following form

$$x_1 = P \operatorname{cn}\left[\omega\,(t - t_0)\right] \tag{3.9a}$$

$$x_2 = Q \operatorname{dn}\left[\omega\,(t - t_0)\right] \tag{3.9b}$$

$$x_3 = -R \operatorname{sn}\left[\omega\,(t - t_0)\right] \tag{3.9c}$$

Substituting these equations into Eqs. (3.1) we obtain the following three relations

$$P\omega = QR \tag{3.10a}$$

$$k^2 Q\omega = PR \tag{3.10b}$$

$$R\omega = 2PQ \tag{3.10c}$$

The requirement to satisfy the initial conditions provides us with three additional equations

$$x_1(0) = P \operatorname{cn}\left(-\omega t_0\right) \tag{3.11a}$$

$$x_2(0) = Q \operatorname{dn}\left(-\omega t_0\right) \tag{3.11b}$$

$$x_3(0) = -R \operatorname{sn}\left(-\omega t_0\right) \tag{3.11c}$$

We now have 6 equations for the six unknowns P, Q, R, ω, t_0 and k. Eqs. (3.10) and (3.11) can be solved for these six unknowns using the identities

$$\mathrm{sn}^2\, u + \mathrm{cn}^2\, u = 1 \tag{3.12a}$$

$$\mathrm{dn}^2\, u + k^2\, \mathrm{sn}^2\, u = 1 \tag{3.12b}$$

An intermediate result is

$$P^2 = x_1^2(0) + \frac{1}{2} x_3^2(0) \tag{3.13a}$$

$$Q^2 = x_2^2(0) + \frac{1}{2} x_3^2(0) \tag{3.13b}$$

$$R^2 = 2x_1^2(0) + x_3^2(0) \tag{3.13c}$$

$$\omega^2 = 2x_2^2(0) + x_3^2(0) \tag{3.13d}$$

$$k^2 = \frac{2x_1^2(0) + x_3^2(0)}{2x_2^2(0) + x_3^2(0)} \tag{3.13e}$$

$$t_0 = -\frac{1}{\omega}\, \mathrm{dn}^{-1}\left(\frac{x_2(0)}{Q}\right) \tag{3.13f}$$

Obviously, Eqs. (3.13a) – (3.13e) have – in principle – two possible solutions. Also, the inverse function $\mathrm{dn}^{-1}(\ldots)$ in Eq. (3.13f) is multivalued. We can, however, pick the appropriate solutions based on the signs of the initial conditions $x_1(0)$ and $x_3(0)$. Let us clarify this by considering the example $x_1(0) > 0$ and $x_3(0) > 0$.

We start off by simply choosing the positive solution for ω. Hence, we have

$$\omega = \sqrt{2x_2^2(0) + x_3^2(0)} \tag{3.14}$$

We require the elliptic modulus k to be positive as well resulting in

$$k = \sqrt{\frac{2x_1^2(0) + x_3^2(0)}{2x_2^2(0) + x_3^2(0)}} \tag{3.15}$$

Note that for the branch of solution curves under consideration we have

$$-\,|x_2(0)| < x_1(0) < |x_2(0)| \tag{3.16}$$

Hence, the elliptic modulus k is real and satisfies $0 < k < 1$.

Since we consider solution curves with $x_2(0) > 0$ and $\mathrm{dn}\, u$ is nonnegative for every u, it follows from Eq. (3.11b) that Q should also be positive. Hence, we have

$$Q = \sqrt{x_2^2(0) + \frac{1}{2} x_3^2(0)} \tag{3.17}$$

We continue by selecting the appropriate solution t_0 in Eq. (3.13f). To this end we consider $\frac{\mathrm{d}x_2}{\mathrm{d}t}|_{t=0} = x_3(0)x_1(0)$. In this example we have $x_3(0)x_1(0) > 0$. Hence, we select t_0 to be the smallest *positive* solution to Eq. (3.13f). With this value of t_0 our expression for x_2 in Eq. (3.9b) will have the desired property $\frac{\mathrm{d}x_2}{\mathrm{d}t}|_{t=0} > 0$.

The correct solution P in Eq. (3.13a) now follows from $\frac{\mathrm{d}x_1}{\mathrm{d}t}|_{t=0} = x_2(0)x_3(0)$. In this example we have $x_2(0)x_3(0) > 0$. We therefore choose

$$P = \sqrt{x_1^2(0) + \frac{1}{2}x_3^2(0)} \tag{3.18}$$

This ensures that our expression for x_1 in Eq. (3.9a) will have the desired property $\frac{\mathrm{d}x_1}{\mathrm{d}t}|_{t=0} > 0$.

Similarly, the correct solution R in Eq. (3.13c) follows from $\frac{\mathrm{d}x_3}{\mathrm{d}t}|_{t=0} = -2x_1(0)x_2(0) < 0$. With

$$R = \sqrt{2x_1^2(0) + x_3^2(0)} \tag{3.19}$$

our expression for x_3 in Eq. (3.9c) has the desired property $\frac{\mathrm{d}x_3}{\mathrm{d}t}|_{t=0} < 0$.

Summarizing, we have determined the analytic solution to Eqs. (3.1) with initial conditions satisfying $x_2(0) > 0$, $x_3(0) > 0$ and $0 < x_1(0) < x_2(0)$. The result is

$$x_1(t) = \sqrt{x_1^2(0) + \frac{1}{2}x_3^2(0)}\,\mathrm{cn}\left[\sqrt{2x_2^2(0) + x_3^2(0)}\,(t - t_0)\ ;\ k\right] \tag{3.20a}$$

$$x_2(t) = \sqrt{x_2^2(0) + \frac{1}{2}x_3^2(0)}\,\mathrm{dn}\left[\sqrt{2x_2^2(0) + x_3^2(0)}\,(t - t_0)\ ;\ k\right] \tag{3.20b}$$

$$x_3(t) = -\sqrt{2x_1^2(0) + x_3^2(0)}\,\mathrm{sn}\left[\sqrt{2x_2^2(0) + x_3^2(0)}\,(t - t_0)\ ;\ k\right] \tag{3.20c}$$

where t_0 is the smallest *positive* solution of

$$t_0 = -\frac{1}{\sqrt{2x_2^2(0) + x_3^2(0)}}\,\mathrm{dn}^{-1}\left(\frac{x_2(0)}{\sqrt{x_2^2(0) + \frac{1}{2}x_3^2(0)}}\right) \tag{3.21}$$

and

$$k = \sqrt{\frac{2x_1^2(0) + x_3^2(0)}{2x_2^2(0) + x_3^2(0)}} \tag{3.22}$$

Analytic solutions for other initial conditions can in principle be determined in a similar fashion. However, we have experienced in this section that obtaining a general analytic solution to the Kraichnan-Orszag problem is a pretty cumbersome task.

Things get even more complicated for stochastic initial conditions like Eq. (3.2). From Eq. (3.22) we see that the elliptic modulus k now becomes a function of ξ and thus becomes a stochastic variable as well. For the case of stochastic initial conditions we will typically be interested in quantities like the mean and variance of x_1, x_2 and x_3. Determining these quantities analytically would require the evaluation of integrals of functions of $\mathrm{sn}\,[u(\boldsymbol{\xi}), k(\boldsymbol{\xi})]$, $\mathrm{cn}\,[u(\boldsymbol{\xi}), k(\boldsymbol{\xi})]$ and $\mathrm{dn}\,[u(\boldsymbol{\xi}), k(\boldsymbol{\xi})]$ over the random variable(s) $\boldsymbol{\xi}$. We see that, in addition to the first argument u, the elliptic modulus k is also dependent on the variable(s) of integration $\boldsymbol{\xi}$. An analytic solution to integrals of this type is unknown to this author. Hence, the mean and variance of x_1, x_2 and x_3 cannot be determined analytically for stochastic initial conditions.

3.4 Effectiveness of gPC

In this section we make use of the analytic result in Eqs. (3.20) – (3.22) to predict the effectiveness of a gPC solution approach to the Kraichnan-Orszag problem (Eq. (3.1)). We consider the stochastic initial conditions presented earlier in Eq. (3.2) for three values of the constant α, namely $\alpha = 0.5$, $\alpha = 0.96$ and $\alpha = 0.99$. Note that with either of these three choices for α Eqs. (3.20) – (3.22) are applicable for every possible realization of the random variable ξ in Eq. (3.2). In other words, every possible realization of the initial conditions will satisfy $x_2(0) > 0$, $x_3(0) > 0$ and $0 < x_1(0) < x_2(0)$.

In the theory of elliptic functions [15] the period of Jacobi's elliptic functions in Eq. (3.20) is often given in terms of a quantity K termed the complete elliptic integral of the first kind. The periods of $\mathrm{sn}(u)$ and $\mathrm{cn}(u)$ are both equal to $4K$, while the period of $\mathrm{dn}(u)$ equals $2K$. K itself may be defined as follows

$$K(k) = \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} \mathrm{d}\theta \tag{3.23}$$

Note from this definition that K is a function of the elliptic modulus k. For our purposes, rather than to consider the exact definition given in Eq. (3.23), it is more helpful to look at the general shape of the function $K(k)$. In Fig. 3.2 we therefore provide a plot. The figure shows that on the somewhat arbitrary interval $0 \leq k \lesssim 0.9$ K increases relatively moderately with increasing k. However, in a relatively narrow interval, say $0.9 \lesssim k < 1$, K increases sharply with increasing k and goes to infinity for $k = 1$.

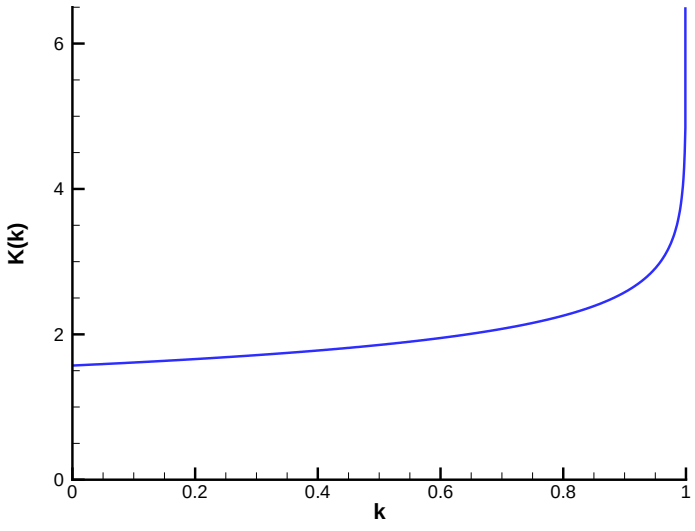

Figure 3.2: Complete elliptic integral of the first kind K versus elliptic modulus k

Let us now substitute the initial conditions given by Eq. (3.2) into the

expression for k in Eq. (3.22). We obtain

$$k = \sqrt{\frac{2}{3}(\alpha + 0.01\xi)^2 + \frac{1}{3}} \tag{3.24}$$

Remember that in the above expression α is a constant and ξ is a random variable uniformly distributed on the interval $[-1, 1]$. If we now consider the case $\alpha = 0.5$ – $x_1(0)$ is uniformly distributed on the interval $[0.49, 0.51]$ – we see from Eq. (3.24) that all possible realizations of k are confined to the narrow interval $0.7024 \lesssim k \lesssim 0.7119$. From Fig. 3.2 we see that this means that realizations of K will also lie within a relatively narrow interval. We have discussed earlier that the periods of Jacobi's elliptic functions are directly related to K. So, from Eqs. (3.20) and the argument above we conclude that for the case $\alpha = 0.5$ the period of the solution will not differ much among all possible realizations $(x_1(t), x_2(t), x_3(t))$.

We illustrate this point in Fig. 3.3 where we see that different realizations of $x_1(t)$ are – on the scale of this plot – almost identical . In this figure we have only considered realizations of $x_1(t)$. The corresponding realizations of $x_2(t)$ and $x_3(t)$ tell the same story. For completeness these are included in Appendix A.1.

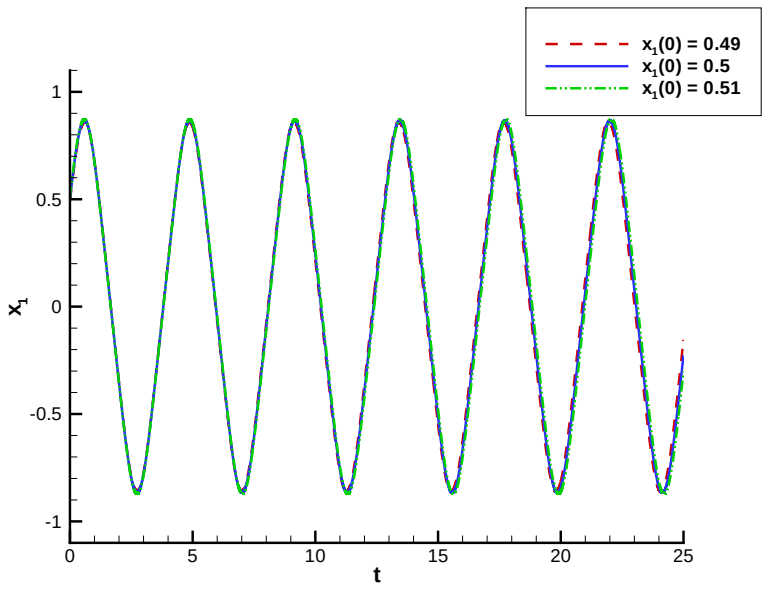

Figure 3.3: Deterministic solutions $x_1(t)$ for various $x_1(0)$ around $x_1(0) = 0.5$ $(x_2(0) = x_3(0) = 1)$

Things change somewhat for the case $\alpha = 0.96$. Now all possible realizations of k are confined to the interval $0.9670 \lesssim k \lesssim 0.9801$. This interval lies much closer to $k = 1$ which means (see Fig. 3.2) that realizations of K will be spread over a much larger interval than for the case $\alpha = 0.5$. So, different realizations $(x_1(t), x_2(t), x_3(t))$ of the solution will also show a much larger spread in terms

of the period. In Fig. 3.4 we have plotted three possible realization of $x1(t)$ for the case $\alpha = 0.96$. The difference in period can clearly be observed.

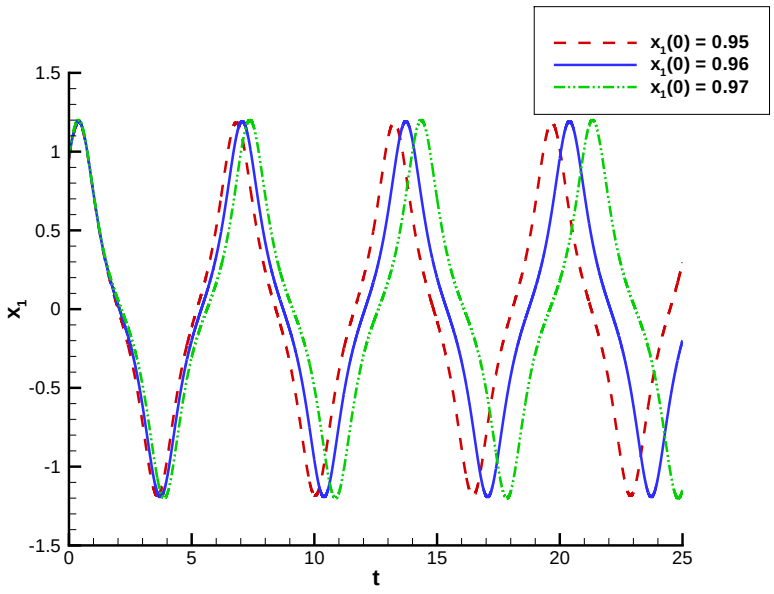

Figure 3.4: Deterministic solutions $x_1(t)$ for various $x_1(0)$ around $x_1(0) = 0.96$ $(x_2(0) = x_3(0) = 1)$

$\alpha = 0.99$ – $x_1(0)$ uniformly distributed on the interval $[0.49, 0.51]$ – can be seen as an extreme case. The possible realizations of k lie in the interval $0.9867 \lesssim k \leq 1$. We have seen above that K goes to infinity for $k = 1$. This means that the possible realizations of K lie on an unbounded interval. As a consequence the periods of possible realizations $(x_1(t), x_2(t), x_3(t))$ of the solution will differ greatly as is illustrated in Fig. 3.5. The interval of possible periods has no upper bound, so even a solution with 'infinite' period can be realized. This means that this solution will be aperiodic.

From Eqs. (3.20) – (3.22) we see that, apart from an effect on k, varying $x_1(0)$ also has an effect on the amplitude of x_1 and x_3. However, for small variations of $x_1(0)$ (like we consider with the initial conditions in Eq. (3.2)) the effect on the amplitude is small. We have seen above that the effect of variations in $x_1(0)$ on the period of the solution depends on the values of k that are attained. If we stay away from $k = 1$ the effect is small, but if we come close to $k = 1$ the effect suddenly becomes very large.

It is precisely in this region close to $k = 1$ where we expect problems with a gPC solution approach to the Kraichnan-Orszag problem. In Fig. 3.4 we have seen that different realizations of the solution diverge from each other with increasing time. With the initial conditions of Eq. (3.2) with values of α around 0.96 we therefore expect difficulties for a gPC solution approach to accurately calculate statistical moments of the solution when we integrate to larger time values.

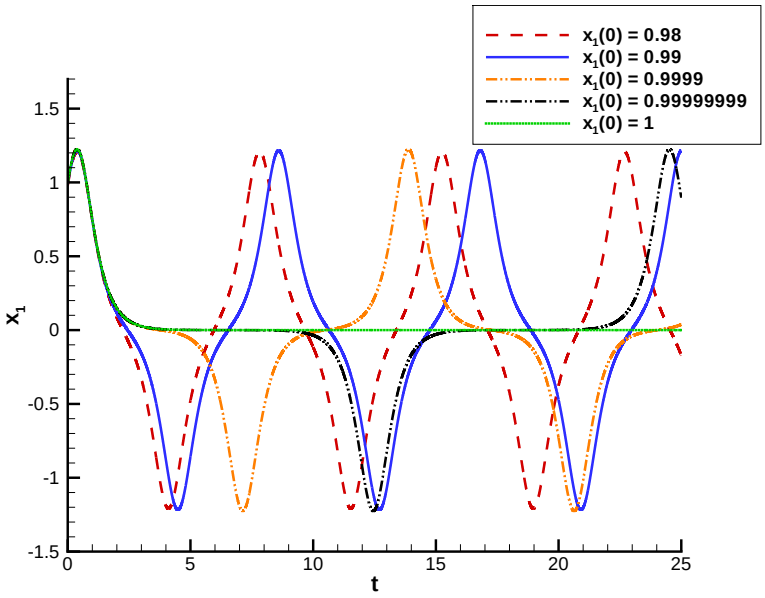

Figure 3.5: Deterministic solutions $x_1(t)$ for various $x_1(0)$ around $x_1(0) = 0.99$ $(x_2(0) = x_3(0) = 1)$

When α is increased to 0.99 we expect the problems with gPC to increase as well. From Fig. 3.5 we have seen that different realizations of the solution often have completely different periods and, as a consequence, diverge from each other at a very early time.

Taking α in the region $0.99 < \alpha < 1.01$ will lead to this same problem with the periods, but we now have the added difficulty that the possible realizations of the solution lie on two completely different branches of solution curves (as was illustrated in Fig. 3.1). Needless to say that we also expect serious problems with gPC for this case.

Chapter 4

Shortcomings of a gPC solution approach

The success of a generalized polynomial chaos (gPC) solution approach to the Kraichnan-Orszag problem defined in Eqs. (3.1) depends to a large extent on the initial conditions. For initial conditions of the form given in Eq. (3.2) the value of α is very important. In this chapter we will look at gPC results for a number of different values of α, namely $\alpha = 0.5$, $\alpha = 0.96$, $\alpha = 0.99$ and $\alpha = 0.995$. We will discuss the accuracy of the calculated solution on the basis of results for x_1. The corresponding gPC results for x_2 and x_3 can be found in Appendix A.2.

To get an idea of the character of the solutions $(x_1(t), x_2(t), x_3(t))$ we present the deterministic solution trajectories for a number of fixed, nonstochastic values of $x_1(0)$ in Fig. 4.1. We can see that for most values of $x_1(0)$ considered individual deterministic solutions follow a closed, repeating trajectory. The case $x_1(0) = 1$ is an exception: this solution does not repeat itself in time but rather converges to a fixed point. Note that, even though solution trajectories lie closely together (see, for instance, the trajectories for $x_1(0) = 0.96$ and $x_1(0) = 0.99$ in Fig. 4.1), we have seen in Figs. 3.4 and 3.5 that the period of the solution may still vary considerably from one trajectory to the other.

Let us first consider the case $\alpha = 0.5$. In Figs. 4.2 and 4.3 we compare the time evolution of the mean and variance, respectively, of x_1 as calculated by a gPC approach to the results from a Monte Carlo analysis. The gPC results were obtained by using basis polynomials up to a maximum degree of three ($P = 3$). For the Monte Carlo analysis a number of 10,000 samples were considered ($N = 10,000$). We observe from Figs. 4.2 and 4.3 that we can generate pretty accurate results with a gPC solution approach when $\alpha = 0.5$. Results for the mean and variance of x_2 and x_3 show a similar accuracy.

Let us proceed by considering the case $\alpha = 0.96$. From Fig. 4.4 we can see that the gPC solution for the mean of x_1 loses accuracy from, approximately, $t = 24$ onwards. Fig. 4.5 shows that the accurate calculation of the variance of x_1 starts to break down at a slightly earlier instant in time.

The performance of the gPC solution approach further deteriorates when we increase α to a value of 0.99. Figs. 4.6 and 4.7 provide results for this case. We

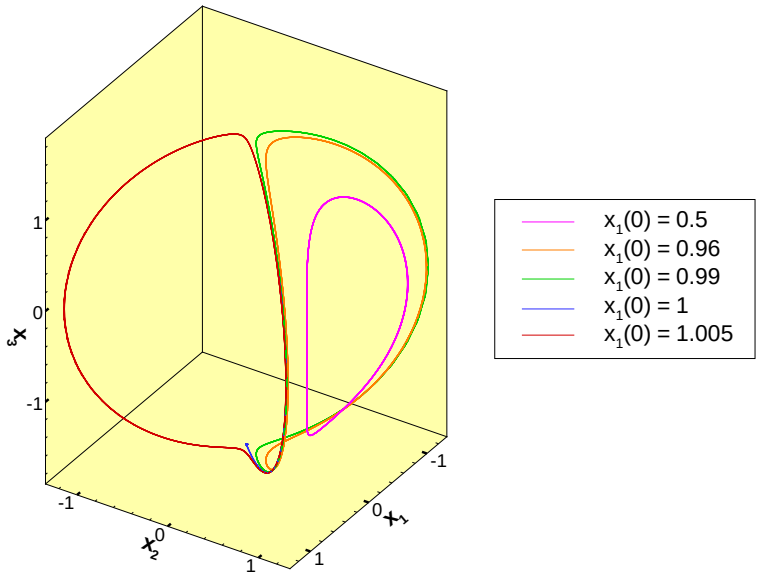

Figure 4.1: Deterministic solutions of the Kraichnan-Orszag problem for various values of $x_1(0)$ ($x_2(0) = x_3(0) = 1$)

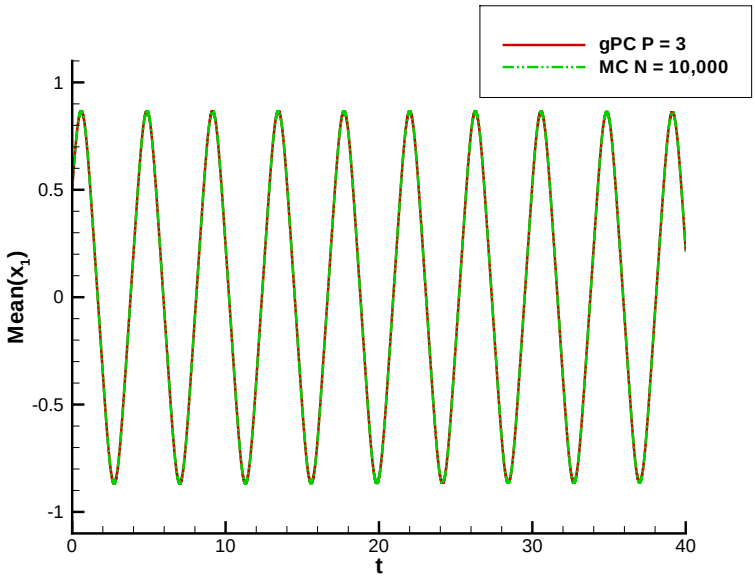

Figure 4.2: Mean of x_1 vs. time for $\alpha = 0.5$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

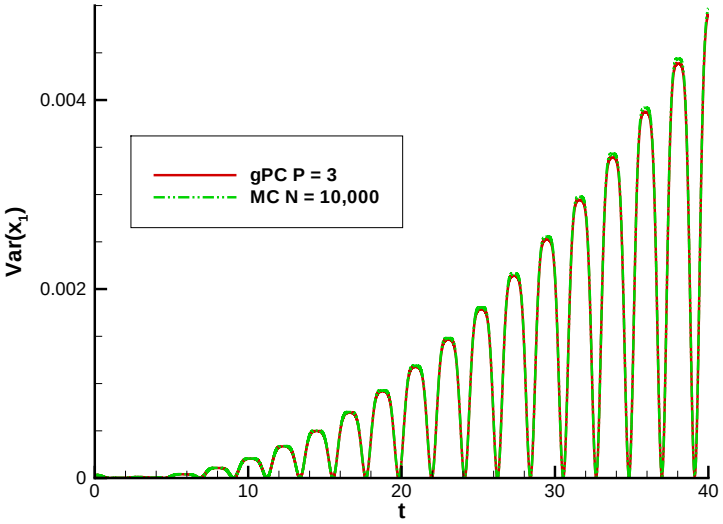

Figure 4.3: Variance of x_1 vs. time for $\alpha = 0.5$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

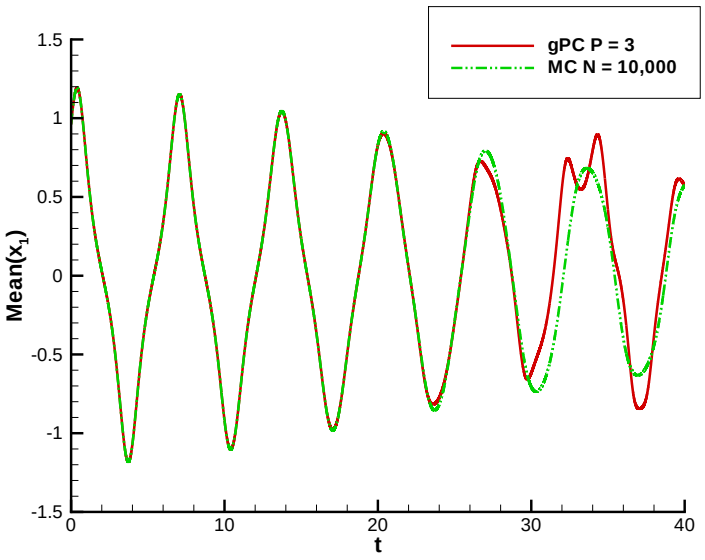

Figure 4.4: Mean of x_1 vs. time for $\alpha = 0.96$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

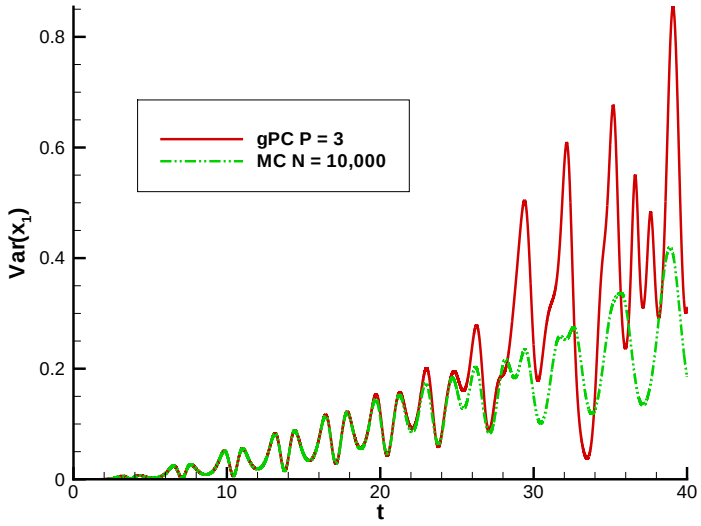

Figure 4.5: Variance of x_1 vs. time for $\alpha = 0.96$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

can see that for $\alpha = 0.99$ the gPC solution breaks down earlier compared to the case where $\alpha = 0.96$. Two additional observations can be made from Figs. 4.6 and 4.7.

Firstly, one might wonder whether increasing the maximum degree of the basis polynomials used in the gPC calculation results in a solution that remains accurate up to larger time values. To investigate this we also plotted gPC results for $P = 6$, $P = 10$ and $P = 22$ in Figs. 4.6 and 4.7. Although gPC results for higher P can be seen to remain accurate up to slightly larger time values than results for $P = 3$, we must conclude that the effect of increasing P is very marginal. Even gPC results calculated with a maximum degree of the basis polynomials as high as $P = 22$ are worthless after, roughly, $t = 15$. So, to our disappointment we have to conclude that increasing the maximum degree of the basis polynomials does not bring us much (apart from a serious increase in calculation time and memory consumption).

Secondly, we check in Figs. 4.6 and 4.7 whether the number of samples N considered for the Monte Carlo analysis is large enough. Since the results for $N = 10,000$ and $N = 100,000$ are virtually identical on the scale of this plot we are led to believe that taking $N = 10,000$ is indeed sufficient for our purposes.

At this point let us also exclude the possibility that the disappointing gPC results are due to incorrectly set integration parameters such as the number of quadrature points Q for integration in random space or the number of stages s and time step Δt for the Runge-Kutta time integration. To this end we consider the conservation property of Eq. (3.5). The (constant) value of $\sum_{i=1}^{3} \langle x_i \rangle^2$ is known exactly from the initial conditions. In Fig. 4.8 we investigate the error that is introduced into this constant due to inaccuracies in the numerical

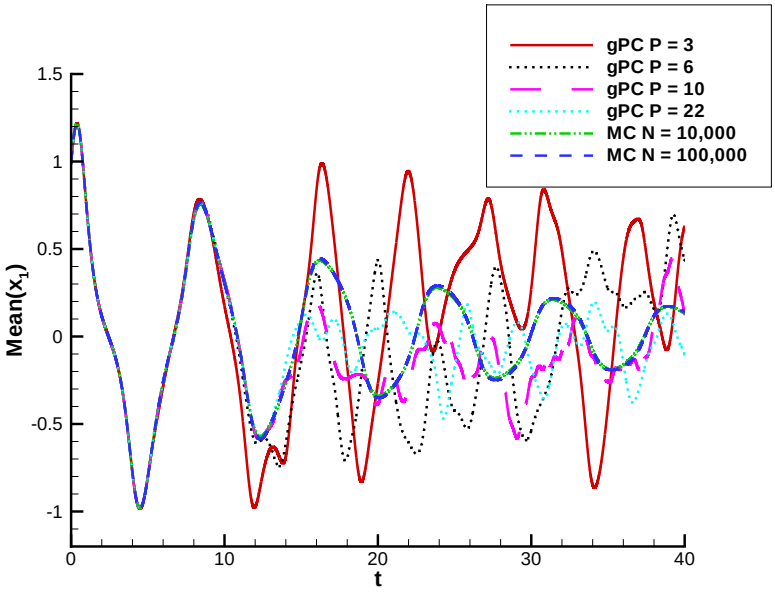

Figure 4.6: Mean of x_1 vs. time for $\alpha = 0.99$: gPC solutions with $P = 3$, $P = 6$, $P = 10$ and $P = 22$ compared to Monte Carlo analyses ($N = 10,000$ and $N = 100,000$)

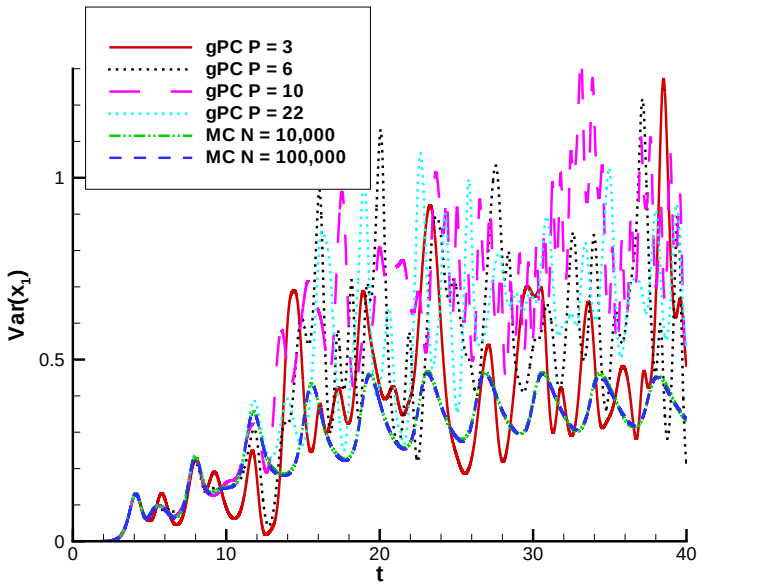

Figure 4.7: Variance of x_1 vs. time for $\alpha = 0.99$: gPC solutions with $P = 3$, $P = 6$, $P = 10$ and $P = 22$ compared to Monte Carlo analyses ($N = 10,000$ and $N = 100,000$)

integration. For P = 3 the gPC algorithm requires the numerical integration of polynomials with a maximum degree of nine. Setting $Q \geq \frac{3}{2}P + \frac{3}{2} = 6$ these integrals are evaluated exactly using Gauss-Lobatto-Legendre integration [16]. The 'benchmark' graph in this figure corresponds to $P = 3$, $Q = 6$, $s = 4$ and $\Delta t = 0.001$. All gPC results in this section were calculated with these values of s and Δt with Q large enough for exact integration. The 'benchmark' graph in Fig. 4.8 shows a very small error: within a few orders of magnitude from machine precision for the entire time interval considered. This leads us to believe that the integration parameters will most likely not be the cause of the disappointing gPC results. Fig. 4.8 also shows results for integration parameters that differ from the benchmark values. It can be seen that relaxing the requirements on time step and number of stages results in a larger error in $\sum_{i=1}^{3} \langle x_i \rangle^2$ that grows slowly with time. Also, taking not enough quadrature points for exact integration in random space results in less accurate results.

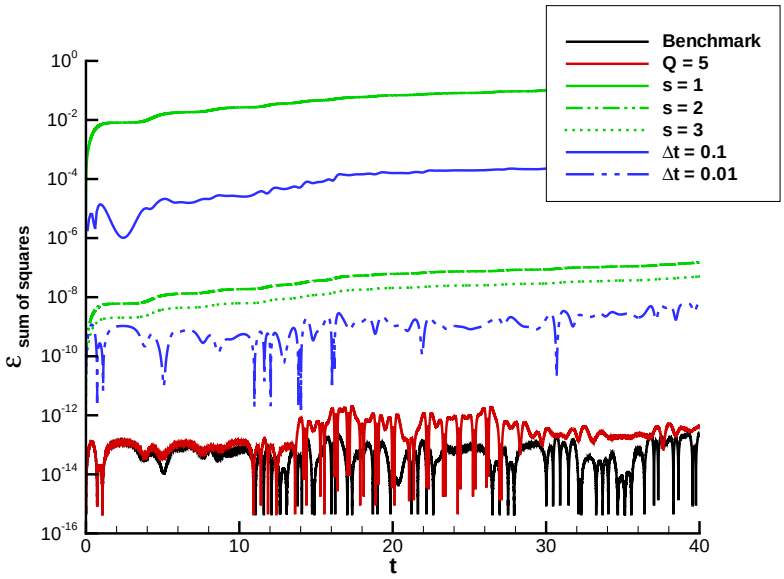

Figure 4.8: Error in $\langle x_1 \rangle^2 + \langle x_2 \rangle^2 + \langle x_3 \rangle^2$ vs. time for $\alpha = 0.99$: comparison of various integration parameter settings (Benchmark: $P = 3$; $Q = 6$; $s = 4$; $\Delta t = 0.001$)

Finally, we look at gPC results for $\alpha = 0.995$. Fig. 4.1 shows why this case is interesting: for $x_1(0) > 1$ a completely different branch of solution trajectories results. In Figs. 4.9 and 4.10 gPC results are presented. Although we still obtain accurate results for small time values, we must conclude that the gPC solution diverges form the Monte Carlo results even sooner than was the case with $\alpha = 0.99$.

