

Uncertainty Analysis of Power Systems Using Collocation

by

Joshua Adam Taylor

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author
Department of Mechanical Engineering
May 9, 2008

Certified by
Franz Hover
Assistant Professor
Thesis Supervisor

Accepted by
Lallit Anand
Chairman, Department Committee on Graduate Students

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE MAY 2008		2. REPORT TYPE		3. DATES COVERED 00-00-2008 to 00-00-2008	
4. TITLE AND SUBTITLE Uncertainty Analysis of Power Systems Using Collocation				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, 02139				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The next-generation all-electric ship represents a class of design and control problems in which the system is too large to approach analytically, and even with many conventional computational techniques. Additionally, numerous environmental interactions and inaccurate system model information make uncertainty a necessary consideration. Characterizing systems under uncertainty is essentially a problem of representing the system as a function over a random space. This can be accomplished by sampling the function, where in the case of the electric ship a ?sample? is a simulation with uncertain parameters set according to the location of the sample. For systems on the scale of the electric ship, simulation is expensive, so we seek an accurate representation of the system from a minimal number of simulations. To this end, collocation is employed to compute statistical moments, from which sensitivity can be inferred and to construct surrogate models with which interpolation can be used to propagate PDF?s. These techniques are applied to three large-scale electric ship models. The conventional formulation for the sparse grid, a collocation algorithm, is modified to yield improved performance. Theoretical bounds and computational examples are given to support the modification. A dimension-adaptive collocation algorithm is implemented in an unscented Kalman filter, and improvement over extended Kalman and unscented filters is seen in two examples.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 97	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Uncertainty Analysis of Power Systems Using Collocation

by

Joshua Adam Taylor

Submitted to the Department of Mechanical Engineering
on May 9, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

The next-generation all-electric ship represents a class of design and control problems in which the system is too large to approach analytically, and even with many conventional computational techniques. Additionally, numerous environmental interactions and inaccurate system model information make uncertainty a necessary consideration. Characterizing systems under uncertainty is essentially a problem of representing the system as a function over a random space. This can be accomplished by sampling the function, where in the case of the electric ship a "sample" is a simulation with uncertain parameters set according to the location of the sample. For systems on the scale of the electric ship, simulation is expensive, so we seek an accurate representation of the system from a minimal number of simulations. To this end, collocation is employed to compute statistical moments, from which sensitivity can be inferred, and to construct surrogate models with which interpolation can be used to propagate PDF's. These techniques are applied to three large-scale electric ship models. The conventional formulation for the sparse grid, a collocation algorithm, is modified to yield improved performance. Theoretical bounds and computational examples are given to support the modification. A dimension-adaptive collocation algorithm is implemented in an unscented Kalman filter, and improvement over extended Kalman and unscented filters is seen in two examples.

Thesis Supervisor: Franz Hover

Title: Assistant Professor

Acknowledgments

I would like to thank my adviser, Professor Franz Hover, for being inspiring and motivating, and, perhaps more importantly, considerate. Thank you to my parents, for more than I could possibly say here. To Rachel and David, for helping ensure that I never actually become an adult. To the rest of my family, for always being unconditionally behind me. And of course, to my friends, new ones I've made here and those in other places. You all have made every step of the way fun.

Support is acknowledged from the Office of Naval Research, Grant N00014-02-1-0623.

Contents

1	Introduction	15
2	Uncertainty in Dynamic Systems: Setting the Stage	17
2.1	Uncertainty in a First Order System	18
2.2	Uncertainty in a Double Pendulum	19
2.3	Uncertainty in the All-Electric Ship	20
2.4	Summary	22
3	Technical Overview	23
3.1	Numerical Integration	23
3.1.1	In One Dimension	24
3.1.2	In Multiple Dimensions	26
3.2	Interpolation	32
3.3	Summary	34
4	Comparison of Integration Methods	37
4.1	Test functions	37
4.2	Convergence of Methods	39
4.2.1	Non-adaptive methods	39
4.2.2	Dimension-adaptivity	42
4.3	Sparse Grids with Shifted Indexing	44
4.3.1	One-Dimensional Quadratures and Shifting	44
4.3.2	Error Bounds	44