Based on the results presented above we can conclude that a gPC solution approach to the Kraichnan-Orszag problem with a stochastic initial condition does not, in general, lead to satisfactory results. For at least some initial con-

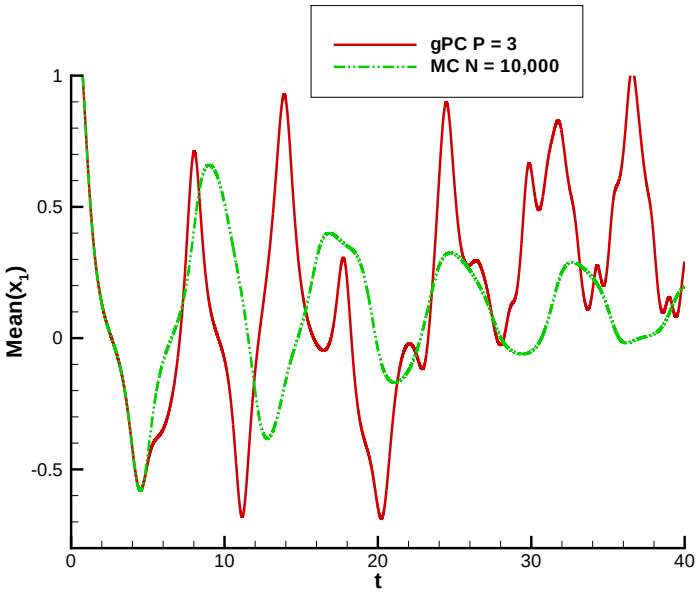

Figure 4.9: Mean of x_1 vs. time for $\alpha = 0.995$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

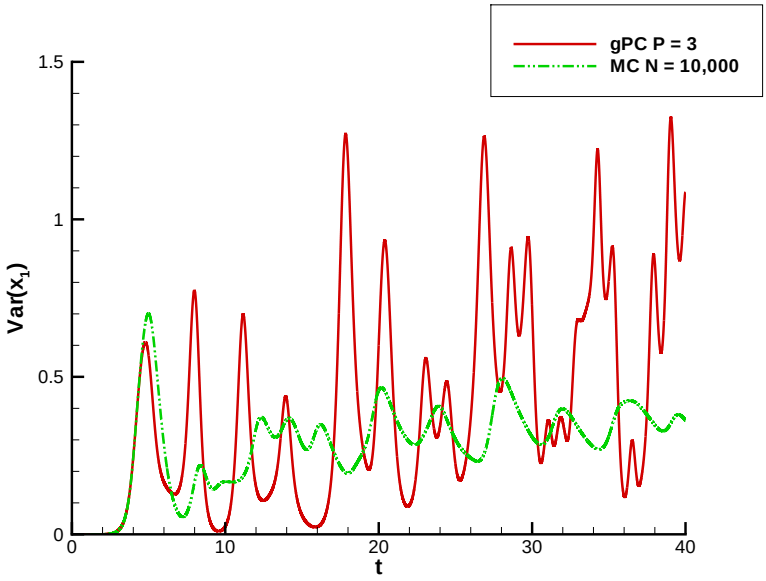

Figure 4.10: Variance of x_1 vs. time for $\alpha = 0.995$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

ditions the gPC solution starts to diverge from the correct solution at a certain instant in time. Also, extending the time interval beyond the values considered in Figs. 4.2 and 4.3, we cannot exclude the possibility that even for $\alpha = 0.5$ we will eventually reach a point in time after which the gPC solution breaks down.

Figs. 4.11 – 4.14 are included to support this presumption. In these figures we present gPC results for the cases $\alpha = 0.7$ and $\alpha = 0.8$ on a more extended time interval than we have considered so far. Notice that, even though these gPC results remain accurate for quite some time, eventually the accuracy of earlier times is lost. It is just a matter of integrating far enough in time. So, it might well be that irrespective of the value of α we can always find an instant in time at which the calculated solution starts to diverge from the correct one.

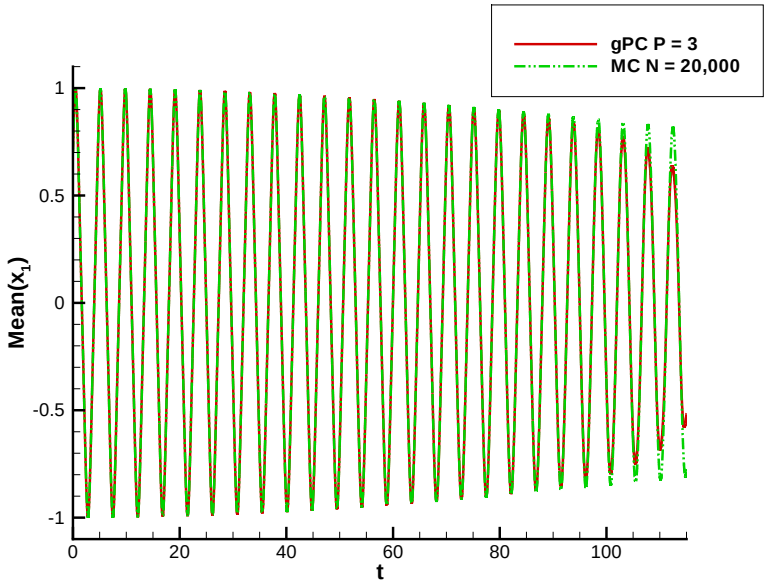

Figure 4.11: Mean of x_1 vs. time for $\alpha = 0.7$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 20,000$)

To overcome this problem we aim to improve the gPC solution approach. We will try to apply a technique proposed by Vos [10] called time-dependent generalized polynomial chaos (TDgPC) to the Kraichnan-Orszag problem.

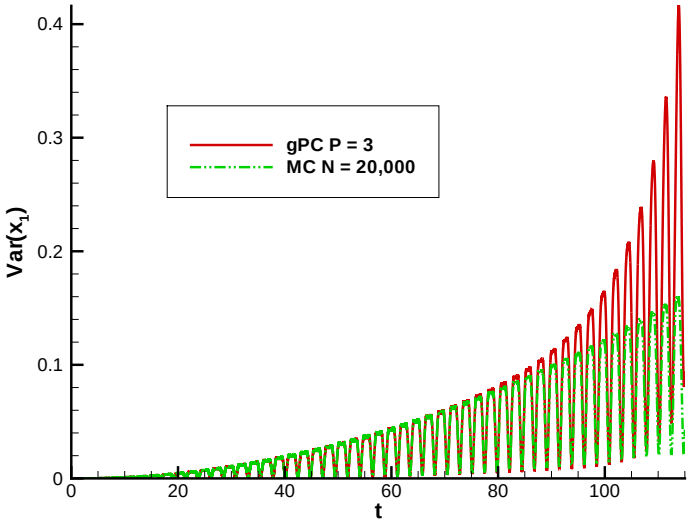

Figure 4.12: Variance of x_1 vs. time for $\alpha = 0.7$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 20,000$)

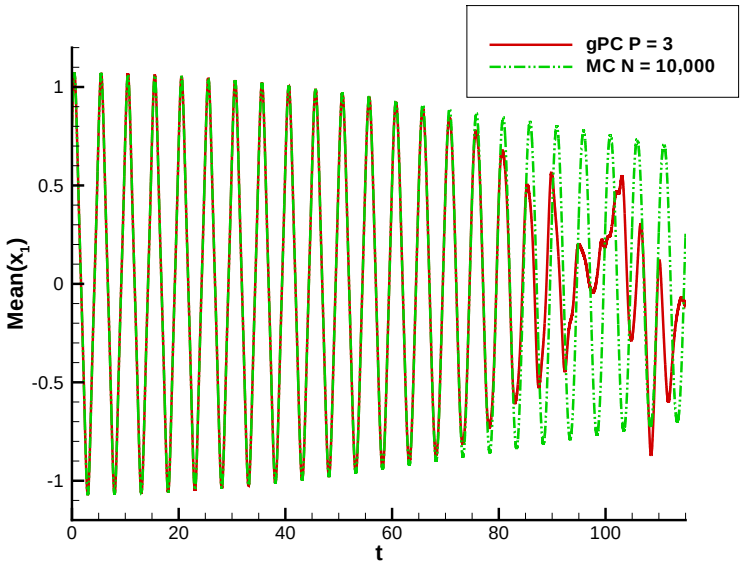

Figure 4.13: Mean of x_1 vs. time for $\alpha = 0.8$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

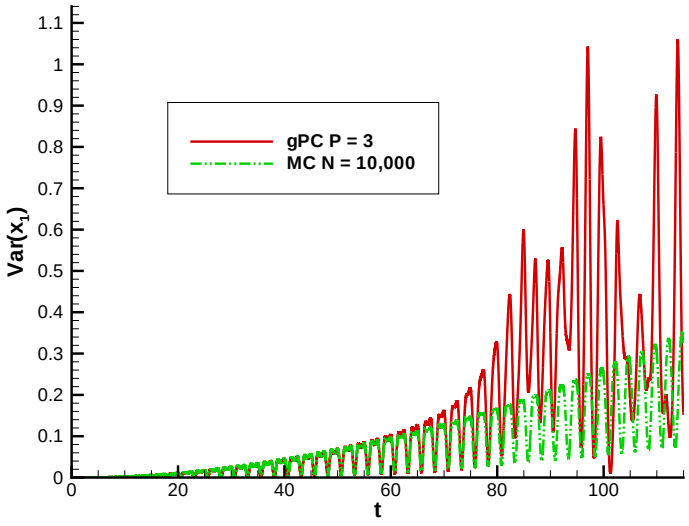

Figure 4.14: Variance of x_1 vs. time for $\alpha = 0.8$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

Chapter 5

TDgPC solution approach: a first try

In Chapter 2 we introduced the generalized polynomial chaos (gPC) method of solving stochastic problems. This method involves the usage of a carefully selected orthogonal polynomial trial basis to spectrally expand the stochastic solution in the random space. To determine which expansion basis is optimal the stochastic distribution of the random inputs is considered (see Eqs. (2.5) and (2.6)).

We already noted in Chapter 2 that the thus obtained polynomial trial basis can be considered optimal based upon the presumption that – at least for early times – the random solution represented by the polynomial chaos expansion has a stochastic distribution roughly identical to the distribution of the random inputs. However, nonlinearities in the problem will in general cause the stochastic distribution of the solution to evolve over time. The character of the solution distribution may therefore have changed radically when we integrate far enough in time. Say, for example, that we have a uniformly distributed random input to a stochastic problem. The problem may be such that the distribution of the resulting solution evolves in time from (near-)uniform to (near-)Gaussian. One could say that with time the uniform random input is 'forgotten' due to nonlinearities in the problem.

An important consequence of this phenomenon is that the initially selected trial basis gradually loses its efficiency. After some time the initial expansion polynomials cannot be considered optimal, or even near-optimal, anymore. Considering that we work with a *finite* number terms in the solution expansion, we not only lose efficiency but also accuracy at later times. For these later times an optimal polynomial trial basis should be based on the *current* distribution of the solution rather than the distribution of the random inputs. Hence, the expansion polynomials should satisfy Eqs. (2.5) and (2.6) with $f_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ replaced by the current PDF of the solution. Of course, a thus calculated expansion basis will also gradually lose its efficiency when we were to integrate the solution in time even further.

Vos [10] was the first to develop the idea of updating the polynomial trial basis at a number of discrete points in time, i.e. calculating new optimal expansion polynomials based upon the instantaneous stochastic distribution of the solu-

tion. Vos termed this technique time-dependent generalized polynomial chaos (TDgPC). In this chapter we will try to apply this idea to the Kraichnan-Orszag problem. This, of course, with the hope of obtaining more satisfactory results for long-term integration than we acquired in Chapter 4 with an ordinary gPC approach.

5.1 Overview of the solution method

We once again consider the Kraichnan-Orszag problem defined in Eqs. (3.1) with the initial conditions of Eq. (3.2). We start out identical to a standard gPC solution approach. So, initially we look for a numerical approximation of the solution in the following form

$$x_i(t,\xi) = \sum_{p=0}^{P} x_p^{(i)}(t) L_p(\xi) \ , \quad i = 1,2,3 \tag{5.1}$$

where L_p is the Legendre polynomial of degree p. Since ξ has a uniform distribution, the Legendre polynomials constitute an optimal trial basis for early times (see Table 2.1). Employing this polynomial chaos expansion of the solution and following the method outlined in Section 2.2 we rather straightforwardly arrive at a system of deterministic ordinary differential equations in time for the coefficients $x_p^{(i)}(t)$. We solve this system by standard fourth-order Runge-Kutta time integration.

From Eq. (5.1) we see that the approximate solutions x_i are polynomials in the random variable ξ. With time the nonlinear coefficients of these solution polynomials increase in size. This is an indication that the stochastic characteristics of the solution are changing. As a consequence the trial basis $\{L_p\}$ is losing its effectiveness. When the nonlinear part of the solution reaches a certain threshold level (say at $t = t_1$) we therefore perform a transformation of the random variable from ξ to ζ_i given by

$$\zeta_i = x_i(t_1,\xi) = \sum_{p=0}^{P} x_p^{(i)}(t_1) L_p(\xi) \ , \quad i = 1,2,3 \tag{5.2}$$

The three new random variables ζ_i have associated PDFs $f_{\zeta_i}(\zeta_i)$. In Section 5.2 we address the problem of calculating these new PDFs.

For each f_{ζ_i} we can, employing a Gram-Schmidt orthogonalization algorithm (see Section 6.3 for details), calculate a set of orthogonal polynomials $\phi_p^{(\zeta_i)}(\zeta_i)$ with $p = 0,\ldots,P$. By $\phi_p^{(\zeta_i)}$ we denote the polynomial of degree p associated with f_{ζ_i}, i.e. f_{ζ_i} acts as the weighting function in the orthogonality relation. At time level $t = t_1$ these polynomials constitute an optimal trial basis again. We therefore use these newly calculated polynomials $\phi_p^{(\zeta_i)}$ and continue to determine a numerical solution to the Kraichnan-Orszag problem in a new form given by

$$x_i(t,\zeta_1,\zeta_2,\zeta_3) = \sum_{0\le l+m+n\le P} x_{lmn}^{(i)}(t) \phi_l^{(\zeta_1)}(\zeta_1)\, \phi_m^{(\zeta_2)}(\zeta_2)\, \phi_n^{(\zeta_3)}(\zeta_3) \ , \quad t \ge t_1 \tag{5.3}$$

The summation in Eq. (5.3) is over all combinations of the integers l, m and n for which $0 \le l+m+n \le P$. The total number of expansion terms $(N+1)$

follows from Eq. (2.11) with $n = 3$ and is given by

$$N + 1 = \binom{P+3}{P} = \frac{(P+3)!}{P!3!} = \frac{1}{6}(P+3)(P+2)(P+1) \sim \frac{P^3}{6} \tag{5.4}$$

Substituting Eq. (5.3) in the problem equations (3.1) we once again follow the standard gPC procedure of Section 2.2. This means we perform a Galerkin projection to end up with a new system of ordinary differential equations for the new expansion coefficients $x^{(i)}_{lmn}(t)$. In Section 5.3 the associated calculations are written out in detail.

We proceed by marching this new system forward in time again from $t = t_1$ onwards using our standard fourth-order Runge-Kutta solver. Note, however, that we need to provide 'initial' conditions (i.e. conditions at $t = t_1$) for all new coefficients $x^{(i)}_{lmn}$. These initial conditions follow from the requirement

$$x_i\left(t_1, \zeta_1, \zeta_2, \zeta_3\right) = \zeta_i \ , \quad i = 1, 2, 3 \tag{5.5}$$

We can arrange for the orthogonal expansion polynomials $\phi^{(\zeta_i)}_p$ to all have unity leading coefficients. Therefore, at $t = t_1$ the coefficients $x^{(i)}_{lmn}$ are given by

$$\begin{aligned}
x^{(1)}_{lmn}\left(t_1\right) &= \begin{cases} -\phi^{(\zeta_1)}_{1,0} & \text{if} \quad l = m = n = 0 \\ 1 & \text{if} \quad l = 1 \wedge m = n = 0 \\ 0 & \text{otherwise} \end{cases} \\
x^{(2)}_{lmn}\left(t_1\right) &= \begin{cases} -\phi^{(\zeta_2)}_{1,0} & \text{if} \quad l = m = n = 0 \\ 1 & \text{if} \quad m = 1 \wedge l = n = 0 \\ 0 & \text{otherwise} \end{cases} \\
x^{(3)}_{lmn}\left(t_1\right) &= \begin{cases} -\phi^{(\zeta_3)}_{1,0} & \text{if} \quad l = m = n = 0 \\ 1 & \text{if} \quad n = 1 \wedge l = m = 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{5.6}$$

where $\phi^{(\zeta_i)}_{1,0}$ denotes the zeroth order term of the expansion polynomial of degree one associated with f_{ζ_i}.

Marching the new system of differential equations forward in time we again keep a close eye on the nonlinear part of the resulting solution. When, by some criterion, this nonlinear part has become too large (say at $t = t_2$) we simply repeat the above procedure in order to regain an optimal trial basis. Hence, we start by introducing the new random variables

$$\zeta^{(2)}_i = x_i\left(t_2, \zeta^{(1)}_1, \zeta^{(1)}_2, \zeta^{(1)}_3\right) \ , \quad i = 1, 2, 3 \tag{5.7}$$

and continue to calculate their PDFs from which the new optimal trial basis is calculated by Gram-Schmidt orthogonalization. Note that we have added a superscript to the random variables in Eq. (5.7) corresponding to the time instant at which they were introduced. Hence, we have rewritten the original variables ζ_i as $\zeta^{(1)}_i$. The process of updating the polynomial trial basis can be performed as many times as is required for the particular problem at hand. So, in general we have

$$\zeta^{(k+1)}_i = x_i\left(t_{k+1}, \zeta^{(k)}_1, \zeta^{(k)}_2, \zeta^{(k)}_3\right) \ , \quad i = 1, 2, 3 \ , \ k = 1, 2, \ldots, K-1 \tag{5.8}$$

with associated PDF $f_{\zeta_i^{(k+1)}}$ and orthogonal polynomials $\phi_p^{\left(\zeta_i^{(k+1)}\right)}$ leading to a polynomial chaos expansion, similar to Eq. (5.3), to be used for $t_{k+1} \leq t \leq t_{k+2}$.

5.2 Transformation of the PDF

In this section we address the problem of calculating the PDF of the newly introduced random variables. We first consider a one-dimensional transformation, like each of the three transformations $\xi \rightarrow \zeta_i$ in Eq. (5.2). After this we look at the general, multidimensional case. This is needed to deal with the transformation $\left(\zeta_1^{(k)}, \zeta_2^{(k)}, \zeta_3^{(k)}\right) \rightarrow \left(\zeta_1^{(k+1)}, \zeta_2^{(k+1)}, \zeta_3^{(k+1)}\right)$ of Eq. (5.8).

5.2.1 One-dimensional transformation

Let ξ be a stochastic variable with associated probability density function $f_\xi(\xi)$. When we transform this stochastic variable according to $\zeta = Z(\xi)$ we would like to know the PDF associated with the transformed random variable ζ. This PDF can be calculated by the following formula

$$f_\zeta(\zeta) = \sum_{\xi_n} \frac{f_\xi\left(\xi_n\right)}{\left|\frac{\mathrm{d}Z(\xi)}{\mathrm{d}\xi}|_{\xi=\xi_n}\right|} \tag{5.9}$$

where the summation is over all solutions ξ_n of the equation

$$\zeta = Z(\xi) \tag{5.10}$$

To understand the idea behind Eqs. (5.9) and (5.10) first note that a particular, realizable value of ζ, say ζ^*, is the image of one or multiple points ξ_n^* in the domain of the transformation Z. The probability that ζ takes on a value in an infinitesimal interval around the point $\zeta = \zeta^*$ can be denoted by $f_\zeta\left(\zeta^*\right)\mathrm{d}\zeta$. Naturally, this probability must be equal to the probability that ξ lies in an infinitesimal intervals around one of the points $\xi = \xi_n^*$. This latter probability can be denoted as $\sum_{\xi_n^*} f_\xi\left(\xi_n^*\right)\mathrm{d}\xi$. Eq. (5.9) now follows after some manipulation. A formal proof of Eqs. (5.9) and (5.10) can be found in [17].

As long as the transformation $\zeta = Z(\xi)$ is a one-to-one increasing or decreasing function, the resulting PDF f_ζ will be smooth and continuous. When, on the other hand, Eq. (5.10) has multiple solutions, f_ζ will be less well-behaved. Let us clarify this by an example.

Suppose ξ with PDF $f_\xi(\xi) = 1/2$ on $[-1, 1]$ is transformed according to

$$\zeta = Z(\xi) = -0.2\xi^2 + 0.2\xi + 0.8 \tag{5.11}$$

This transformation is visualized in Fig. 5.1. We see that for $\zeta \geq 0.8$ Eq. (5.11) has two solutions, while for $0.4 \leq \zeta < 0.8$ only a single solution exists. Also, the transformation has a maximum at $\zeta = 0.85$. These two properties influence the probability density function associated with ζ (calculated according to Eq. (5.9)) which is plotted in Fig. 5.2. We see that f_ζ exhibits a jump discontinuity at $\zeta = 0.8$ and a vertical asymptote at $\zeta = 0.85$. The jump discontinuity is found at the value of ζ where Eq. 5.11 changes form having one solution ξ_1 to having two solutions ξ_1 and ξ_2. This means that for $\zeta \geq 0.8$ the sum in

Eq. (5.9) consists of two terms instead of one. This sudden extra contribution causes the jump discontinuity at $\zeta = 0.8$. The asymptote is of course a result of the vanishing of $\frac{\mathrm{d}Z(\xi)}{\mathrm{d}\xi}$ at the value of ξ associated with $\zeta = 0.85$.

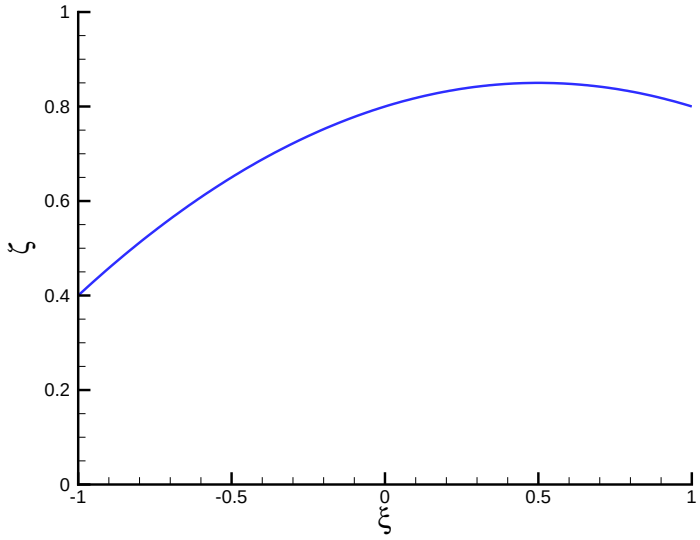

Figure 5.1: The random variable transformation $\zeta = -0.2\xi^2 + 0.2\xi + 0.8$

So, we now understand that (local) extremes in the transformation function Z result in jump discontinuities and vertical asymptotes in the PDF associated with the transformed random variable. Above we considered a second order transformation. A higher order transformation can contain more local extremes and could therefore result in a transformed PDF that has multiple vertical asymptotes and jump discontinuities.

We already mentioned in Chapter 4 that we use Gauss-Lobatto-Legendre numerical integration to evaluate all integrals in random space. A function with jump discontinuities can perfectly well be treated by Gauss-Lobatto-Legendre integration: we just have to split the integration domain at the discontinuities and perform a piecewise integration. For functions with a vertical asymptote, however, this integration method leads to poor results.

So, we conclude that a random variable transformation with local extremes will lead to problems in our TDgPC algorithm since the transformed PDF is subsequently used as a weighting function in integrals. There exist, of course, integration methods capable of handling vertical asymptotes. However, we will later see that there is a possibility to keep using the Gauss-Lobatto-Legendre method and work around the difficulties discussed above.

5.2.2 Multidimensional transformation

Let us now focus on the generalization of Eqs. (5.9) and (5.10) to multiple dimensions. This problem is discussed in [18]. In this section we briefly present

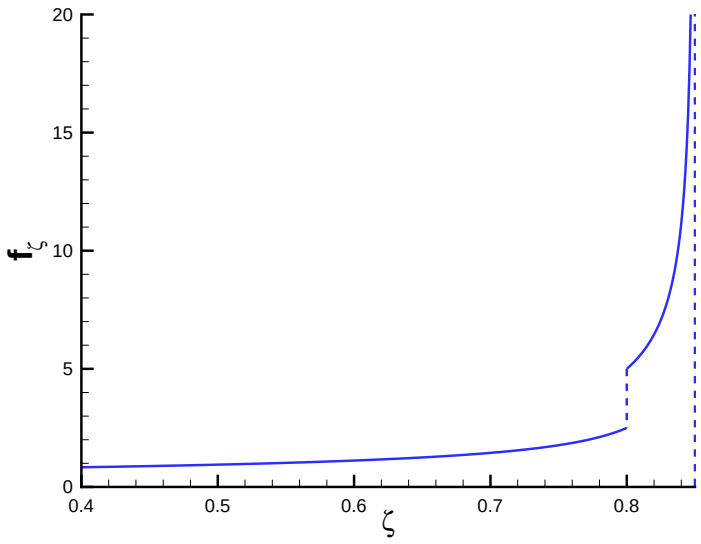

Figure 5.2: Probability density function associated with the transformed random variable

results from this work that can be applied to Eq. (5.8).

Let $\xi_1, \xi_2, \ldots, \xi_n$ be jointly continuous stochastic variables with joint PDF $f_{\xi_1,\xi_2,\ldots,\xi_n}(\xi_1, \xi_2, \ldots, \xi_n)$. Let these random variables be transformed according to $\zeta_1 = Z_1(\xi_1, \xi_2, \ldots, \xi_n)$, $\zeta_2 = Z_2(\xi_1, \xi_2, \ldots, \xi_n)$, ..., $\zeta_k = Z_k(\xi_1, \xi_2, \ldots, \xi_n)$ where k is some integer satisfying $1 \leq k \leq n$. We are interested in calculating the joint PDF $f_{\zeta_1,\zeta_2,\ldots,\zeta_k}(\zeta_1, \zeta_2, \ldots, \zeta_k)$. Let us start, however, by making two introductory remarks.

Firstly, note that we work with *joint* PDFs in this section, both for the original as well as for the transformed random variables. We have, however, seen in Section 5.1 that we rather need the *marginal* distributions $f_{\zeta_1}(\zeta_1)$, ..., $f_{\zeta_n}(\zeta_n)$ for the construction of the optimal polynomial trial basis. If ζ_1, ..., ζ_n are statistically independent we have the relation

$$f_{\zeta_1,\zeta_2,\ldots,\zeta_n}(\zeta_1, \zeta_2, \ldots, \zeta_n) = \prod_{i=1}^{n} f_{\zeta_i}(\zeta_i) \tag{5.12}$$

linking joint and marginal distributions. But even for statistically dependent random variables we can always determine a particular marginal distribution from the joint distribution by 'integrating out' the other random variables as follows

$$f_{\zeta_k}(\zeta_k) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\zeta_1,\zeta_2,\ldots,\zeta_n}(\zeta_1, \zeta_2, \ldots, \zeta_n) \mathrm{d}\zeta_1 \ldots \mathrm{d}\zeta_{k-1} \mathrm{d}\zeta_{k+1} \ldots \mathrm{d}\zeta_n \tag{5.13}$$

Secondly, note that if $k < n$ we can introduce auxiliary random variables $\zeta_{k+1} = Z_{k+1}(\xi_1, \xi_2, \ldots, \xi_n)$, ..., $\zeta_n = Z_n(\xi_1, \xi_2, \ldots, \xi_n)$ for judiciously selected

functions Z_{k+1}, ..., Z_n and calculate the joint PDF $f_{\zeta_1,\zeta_2,\ldots,\zeta_n}(\xi_1,\xi_2,\ldots,\xi_n)$. By integrating out the auxiliary random variables ζ_{k+1}, ..., ζ_n the desired marginal PDF $f_{\zeta_1,\zeta_2,\ldots,\zeta_k}(\xi_1,\xi_2,\ldots,\xi_k)$ can now be determined. Hence, we can cover all transformations for which k satisfies $1 \leq k \leq n$ by assuming that $k = n$. We will make this assumption from this point onwards, i.e. we consider a transformation from n-dimensional space to n-dimensional space.

Now, let us continue with the calculation of $f_{\zeta_1,\zeta_2,\ldots,\zeta_n}(\zeta_1,\zeta_2,\ldots,\zeta_n)$. We define the domain $\mathcal{D}$ of the transformation by

$$\mathcal{D} = \{(\xi_1,\ldots,\xi_n) : f_{\xi_1,\ldots,\xi_n}(\xi_1,\ldots,\xi_n) > 0\} \tag{5.14}$$

Assume that $\mathcal{D}$ can be decomposed into subdomains $\mathcal{D}_1,\ldots,\mathcal{D}_m$ such that $\zeta_1 = Z_1(\xi_1,\ldots,\xi_n)$, $\zeta_2 = Z_2(\xi_1,\ldots,\xi_n)$, ..., $\zeta_n = Z_n(\xi_1,\ldots,\xi_n)$ are all one-to-one transformations from $\mathcal{D}_i$ onto $\mathcal{R}$ for $i = 1,\ldots,m$. Let $\xi_1 = Z_{1i}^{-1}(\zeta_1,\ldots,\zeta_n)$, ..., $\xi_n = Z_{ni}^{-1}(\zeta_1,\ldots,\zeta_n)$ denote the inverse transformation of $\mathcal{R}$ onto $\mathcal{D}_i$, $i = 1,\ldots,m$. Now define the determinants

$$J_i = \begin{vmatrix} \frac{\partial Z_{1i}^{-1}}{\partial \zeta_1} & \frac{\partial Z_{1i}^{-1}}{\partial \zeta_2} & \cdots & \frac{\partial Z_{1i}^{-1}}{\partial \zeta_n} \\ \frac{\partial Z_{2i}^{-1}}{\partial \zeta_1} & \frac{\partial Z_{2i}^{-1}}{\partial \zeta_2} & \cdots & \frac{\partial Z_{2i}^{-1}}{\partial \zeta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Z_{ni}^{-1}}{\partial \zeta_1} & \frac{\partial Z_{ni}^{-1}}{\partial \zeta_2} & \cdots & \frac{\partial Z_{ni}^{-1}}{\partial \zeta_n} \end{vmatrix}, \quad i = 1,\ldots,m \tag{5.15}$$

Assuming that all partial derivatives in Eq. (5.15) are continuous and the determinants J_i are nonzero for $i = 1,\ldots,m$, we have

$$f_{\zeta_1,\ldots,\zeta_n}(\zeta_1,\ldots,\zeta_n) = \sum_{i=1}^{m} |J_i| \, f_{\xi_1,\ldots,\xi_n}\left(Z_{1i}^{-1}(\zeta_1,\ldots,\zeta_n),\ldots,Z_{ni}^{-1}(\zeta_1,\ldots,\zeta_n)\right) \tag{5.16}$$

for $(\zeta_1,\ldots,\zeta_n) \in \mathcal{R}$.

Eq. (5.16) together with Eq. (5.15) are based on the same principle as their one-dimensional counterparts Eqs. (5.9) and (5.10). The more complex mathematical implementation is actually very well-known from variable transformations in multiple integrals. A comprehensive discussion of these transformations can be found in many standard calculus textbooks.

5.3 System of differential equations after a random variable transformation

Having made the transformation of Eq. (5.2) from the single initial random variable ξ to the three new random variables ζ_i – note that we have dropped the superscript $^{(1)}$ again for clarity – we approximate the solution to the Kraichnan-Orszag problem by Eq. (5.3). When we substitute this expression into Eq. (3.1a) we obtain

$$\sum_{0\leq i+j+k\leq P} \frac{dx_{ijk}^{(1)}}{dt} \phi_i^{(\zeta_1)} \phi_j^{(\zeta_2)} \phi_k^{(\zeta_3)} = \sum_{0\leq p+q+r\leq P} \sum_{0\leq u+v+w\leq P} x_{pqr}^{(2)} x_{uvw}^{(3)} \phi_p^{(\zeta_1)} \phi_q^{(\zeta_2)} \phi_r^{(\zeta_3)} \phi_u^{(\zeta_1)} \phi_v^{(\zeta_2)} \phi_w^{(\zeta_3)} \tag{5.17}$$

We multiply this equation by $\phi_l^{(\zeta_1)} f_{\zeta_1} \phi_m^{(\zeta_2)} f_{\zeta_2} \phi_n^{(\zeta_3)} f_{\zeta_3}$ and perform a triple integration w.r.t. ζ_1, ζ_2 and ζ_3. Taking into account the orthogonality of the basis functions we arrive at

$$\frac{dx_{lmn}^{(1)}}{dt} = \frac{1}{\left\langle {\phi_l^{(\zeta_1)}}^2 \right\rangle \left\langle {\phi_m^{(\zeta_2)}}^2 \right\rangle \left\langle {\phi_n^{(\zeta_3)}}^2 \right\rangle} \sum_{0 \le p+q+r \le P} \sum_{0 \le u+v+w \le P} x_{pqr}^{(2)} x_{uvw}^{(3)} \left\langle \phi_p^{(\zeta_1)} \phi_u^{(\zeta_1)} \phi_l^{(\zeta_1)} \right\rangle \left\langle \phi_q^{(\zeta_2)} \phi_v^{(\zeta_2)} \phi_m^{(\zeta_2)} \right\rangle \left\langle \phi_r^{(\zeta_3)} \phi_w^{(\zeta_3)} \phi_n^{(\zeta_3)} \right\rangle \quad (5.18)$$

for $l, m, n = 0, \ldots, P$ with

$$\langle I(\zeta_i) \rangle = \int_{-\infty}^{\infty} I(\zeta_i) f_{\zeta_i}(\zeta_i) \, \mathrm{d}\zeta_i \quad (5.19)$$

for some function $I(\zeta_i)$. Substituting Eq. (5.3) into Eqs. (3.1b) and (3.1c) gives a result similar to Eq. (5.18). Together these three equations – note that each of these three equations in turn represents a number of scalar-valued DEs – constitute the governing *deterministic* system of differential equations in time for the coefficients $x_{lmn}^{(1)}(t)$, $x_{lmn}^{(2)}(t)$ and $x_{lmn}^{(3)}(t)$ with $0 \le l+m+n \le P$. As noted above, we solve this system by standard fourth-order Runge-Kutta time integration.

5.4 Calculation of mean and variance

We are interested in the mean and variance of $x_1(t, \zeta_1, \zeta_2, \zeta_3)$, $x_2(t, \zeta_1, \zeta_2, \zeta_3)$ and $x_3(t, \zeta_1, \zeta_2, \zeta_3)$. Once we have solved for the time histories of the solution coefficients $x_{lmn}^{(i)}(t)$ (see Eq. (5.3)) the mean and variance of $x_i(t, \zeta_1, \zeta_2, \zeta_3)$ can be calculated as follows.

5.4.1 Mean

The mean of x_i is defined as

$$\bar{x}_i(t) = E\left[x_i(t, \zeta_1, \zeta_2, \zeta_3)\right] \quad (5.20)$$

Substituting Eq. (5.3) into Eq. (5.20) we get

$$\begin{aligned}
\bar{x}_i(t) &= E\left[\sum_{0 \le l+m+n \le P} x_{lmn}^{(i)}(t) \phi_l^{(\zeta_1)}(\zeta_1) \phi_m^{(\zeta_2)}(\zeta_2) \phi_n^{(\zeta_3)}(\zeta_3)\right] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{0 \le l+m+n \le P} x_{lmn}^{(i)}(t) \phi_l^{(\zeta_1)} \phi_m^{(\zeta_2)} \phi_n^{(\zeta_3)} f_{\zeta_1,\zeta_2,\zeta_3} \mathrm{d}\zeta_1 \mathrm{d}\zeta_2 \mathrm{d}\zeta_3 \\
&\approx \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{0 \le l+m+n \le P} x_{lmn}^{(i)}(t) \phi_l^{(\zeta_1)} \phi_m^{(\zeta_2)} \phi_n^{(\zeta_3)} f_{\zeta_1} f_{\zeta_2} f_{\zeta_3} \mathrm{d}\zeta_1 \mathrm{d}\zeta_2 \mathrm{d}\zeta_3 \\
&= \sum_{0 \le l+m+n \le P} x_{lmn}^{(i)}(t) \int_{-\infty}^{\infty} \phi_l^{(\zeta_1)} f_{\zeta_1} \mathrm{d}\zeta_1 \int_{-\infty}^{\infty} \phi_m^{(\zeta_2)} f_{\zeta_2} \mathrm{d}\zeta_2 \int_{-\infty}^{\infty} \phi_n^{(\zeta_3)} f_{\zeta_3} \mathrm{d}\zeta_3 \\
&= x_{000}^{(i)}(t)
\end{aligned} \quad (5.21)$$

Here we have assumed that ζ_1, ζ_2 and ζ_3 are (practically) statistically independent variables. This means we can make the following approximation

$$f_{\zeta_1,\zeta_2,\zeta_3}\left(\zeta_1,\zeta_2,\zeta_3\right) \approx f_{\zeta_1}\left(\zeta_1\right) f_{\zeta_2}\left(\zeta_2\right) f_{\zeta_3}\left(\zeta_3\right) \tag{5.22}$$

Furthermore, note that the last step in Eq. (5.21) follows from the orthogonality of the basis polynomials, i.e.

$$\int_{-\infty}^{\infty} \phi_i^{(\zeta_i)} f_{\zeta_i} \mathrm{d}\zeta_i = \begin{cases} 1 & \text{if} \quad i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.23}$$

5.4.2 Variance

The variance of x_i is defined as

$$\begin{aligned} \mathrm{Var}\left(x_i(t)\right) &= E\left[\left(x_i\left(t,\zeta_1,\zeta_2,\zeta_3\right) - \bar{x}_i(t)\right)^2\right] \\ &= E\left[x_i^2\left(t,\zeta_1,\zeta_2,\zeta_3\right) - 2x_i\left(t,\zeta_1,\zeta_2,\zeta_3\right)\bar{x}_i(t) + \bar{x}_i^2(t)\right] \\ &= E\left[x_i^2\left(t,\zeta_1,\zeta_2,\zeta_3\right)\right] - 2E\left[x_i\left(t,\zeta_1,\zeta_2,\zeta_3\right)\right]\bar{x}_i(t) + \bar{x}_i^2(t) \\ &= E\left[x_i^2\left(t,\zeta_1,\zeta_2,\zeta_3\right)\right] - \bar{x}_i^2(t) \end{aligned} \tag{5.24}$$

Substituting the numerical approximation (Eq. (5.3)) and Eq. (5.21) into the definition of Eq. (5.24) we obtain

$$\begin{aligned} \mathrm{Var}\left(x_i(t)\right) &= E\left[\left(\sum_{0\le l+m+n\le P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\phi_m^{(\zeta_2)}\phi_n^{(\zeta_3)}\right)^2\right] - {x_{000}^{(i)}}^2(t) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\sum_{0\le l+m+n\le P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\phi_m^{(\zeta_2)}\phi_n^{(\zeta_3)}\right)^2 f_{\zeta_1,\zeta_2,\zeta_3}\mathrm{d}\zeta_1\mathrm{d}\zeta_2\mathrm{d}\zeta_3 \\ &\quad - {x_{000}^{(i)}}^2(t) \\ &\approx \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\sum_{0\le l+m+n\le P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\phi_m^{(\zeta_2)}\phi_n^{(\zeta_3)}\right)^2 f_{\zeta_1}f_{\zeta_2}f_{\zeta_3}\mathrm{d}\zeta_1\mathrm{d}\zeta_2\mathrm{d}\zeta_3 \\ &\quad - {x_{000}^{(i)}}^2(t) \\ &= \sum_{0\le l+m+n\le P}\left[{x_{lmn}^{(i)}}^2(t)\int_{-\infty}^{\infty}{\phi_l^{(\zeta_1)}}^2 f_{\zeta_1}\mathrm{d}\zeta_1\int_{-\infty}^{\infty}{\phi_m^{(\zeta_2)}}^2 f_{\zeta_2}\mathrm{d}\zeta_2\int_{-\infty}^{\infty}{\phi_n^{(\zeta_3)}}^2 f_{\zeta_3}\mathrm{d}\zeta_3\right] \\ &\quad - {x_{000}^{(i)}}^2(t) \\ &= \sum_{0\le l+m+n\le P}\left[{x_{lmn}^{(i)}}^2(t)\left\langle{\phi_l^{(\zeta_1)}}^2\right\rangle\left\langle{\phi_m^{(\zeta_2)}}^2\right\rangle\left\langle{\phi_n^{(\zeta_3)}}^2\right\rangle\right] - {x_{000}^{(i)}}^2(t) \end{aligned} \tag{5.25}$$

The fourth line in Eq. (5.25) follows from the orthogonality of the basis polynomials. Also note that again we have made use of the assumption of statistical independence from Eq. (5.22).

5.5 First TDgPC results

To test the TDgPC solution approach presented in this chapter we look at the case $\alpha = 0.99$. In Figs. 4.6 and 4.7 we already compared results for a gPC solution approach to a Monte Carlo analysis for this value of α.

5.5.1 One forced transformation at $t = 0.5$

In Section 5.2 we have seen that calculating the PDFs of the transformed random variables ζ_1, ζ_2 and ζ_3 becomes more complicated when the random variable transformations are not one-to-one. To circumvent this complicating factor for the moment we start off with a calculation with a single *forced* random variable transformation. This transformation is carried out at a preselected instant in time when we know the transformations $\xi \rightarrow \zeta_i$ in Eq. (5.2) to be one-to-one for $i = 1, 2, 3$. We call this instant in time $t = t_1$. So, to keep things relatively simple, we do not yet use a transformation criterion based on the size of the nonlinear terms in the expansion of the solution.

Figs. 5.3 and 5.4 show TDgPC results for one *forced* random variable transformation at $t = 0.5$. From Fig. 5.3 we learn that the calculation of the mean of x_1 with a TDgPC solution approach remains accurate after the random variable transformation. Unfortunately, the result for the variance of x_1 is less promising. From Fig. 5.4 it can be seen that the variance calculated by the TDgPC approach starts to deviate drastically from the correct values after the random variable transformation. Results for x_2 and x_3 (which are not included) have the same character: the mean is accurate, the variance is not. The cause for this inaccurate variance is almost certainly the assumption of statistical independence (Eq. (5.22)) used in Eq. (5.25). Fig. 5.4 gives a strong indication that Eq. (5.22) does not hold for $t_1 = 0.5$.

5.5.2 One forced transformation at other times

At this moment it would be interesting to see whether another choice of t_1 leads to more accurate results for the variance. The assumption of statistical independence of ζ_1, ζ_2 and ζ_3 might be justified for a different value of t_1. In Figs. 5.5 and 5.6 we therefore look at two alternative cases. Fig. 5.5 shows results for the variance of x_1 for an earlier transformation, namely at $t_1 = 0.1$. Fig. 5.6 presents results for a later transformation at $t_1 = 3$. Sadly, we must conclude that for both $t_1 = 0.1$ as well as $t_1 = 3$ the variance is inaccurate.

5.5.3 Difficulties with variance calculation

Having established that the assumption of statistical independence of the transformed random variables (Eq. (5.22)) is not justified we have to revert to the second line in Eq. (5.25) for the calculation of the variance. Hence, we have to calculate the variance from the joint PDF $f_{\zeta_1,\zeta_2,\zeta_3}$ rather than from the three marginal PDFs f_{ζ_1}, f_{ζ_2} and f_{ζ_3}. From the second random variable transformation onwards (Eq. (5.7)) this would not be much of a problem. We have seen in Section 5.2.2 that we must compute the joint PDF anyway. In other words: it is not possible to calculate the marginal PDFs without first determining the

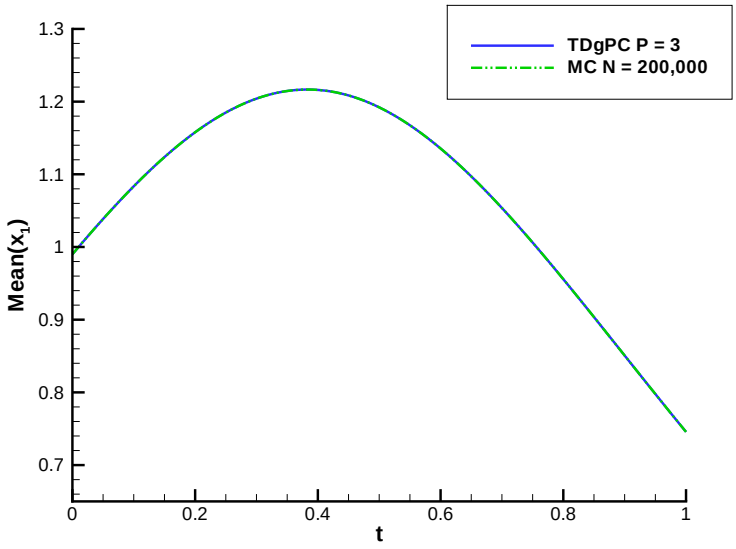

Figure 5.3: Mean of x_1 vs. time for $\alpha = 0.99$ and $t_1 = 0.5$: TDgPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 200,000$)

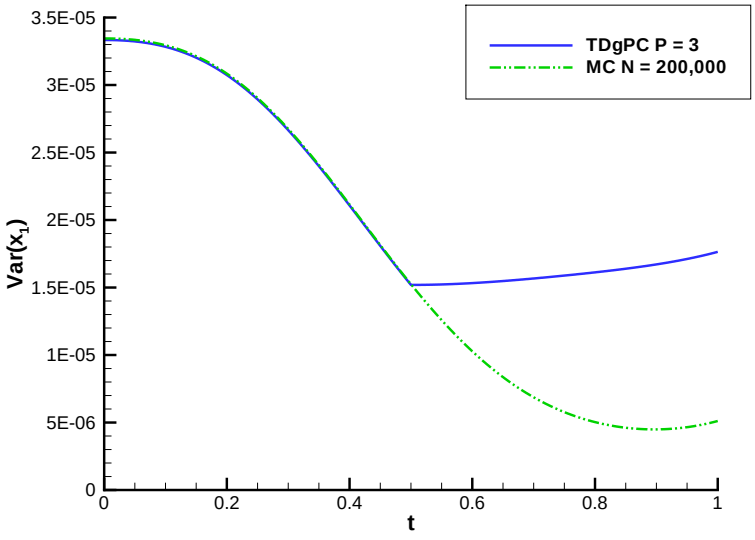

Figure 5.4: Variance of x_1 vs. time for $\alpha = 0.99$ and $t_1 = 0.5$: TDgPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 200,000$)

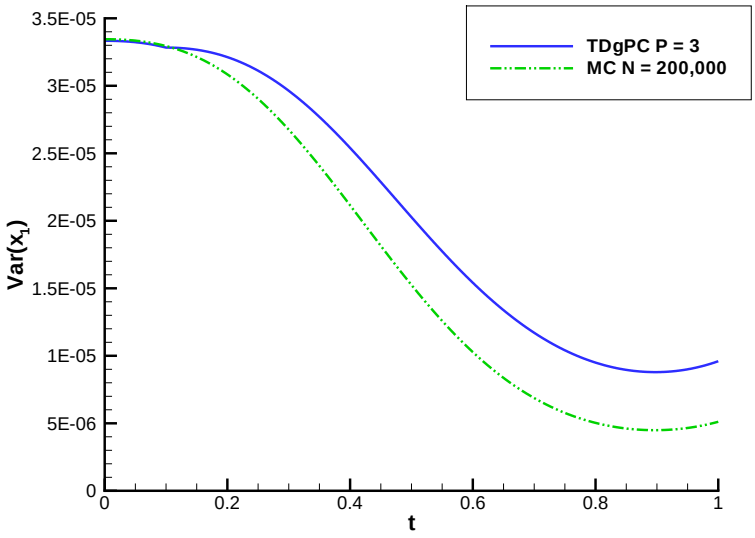

Figure 5.5: Variance of x_1 vs. time for $\alpha = 0.99$ and $t_1 = 0.1$: TDgPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 200,000$)

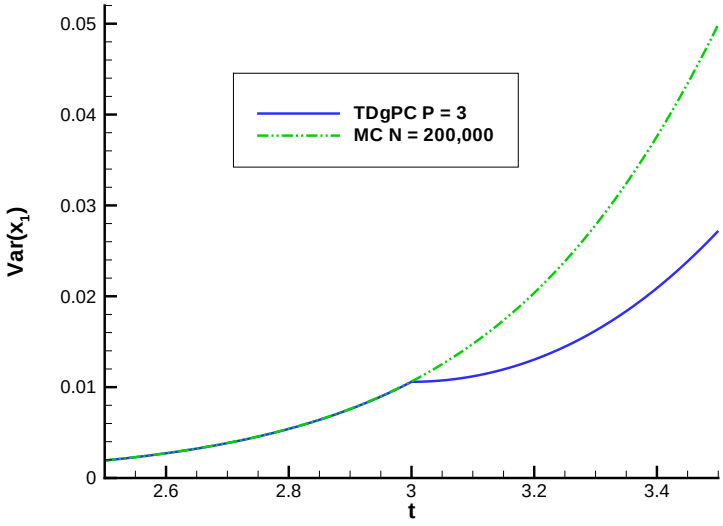

Figure 5.6: Variance of x_1 vs. time for $\alpha = 0.99$ and $t_1 = 3$: TDgPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 200,000$)

joint PDF. At the first random variable transformation, however, we are faced with some difficulty.

Since we start out at $t = 0$ with a one-dimensional random input, the first transformation (Eq. (5.2)) is from one-dimensional to three-dimensional random space. The joint PDF of the transformed variables $f_{\zeta_1,\zeta_2,\zeta_3}$ is presumably not very well-behaved. We can use Eqs. (5.9) and (5.10) to calculate f_{ζ_1}, f_{ζ_2} and f_{ζ_3}, but not $f_{\zeta_1,\zeta_2,\zeta_3}$. Obviously, Eq. (5.15) and Eq. (5.16) do not apply since the transformed random space is of higher dimensionality than the original random space. So, after the first random variable transformation $f_{\zeta_1,\zeta_2,\zeta_3}$ cannot readily be determined. This prevents an accurate calculation of the variance. In the next chapter we discuss a way to overcome this problem.

Chapter 6

Modifications to the TDgPC approach

We have seen that the solution procedure presented in Chapter 5 leads to erroneous results for the variance of the solution (Figs. 5.4–5.6). Apparently, the transformed random variables ζ_1, ζ_2 and ζ_3 are not statistically independent (Eq. (5.22) is not justified). Hence, the calculation of the variance according to Eq. (5.25) leads to erroneous results. Since the joint probability density function $f_{\zeta_1,\zeta_2,\zeta_3}$ cannot readily be determined we need to look for an alternative approach.

6.1 Integration over the original random variable

The integrand in Eq. (5.25) for the variance of x_i is a function of the transformed random variables ζ_1, ζ_2 and ζ_3. These transformed random variables in turn are all functions of the original random variable ξ: $\zeta_1 = Z_1(\xi)$, $\zeta_2 = Z_2(\xi)$ and $\zeta_3 = Z_3(\xi)$. Hence, the integrand in Eq. (5.25) can also be seen as a function solely dependent on ξ. To avoid the calculation of $f_{\zeta_1,\zeta_2,\zeta_3}$ we can transform the triple integral over ζ_1, ζ_2 and ζ_3 in Eq. (5.25) to a single integral over ξ.

We do this by recognizing that the following relation should be valid for every realisable point $(\zeta_1^*, \zeta_2^*, \zeta_3^*)$

$$f_{\zeta_1,\zeta_2,\zeta_3}\left(\zeta_1^*, \zeta_2^*, \zeta_3^*\right) \mathrm{d}\zeta_1 \mathrm{d}\zeta_2 \mathrm{d}\zeta_3 = \sum_{\xi^*} f_\xi\left(\xi^*\right) \mathrm{d}\xi \tag{6.1}$$

where the summation is over all points ξ^* for which $Z_1(\xi^*) = \zeta_1^*$, $Z_2(\xi^*) = \zeta_2^*$ and $Z_3(\xi^*) = \zeta_3^*$. Eq. (6.1) merely states that, given the transformation $\xi \rightarrow (\zeta_1, \zeta_2, \zeta_3)$, the probability that $(\zeta_1, \zeta_2, \zeta_3)$ lies within an infinitesimal volume around $(\zeta_1^*, \zeta_2^*, \zeta_3^*)$ should be equal to the probability that ξ lies within the (possibly multiple) corresponding infinitesimal interval(s) around ξ^*. It follows that the following relation should then also be valid

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \ldots f_{\zeta_1,\zeta_2,\zeta_3} \mathrm{d}\zeta_1 \mathrm{d}\zeta_2 \mathrm{d}\zeta_3 = \int_{-1}^{1} \ldots f_\xi \mathrm{d}\xi \tag{6.2}$$

Hence, with the help of Eq. (6.2) we can calculate the variance of x_i according to

$$\begin{aligned}\mathrm{Var}\left(x_i(t)\right) &= E\left[\left(\sum_{0\leq l+m+n\leq P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\phi_m^{(\zeta_2)}\phi_n^{(\zeta_3)}\right)^2\right] - \bar{x}_i^2(t)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\sum_{0\leq l+m+n\leq P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\phi_m^{(\zeta_2)}\phi_n^{(\zeta_3)}\right)^2 f_{\zeta_1,\zeta_2,\zeta_3}\mathrm{d}\zeta_1\mathrm{d}\zeta_2\mathrm{d}\zeta_3 - \bar{x}_i^2(t)\\ &= \int_{-1}^{1}\left(\sum_{0\leq l+m+n\leq P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\left(Z_1(\xi)\right)\phi_m^{(\zeta_2)}\left(Z_2(\xi)\right)\phi_n^{(\zeta_3)}\left(Z_3(\xi)\right)\right)^2 f_\xi(\xi)\mathrm{d}\xi\\ &\quad - \bar{x}_i^2(t)\end{aligned} \tag{6.3}$$

Transforming an integral over the transformed random variables to an integral over the original random variable is a technique that can be used to good advantage more often than just for the calculation of the variance of x_i. We can also circumvent the approximation in Eq. (5.22) in the calculation of the mean of the solution. This leads to

$$\begin{aligned}\bar{x}_i(t) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{0\leq l+m+n\leq P} x_{lmn}^{(i)}(t)\phi_l^{(\zeta_1)}\phi_m^{(\zeta_2)}\phi_n^{(\zeta_3)} f_{\zeta_1,\zeta_2,\zeta_3}\mathrm{d}\zeta_1\mathrm{d}\zeta_2\mathrm{d}\zeta_3\\ &= \sum_{0\leq l+m+n\leq P} x_{lmn}^{(i)}(t)\int_{-1}^{1}\phi_l^{(\zeta_1)}\left(Z_1(\xi)\right)\phi_m^{(\zeta_2)}\left(Z_2(\xi)\right)\phi_n^{(\zeta_3)}\left(Z_3(\xi)\right) f_\xi(\xi)\mathrm{d}\xi\end{aligned} \tag{6.4}$$

Furthermore, we can just as well transform a *single* integral over a transformed random variable to a single integral over the original random variable. For similarly to Eq. (6.1) we can also say that

$$f_{\zeta_i}\left(\zeta_i^*\right)\mathrm{d}\zeta_i = \sum_{\xi^*} f_\xi\left(\xi^*\right)\mathrm{d}\xi \tag{6.5}$$

so

$$\int_{-\infty}^{\infty}\ldots f_{\zeta_i}\mathrm{d}\zeta_i = \int_{-1}^{1}\ldots f_\xi\mathrm{d}\xi \tag{6.6}$$

With the help of Eq. (6.6) we can transform all integrals needed for the determination of the governing system of differential equations (Eqs. (5.18) and (5.19)) to integrals over the original random variable ξ. The integrals in the Gram-Schmidt orthogonalization algorithm (to calculate the orthogonal polynomials $\phi_p^{(\zeta_i)}\left(\zeta_i\right)$) can similarly be transformed to integrals over ξ.

To conclude, we make the following important point. Performing all integrations in ξ-space has a major advantage: there is no need to calculate the probability density functions of the transformed random variables anymore! This means we do not have to deal with all the complications described in Section

5.2. Furthermore, since the calculation of the transformed PDFs accounts for a large portion of the total computation time, we observe a dramatic increase in computation speed.

6.1.1 Results after modification

Performing all integrations in ξ-space and calculating the variance according to Eq. (6.3) hopefully fixes the problem of inaccurate results for the variance after the random variable transformation (see Fig. 5.4). We check this in Fig. 6.1. We see that the variance is now indeed accurately calculated for $t > t_1 = 0.5$.

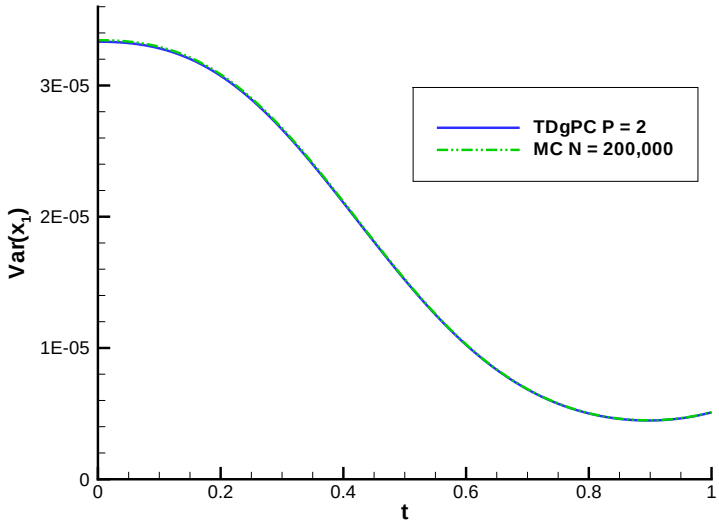

Figure 6.1: Variance of x_1 vs. time for $\alpha = 0.99$ and $t_1 = 0.5$: TDgPC solution with $P = 2$ compared to a Monte Carlo analysis ($N = 200,000$)

With this issue resolved the question now rises whether the TDgPC solution approach actually leads to favourable results as compared to a standard gPC approach. In other words, do these random variable transformations (Eqs. (5.2) and (5.8)) result in a calculated solution that remains accurate for larger values of time? In Figs. 6.2 and 6.3 we present results for the mean and variance of x_1 over a larger time scale. A comparison is made between a TDgPC and a gPC solution approach. Results from a Monte Carlo analysis are also shown for reference. The TDgPC results were calculated by performing random variable transformations at t-intervals of 0.5 starting at $t = 0.5$. This way the nonlinear part of the solution expansion remains relatively small. Sadly, we must conclude form Figs. 6.2 and 6.3 that the TDgPC results presented are, contrary to our anticipation, even less accurate than the gPC solution shown. Also, the TDgPC solution is seen to blow up around $t = 13$.

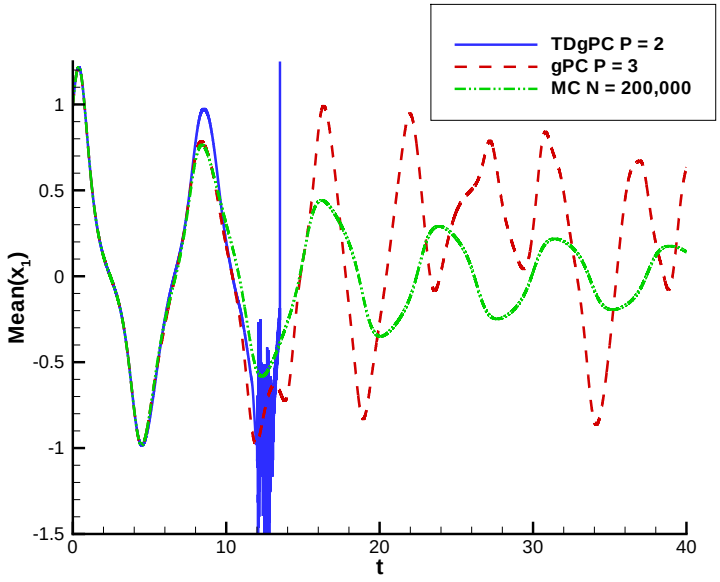

Figure 6.2: Mean of x_1 vs. time for $\alpha = 0.99$: TDgPC solution with $P = 2$ compared to a gPC solution with $P = 3$ and a Monte Carlo analysis ($N = 200,000$)

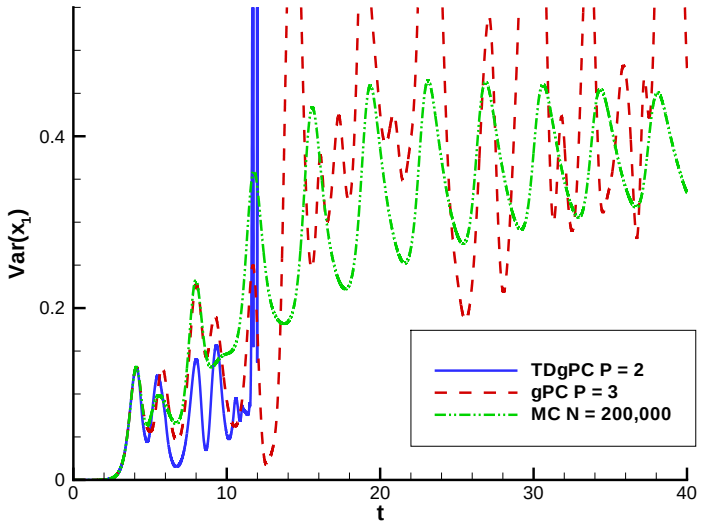

Figure 6.3: Variance of x_1 vs. time for $\alpha = 0.99$: TDgPC solution with $P = 2$ compared to a gPC solution with $P = 3$ and a Monte Carlo analysis ($N = 200,000$)

6.2 Transforming every time step

Do the results from Figs. 6.2 and 6.3 mean we have to close the curtain on the TDgPC solution approach to the Kraichnan-Orszag problem? Not quite yet. We can, of course, increase the maximum degree of the expansion polynomials and see what this brings us. But first, let us investigate another option.

Remember that the transformation of all integrals to ξ-space led to a huge reduction of the computation time. Since we do not need to calculate new PDFs with a random variable change anymore, we are now in a position to transform the random variables much more often. We can let go of our original idea of performing a random variable change every time the nonlinear part of the solution reaches a certain threshold value. In fact, it now has become practically feasible to execute a random variable transformation at every time step! The following section presents some results with this modification to the TDgPC solution approach.

6.2.1 Results after modification

Figs. 6.4 and 6.5 show results for the mean and variance, respectively, of x_1 calculated using the aforementioned modification to the TDgPC solution approach (again, results for x_2 and x_3 can be found in Appendix A.3). Adjusting the random variables and corresponding expansion polynomials at every time step can be seen to have a dramatic effect on the accuracy of the solution. Not long after $t = 13$ the results generated with a gPC solution approach stop to bear any resemblance to the correct solution. However, using a TDgPC solution approach with expansion polynomials having a maximum degree of only two ($P = 2$) already leads to pretty accurate results. The calculated solution can be seen to have the same characteristics as the results from the Monte Carlo analysis for the entire range of t displayed. Increasing the maximum degree of the expansion polynomials to $P = 3$ can be seen to results in a significantly higher accuracy. In fact, the TDgPC results with $P = 3$ are graphically indistinguishable from the Monte Carlo results on the scale of these plots.

In Figs. 6.6 and 6.7 the time evolution of the 'error' in the mean and variance, respectively, of x_1 is shown for various values of P. We write error with quotation marks since the exact solution of the problem is unknown. So, we rather take the results from a Monte Carlo analysis with 200,000 samples as a reference. Hence, we calculate the 'error' in the mean according to

$$\epsilon_{\bar{x}_i}(t) = \bar{x}_i^{TDgPC}(t) - \bar{x}_i^{MC}(t) \tag{6.7}$$

and likewise for the error in the variance. The error plots more clearly show the accuracy we gain by going from $P = 2$ to $P = 3$. The accuracy of a TDgPC solution with $P = 4$ can be seen to be almost exactly equal to a solution with $P = 3$. Presumably, the higher accuracy that is to be expected from the use of higher degree expansion polynomials is nullified due to inaccuracies in the numerical time and/or random space integration that now become the dominant source of error. Also, remember that our reference solution is not exact. So, it may be that the TDgPC results for $P = 3$ and $P = 4$ already lie within the margin of error of this the Monte Carlo reference solution. Then it would make little sense to compare the accuracy of these TDgPC solutions on the basis of Eq. (6.7).

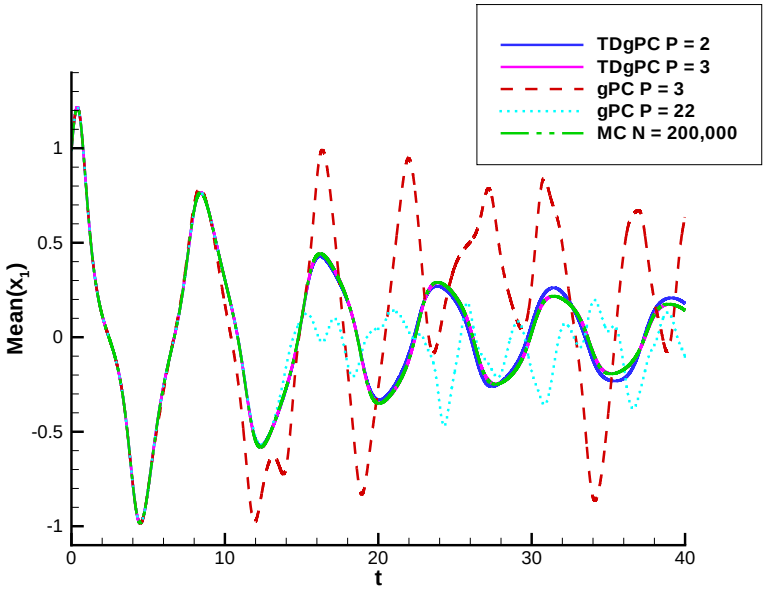

Figure 6.4: Mean of x_1 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$ and $P = 3$ compared to gPC solutions with $P = 3$ and $P = 22$ and a Monte Carlo analysis ($N = 200,000$)

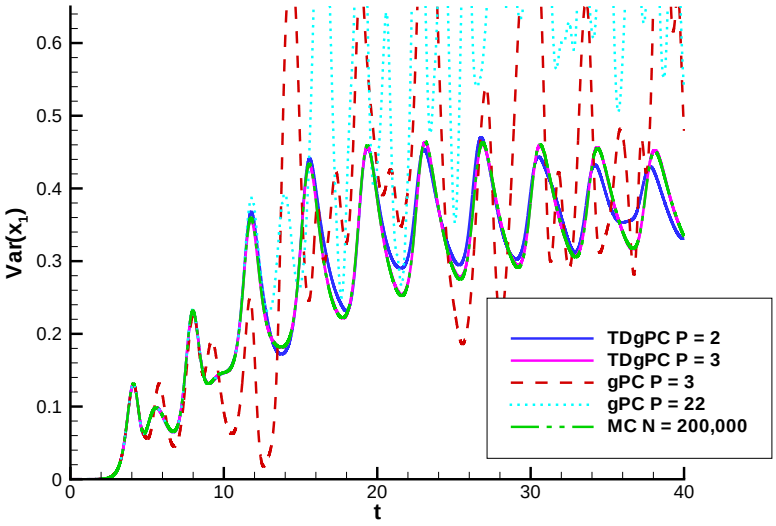

Figure 6.5: Variance of x_1 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$ and $P = 3$ compared to gPC solution with $P = 3$ and $P = 22$ and a Monte Carlo analysis ($N = 200,000$)

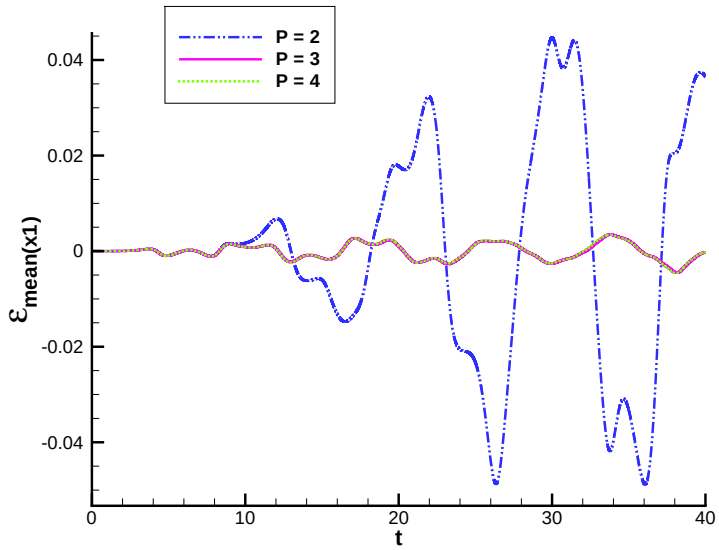

Figure 6.6: Error in the mean of x_1 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$, $P = 3$ and $P = 4$

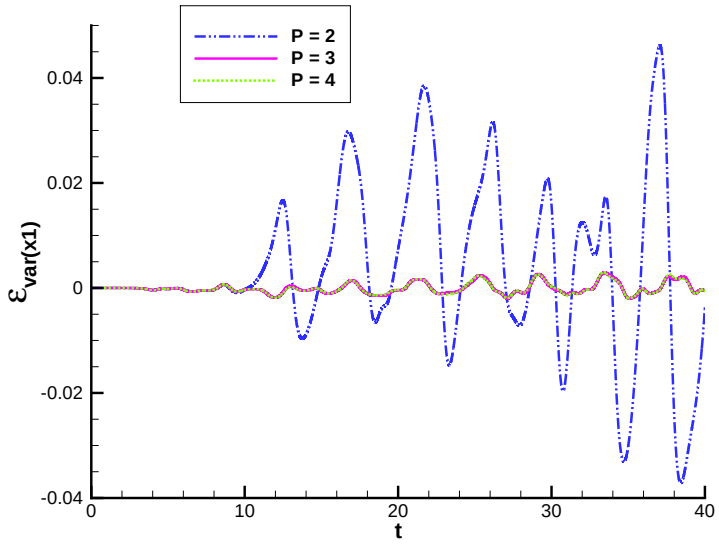

Figure 6.7: Error in the variance of x_1 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$, $P = 3$ and $P = 4$

6.3 Modified Gram-Schmidt algorithm

Up to this point all TDgPC results presented were calculated using a classical Gram-Schmidt algorithm to compute the expansion polynomials. Thus, taking $\phi_0 = 1$, all higher order monic orthogonal polynomials are generated recursively according to

$$\phi_p(\zeta) = \zeta^p - \sum_{k=0}^{p-1} c_{pk}\phi_k(\zeta) \ , \quad p = 1, 2, \ldots, P \tag{6.8}$$

with

$$c_{pk} = \frac{\langle \zeta^p \phi_k(\zeta) \rangle}{\langle \phi_k^2(\zeta) \rangle} \tag{6.9}$$

Note that $c_{pk}\phi_k$ is the orthogonal projection of ζ^p onto ϕ_k. Hence, the summation in Eq. (6.8) removes the components of ζ^p in the direction of all lower order expansion polynomials $\phi_0, \phi_1, \ldots, \phi_{k-1}$. As discussed in Section 6.1 the ensemble averages $\langle \ldots \rangle$ are numerically evaluated in ξ-space.

It is well known (see, for instance, [19]) that the classical Gram-Schmidt algorithm presented in Eqs. (6.8) and (6.9) is numerically unstable, i.e. finite-precision arithmetic can result in a computed set of polynomials $(\phi_0, \phi_1, \ldots, \phi_P)$ that is not completely orthogonal. Fortunately, numerical stability can be achieved in a number of ways.

We can, for instance, incorporate a reorthogonalization step (see [19]) in the Gram-Schmidt algorithm: we feed back the first series of calculated 'orthogonal' polynomials into Eqs. (6.8) and (6.9) (replacing the monomials ζ^p) and carry out the orthogonalization process a second time. Since the first series of calculated polynomials is already close to orthogonal, the second series of polynomials will be truly orthogonal. However, the reorthogonalization step obviously doubles the computing time required.

Another, computationally cheaper way to achieve numerical stability is the modified Gram-Schmidt algorithm introduced below (see, for instance, [20]). To compute a set of $P+1$ orthogonal polynomials ϕ_p both the classical as well as the modified Gram-Schmidt algorithm start out with a set of monomials $\left(1, \zeta, \zeta^2, \ldots, \zeta^P\right)$. The classical algorithm of Eqs. (6.8) and (6.9) first considers ϕ_1 and updates this polynomial to become orthogonal to all polynomials of *lower* degree. This process is then repeated all the way up to ϕ_P. Following the modified algorithm, on the other hand, we take the lowest degree polynomial ϕ_0 and update all polynomials of *higher* degree $\phi_1, \ldots, \phi_P$ to become orthogonal to ϕ_0. We proceed by making the updated polynomials $\phi_2, \ldots, \phi_P$ orthogonal to the updated ϕ_1. This process is repeated until, in the final step, we only have to update ϕ_P to become orthogonal to ϕ_{P-1}. Using exact arithmetic the orthogonal polynomials calculated according to both algorithms would be identical. The modified algorithm, however, is less sensitive to rounding errors and produces more accurate results using finite-precision arithmetic.

Do we get more accurate TDgPC results if we use the modified Gram-Schmidt algorithm instead of the classical one? Does an extra reorthogonalization step improve the accuracy? In Figs. 6.8 and 6.9 we show the errors in TDgPC calculations with $P = 2$ using the three different options above for the Gram-Schmidt orthogonalization. At least for $P = 2$ we do not observe any difference between the various options. However, it could be that the modified

Gram-Schmidt algorithm proves to be more accurate for computations employing basis polynomials of higher maximum degree. In any case, it does not hurt to use the modified algorithm instead of the classical one.

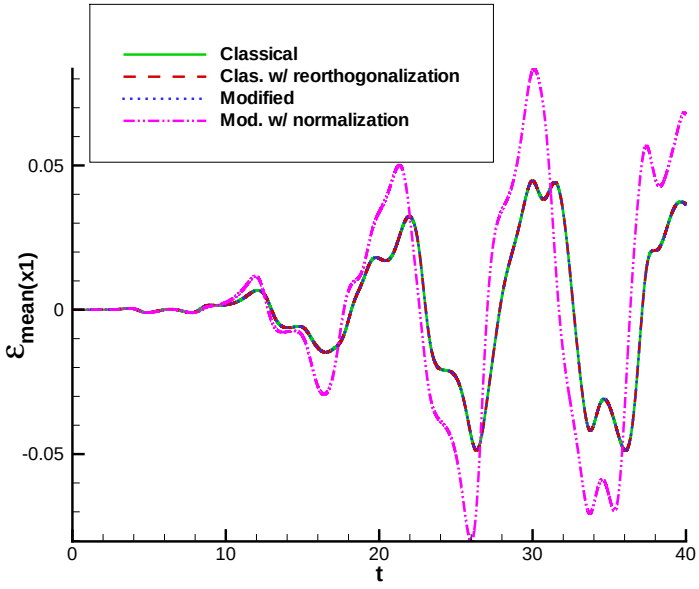

Figure 6.8: Error in the mean of x_1 vs. time for $\alpha = 0.99$: TDgPC results using various Gram-Schmidt orthogonalization algorithms

For both the classical as well as the modified Gram-Schmidt algorithm we have the possibility of adding a normalization step. This results in expansion polynomials that are not only orthogonal but also have the property

$$\left\langle \phi_p^2(\zeta) \right\rangle = 1 \ , \quad (p = 1, 2, \ldots, P) \tag{6.10}$$

Hence, the expansion polynomials are ortho*normal*. As can be seen in Figs. 6.8 and 6.9 such a normalization step has a negative influence on the accuracy of the results.

Witteveen and Bijl [7] suggest that the Gram-Schmidt orthogonalization can be done efficiently by first calculating the ensemble averages of the monomials ζ^p, for $p = 1, 2, \ldots, 2P$. All more complicated ensemble averages, like, for instance, in Eq. (6.9), can then be calculated by a summation of the ensemble averages of the monomials ζ^p. For our purposes this works fine for $P = 2$. However, if we set $P = 3$ this strategy leads to the breakdown of the orthogonalization algorithm due to rounding errors. $\left\langle \phi_3^2(\zeta) \right\rangle$ in Eq. (6.9), for instance, is typically very small for our problem. Using the strategy of Witteveen and Bijl in combination with double precision arithmetic, this ensemble average is sometimes calculated to be zero, leading to a division by zero and the breakdown of the algorithm. Luckily, the approach of calculating all ensemble averages directly, rather than as the summation of the ensemble averages of monomials, does not suffer from this problem.

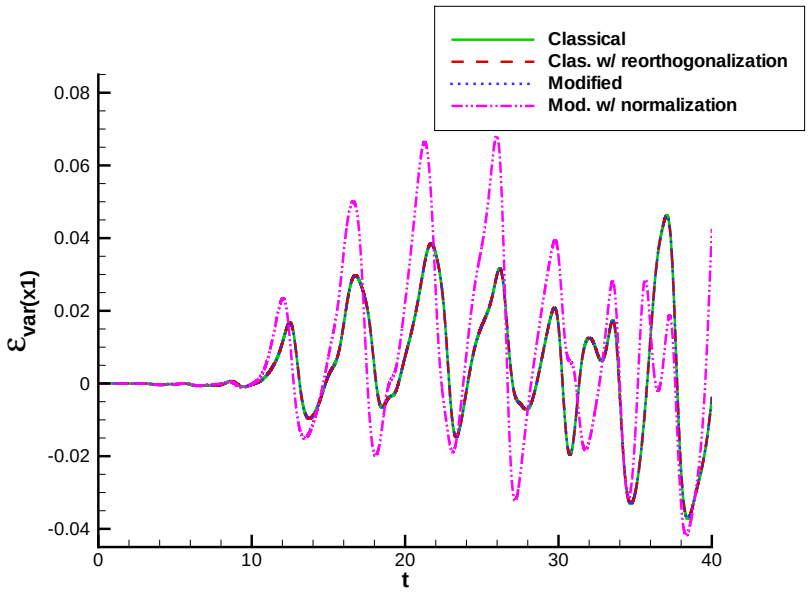

Figure 6.9: Error in the variance of x_1 vs. time for $\alpha = 0.99$: TDgPC results using various Gram-Schmidt orthogonalization algorithms

6.4 Effect of integration parameters

In this section we examine the effect of the number of Gauss-Lobatto-Legendre quadrature points Q (for integration in the random space) and the time step Δt on the accuracy of a TDgPC solution of the Kraichnan-Orszag problem.

6.4.1 Number of quadrature points

We have mentioned before that we use Gauss-Lobatto-Legendre numerical integration [16] to evaluate all integrals in random space. In Chapter 4 we noted that for a plain gPC solution approach taking a number of quadrature points $Q \geq \frac{3}{2}P + \frac{3}{2}$ results in exact integration in the random space. This follows from the fact that all integrands are polynomials with a maximum degree of $3P$. For a TDgPC solution approach, however, the situation is different.

In our TDgPC algorithm all integrations in the random space are executed in terms of the original random variable ξ. After the first random variable transformation (Eq. (5.2)) new expansion polynomials of maximum degree P are determined in terms of the transformed random variables $\zeta_1^{(1)}$, $\zeta_2^{(1)}$ and $\zeta_3^{(1)}$, which are themselves polynomials of degree P in terms of ξ. Hence, the expansion polynomials after the first random variable transformation can also be seen as polynomials of maximum degree P^P in terms of ξ. By the same reasoning we find that the expansion polynomials after the kth random variable transformation can both be seen as polynomials of maximum degree P in terms of $\zeta_1^{(k)}$, $\zeta_2^{(k)}$ and $\zeta_3^{(k)}$ as well as polynomials of maximum degree P^{kP} in terms of ξ. So, after k random variable transformations we need to evaluate integrals of

polynomials with a maximum degree of $3P^{kP}$ in ξ-space. Obviously, employing enough quadrature points to evaluate these integrals exactly ($Q \geq \frac{3}{2}P^{kP} + \frac{3}{2}$) ceases to be practically feasible after only a couple of random variable transformations. We will have to settle for less quadrature points and approximate the required integrals rather than to evaluate them exactly. However, we expect this approximation to be very good as long as we increase the number of quadrature points somewhat over the values of Q we used for the gPC calculations.

In Figs. 6.10 and 6.11 we compare the error evolution of TDgPC calculations with $P = 2$ for various numbers of quadrature points. We see that taking $Q = 10$ is clearly not sufficient. Setting $Q = 20$ shows a big improvement; $Q = 30$ is perhaps still a little bit better. We also see that increasing Q above a value of (approximately) 30 has no additional effect and does not lead to a further reduction of the error.

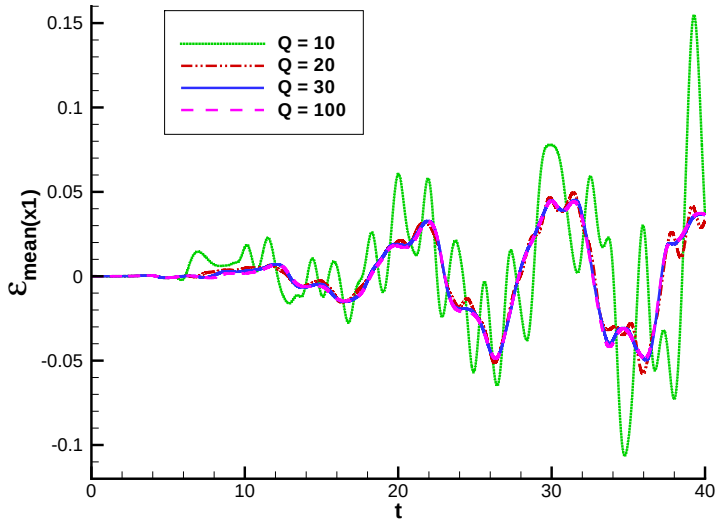

Figure 6.10: Error in the mean of x_1 vs. time for $\alpha = 0.99$: TDgPC results with $P = 2$ using various amounts of quadrature points Q

6.4.2 Time step

In Figs. 6.12 and 6.13 we investigate the effect of the time step on the error evolution of TDgPC calculations with $P = 3$. Note that we employ a semilog-scale in these figures so that we can better discriminate between the various curves. We see that reducing the time step from $\Delta t = 0.1$ to $\Delta t = 0.01$ results in a large improvement in accuracy. A further reduction of the time step to $\Delta t = 0.001$ results in an even more accurate solution, albeit that the gain in accuracy is smaller. Up to this point all gPC and TDgPC results presented in this work were calculated using $\Delta t = 0.001$.

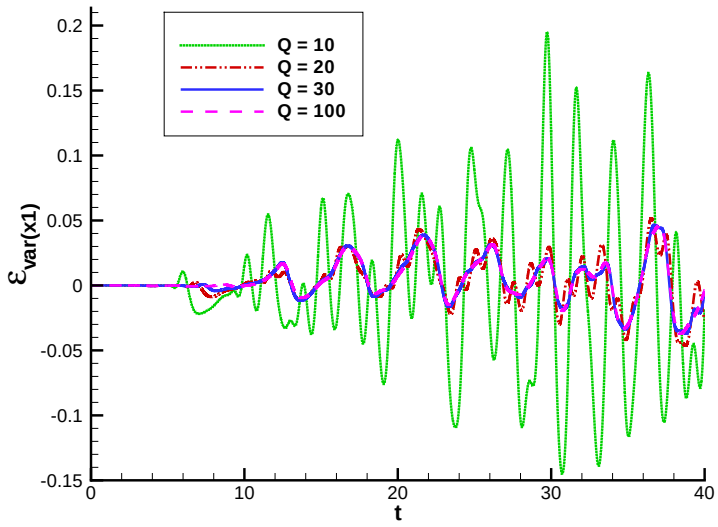

Figure 6.11: Error in the variance of x_1 vs. time for $\alpha = 0.99$: TDgPC results with $P = 2$ using various amounts of quadrature points Q

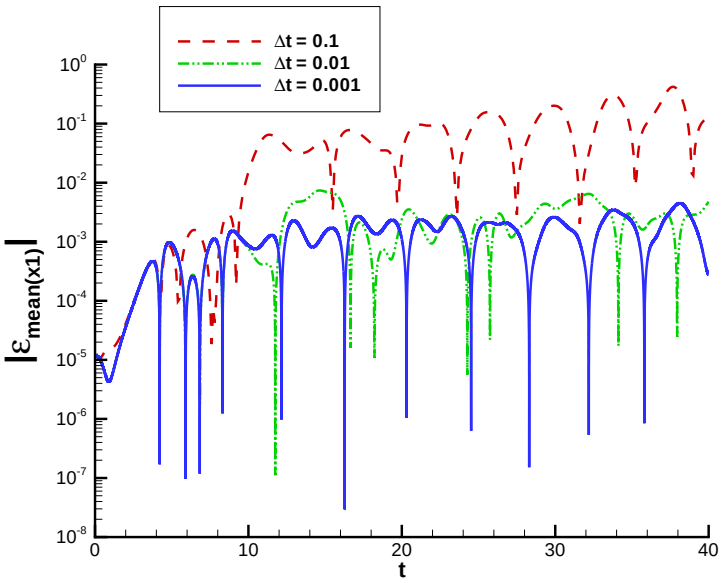

Figure 6.12: Error in the mean of x_1 vs. time for $\alpha = 0.99$: TDgPC results with $P = 3$ for various time steps Δt

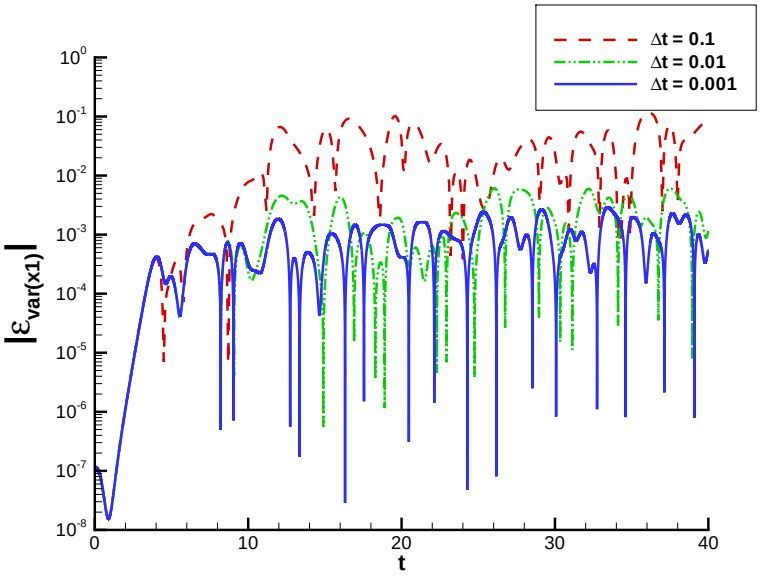

Figure 6.13: Error in the variance of x_1 vs. time for $\alpha = 0.99$: TDgPC results with $P = 3$ for various time steps Δt

6.5 Distribution functions

Statistical moments of the solution $(x_1(t), x_2(t), x_3(t))$ like the mean or the variance can be computed from the solution coefficients $x^{(i)}_{lmn}(t)$ only, as we have seen in Eqs. (6.4) and (6.3). But suppose now that we are interested in the distribution functions of x_1, x_2 and x_3 at a certain time level t_k. Remember that we calculate the solution in terms of polynomials of the random variables $\zeta_1^{(k)}$, $\zeta_2^{(k)}$ and $\zeta_3^{(k)}$ (see Eq. (5.3)) which are transformed with every time step. We noted before that with our TDgPC algorithm there is no need to compute the probability density functions of these transformed random variables. But how then can the distribution functions of x_1 x_2 and x_3 be determined without knowledge of the PDFs of the transformed random variables?

One probability density function that *is* known is the PDF of the original random variable ξ. In Section 6.4.1 we already discussed that all transformed random variables $\zeta_1^{(k)}$, $\zeta_2^{(k)}$ and $\zeta_3^{(k)}$ are ultimately polynomial functions of ξ. These polynomial functions get very complex after only a couple of time steps, so it is unpractical to exactly keep track of these functional relationships between the transformed and the original random variables. Rather, we only calculate the transformed random variables at a limited number of points ξ_q $(q = 1 \ldots Q)$: the Gauss-Lobatto-Legendre quadrature points. With the help of Eq. (5.3) we can now construct graphs like Fig. 6.14: the value of x_1 corresponding to the points ξ_q at a certain instant in time ($t = 6$ in this figure).

Since the exact function $x_1(\xi)$ is too complex we interpolate between the points $(\xi_q, x_1(\xi_q))$. In Fig. 6.14 we compare two interpolation methods: a polynomial fit of degree 10 and a natural cubic spline. Polynomial fitting is

computationally cheaper but can be seen to lead to undesired wiggles. The natural cubic spline interpolation results in a far more natural fit. From the interpolation function – preferably the natural cubic spline – and the PDF of ξ we can now calculate the distribution function of x_1, denoted by $F_{x_1}(x_1)$, at any desired point. A plot of $F_{x_1}(x_1)$ at $t = 6$ can be found in Fig. 6.15.

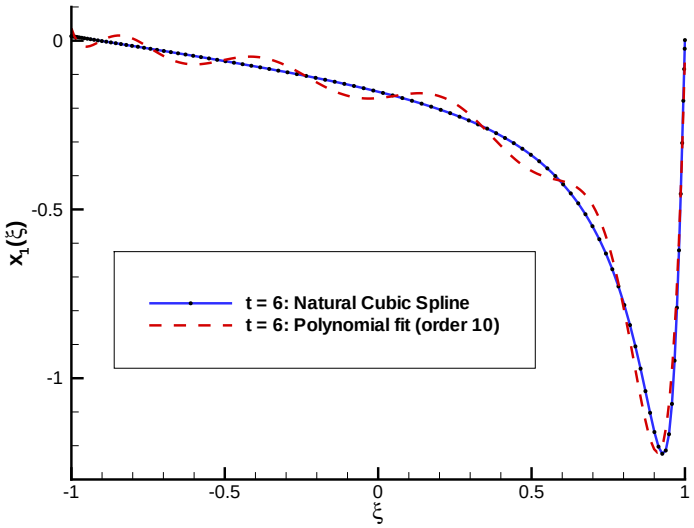

Figure 6.14: TDgPC calculation for $\alpha = 0.99$ with $P = 2$: x_1 as a function of ξ at $t = 6$

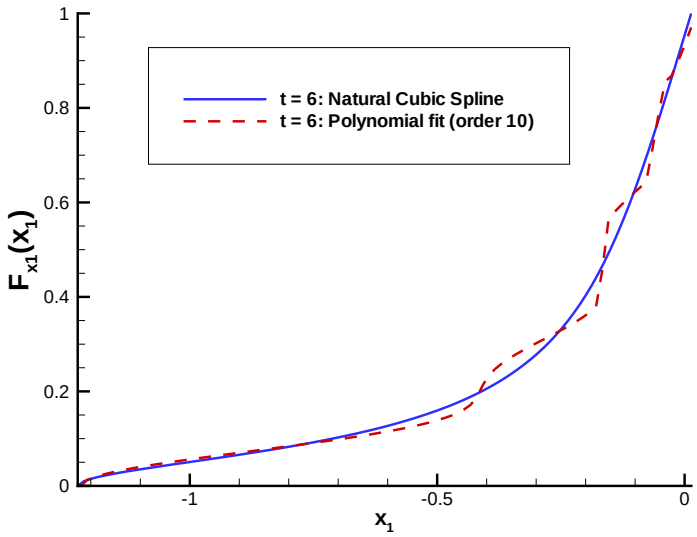

Figure 6.15: TDgPC calculation for $\alpha = 0.99$ with $P = 2$: Distribution function of x_1 at $t = 6$

6.6 Results for $\alpha = 0.995$

In this section we look at TDgPC results for $\alpha = 0.995$ in Eq. (3.2). We have seen in Fig. 4.1 that for $x_1(0) > 1$ the solution curve belongs to a completely different branch of trajectories than for $x_1(0) < 1$. Note that $P(x_1(0) > 1) = 0.25$ and $P(x_1(0) < 1) = 0.75$ for $\alpha = 0.995$. Hence, this is a challenging problem and we have seen in Figs. 4.9 and 4.10 that a plain gPC solution breaks down very quickly.

In Figs. 6.16 and 6.17 TDgPC results are presented for the mean and variance of x_1, respectively. We again compare the TDgPC solutions with results from a Monte Carlo simulation. Also for $\alpha = 0.995$ TDgPC can be seen to produce very decent results. The accuracy of the solution with $P = 2$ is comparable to the case $\alpha = 0.99$. Again there is a significant improvement going from $P = 2$ to $P = 3$. However, the solution for $P = 3$ is not quite as accurate as in the case $\alpha = 0.99$, presumably due to the higher complexity of the problem with $\alpha = 0.995$. In Figs. 6.18 and 6.19 we consider the 'error' in the mean and variance of x_1, respectively, taking the Monte Carlo simulation with $N = 100,000$ as a reference. These figures more clearly show that for $\alpha = 0.995$, going from $P = 2$ to $P = 3$, the gain in accuracy is not as large as for the case $\alpha = 0.995$ (cf. Figs. 6.6 and 6.7).

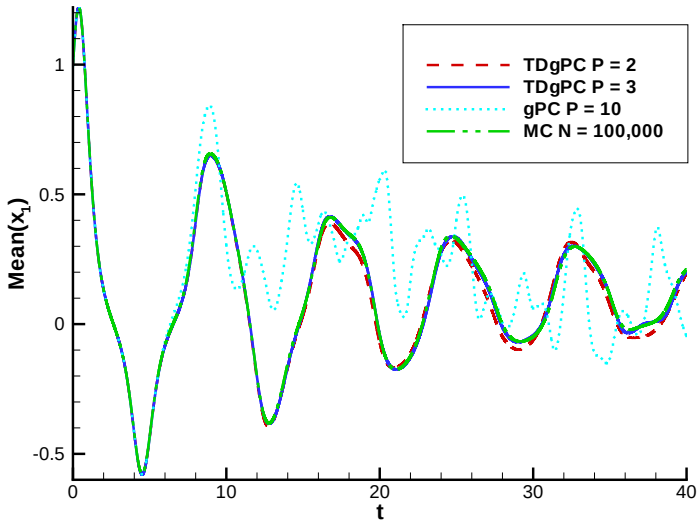

Figure 6.16: Mean of x_1 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$ compared to a gPC solution with $P = 10$ and a Monte Carlo analysis ($N = 100,000$)

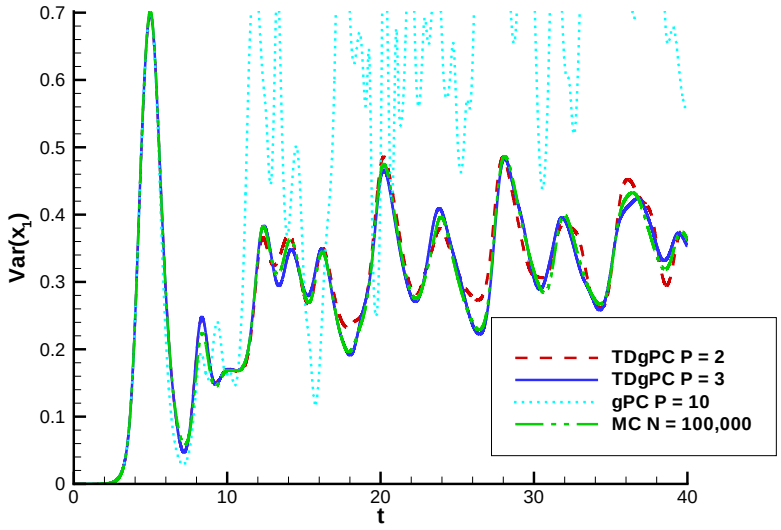

Figure 6.17: Variance of x_1 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$ compared to a gPC solution with $P = 10$ and a Monte Carlo analysis ($N = 100,000$)

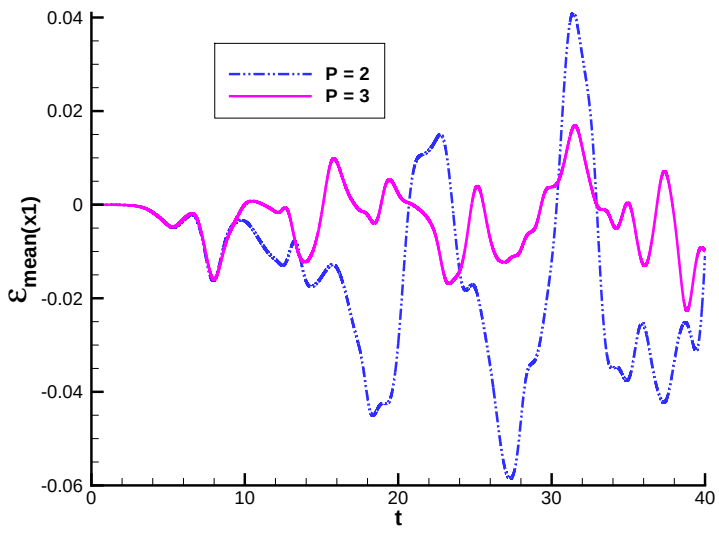

Figure 6.18: Error in the mean of x_1 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$

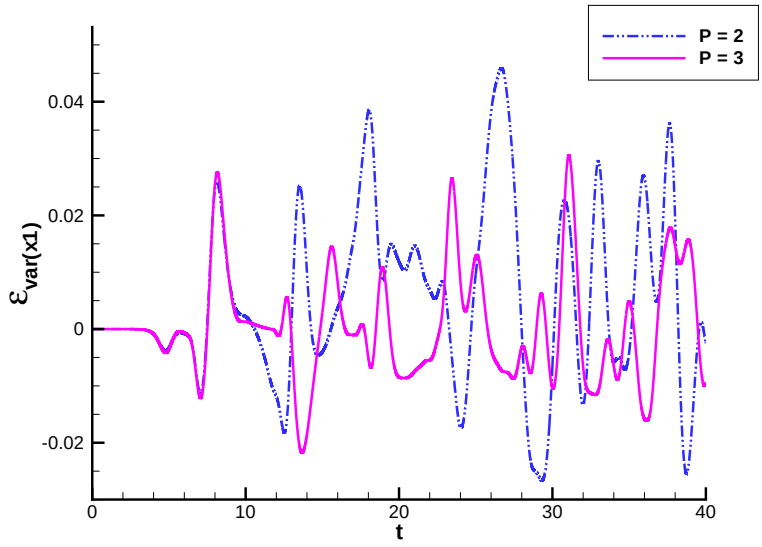

Figure 6.19: Error in the variance of x_1 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$

Chapter 7

A three-dimensional random space

In the previous chapters we have studied the Kraichnan-Orszag problem (Eq. (3.1)) subject to initial conditions depending on a single random variable (Eq. (3.2)). Another interesting set of initial conditions is the following

$$x_1(0) = \alpha + 0.01\xi_1 \ , \quad x_2(0) = \beta + 0.01\xi_2 \ , \quad x_3(0) = \gamma + 0.01\xi_3 \tag{7.1}$$

where α, β and γ are constants and ξ_1, ξ_2 and ξ_3 are uniformly distributed random variables on the interval $[-1, 1]$. ξ_1, ξ_2 and ξ_3 are statistically independent. Can the TDgPC solution approach discussed in Chapters 5 and 6 also be successfully applied to Eq. (3.1) subject to Eq. (7.1)? In this chapter we study this problem taking $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$.

7.1 Alterations to the solution approach

Conceptually we do not need to adapt the solution approach very much. Of course we now directly start out with a three-dimensional expansion in terms of ξ_1, ξ_2 and ξ_3 analogous to Eq. (5.3). We introduce transformed random variables according to Eq. (5.8) and calculate new expansion polynomials just like for the single random variable case. Also similar to the singular random variable case we transform all integrals occurring in the solution algorithm to integrals over the original independent random variables ξ_1, ξ_2 and ξ_3. Since we now have three original random variables instead of one Eq. (6.1) is rewritten as

$$\begin{aligned} f_{\zeta_1,\zeta_2,\zeta_3}\left(\zeta_1^*, \zeta_2^*, \zeta_3^*\right) \mathrm{d}\zeta_1 \mathrm{d}\zeta_2 \mathrm{d}\zeta_3 &= \sum_{\left(\xi_1^*, \xi_2^*, \xi_3^*\right)} f_{\xi_1,\xi_2,\xi_3}\left(\xi_1^*, \xi_2^*, \xi_3^*\right) \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3 \\ &= \sum_{\left(\xi_1^*, \xi_2^*, \xi_3^*\right)} f_{\xi_1}\left(\xi_1^*\right) f_{\xi_2}\left(\xi_2^*\right) f_{\xi_3}\left(\xi_3^*\right) \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3 \end{aligned} \tag{7.2}$$

for every realisable point $(\zeta_1^*, \zeta_2^*, \zeta_3^*)$. The summation in Eq. (7.2) is over all points $(\xi_1^*, \xi_2^*, \xi_3^*)$ for which $Z_1\left(\xi_1^*, \xi_2^*, \xi_3^*\right) = \zeta_1^*$, $Z_2\left(\xi_1^*, \xi_2^*, \xi_3^*\right) = \zeta_2^*$ and $Z_3\left(\xi_1^*, \xi_2^*, \xi_3^*\right) = \zeta_3^*$. Note that in Eq. (7.2) we have made use of the statistical

independence of ξ_1, ξ_2 and ξ_3. It follows from Eq. (7.2) that integrals over the new random variables can be transformed according to

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \dots f_{\zeta_1,\zeta_2,\zeta_3} \mathrm{d}\zeta_1 \mathrm{d}\zeta_2 \mathrm{d}\zeta_3 = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \dots f_{\xi_1} f_{\xi_2} f_{\xi_3} \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3 \tag{7.3}$$

Unlike the single random variable case we still have to deal with a three-dimensional integral after transformation. We will treat this integral as a repeated one-dimensional integral. Obviously this will lead to a serious increase in computation time. To counteract this effect we bring back the number of Gauss-Lobatto-Legendre quadrature points to 20. This, in turn, results in a significant reduction of the computation time. Also, as we will see later, it appears that the accuracy of the calculation results does not suffer from this reduction of the number of quadrature points.

7.2 Results

In Figs. 7.1 and 7.2 we compare the results from a TDgPC solution approach to gPC results and a Monte-Carlo analysis. We choose values of $P = 2$ and $P = 3$ for the two TDgPC solutions in this comparison. It is immediately clear that the TDGPC approach performs a lot better than the gPC solution approach. From approximately $t = 12$ onwards the gPC results for the mean of x_1 lose any resemblance to the correct solution. When looking at the variance of x_1 this point is already reached at $t = 4$. From our experience of Section 3.4 and Chapter 4 we know that increasing the maximum degree of the expansion polynomials will not lead to significantly better gPC results.

The TDgPC results with $P = 2$ remain reasonably close to the Monte Carlo analysis results for the entire time interval considered, although the curves can be seen to lose some of their accuracy as time progresses. Raising P from $P = 2$ to $P = 3$ results in an increase in accuracy: TDgPC results for the mean of x_1 are now graphically indistinguishable from the Monte Carlo results for the entire time interval displayed. The accuracy of the variance of x_1 goes up as well, but the TDgPC curve is not precisely on top of the Monte Carlo curve as was the case for a one-dimensional random input (cf. Fig. 6.5). A comparison of TDgPC, gPC and Monte Carlo results for x_2 and x_3 shows similar characteristics as the results for x_1 (see Appendix A.4).

7.2.1 Varying solution parameters

There are a number of parameters that can be adjusted when performing a TDgPC calculation, for instance:

- the maximum degree of the expansion polynomials P
- the number of Gauss-Lobatto-Legendre quadrature points Q
- the time step Δt in the numerical integration

Obviously, all of these three parameters have a profound effect on the computation time. Furthermore, it is of course interesting to look at the effect of these

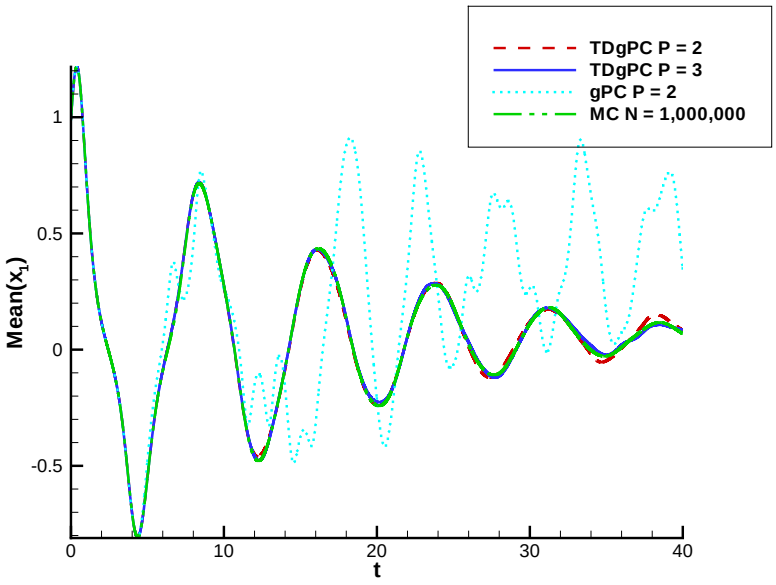

Figure 7.1: Mean of x_1 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions ($P = 2$ and $P = 3$) compared to a gPC solution ($P = 2$) and a Monte Carlo analysis ($N = 1,000,000$)

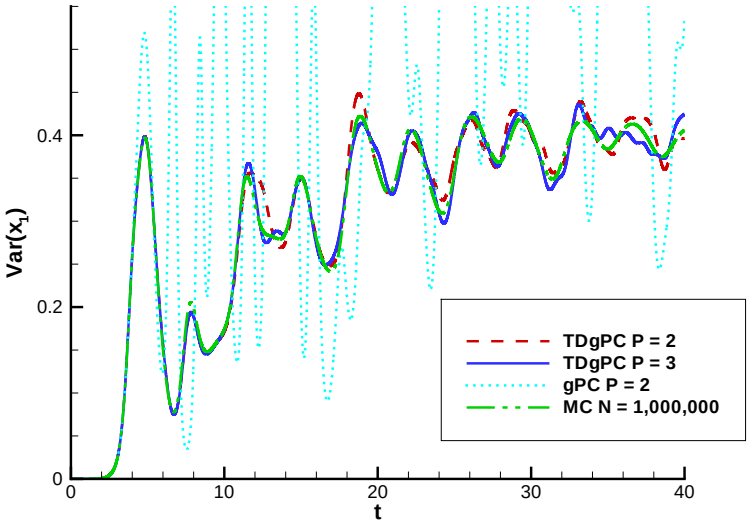

Figure 7.2: Variance of x_1 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions ($P = 2$ and $P = 3$) compared to a gPC solution ($P = 2$) and a Monte Carlo analysis ($N = 1,000,000$)

parameters on the accuracy of the calculated solution. Rather than conducting an exhaustive study on this issue we will compare the results for a couple of different sets P, Q and Δt in Figs. 7.3 and 7.4. For these figures we have assumed that the Monte Carlo result is the most accurate representation of the exact solution. In Figs. 7.3 and 7.4 we consider the evolution of the 'error' (in the sense of Eq. (6.7)) in the mean and the variance of x_1, respectively. Taking the case $P = 2$, $Q = 20$ and $\Delta t = 0.01$ as a (computationally relatively cheap) reference solution we can make the following observations:

- Doubling the number of quadrature points to $Q = 40$ does *not* result in a more accurate solution. The result for the mean with $Q = 40$ is practically identical to the reference solution with $Q = 20$. The result for the variance differs somewhat from the reference solution but is of the same accuracy. Increasing Q is also very unattractive from a computational viewpoint since this leads to a sharp rise of the computation time. Therefore we only calculated the solution with $Q = 40$ up to $t = 10$.

- Increasing the maximum degree of the expansion polynomials to $P = 3$ *does* result in a significantly more accurate solution. A comparable increase in accuracy can be obtained by decreasing the time step by a factor 10 to $\Delta t = 0.001$, with the former option of increasing P perhaps performing slightly better. Also, increasing P by one brings with it only roughly 50% of the rise in computation time that we experience due to a reduction of the time step to $\Delta t = 0.001$.

- A calculation with both third degree expansion polynomials as well as a time step of $\Delta t = 0.001$ is hardly more accurate than the results for the cases $P = 2$, $\Delta t = 0.001$ and $P = 3$, $\Delta t = 0.01$.

- Increasing P further to $P = 4$ does not lead to a significantly more accurate solution than for $P = 3$. It is remarkable, however, that the curves for $P = 4$, $\Delta t = 0.01$ and $P = 3$, $\Delta t = 0.001$ are almost exactly on top of each other. This *may* indicate that the Monte Carlo calculation on which the 'error' calculation in Figs. 7.3 and 7.4 is based is in fact more inaccurate than these higher accuracy TDgPC solutions. In other words, imagine a situation where the TDgPC solutions with $P = 4$, $\Delta t = 0.01$ and $P = 3$, $\Delta t = 0.001$ have converged strongly to the exact solution and the Monte Carlo solution is still a little bit off this exact solution. This situation would also result in an 'error' plot similar to Figs. 7.3 and 7.4. Perhaps $N = 1,000,000$ are still too little samples to obtain Monte Carlo results that are more accurate than the higher accuracy TDgPC solutions. Remember that we have a three-dimensional random input. So, $N = 1,000,000$ can also be seen as only $\sqrt[3]{1,000,000} = 100$ samples in each dimension. A more thorough investigation is needed before we can say anything decisive on this matter.

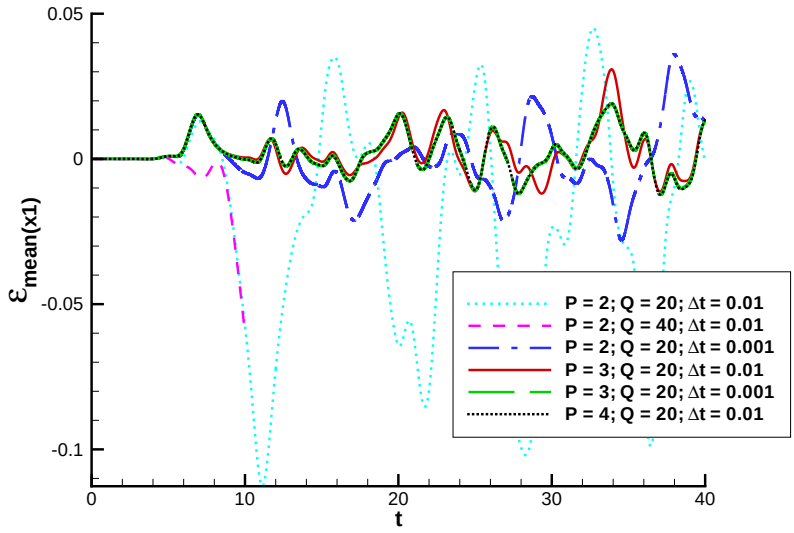

Figure 7.3: Error in the mean of x_1 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions with different P, Q and Δt

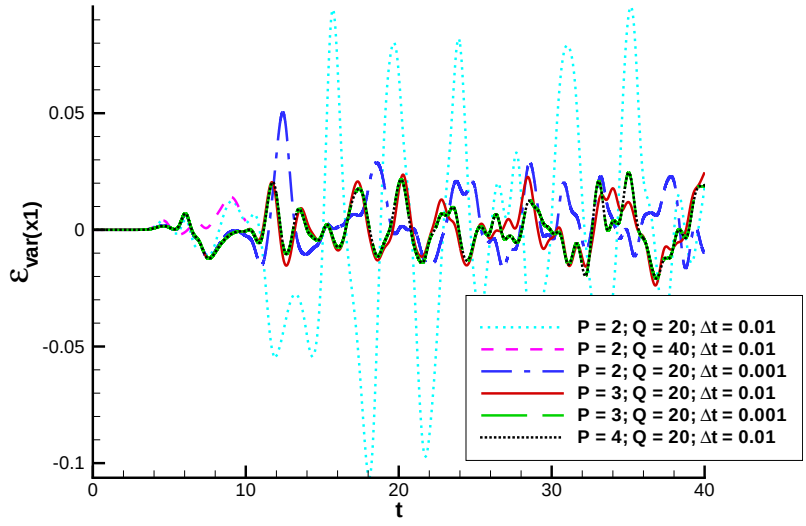

Figure 7.4: Error in the variance of x_1 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions with different P, Q and Δt

Part II

The growth equation

Chapter 8

Problem definition

In Chapter 6 we have refined Vos's original TDgPC method in order to make it applicable to the Kraichnan-Orszag three-mode problem. In the remaining chapters of this work we revisit the single differential equation test problem employed by Vos. We solve this problem using our modified TDgPC algorithm. With our refinements we expect to achieve the same accuracy as obtained by Vos, but at a substantially lower computational cost.

8.1 The growth equation

While Vos [10] was in the process of developing his time-dependent generalized polynomial chaos solution technique he concerned himself with the the following stochastic ordinary differential equation

$$\frac{\mathrm{d}u(t,\xi)}{\mathrm{d}t} + k(\xi)u(t,\xi) = 0\,, \quad u(0) = 1 \tag{8.1}$$

Since this equation can be thought of as modelling exponential population growth (with k representing the reproduction rate) we will call Eq. (8.1) the growth equation. Vos considers the case where the randomness enters the problem through k according to

$$k(\xi) = \frac{1}{2}\xi + \frac{1}{2} \tag{8.2}$$

where ξ is a uniformly distributed random variable on the interval $[-1, 1]$. Hence, ξ has the following PDF

$$f_\xi(\xi) = \frac{1}{2}\,, \quad -1 \leq \xi \leq 1 \tag{8.3}$$

In contrast to the Kraichnan-Orszag problem, the exact solution to Eq. (8.1) can of course be determined in a straightforward manner. We almost trivially find

$$u(t,\xi) = e^{-k(\xi)t} \tag{8.4}$$

Taking note of Eqs. (8.2) and (8.3) the exact mean and variance can also easily be calculated according to

$$\overline{u}_{exact}(t) = E\left[u_{exact}(t)\right] = \int_{-1}^{1} e^{-k(\xi)t} f_\xi(\xi)\mathrm{d}\xi = \frac{1 - e^{-t}}{t} \tag{8.5}$$

and

$$\sigma_{exact}(t) = E\left[\left(u_{exact}(t) - \overline{u}_{exact}(t)\right)^2\right] = \frac{1 - e^{-2t}}{2t} - \left(\frac{1 - e^{-t}}{t}\right)^2 \tag{8.6}$$

We have plotted these exact results for $\overline{u}$ and σ as a function of t in Figs. 8.1 and 8.2, respectively.

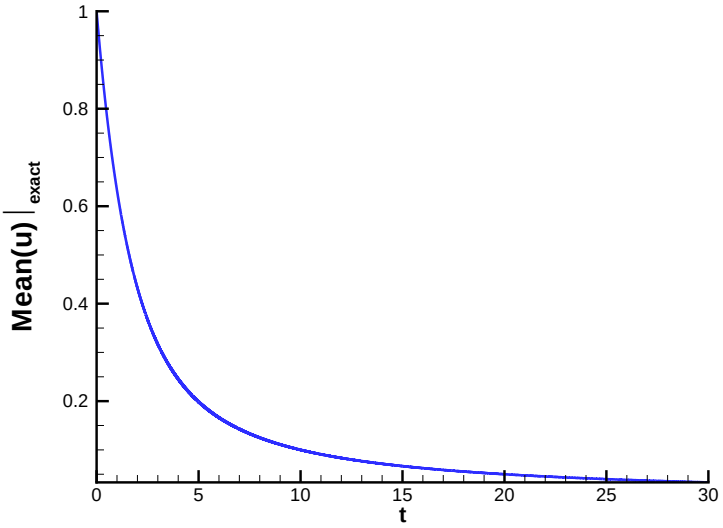

Figure 8.1: Mean of u vs. time for $u_0 = 1$: exact solution

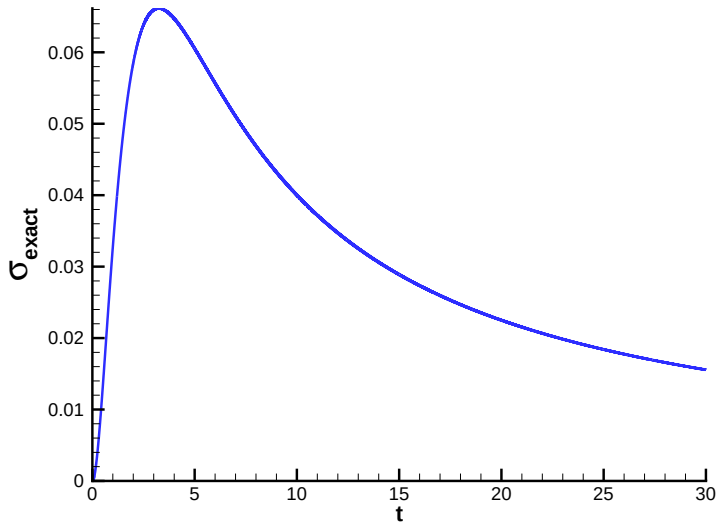

Figure 8.2: Variance of u vs. time for $u_0 = 1$: exact solution

Chapter 9

Vos's solution algorithm

Vos [10] does not adapt the expansion basis of the calculated solution at every time step. Instead, he employs a reinitialisation criterion: a new expansion basis is calculated when the nonlinear part of the solution reaches a certain size relative to the linear part of the solution. The reinitialisation process starts with the introduction of a new random variable $\zeta = Z(\xi)$ being equal to the solution at the reinitialisation time $u(t_1, \xi)$. Vos then proceeds by calculating the PDF of this new random variable according to

$$f_\zeta(\zeta) = \sum_{\xi_n} \frac{f_\xi(\xi_n)}{\left| \frac{\mathrm{d}Z(\xi)}{\mathrm{d}\xi} \big|_{\xi=\xi_n} \right|} \tag{9.1}$$

where the summation is over all solutions ξ_n of the equation

$$\zeta = Z(\xi) \tag{9.2}$$

The transition from the original random variable ξ to the first new random variable ζ can be performed smoothly with the help of these equations. However, the use of Eq. (9.1) for subsequent transformations of the random variable becomes less and less attractive from a computational point of view. This can be understood as follows.

After a couple of random variable transformations the associated PDF becomes a very complex function. During the TDgPC computation the PDF is not stored into memory as a function. Instead the PDF is evaluated (and subsequently stored into memory) at a limited number of points only, the so-called Gauss-Lobatto-Legendre quadrature points. Note that this makes sense since the PDF is only needed as a weighting function in a number of integrals. But what happens when we want to evaluate the PDF for the random variable after, say, five transformations? From Eq. (9.1) we see that this 'new' PDF can be expressed in the 'previous' PDF, i.e. the PDF for the random variable after four transformations. We need values for this 'previous' PDF at the solutions to Eq. (9.2). However, the 'previous' PDF has only been calculated at the quadrature points. In general, this set of quadrature points does not contain the solutions to Eq. (9.2). Vos overcomes this problem by repeatedly applying Eq. (9.1). The value of a PDF at a certain point can always be expressed in the 'previous' PDF. So, to evaluate the PDF for the random variable after five transformations

at a certain quadrature point one can 'backtrack' all the way to the PDF of the original variable in Eq. (8.3). Being a constant, this original PDF is of course known at every point and not only at the quadrature points. Note that this 'backtracking' can become very costly from a computational point of view: the set of old PDFs to track back through gets longer and longer with every new random variable transformation.

Luckily, we have learned in Section 6.1 that there is absolutely no need to calculate the PDF of the newly introduced random variables! In the TDgPC solution algorithm of the Kraichnan-Orszag problem all integrals can be evaluated in ξ-space. There is no reason to assume that this trick would not work for the TDgPC solution of the growth equation.

In the next section we will present TDgPC solutions of the growth equation with all integrals evaluated in ξ-space. We will initially use Vos's reinitialisation criterion

$$\frac{\max\left(\left|u_2(t)\right|,\ldots,\left|u_P(t)\right|\right)}{\left|u_1(t)\right|} \geq \theta \tag{9.3}$$

where $u_i(t)$ are the expansion coefficients of the calculated solution $u(t)$. Vos uses $\theta = 1/6$. We will use almost the same value and set $\theta = 0.15$.

Chapter 10

Results

In this section we compare TDgPC computations for Eq. (8.1) together with Eqs. (8.2) and (8.3) using various amounts of quadrature points, setting a number of different time steps and employing different maximum degrees of the expansion polynomials. Also, we will try to establish if it is preferable to use a reinitialisation criterion (like Vos did) or to just adapt the expansion polynomials at every time step (as we did in the case of the Kraichnan-Orszag problem).

We will compare the different TDgPC computations on the basis of the error in both mean and variance of the calculated solution. We define these errors as follows

$$\epsilon_{mean}(t) = \left| \frac{\overline{u}(t) - \overline{u}_{exact}(t)}{\overline{u}_{exact}(t)} \right| \tag{10.1}$$

and

$$\epsilon_{var}(t) = \left| \frac{\sigma(t) - \sigma_{exact}(t)}{\sigma_{exact}(t)} \right| \tag{10.2}$$

10.1 Different quantities of quadrature points

Vos used a number of $Q = 300$ for his TDgPC computations on the growth equation. Setting Q this high may have been motivated by the idea to have enough resolution near the boundaries of the domain. This might have been important since the corresponding PDF has a tendency to form a sharper and sharper peak at one boundary of the domain through the random variable transformations.

Remember that, contrary to Vos, we do not calculate these PDFs of the transformed random variables. We evaluate all integrals in ξ-space using the original PDF (Eq. (8.3)). So, it may very well be possible to achieve accurate TDgPC results with fewer quadrature points. We investigate this in Figs. 10.1 and 10.2 where we show the error in the mean and variance, respectively, of TDgPC calculations using different numbers of quadrature points.

From these two figures it becomes clear that as long as $Q \geq 10$ the precise number of quadrature points is largely immaterial to the accuracy of the solution. The only exception is the ϵ_{mean} curve for $Q = 10$ which exhibits a strange spike around $t = 25$. Setting $Q < 10$ can be seen to result in a loss in accuracy for both mean and variance of the solution. The results in Figs. 10.1 and 10.2 are for $P = 3$. It may well be that we need more quadrature points

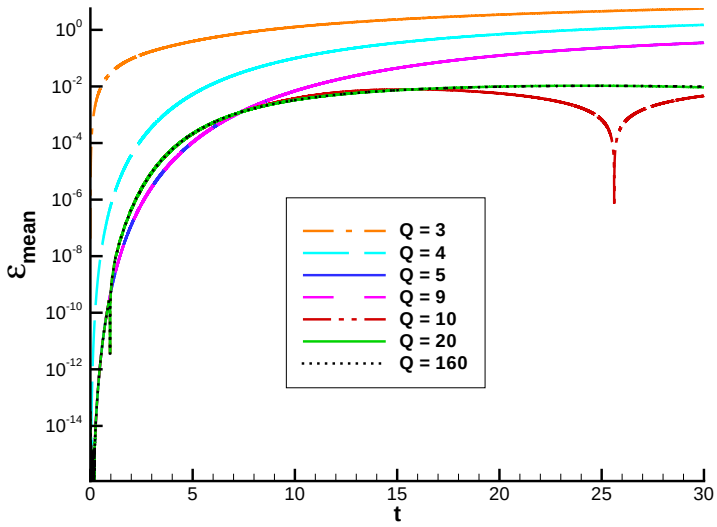

Figure 10.1: Error in the mean of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 3$ for different amounts of quadrature points Q

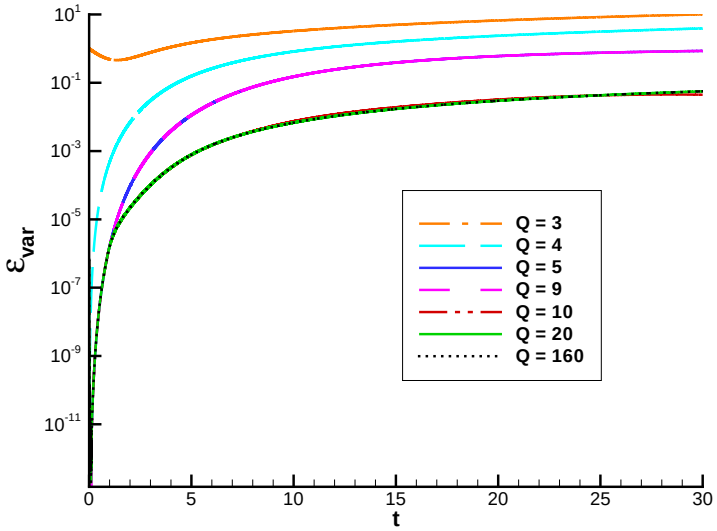

Figure 10.2: Error in the variance of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 3$ for different amounts of quadrature points Q

when P is set to a higher value. We can conclude, however, that we can do with substantially less quadrature points than Vos's $Q = 300$ resulting in an increase in computation speed. To be on the safe side, also for higher values of P, we have used $Q = 5P$ to compute the results in the following sections.

10.2 Different time steps

In Figs. 10.3 and 10.4 we investigate the influence of the time step Δt used in the fourth-order Runge-Kutta time integration on the accuracy of a TDgPC computation with $P = 4$. We see that if we are only interested in later time values the time step can be set as high as $\Delta t = 1$ without compromising the accuracy of the results for mean and variance. Reducing the time step, however, leads to a large increase in accuracy for early times. Setting $\Delta t = 0.001$ (the value that we have used for the rest of this chapter) can be seen to deliver the most accurate results. Decreasing the time step beyond this value has no effect on the accuracy.

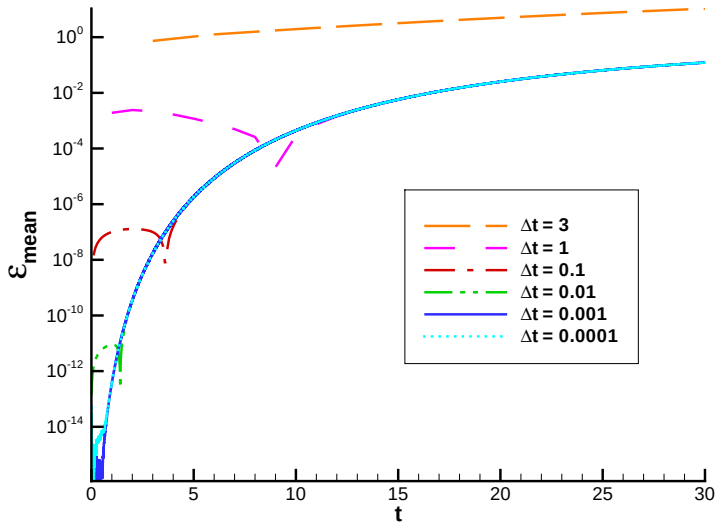

Figure 10.3: Error in the mean of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 4$ for different time steps Δt

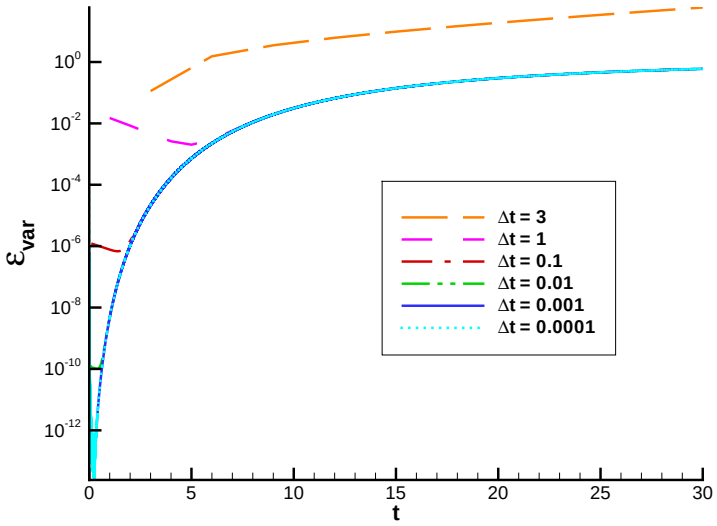

Figure 10.4: Error in the variance of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 4$ for different time steps Δt

10.3 Different maximum expansion polynomial degrees

In Figs. 10.5 and 10.6 we compare the accuracy of mean and variance, respectively, for different values of the maximum degree of the expansion polynomials. Vos presented results for $P = 2, \ldots, 6$. In this range of P our results are practically identical to Vos's calculations. However, since we do not calculate the PDF of the transformed random variable, our solution algorithm is substantially faster. This means that calculations with higher values of P become more attractive. In Figs. 10.5 and 10.6 we therefore present results for selected values of P in the range $P = 2, \ldots, 18$. It can be seen that, for larger time values, increasing P reduces the error in both mean and variance. We can see this more clearly in Fig. 10.7 where we have plotted the error in both mean and variance at $t = 30$ against P. For smaller time values (roughly under $t = 10$) we see that the accuracy first increases with P, but for values larger than $P = 9$ the accuracy starts to slightly decreases again. Fig. 10.7 also shows that at $t = 30$ the calculated mean of the solution converges exponentially to the exact value with increasing P. The calculated variance shows exponential P-convergence in the lower P-range only. For higher values of P the speed of convergence of the variance diminishes.

Finally, to get an idea of the gain in computation speed when using the present solution algorithm instead of Vos's, consider the following. Every graph in Figs. 10.5 and 10.6 has been calculated in less than one minute. Results for $P = 2$ took 4.9 seconds to generate, for instance. Setting $P = 18$ led to a calculation time of 53.4 seconds. If we would use Vos's algorithm to generate

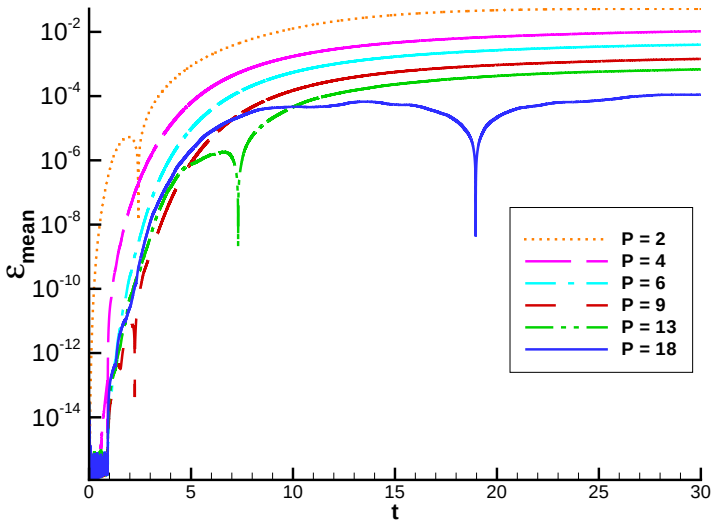

Figure 10.5: Error in the mean of u vs. time for $u_0 = 1$: TDgPC solutions for different maximum degrees of the expansion polynomials P

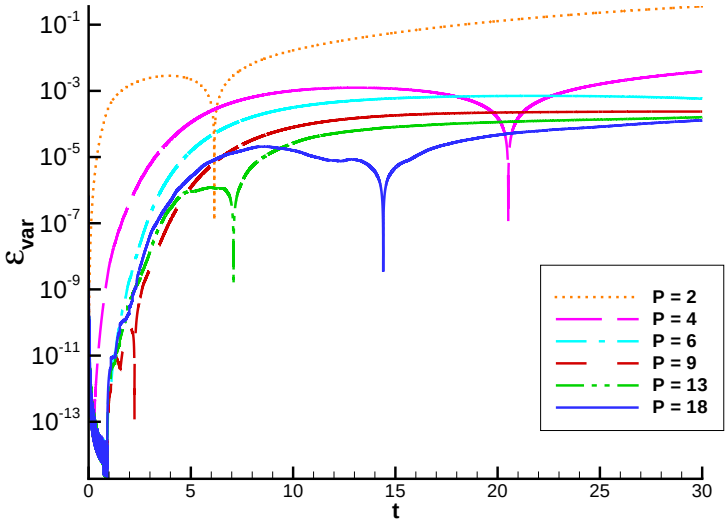

Figure 10.6: Error in the variance of u vs. time for $u_0 = 1$: TDgPC solutions for different maximum degrees of the expansion polynomials P

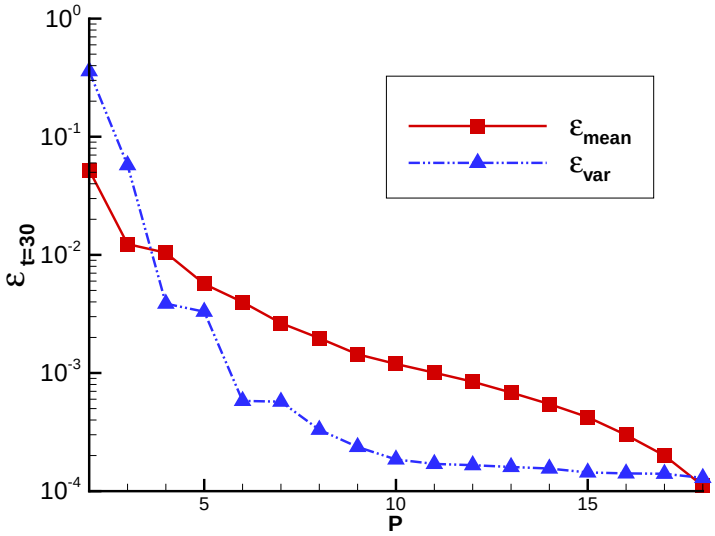

Figure 10.7: Error in the mean and variance of u at $t = 30$ vs. maximum degree of the expansion polynomials P ($u_0 = 1$)

results for $P = 4$ we would already be looking at a calculation time around 10 minutes. Setting $P = 6$ would increase the calculation time further to several hours. Of course, actual computation times depend a lot on machine specifications. However, we merely want to give an indication of the huge gain in computation speed that results when performing all integrations in ξ-space.

10.4 Comparing different reinitialisation criteria

Up to this point we have used the reinitialisation criterion proposed by Vos (Eq. (9.3)) with $\theta = 0.15$ to generate the TDgPC solutions of the growth equation. In this approach new expansion polynomials are calculated only when this criterion is met. In Section 6.2, however, we came to the conclusion that in the case of the Kraichnan-Orszag problem the accuracy of the calculated solution increases when we just compute a new set of expansion polynomials *every time step*. Would this also be the case for a TDgPC solution of the growth equation? Yet another, sort of in-between way to approach this issue of expansion polynomial reinitialisation would be to calculate new expansion polynomials after a certain, fixed time interval Δt_{Reinit}, letting go of the idea behind Eq. (9.3) that we need to reinitialise when the nonlinear terms reach a certain size.

In this section we will compare the error in mean and variance of TDgPC solutions calculated using different reinitialisation criteria, either in the form of Eq. (9.3) (*type I*) or using a fixed Δt_{Reinit} (*type II*). We have experimented with different values for θ and Δt_{Reinit}. The optimal values of these parameters were

found to be dependent on the maximum degree of the expansion polynomials P. In this section we present results for $P = 3$, $P = 10$ and $P = 17$. Also, it became clear that reinitialising every time step (as we did in the case of the Kraichnan-Orszag problem) does *not* lead to the most accurate results. We come back to this point at the end of this section.

In Figs. 10.8 and 10.9 we look at the case $P = 3$. Both for reinitialisation criterion type I as well as type II we have plotted one or two graphs corresponding to those values of θ or Δt_{Reinit} that lead to the most accurate results on the time interval $0 \leq t \leq 100$. Results from a gPC calculation (where we do not reinitialise the polynomial expansion basis) are also shown for comparison. From $t = 10$ onwards the TDgPC results are roughly an order of magnitude more accurate than the gPC calculation. It can also be seen that the 'optimal' reinitialisation criterion depends on the time interval we find most important. The calculation with $\Delta t_{Reinit} = 2$ seems to perform best on average.

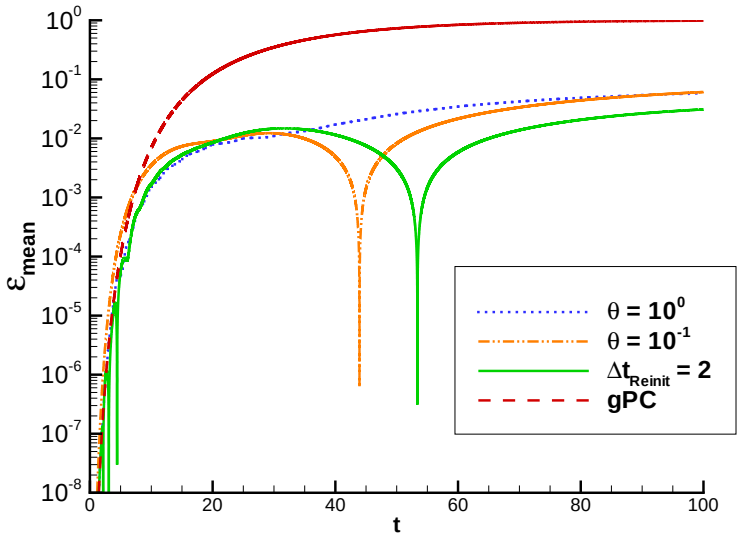

Figure 10.8: Error in the mean of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 3$ for various reinitialisation criteria

The case $P = 10$ is considered in Figs. 10.10 and 10.11. We again have only plotted results for the 'optimal' values of θ and Δt_{Reinit}. Note that these optimal values are higher than they were for $P = 3$. So, when increasing P we get the most accurate results when the frequency of polynomial basis reinitialisation is decreased. Also note the different shape of the gPC error curves compared to the TDgPC curves. If we would mainly be interested in the mean of the solution on the interval $0 \leq t \leq 40$ the gPC calculation would be more accurate than the TDgPC calculations for $P = 10$.

Finally, we look at the case $P = 17$ in Figs. 10.12 and 10.13. To achieve optimal results we once again need to decrease the frequency of polynomial basis reinitialisation. Since this is easier to accomplish by using a type II reini-

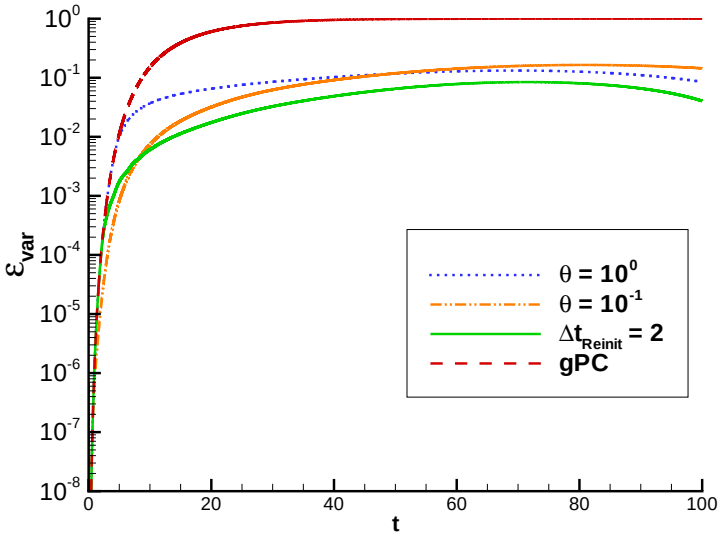

Figure 10.9: Error in the variance of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 3$ for various reinitialisation criteria

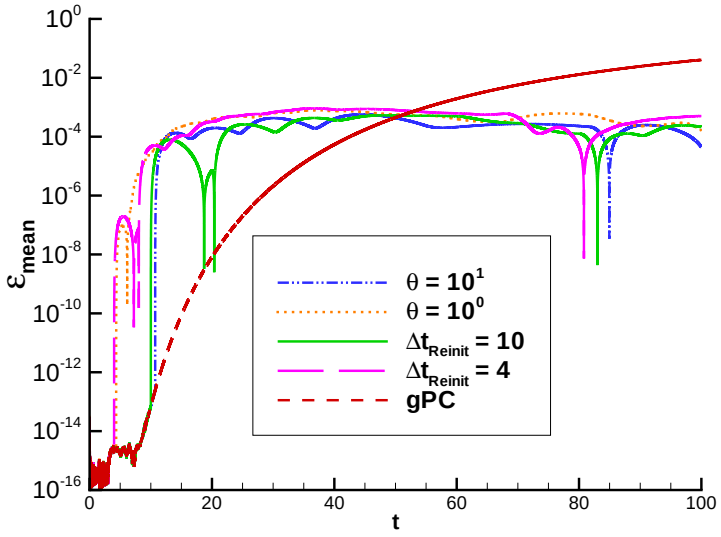

Figure 10.10: Error in the mean of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 10$ for various reinitialisation criteria

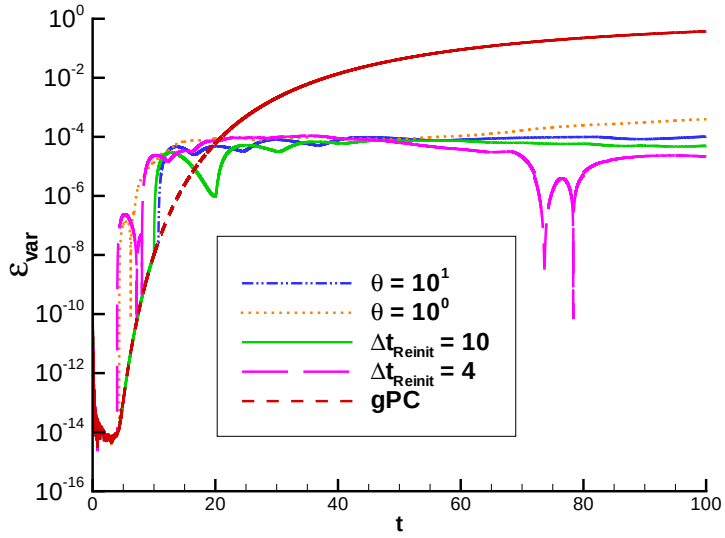

Figure 10.11: Error in the variance of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 10$ for various reinitialisation criteria

tialisation criterion (and since we have seen that with both types roughly the same accuracy can be achieved) we only present calculation results using 'optimal' values of Δt_{Reinit}. For comparison gPc results are plotted as well. We see that for $P = 17$ using a gPC instead of a TDgPC approach leads to more accurate results for large parts of the time interval considered. Performing the first expansion polynomial reinitialisation can be seen to result in a sudden increase of the error. However, once this first reinitialisation has been executed the error remains approximately at the same level, even through subsequent polynomial basis reinitialisations. The error of the gPC solution, on the other hand, remains low for a substantial time interval for this high value of $P = 17$. Eventually, however, the error of the gPC solution starts to increase at a faster rate than the error of the TDgPC solution. Based on these characteristics we expect that there always is an instant in time beyond which a TDgPC solution is more accurate than a gPC solution. So, which approach leads to the best results depends on the time interval we are interested in.

To conclude let us come back to the fact that a TDgPC solution of the Kraichnan-Orszag problem produces the most accurate results when we reinitialise the expansion polynomials every time step, while a TDgPC solution of the growth equation is more accurate when we reinitialise less often. This difference may (in part) stem from the fact that the randomness enters the Kraichnan-Orszag problem through the initial conditions, while for the growth equation the randomness enters the problem through the parameter k. This issue would require further investigation, for instance by considering the growth equation with deterministic k and random initial condition. We have found, however, that this problem is already accurately solved by a plain gPC approach. Hence,

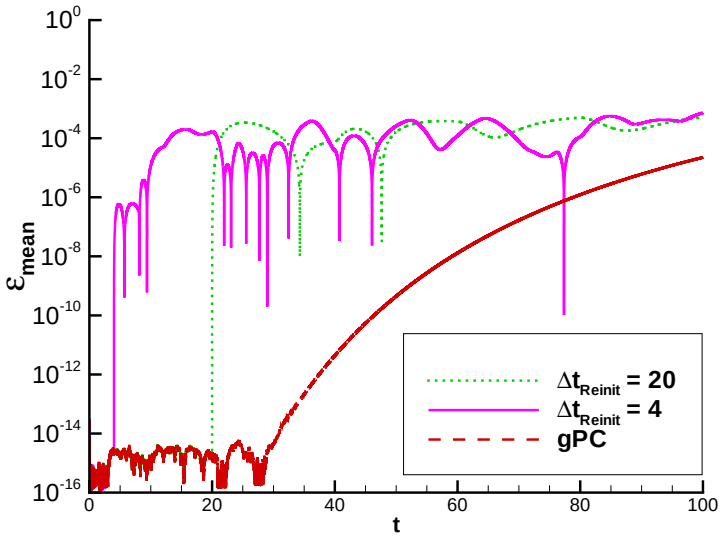

Figure 10.12: Error in the mean of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 17$ for various reinitialisation criteria

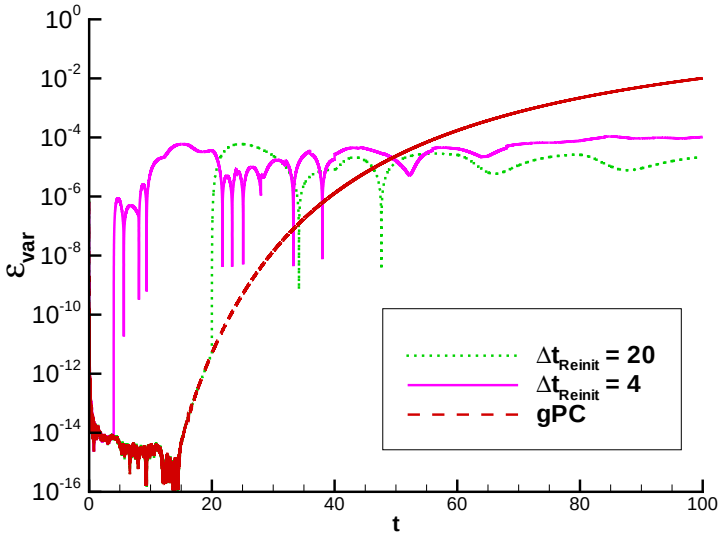

Figure 10.13: Error in the variance of u vs. time for $u_0 = 1$: TDgPC solutions with $P = 17$ for various reinitialisation criteria

it would be necessary to look at other problems to shed some more light on this issue, for instance a Kraichnan-Orszag-type problem with random parameters instead of random initial conditions.

Chapter 11

Conclusions and recommendations

11.1 Conclusions

In this work we have successfully developed a time-dependent generalized polynomial chaos method (TDgPC) for a system of coupled stochastic differential equation. The solution technique is an extension of the generalized polynomial chaos method (gPC). From a computational point of view TDgPC is a potentially attractive alternative to Monte Carlo simulations that typically require immense amounts of computer power. We have applied the TDgPC method to the Kraichnan-Orszag three-mode problem.

A gPC method approximates the solution of a stochastic problem by a polynomial expansion employing deterministic coefficients in combination with mutually orthogonal expansion polynomials in terms of random variables. With TDgPC these expansion polynomials are adapted over time based on the current stochastic properties of the solution. This constantly updated expansion basis helps to generate accurate results for long-term integration.

The Kraichnan-Orszag problem can be very sensitive to small changes in the initial conditions. Small fluctuations in the initial conditions can in this case lead to solutions with substantially different frequencies. It is also possible that a discontinuity occurs. We have seen that a plain gPC solution approach (employing a fixed polynomial expansion basis) is not capable of calculating accurate long-term results for mean and variance of the solution when the initial conditions are randomly distributed over such a sensitive area. Increasing the maximum degree of the expansion polynomials is not a solution to this problem. TDgPC, on the other hand, produces accurate long-term results in these difficult situations by adapting the polynomial expansion basis at every time step. There is no need to employ expansion polynomials of high degree with the TDgPC method. Furthermore, we have established that the TDgPC method works accurately for both one- as well as multidimensional random inputs.

Vos [10] introduced the idea of adapting the expansion expansion basis over time. He also developed the first TDgPC algorithm for the solution of a *single* stochastic differential equation. For the TDgPC solution method developed

in this work – applicable to *systems* of differential equations – a number of modifications were made to Vos's TDgPC algorithm, the most important being a different way of integrating in random space which renders the calculation of the PDF of the solution unnecessary. These modifications result in a spectacular speed-up of the algorithm when applied to Vos's test problem: the growth equation. We have established exponential P-convergence for the mean of a TDgPC solution of the growth equation. For this growth equation we have also shown that, even for later times, TDgPC results for mean and variance of the solution can achieve an accuracy of $O\left(10^{-4}\right)$.

Contrary to the Kraichnan-Orszag problem a TDgPC solution of the growth equation does not produce its most accurate results when we adapt the expansion polynomials every time step. At the moment the reason for this has not been researched well enough. We *have* established that an expansion basis with polynomials of higher degree needs to be updated less frequently to lead to optimal TDgPC results for the growth equation.

11.2 Suggestions for further research

In this section we propose a number of directions for future research. First off, we have only considered uniformly distributed random inputs on bounded domains in this work. This was motivated by the fact that we have the Gauss-Lobatto-Legendre numerical integration scheme to adequately integrate on bounded domains in the random space. It might be useful to also look at random inputs with distributions on unbounded domains (a Gaussian distribution, for instance). This should not be too difficult to implement, either through a transformation of the random variables to a bounded domain or a different quadrature scheme.

Secondly, it would be useful to apply the TDgPC method to a system of stochastic differential equations for which the exact solution is known in analytical form. This would help with the assessment of accuracy and convergence behaviour of the TDgPC algorithm. In case of the Kraichnan-Orszag problem we can never be completely sure about the accuracy of the Monte Carlo analyses. Too small a number of samples or inaccuracies in the time integration introduce errors in the Monte Carlo results. This makes it harder to verify the accuracy of the TDgPC results.

A third issue on which some further research is welcome is the frequency of expansion polynomial updates. Examples of questions that still need an answer are the following. What determines the optimal instants in time to update the expansion polynomials? How is this dependent on the type of problem (random initial conditions vs. random problems parameters)? In which cases does updating every time step lead to the best results? Can an optimal reinitialisation criterion be constructed for a certain problem?

We noted in the introduction to this work that Wan [9] has constructed a multi-element generalized polynomial chaos method in order to deal with the long-term integration issues of plain gPC. It would be very interesting too combine the multi-element approach of Wan with the time-dependent gPC approach discussed in this work. In Wan's approach as time progresses larger elements are split into smaller elements on which a new expansion basis is constructed. Wan, however, bases this new expansion polynomials on the original random input, while we have seen in this work that it might be beneficial to base the

new expansion polynomials on the current statistical properties of the solution.

Finally, a very interesting direction for further research would be to apply the TDgPC approach discussed in this work to partial differential equations. The heat equation might be a good candidate to start on. Vos developed his solution algorithm for a single stochastic DE. In this work we have extended the approach to a system of coupled, stochastic DEs. So, applying TDgPC to a PDE is the natural yet very exciting next step.

Appendix A

Additional figures

A.1 Kraichnan-Orszag: deterministic solutions

In Section 3.4 we have presented deterministic solutions to the Kraichnan-Orszag problem for various initial condtions. However, in this section only results for x_1 were plotted. In this appendix plots of the corresponding solutions $x_2(t)$ and $x_3(t)$ can be found.

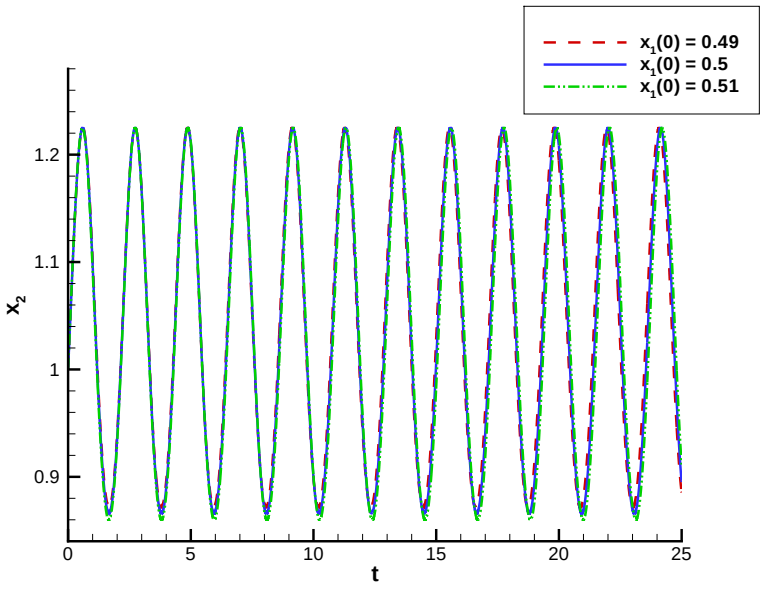

Figure A.1: Deterministic solutions $x_2(t)$ for various $x_1(0)$ around $x_1(0) = 0.5$ $(x_2(0) = x_3(0) = 1)$

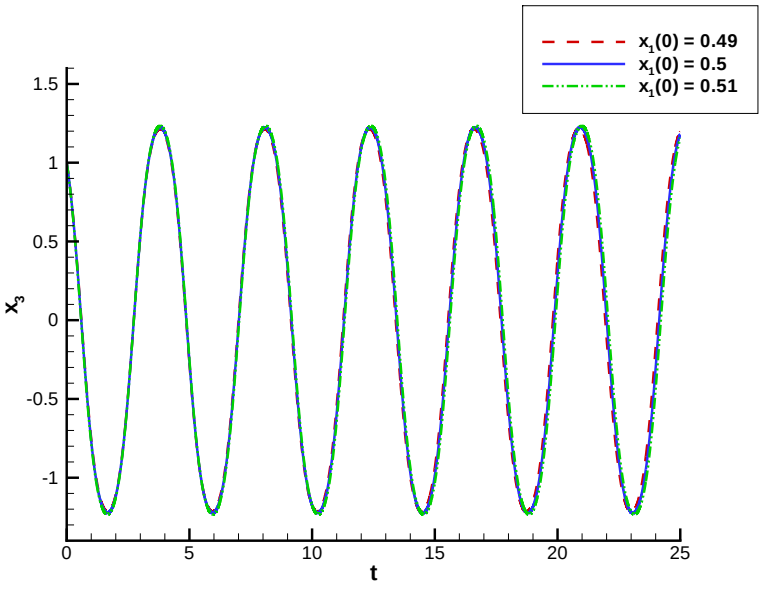

Figure A.2: Deterministic solutions $x_3(t)$ for various $x_1(0)$ around $x_1(0) = 0.5$ $(x_2(0) = x_3(0) = 1)$

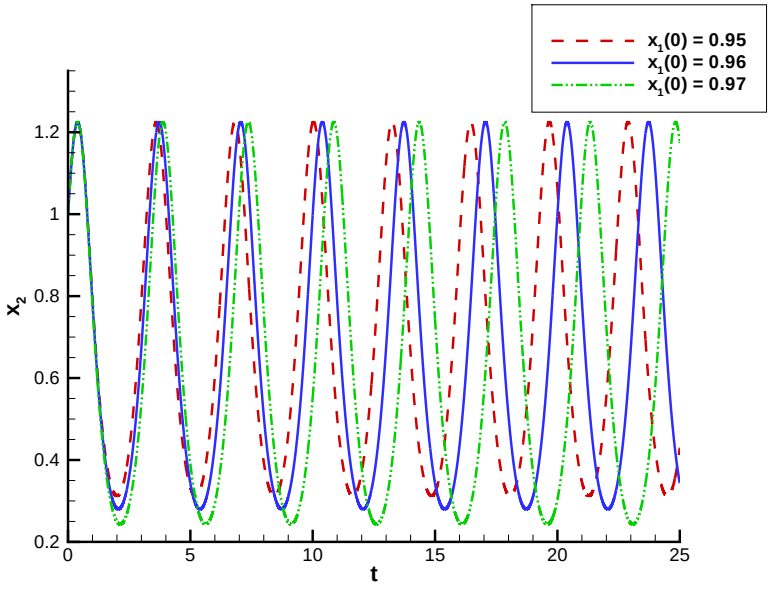

Figure A.3: Deterministic solutions $x_2(t)$ for various $x_1(0)$ around $x_1(0) = 0.96$ $(x_2(0) = x_3(0) = 1)$

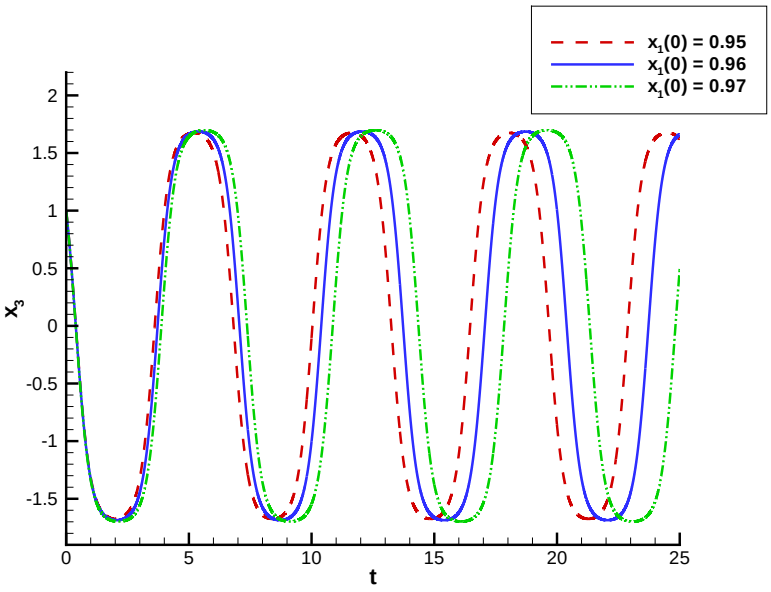

Figure A.4: Deterministic solutions $x_3(t)$ for various $x_1(0)$ around $x_1(0) = 0.96$ $(x_2(0) = x_3(0) = 1)$

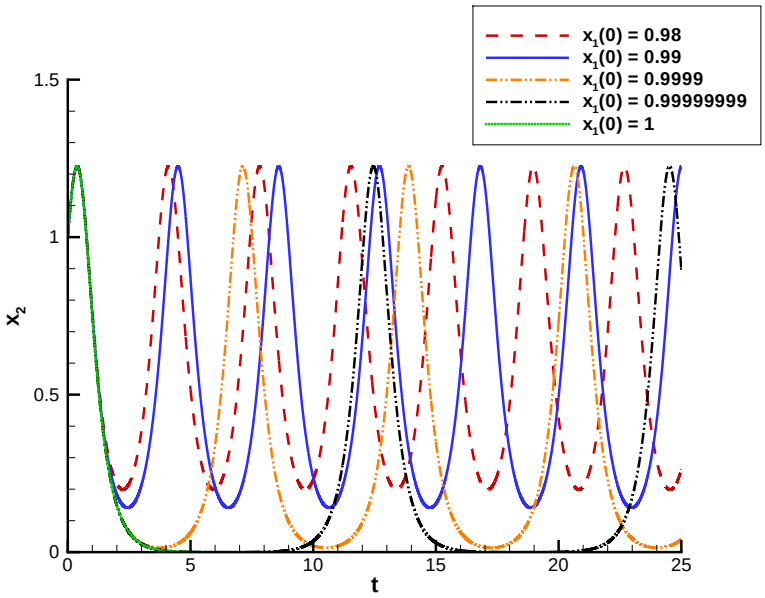

Figure A.5: Deterministic solutions $x_2(t)$ for various $x_1(0)$ around $x_1(0) = 0.99$ $(x_2(0) = x_3(0) = 1)$

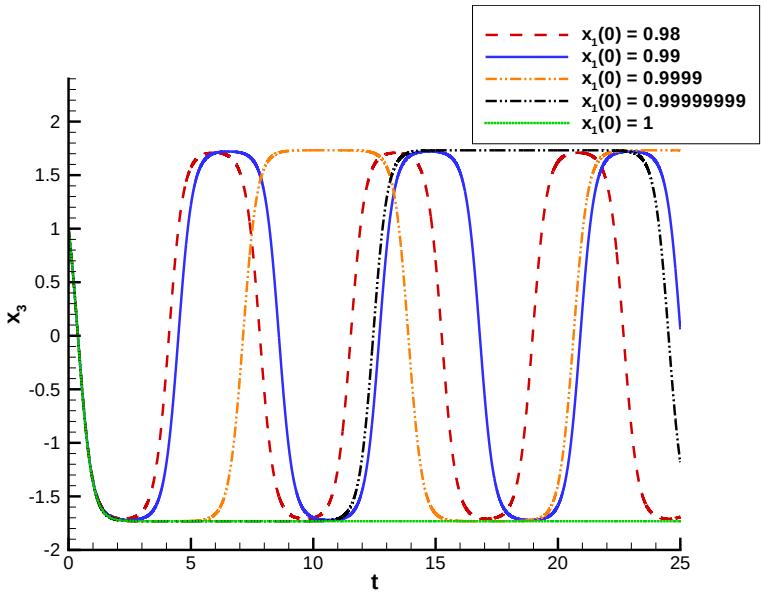

Figure A.6: Deterministic solutions $x_3(t)$ for various $x_1(0)$ around $x_1(0) = 0.99$ $(x_2(0) = x_3(0) = 1)$

A.2 Kraichnan-Orszag: gPC approach

In Chapter 4 we have discussed results of a gPC solution approach to the Kraichnan-Orszag problem, presenting results for x_1 only. In this appendix the corresponding gPC results for x_2 and x_3 can be found.

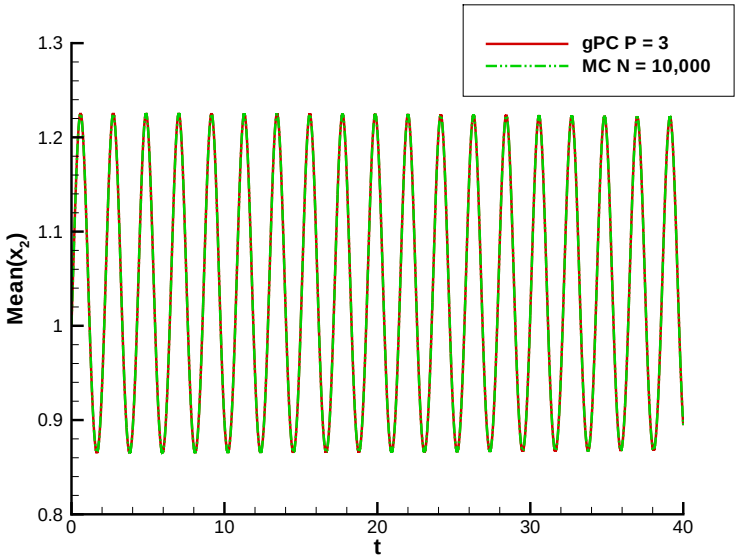

Figure A.7: Mean of x_2 vs. time for $\alpha = 0.5$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

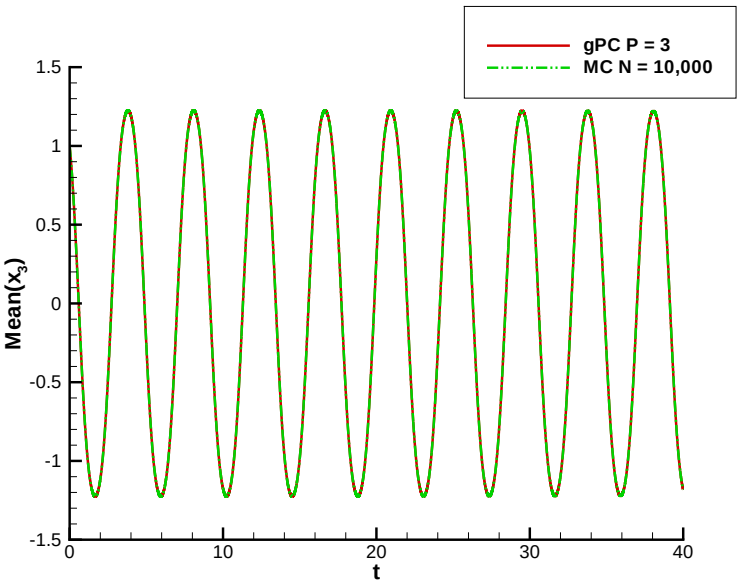

Figure A.8: Mean of x_3 vs. time for $\alpha = 0.5$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

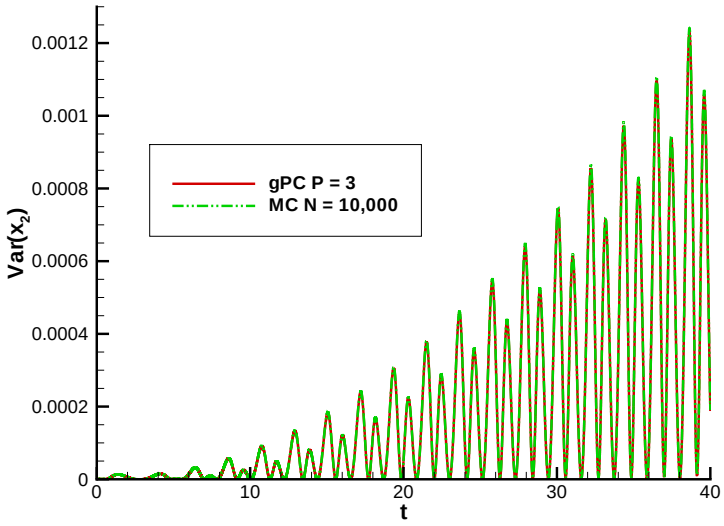

Figure A.9: Variance of x_2 vs. time for $\alpha = 0.5$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

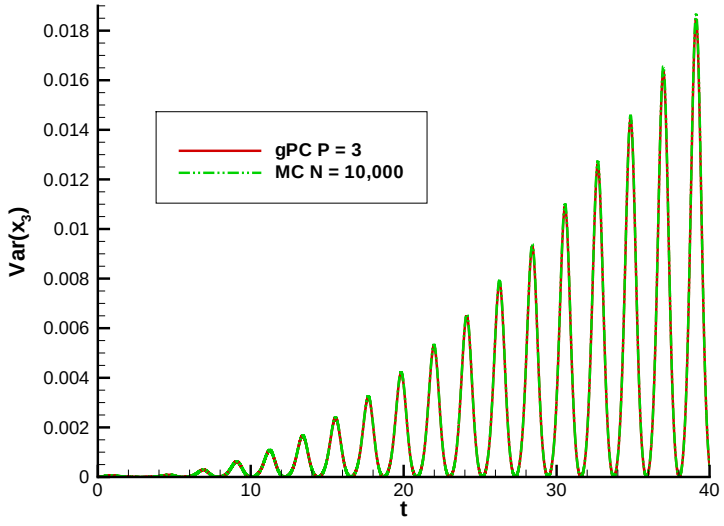

Figure A.10: Variance of x_3 vs. time for $\alpha = 0.5$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

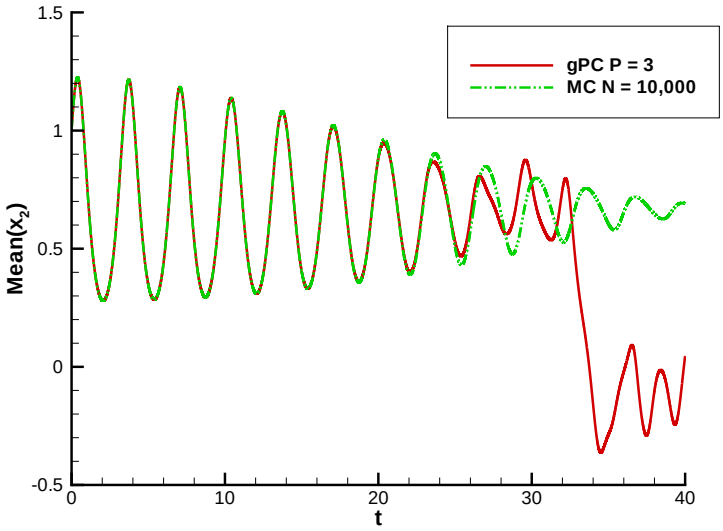

Figure A.11: Mean of x_2 vs. time for $\alpha = 0.96$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

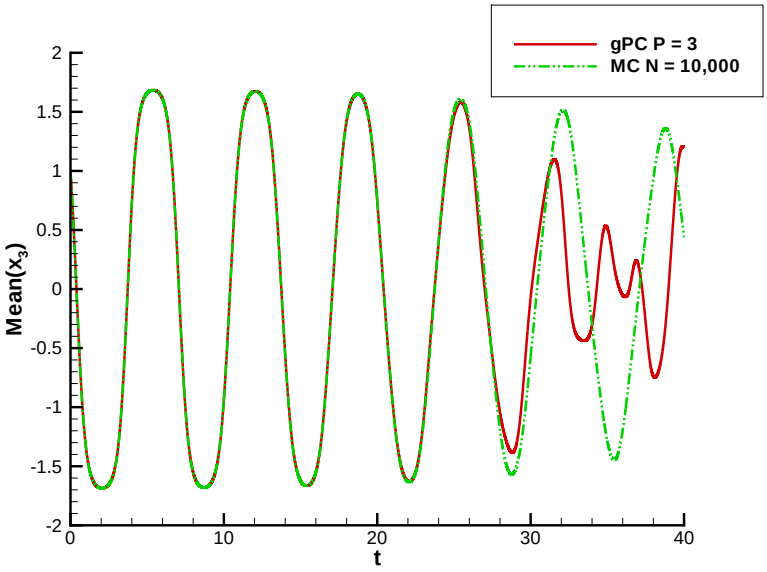

Figure A.12: Mean of x_3 vs. time for $\alpha = 0.96$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

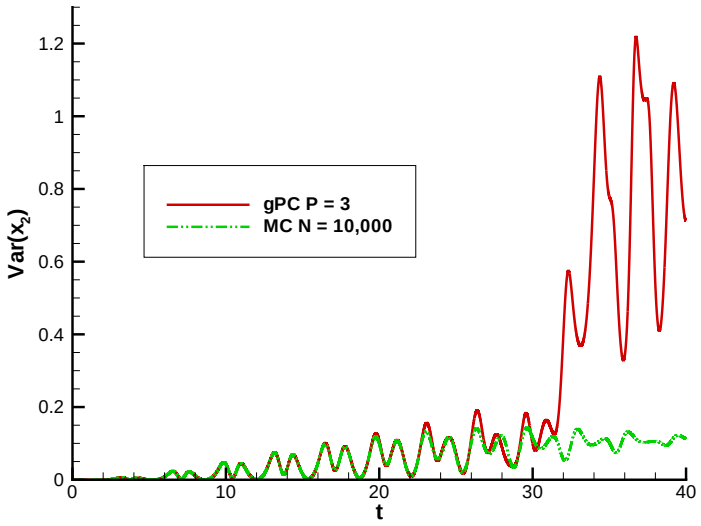

Figure A.13: Variance of x_2 vs. time for $\alpha = 0.96$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

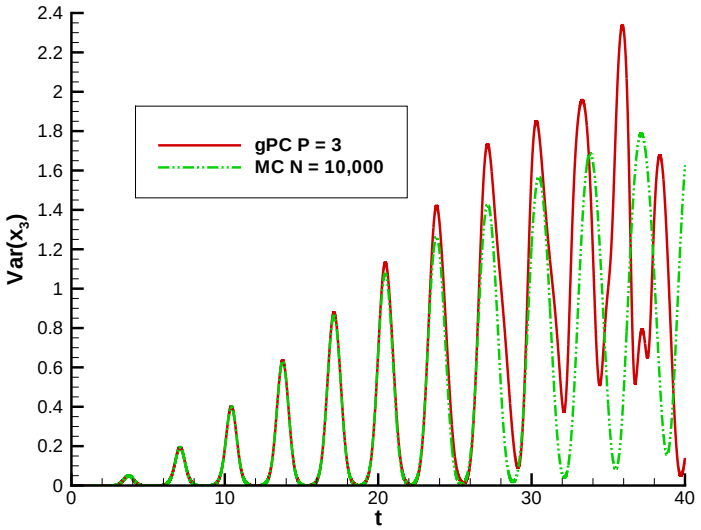

Figure A.14: Variance of x_3 vs. time for $\alpha = 0.96$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

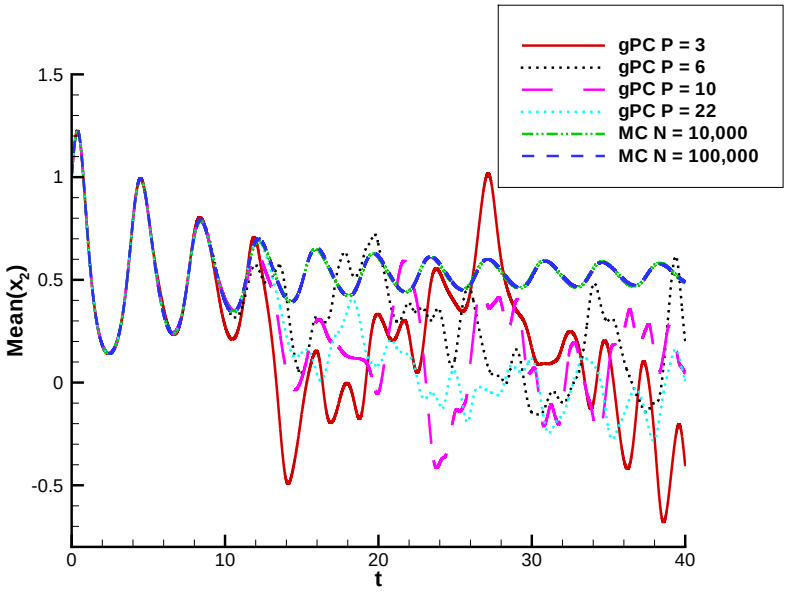

Figure A.15: Mean of x_2 vs. time for $\alpha = 0.99$: gPC solutions with $P = 3$ and $P = 4$ compared to Monte Carlo analyses ($N = 10,000$ and $N = 100,000$)

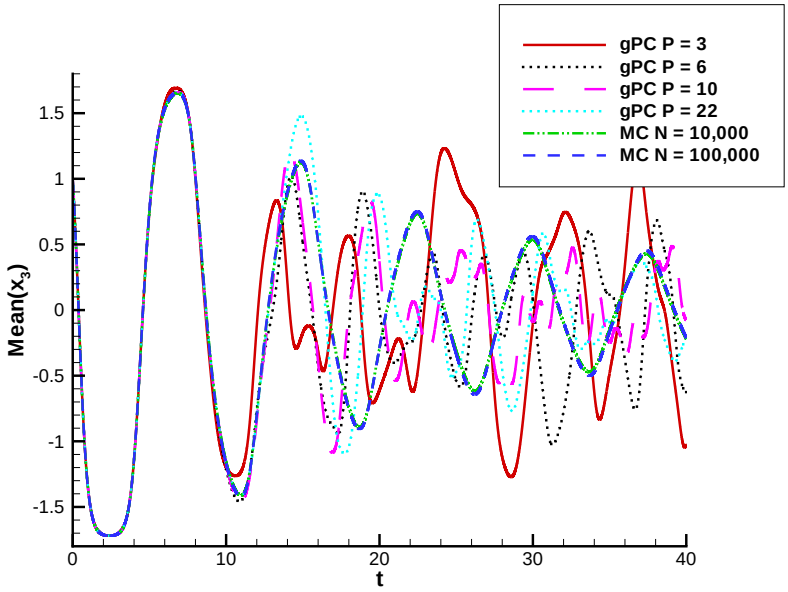

Figure A.16: Mean of x_3 vs. time for $\alpha = 0.99$: gPC solutions with $P = 3$ and $P = 4$ compared to Monte Carlo analyses ($N = 10,000$ and $N = 100,000$)

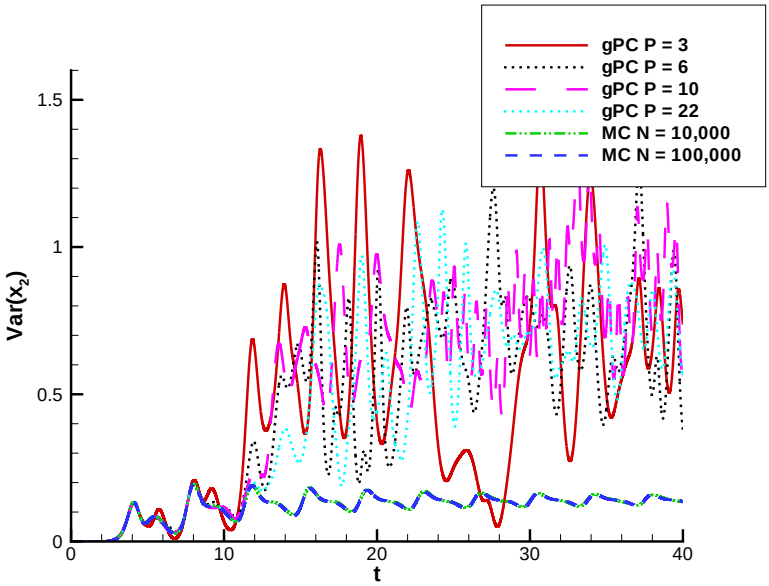

Figure A.17: Variance of x_2 vs. time for $\alpha = 0.99$: gPC solutions with $P = 3$ and $P = 4$ compared to Monte Carlo analyses ($N = 10,000$ and $N = 100,000$)

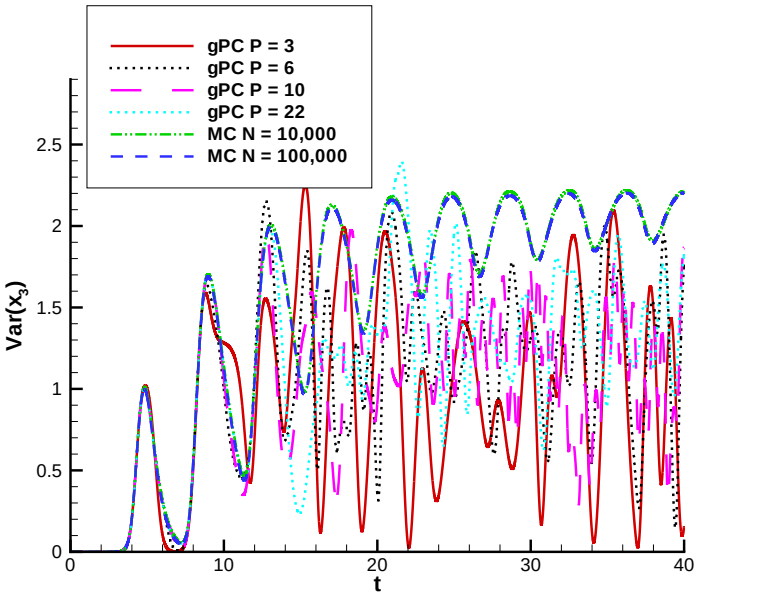

Figure A.18: Variance of x_3 vs. time for $\alpha = 0.99$: gPC solutions with $P = 3$ and $P = 4$ compared to Monte Carlo analyses ($N = 10,000$ and $N = 100,000$)

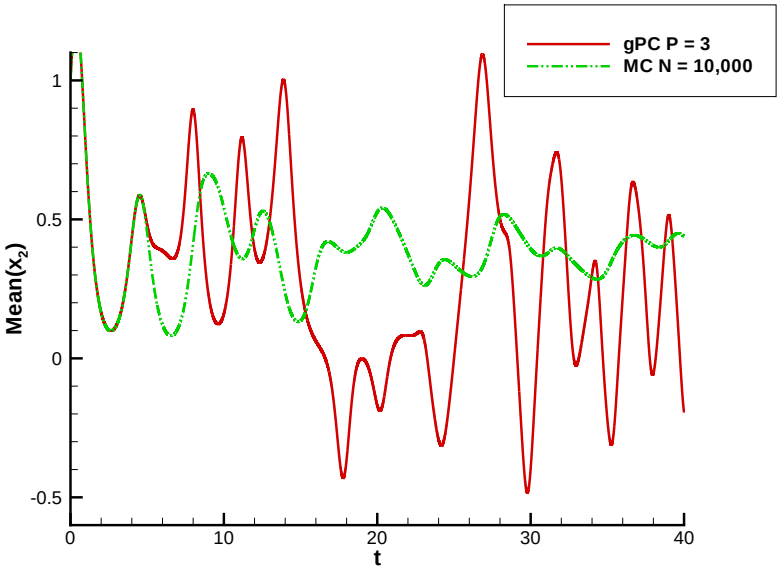

Figure A.19: Mean of x_2 vs. time for $\alpha = 0.995$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

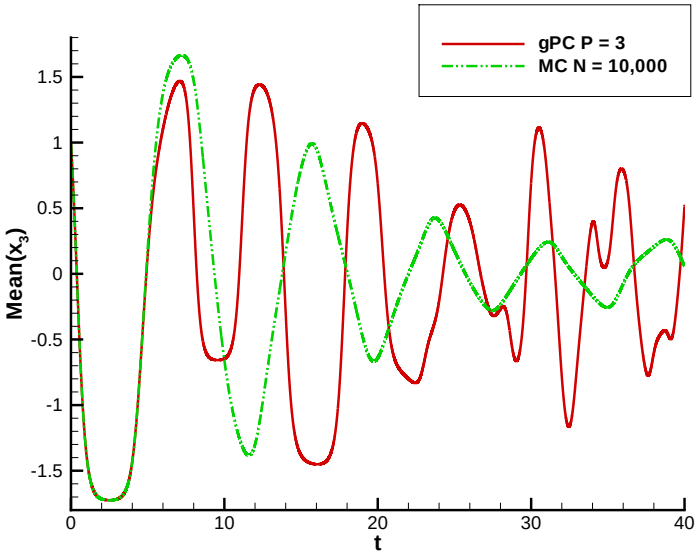

Figure A.20: Mean of x_3 vs. time for $\alpha = 0.995$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

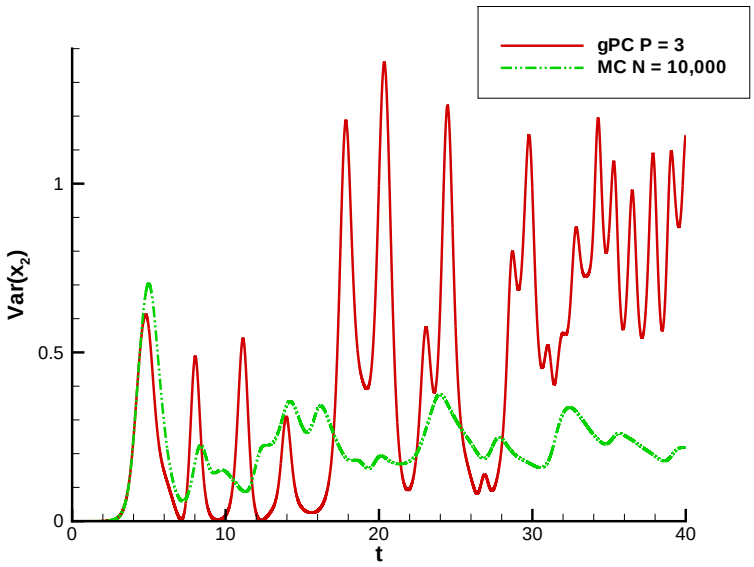

Figure A.21: Variance of x_2 vs. time for $\alpha = 0.995$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

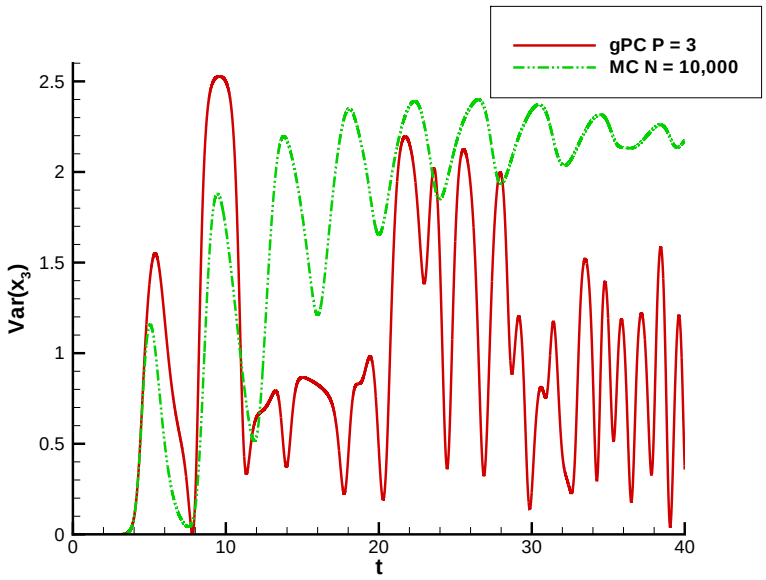

Figure A.22: Variance of x_3 vs. time for $\alpha = 0.995$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

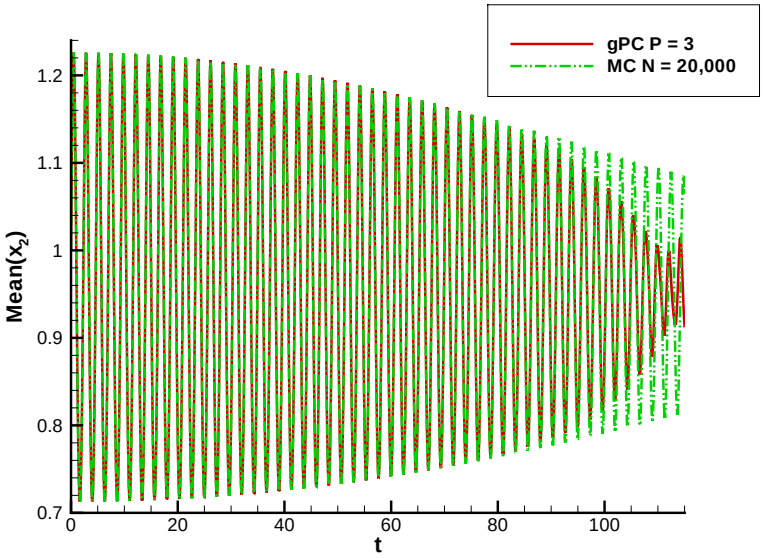

Figure A.23: Mean of x_2 vs. time for $\alpha = 0.7$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 20,000$)

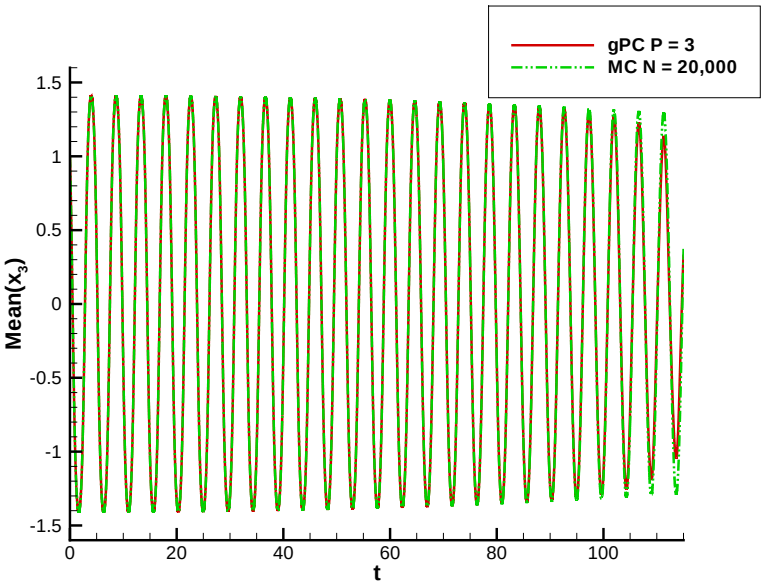

Figure A.24: Mean of x_3 vs. time for $\alpha = 0.7$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 20,000$)

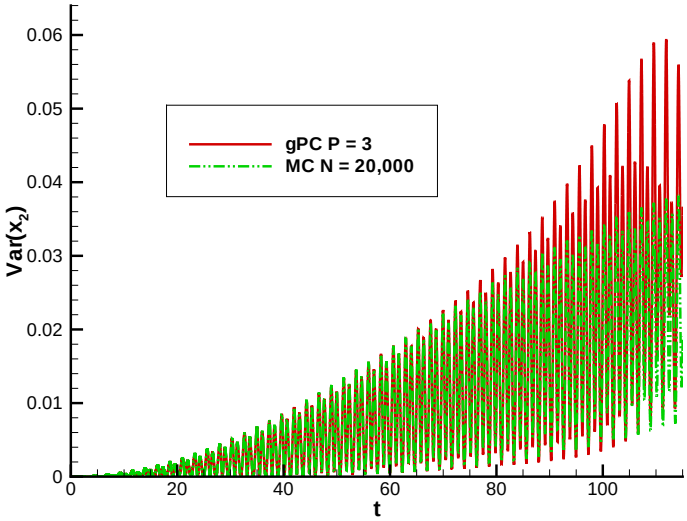

Figure A.25: Variance of x_2 vs. time for $\alpha = 0.7$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 20,000$)

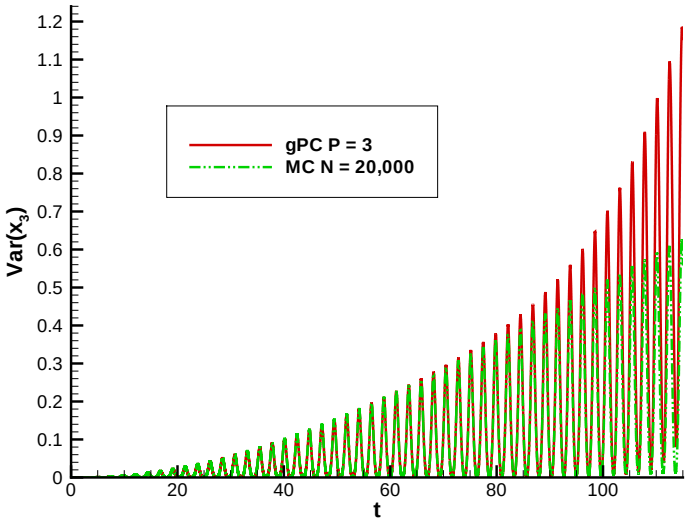

Figure A.26: Variance of x_3 vs. time for $\alpha = 0.7$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 20,000$)

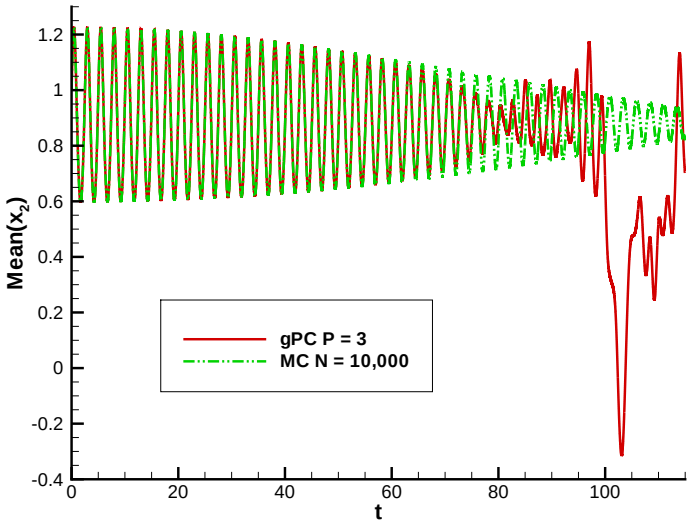

Figure A.27: Mean of x_2 vs. time for $\alpha = 0.8$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

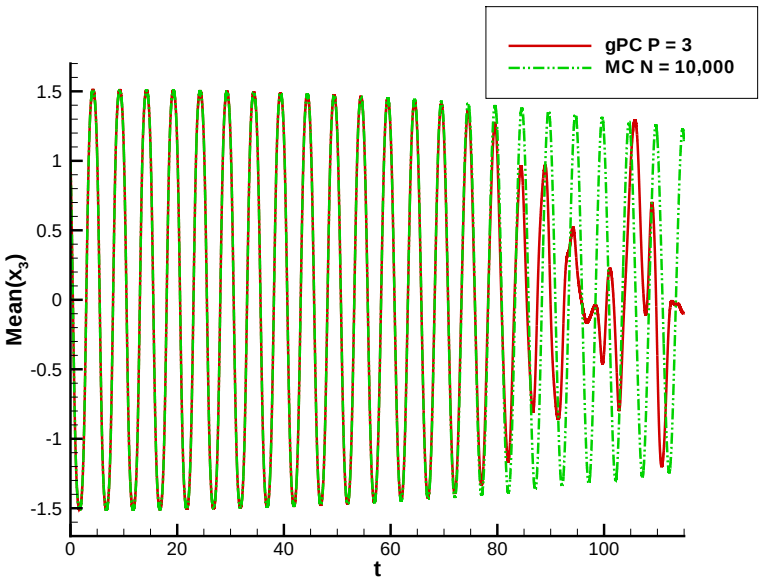

Figure A.28: Mean of x_3 vs. time for $\alpha = 0.8$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

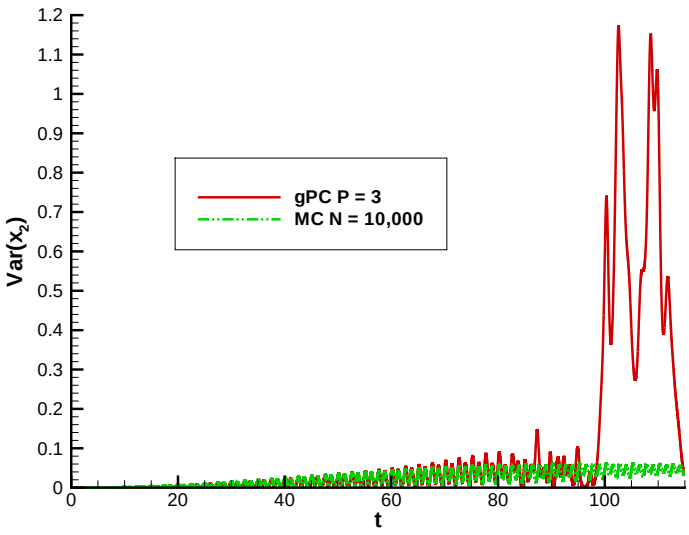

Figure A.29: Variance of x_2 vs. time for $\alpha = 0.8$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

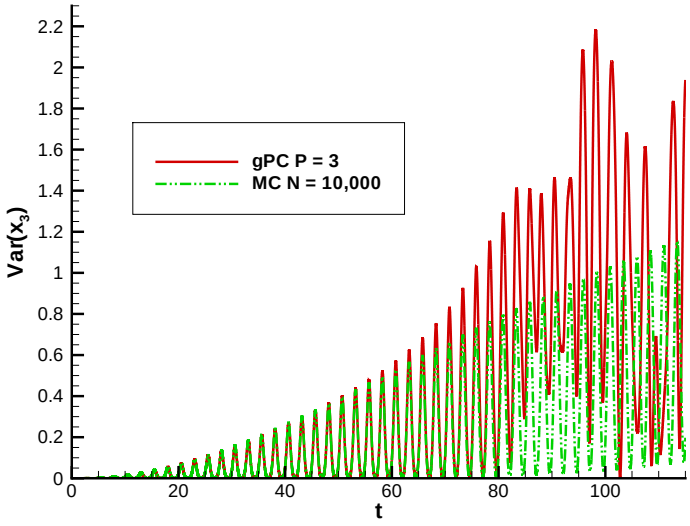

Figure A.30: Variance of x_3 vs. time for $\alpha = 0.8$: gPC solution with $P = 3$ compared to a Monte Carlo analysis ($N = 10,000$)

A.3 Kraichnan-Orszag: TDgPC approach

In Chapter 6 we have discussed results of a TDgPC solution approach to the Kraichnan-Orszag problem, presenting results for x_1 only. In this appendix the corresponding TDgPC results for x_2 and x_3 can be found.

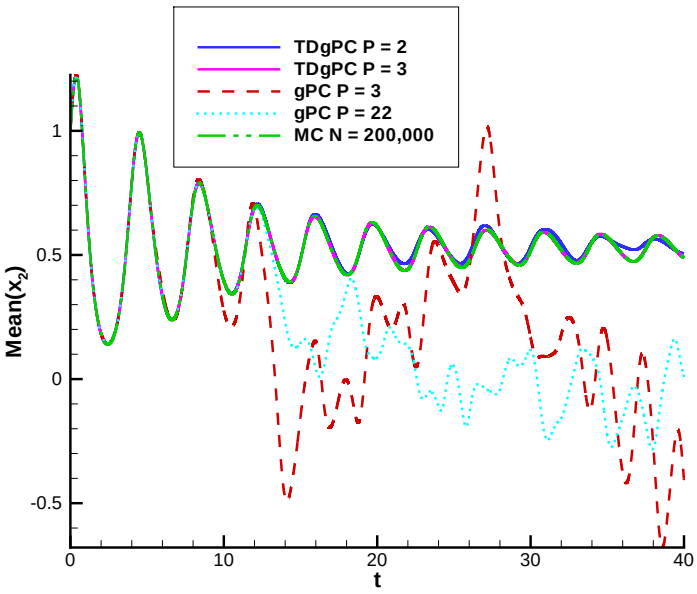

Figure A.31: Mean of x_2 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$ and $P = 3$ compared to gPC solutions with $P = 3$ and $P = 22$ and a Monte Carlo analysis ($N = 200,000$)

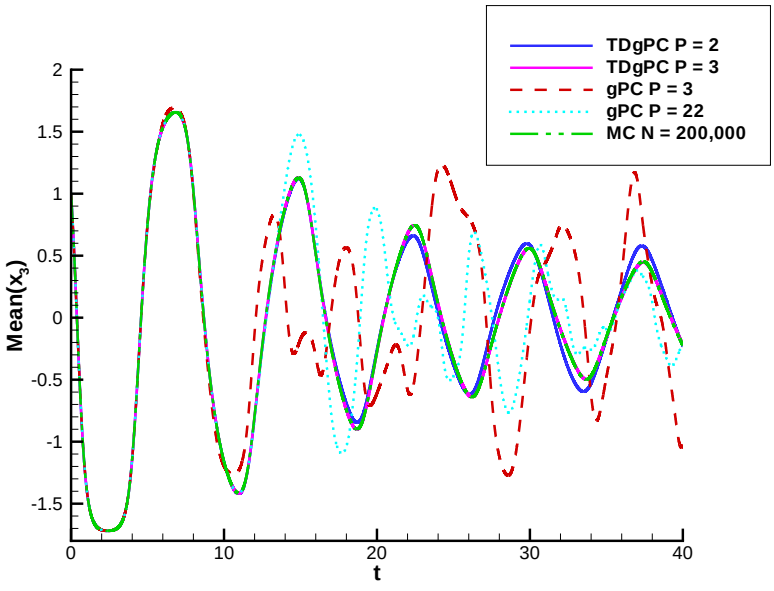

Figure A.32: Mean of x_3 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$ and $P = 3$ compared to gPC solutions with $P = 3$ and $P = 22$ and a Monte Carlo analysis ($N = 200,000$)

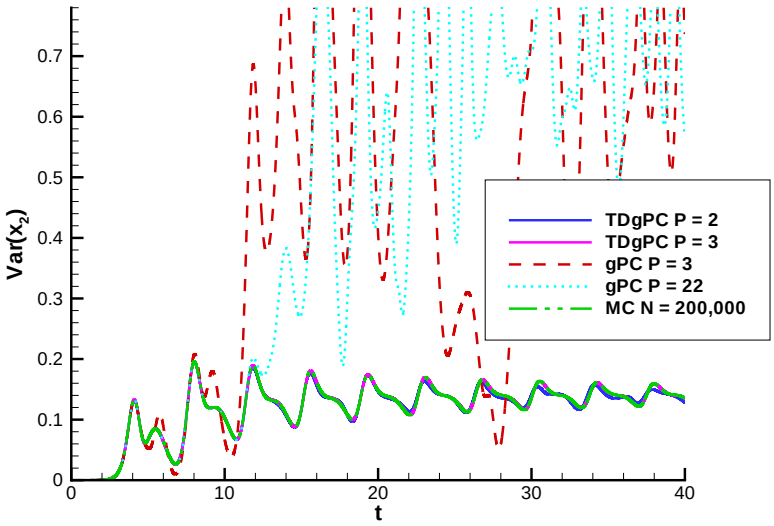

Figure A.33: Variance of x_2 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$ and $P = 3$ compared to gPC solution with $P = 3$ and $P = 22$ and a Monte Carlo analysis ($N = 200,000$)

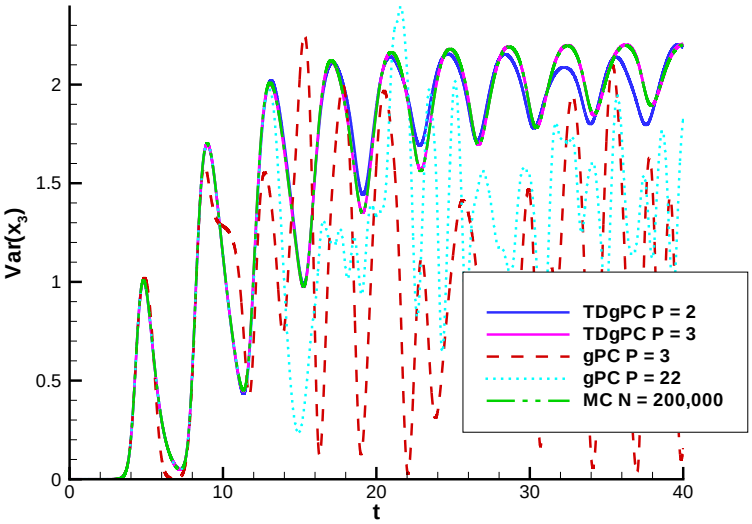

Figure A.34: Variance of x_3 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$ and $P = 3$ compared to gPC solution with $P = 3$ and $P = 22$ and a Monte Carlo analysis ($N = 200,000$)

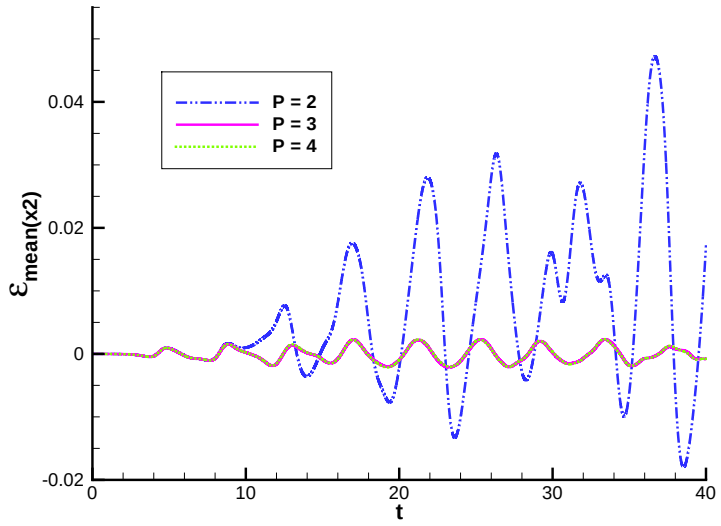

Figure A.35: Error in the mean of x_2 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$, $P = 3$ and $P = 4$

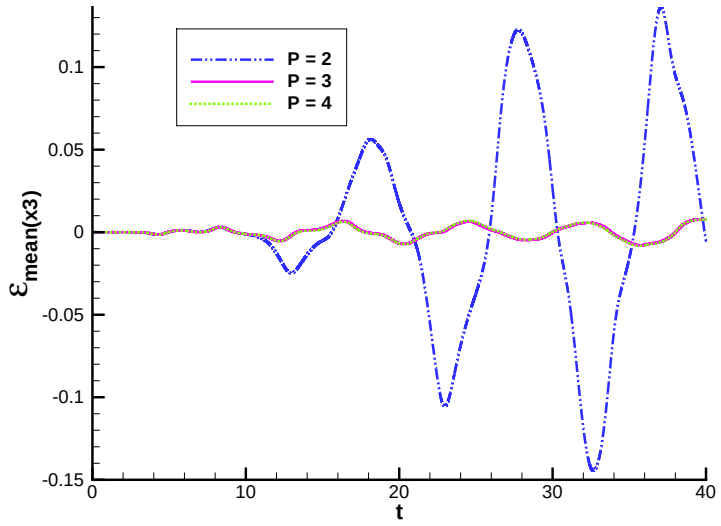

Figure A.36: Error in the mean of x_3 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$, $P = 3$ and $P = 4$

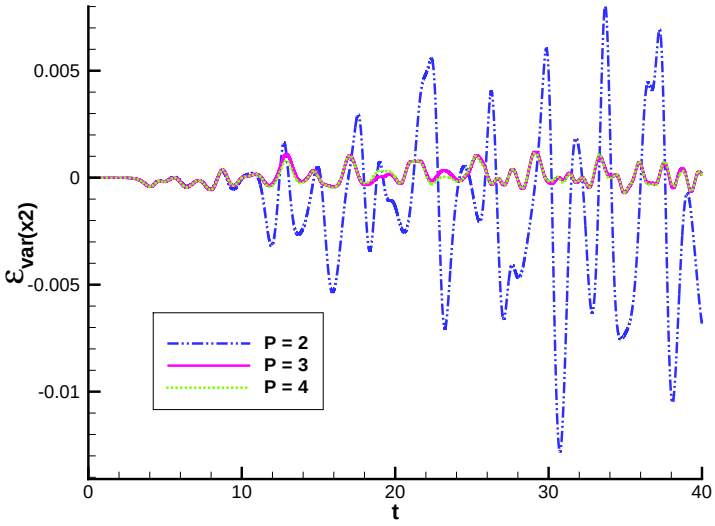

Figure A.37: Error in the variance of x_2 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$, $P = 3$ and $P = 4$

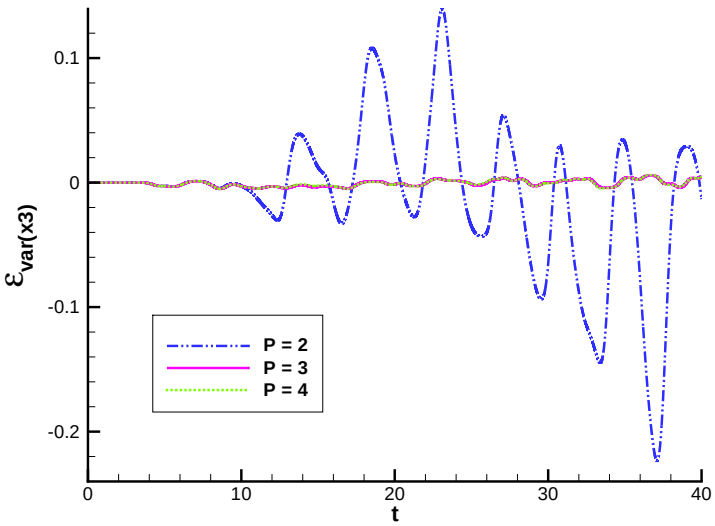

Figure A.38: Error in the variance of x_3 vs. time for $\alpha = 0.99$: TDgPC solutions with $P = 2$, $P = 3$ and $P = 4$

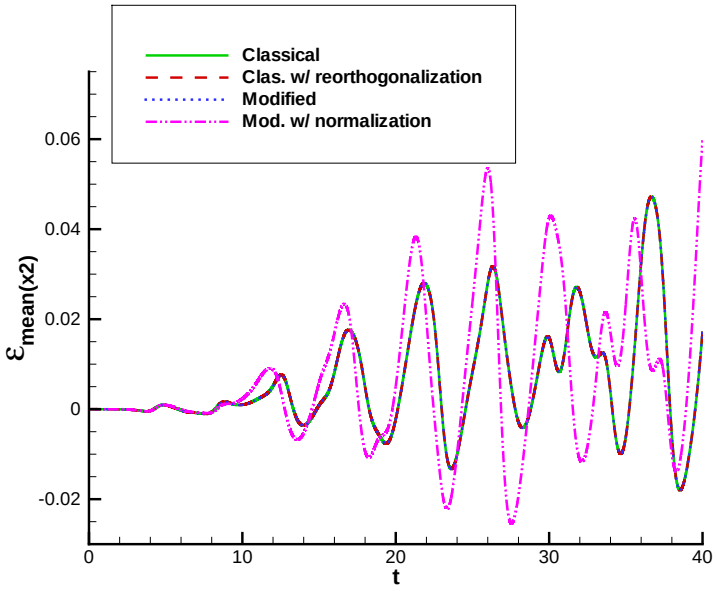

Figure A.39: Error in the mean of x_2 vs. time for $\alpha = 0.99$: TDgPC results using various Gram-Schmidt orthogonalization algorithms

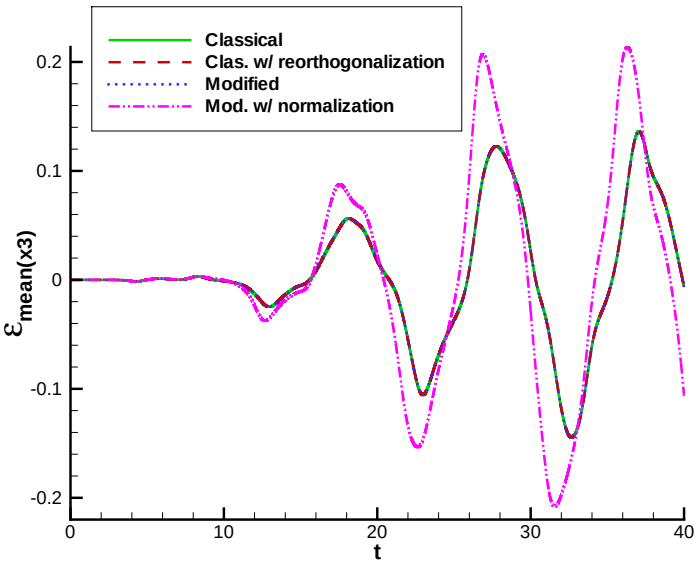

Figure A.40: Error in the mean of x_3 vs. time for $\alpha = 0.99$: TDgPC results using various Gram-Schmidt orthogonalization algorithms

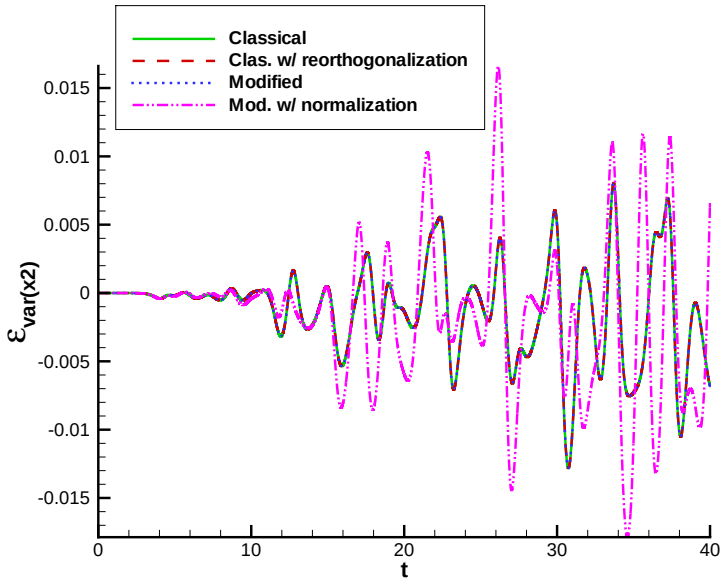

Figure A.41: Error in the variance of x_2 vs. time for $\alpha = 0.99$: TDgPC results using various Gram-Schmidt orthogonalization algorithms

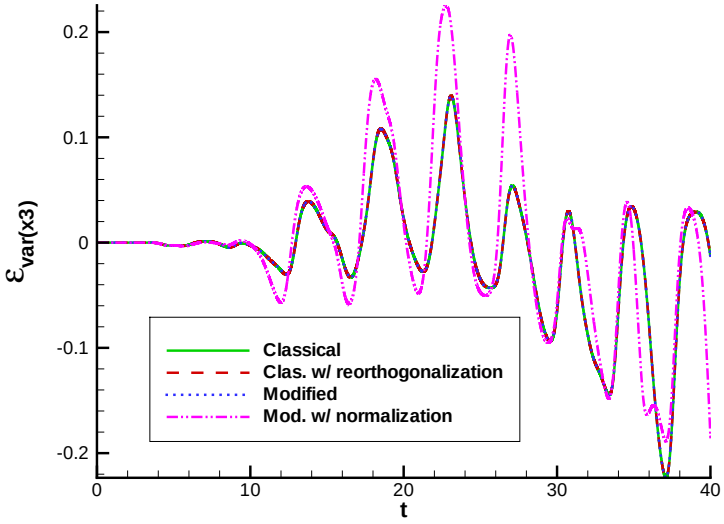

Figure A.42: Error in the variance of x_3 vs. time for $\alpha = 0.99$: TDgPC results using various Gram-Schmidt orthogonalization algorithms

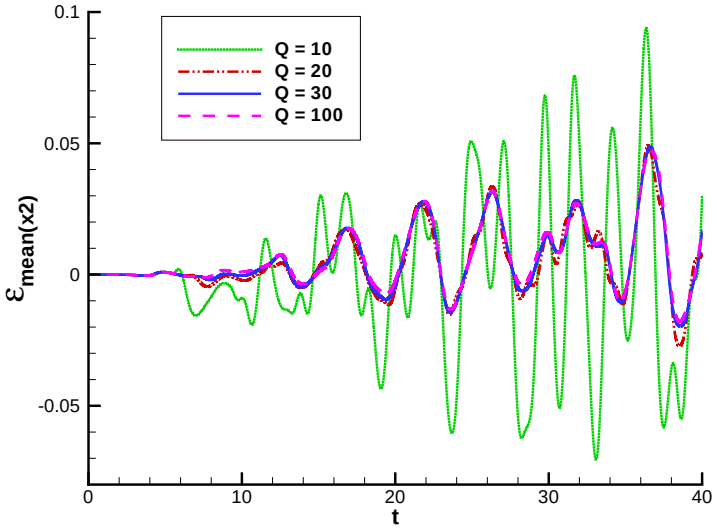

Figure A.43: Error in the mean of x_2 vs. time for $\alpha = 0.99$: TDgPC results with $P = 2$ using various amounts of quadrature points Q

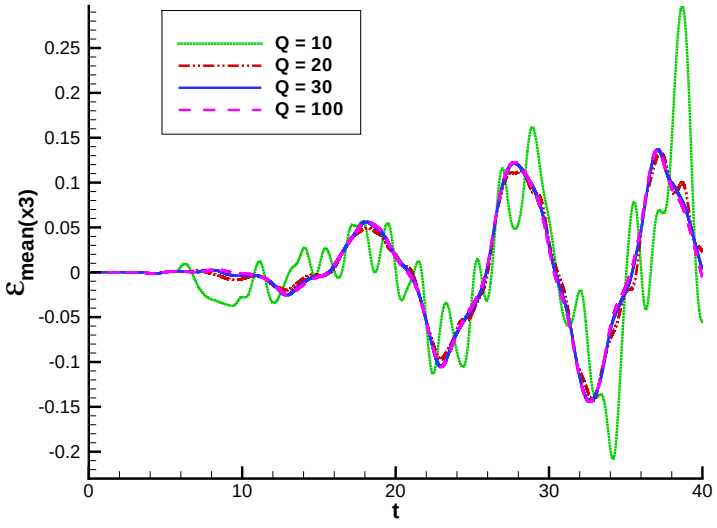

Figure A.44: Error in the mean of x_3 vs. time for $\alpha = 0.99$: TDgPC results with $P = 2$ using various amounts of quadrature points Q

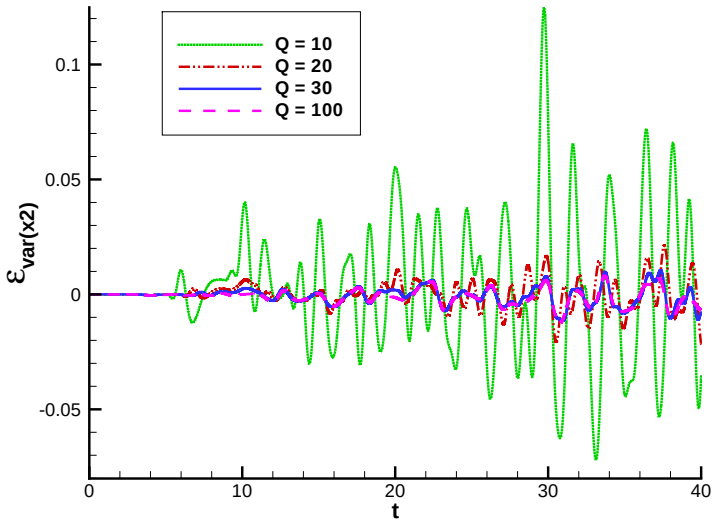

Figure A.45: Error in the variance of x_2 vs. time for $\alpha = 0.99$: TDgPC results with $P = 2$ using various amounts of quadrature points Q

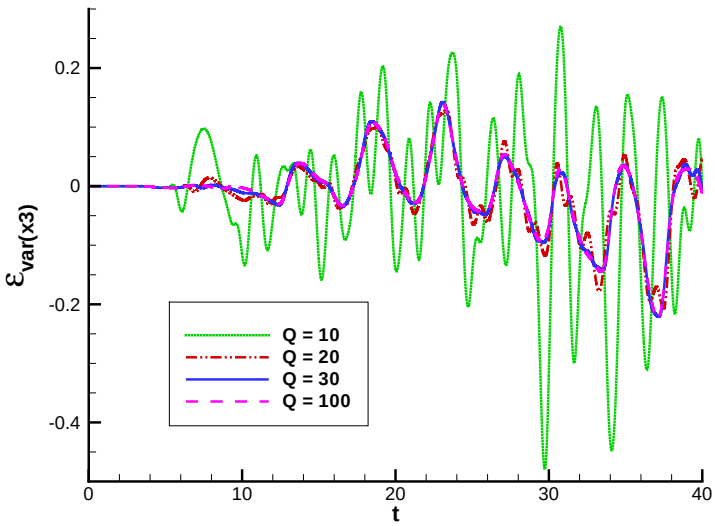

Figure A.46: Error in the variance of x_3 vs. time for $\alpha = 0.99$: TDgPC results with $P = 2$ using various amounts of quadrature points Q

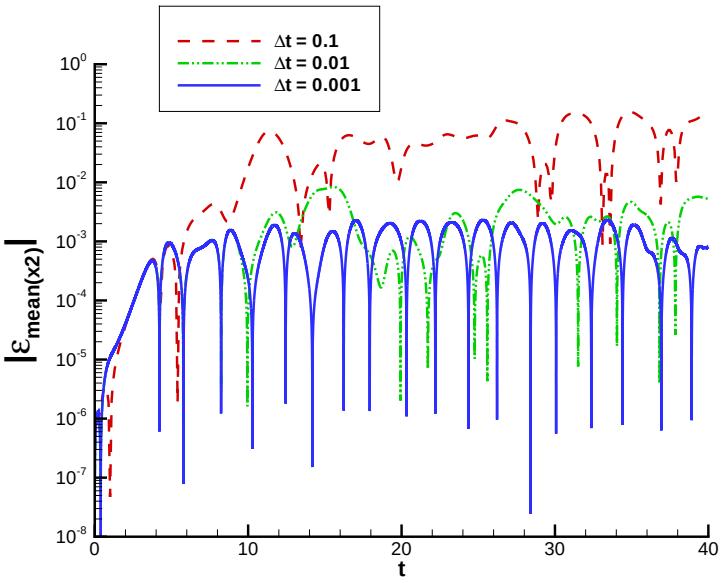

Figure A.47: Error in the mean of x_2 vs. time for $\alpha = 0.99$: TDgPC results with $P = 3$ for various time steps Δt

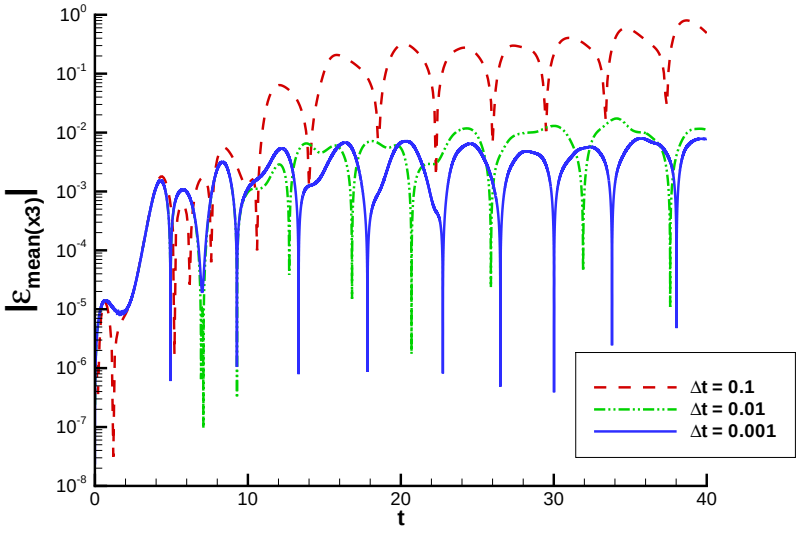

Figure A.48: Error in the mean of x_3 vs. time for $\alpha = 0.99$: TDgPC results with $P = 3$ for various time steps Δt

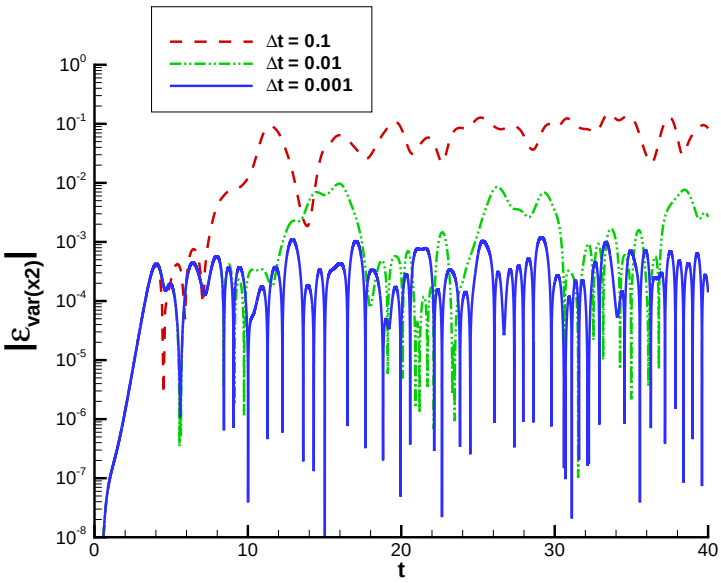

Figure A.49: Error in the variance of x_2 vs. time for $\alpha = 0.99$: TDgPC results with $P = 3$ for various time steps Δt

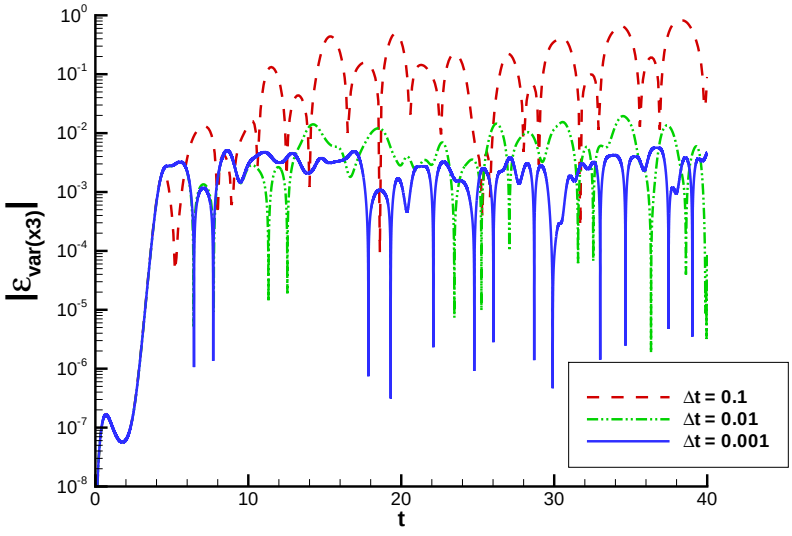

Figure A.50: Error in the variance of x_3 vs. time for $\alpha = 0.99$: TDgPC results with $P = 3$ for various time steps Δt

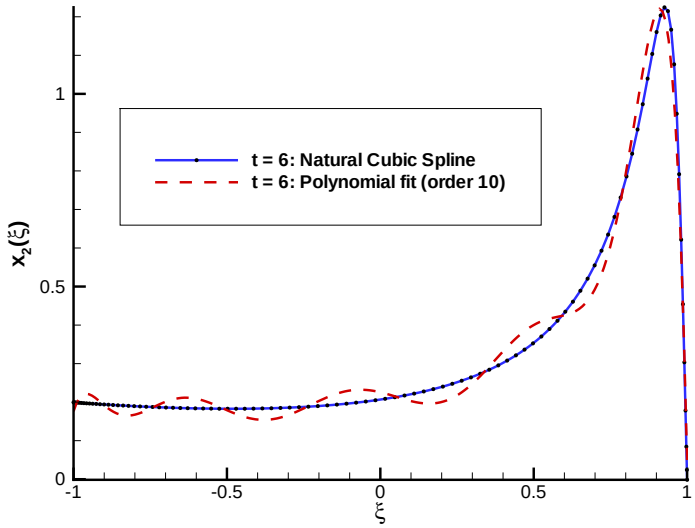

Figure A.51: TDgPC calculation for $\alpha = 0.99$ with $P = 2$: x_2 as a function of ξ at $t = 6$

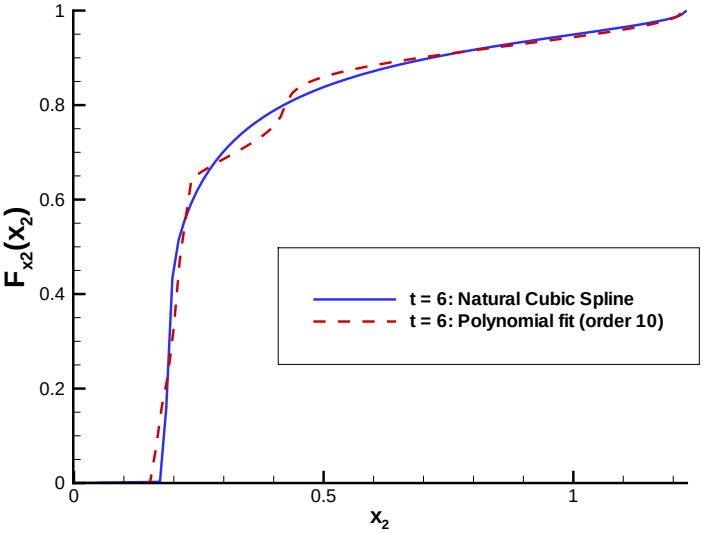

Figure A.52: TDgPC calculation for $\alpha = 0.99$ with $P = 2$: Distribution function of x_2 at $t = 6$

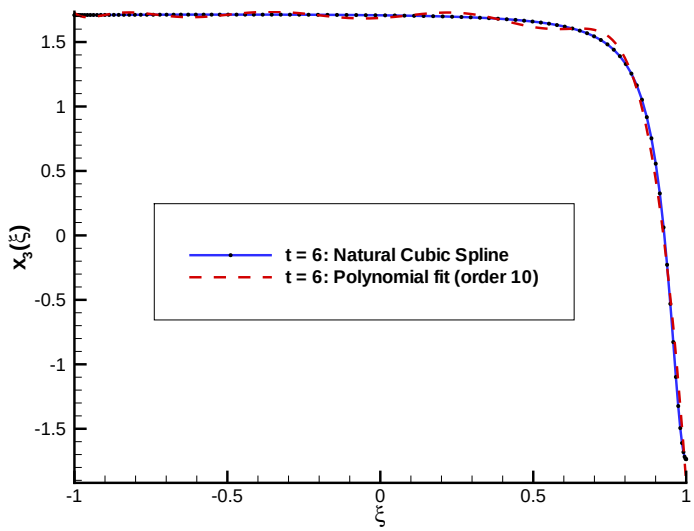

Figure A.53: TDgPC calculation for $\alpha = 0.99$ with $P = 2$: x_3 as a function of ξ at $t = 6$

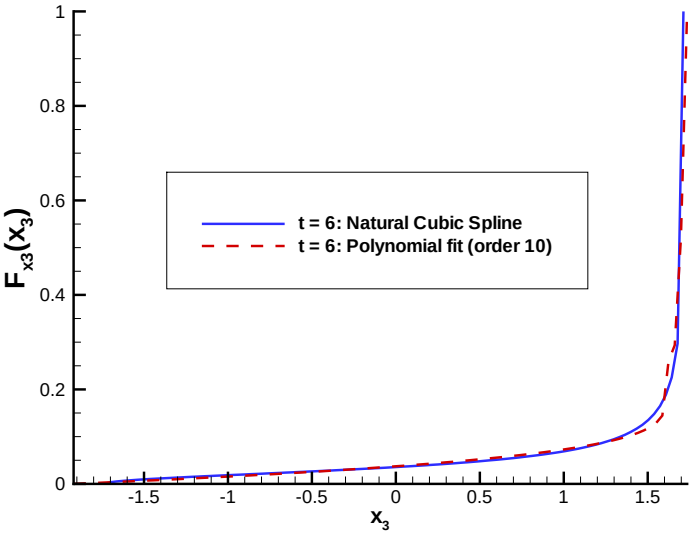

Figure A.54: TDgPC calculation for $\alpha = 0.99$ with $P = 2$: Distribution function of x_3 at $t = 6$

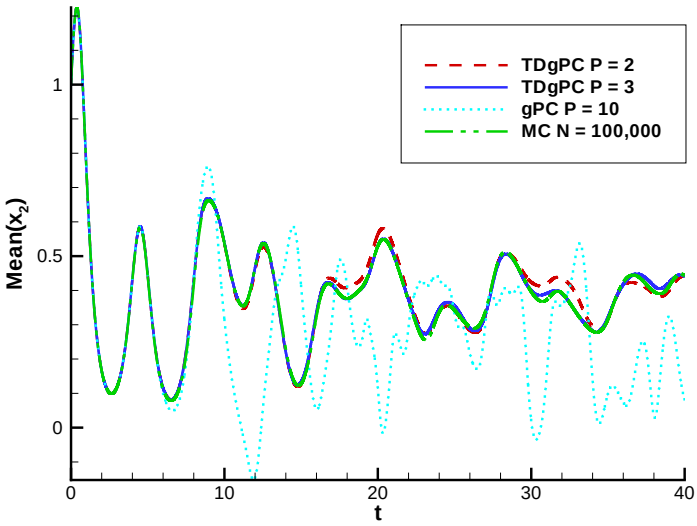

Figure A.55: Mean of x_2 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$ compared to a gPC solution with $P = 10$ and a Monte Carlo analysis ($N = 100,000$)

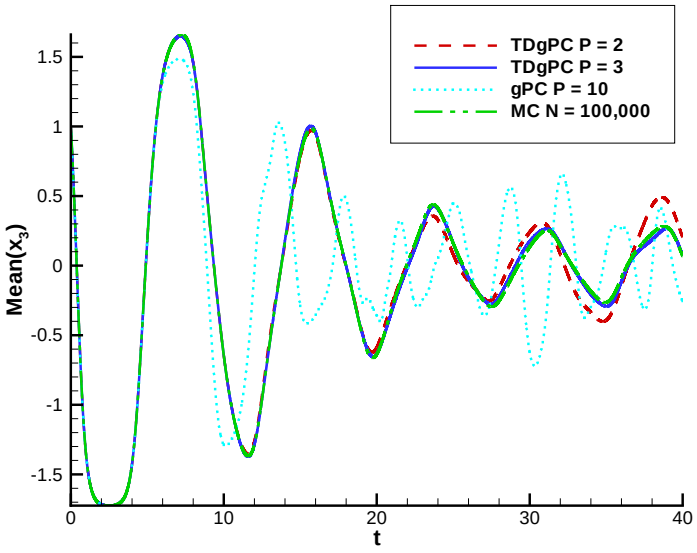

Figure A.56: Mean of x_3 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$ compared to a gPC solution with $P = 10$ and a Monte Carlo analysis ($N = 100,000$)

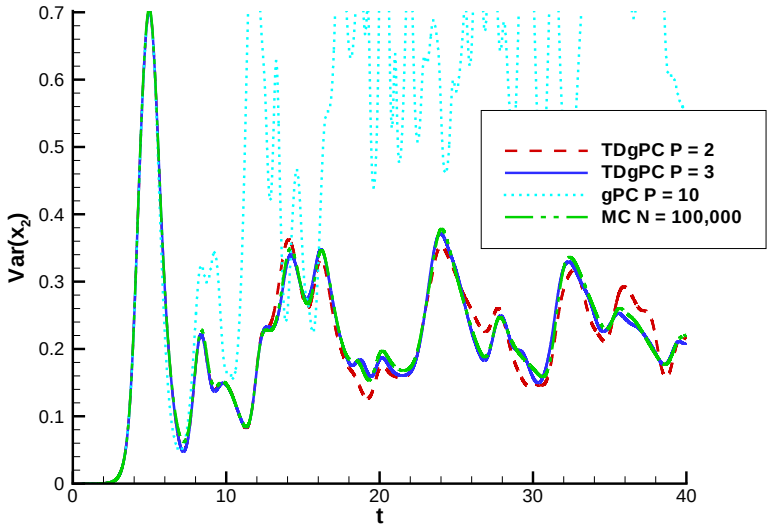

Figure A.57: Variance of x_2 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$ compared to a gPC solution with $P = 10$ and a Monte Carlo analysis ($N = 100,000$)

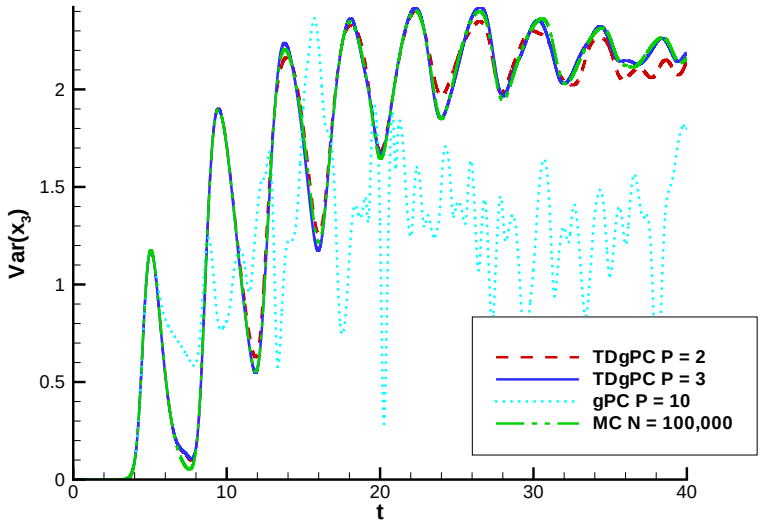

Figure A.58: Variance of x_3 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$ compared to a gPC solution with $P = 10$ and a Monte Carlo analysis ($N = 100,000$)

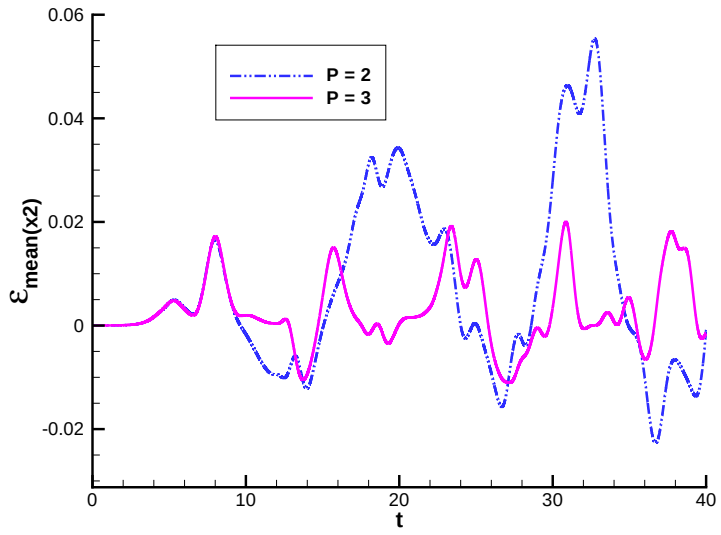

Figure A.59: Error in the mean of x_2 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$

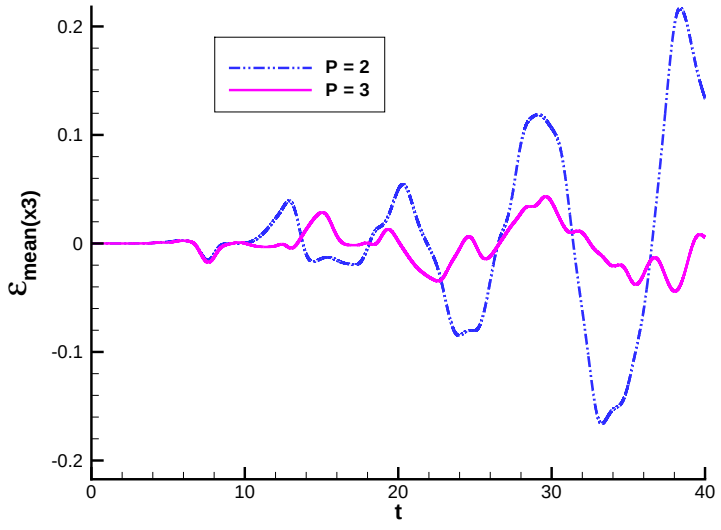

Figure A.60: Error in the mean of x_3 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$

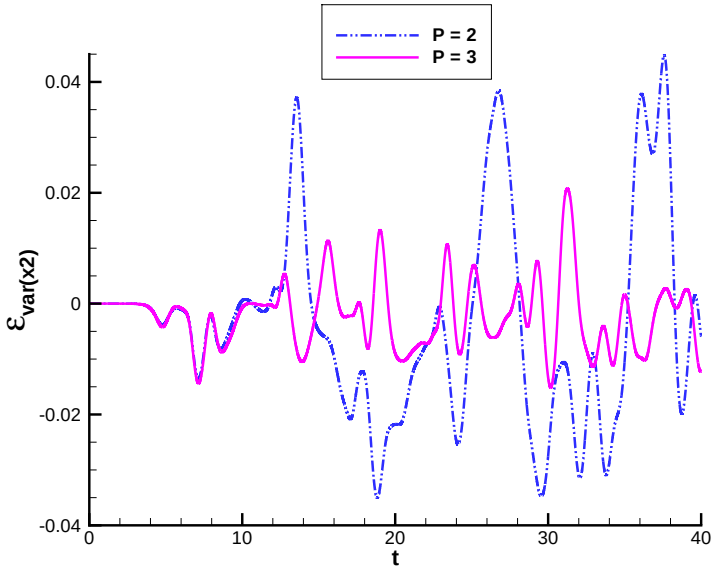

Figure A.61: Error in the variance of x_2 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$

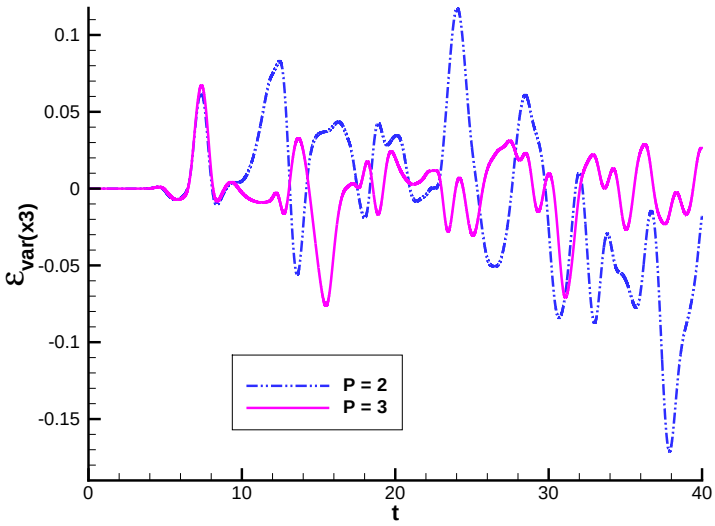

Figure A.62: Error in the variance of x_3 vs. time for $\alpha = 0.995$: TDgPC solutions with $P = 2$ and $P = 3$

A.4 Kraichnan-Orszag 3D: TDgPC approach

In Chapter 7 we have discussed results of a TDgPC solution approach to the Kraichnan-Orszag problem with a three-dimensional random input, presenting results for x_1 only. In this appendix the corresponding TDgPC results for x_2 and x_3 can be found.

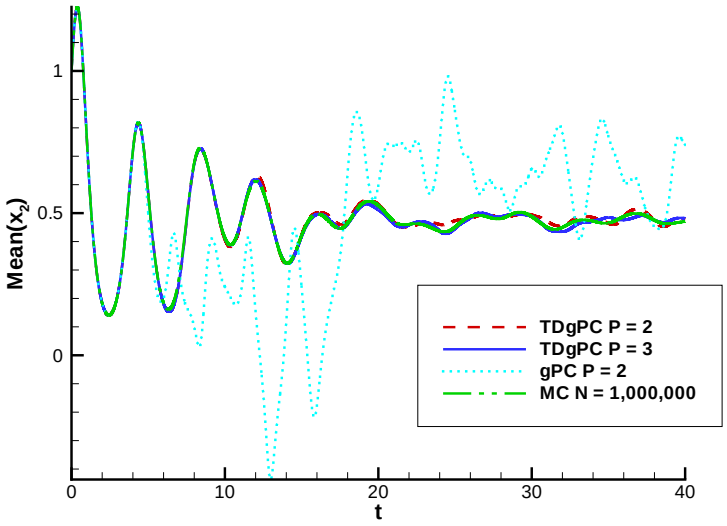

Figure A.63: Mean of x_2 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions ($P = 2$ and $P = 3$) compared to a gPC solution ($P = 2$) and a Monte Carlo analysis ($N = 1,000,000$)

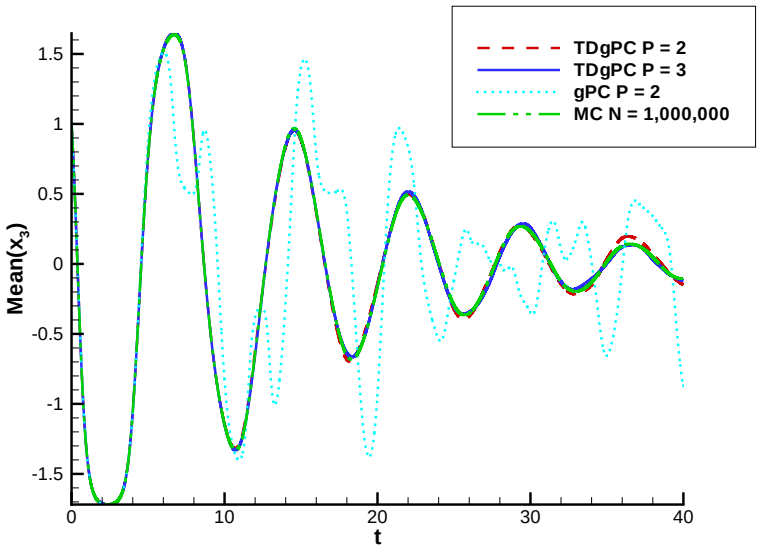

Figure A.64: Mean of x_3 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions ($P = 2$ and $P = 3$) compared to a gPC solution ($P = 2$) and a Monte Carlo analysis ($N = 1,000,000$)

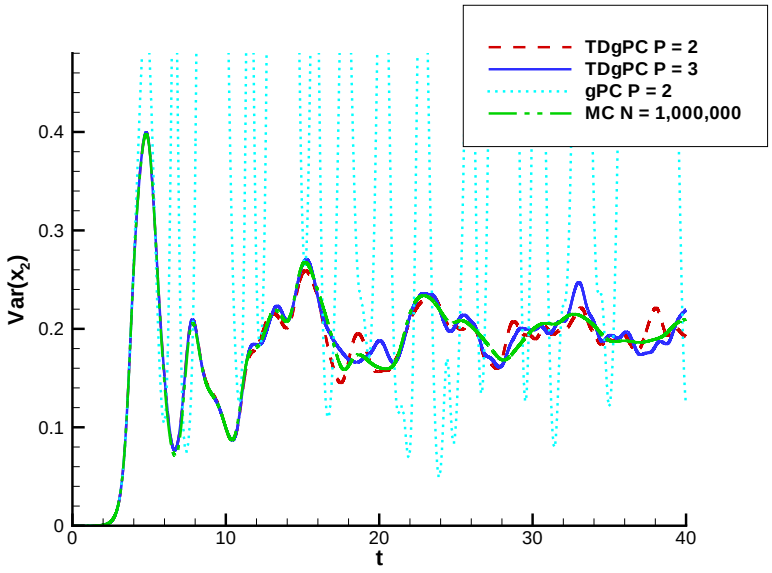

Figure A.65: Variance of x_2 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions ($P = 2$ and $P = 3$) compared to a gPC solution ($P = 2$) and a Monte Carlo analysis ($N = 1,000,000$)

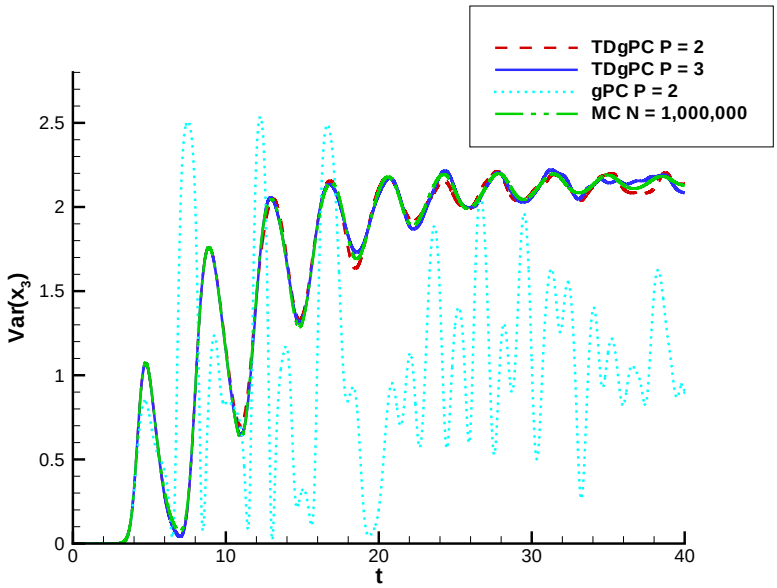

Figure A.66: Variance of x_3 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions ($P = 2$ and $P = 3$) compared to a gPC solution ($P = 2$) and a Monte Carlo analysis ($N = 1,000,000$)

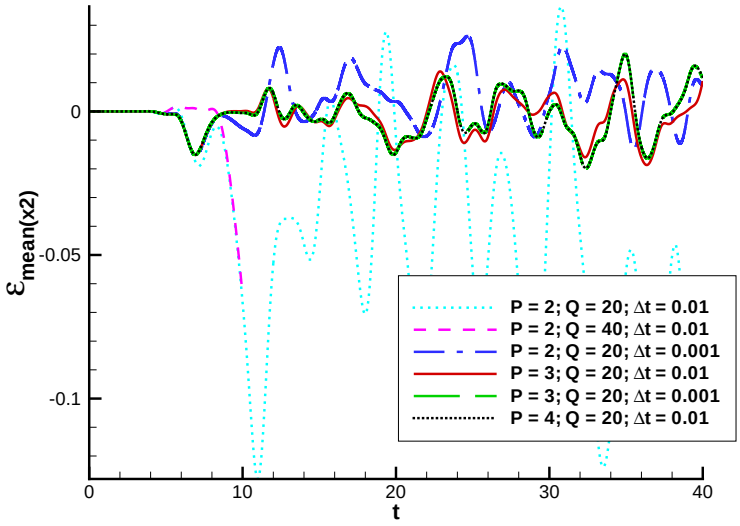

Figure A.67: Error in the mean of x_2 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions with different P, Q and Δt

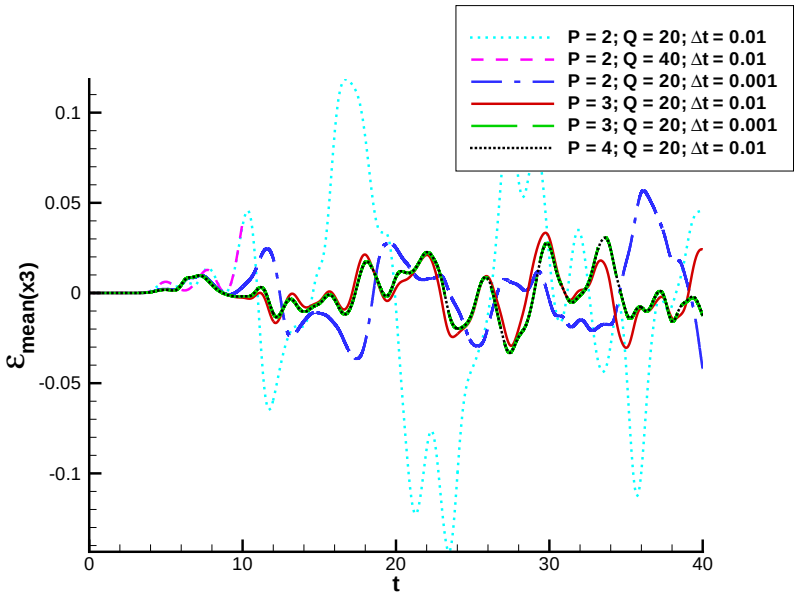

Figure A.68: Error in the mean of x_3 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions with different P, Q and Δt

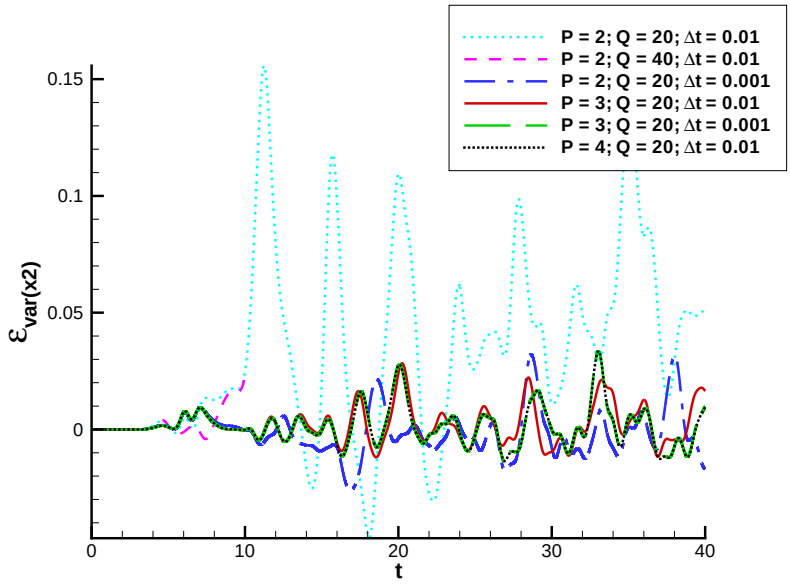

Figure A.69: Error in the variance of x_2 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions with different P, Q and Δt

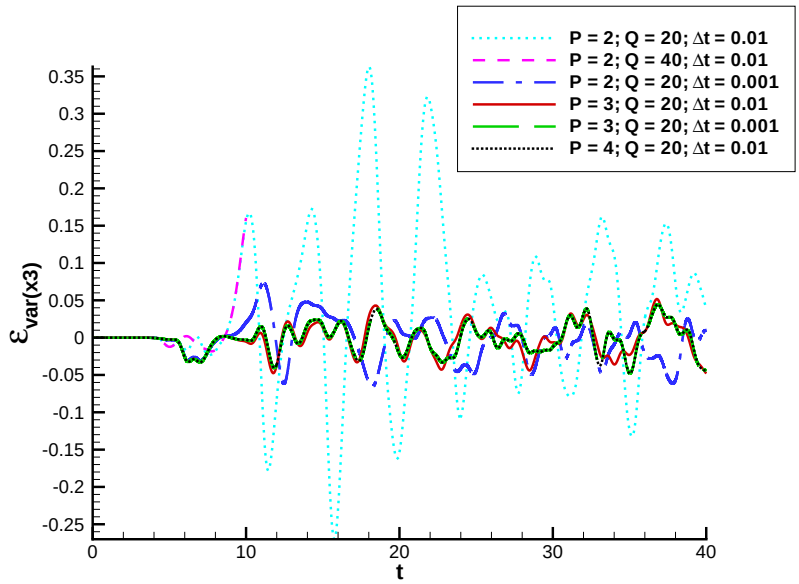

Figure A.70: Error in the variance of x_3 vs. time for $\alpha = 0.99$, $\beta = 1$ and $\gamma = 1$: TDgPC solutions with different P, Q and Δt

Bibliography

[1] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, October 1938.

[2] Robert H. Kraichnan. Direct-interaction approximation for a system of several interacting simple shear waves. *The Physics of Fluids*, 6(11):1603–1609, November 1963.

[3] Steven A. Orszag and L.R. Bissonnette. Dynamical properties of truncated Wiener-Hermite expansions. *The Physics of Fluids*, 10(12):2603–2613, December 1967.

[4] Roger G. Ghanem and Pol D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, New York, 1991.

[5] Dongbin Xiu and George Em Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[6] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, May 2003.

[7] Jeroen A.S. Witteveen and Hester Bijl. Modeling arbitrary uncertainties using Gram-Schmidt polynomial chaos. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, January 2006. AIAA 2006-896.

[8] Xiaoliang Wan and George Em Karniadakis. Beyond Wiener-Askey expansions: Handling arbitrary PDFs. *Journal of Scientific Computing*, 27(1–3): 455–464, June 2006.

[9] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, November 2005.

[10] Peter Vos. Time-dependent polynomial chaos. Master's thesis, Delft University of Technology, November 2006.

[11] Marc Gerritsma, Peter Vos, and Jan-Bart van der Steen. Time-dependent polynomial chaos. In Theodore E. Simos, George Psihoyios, and Ch. Tsitouras, editors, *Numerical Analysis and Applied Mathematics: International Conference on Numerical Analysis and Applied Mathematics 2008*, pages 221–224, Melville, NY, 2008. European Society of Computational Methods in Sciences and Engineering, American Institute of Physics.

[12] R.H. Cameron and W.T. Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, 48(2):385–392, April 1947.

[13] H. Ogura. Orthogonal functionals of the poisson process. *IEEE Transactions on Information Theory*, 18(4):473–481, July 1972.

[14] Wikipedia: Multiset. `http://en.wikipedia.org/wiki/Multiset`, February 2008.

[15] Derek F. Lawden. *Elliptic Functions and Applications*. Applied Mathematical Sciences 80. Springer-Verlag, New York, 1989.

[16] George Em Karniadakis and Spencer Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, second edition, 2005.

[17] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Series in Systems Science. McGraw-Hill, New York, 1965.

[18] Alexander M. Mood, Franklin A. Graybill, and Duane C. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill Series in Probability and Statistics. McGraw-Hill, Singapore, third edition, 1974.

[19] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Texts in Applied Mathematics 12. Springer-Verlag, New York, second edition, 1993.

[20] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. John Hopkins Studies in the Mathematical Sciences. John Hopkins University Press, Baltimore, third edition, 1996.