4.3.3	Numerical Results	48
4.3.4	Discussion	50
4.4	Summary	51
5	Stochastic Simulation of the All-Electric Ship	53
5.1	ONR IPS Testbed	53
5.1.1	Deterministic Simulation	53
5.1.2	Probabilistic Simulation	54
5.1.3	Adaptive Simulation	59
5.2	Pulse Power System	61
5.3	RTDS	61
5.4	Discussion	64
6	Unscented Kalman Filtering with Dimension-Adaptive Numerical Integration	65
6.1	Background	65
6.1.1	The Extended Kalman Filter	66
6.1.2	The Unscented Kalman Filter	66
6.2	Dimension-Adaptive Unscented Kalman Filtering	67
6.3	Performance of the New Filter	68
6.3.1	Falling Body Problem	68
6.3.2	The Kraichnan-Orszag System	69
6.3.3	The Lorenz system	74
6.4	Summary	77
7	Summary	79
A	Figures from FSU CAPS Collaboration	81
B	Readme for Matlab implementation of Collocation Algorithms	83
B.1	Overview and introduction	83
B.2	Using <i>Collocation.m</i>	84

B.2.1	A warning about computational costs	88
B.3	Description of each program	88
B.4	Test functions	91
B.5	Running <i>CollocationDemo.m</i>	92

List of Figures

2-1	Double pendulum with uniform uncertainty in the initial positions. . .	20
2-2	Mean solutions with standard deviation envelopes for a double pendulum with uniform initial position uncertainty.	21
3-1	Left: Full grid with Legendre polynomial Gaussian quadrature. Right: Sparse grid with Gauss-Kronrod-Patterson quadrature.	29
3-2	Nine element Clenshaw-Curtis dimension-adaptive integration of the two dimensional instantiation of test function five from [11]. The nodes chosen by the algorithm are shown on the left, and the function evaluations at those points on the right.	30
3-3	Left: 100 pseudo-random points. Right: 100 quasi-random points from the Sobol sequence.	32
3-4	Interpolated and directly evaluated trajectories of double pendulum states.	34
3-5	Interpolated and directly evaluated histograms of double pendulum states at 2.5 seconds.	35
4-1	Two dimensional test functions	38
4-2	Log-log plots of error versus number of function evaluations for product peak, oscillatory, and corner peak test functions. Legend entries correspond to: FG-GQ - Full grid with Gaussian Quadrature, SG-CC - Sparse grid with Clenshaw-Curtis, MC - Monte Carlo, QMC - Quasi-Monte Carlo.	40

4-3	Log-log plots of error versus number of function evaluations for Gaussian, C^0 , and discontinuous test functions.	41
4-4	Log-log plots of error versus number of function evaluations. Sparse grid with Gauss-Kronrod-Patterson (SG-KP), full grid with Gaussian Quadrature (FG-GQ), and dimension-adaptivity with Gauss-Kronrod-Patterson (DA-KP) in three dimensions are shown.	42
4-5	Log-log plots of error versus number of function evaluations for algorithms in six dimensions are shown.	43
4-6	Theoretical error convergence of conventional, once-shifted, and twice-shifted sparse grids with Clenshaw-Curtis and Kronrod-Patterson basis quadratures.	49
4-7	Error versus function evaluations for regular and shifted sparse grids using Clenshaw-Curtis and Kronrod-Patterson quadratures	50
5-1	Schematic for ONR IPS Testbed.	54
5-2	Deterministic trajectories for ONR IPS.	55
5-3	Starboard AC system generator abc -reference frame c current mean (curve) and standard deviation (error bars) solutions for the triple containing starboard bus shunt capacitance, filter capacitance and filter inductance for ONR IPS.	56
5-4	Two dimensional integrated variance analysis of the starboard AC system generator abc -reference frame c current: all triples containing the starboard bus filter inductance (3) for ONR IPS.	57
5-5	Starboard AC system normalized integrated variances and multi-dimensional sensitivity (MS) for ONR IPS.	58
5-6	Multi-dimensional sensitivity (MS) of starboard power supply currents for $\pm 1 - 17\%$ uncertainty for ONR IPS.	59

5-7	Record of dimension adaptive behavior in each element for a two dimensional, four element simulation with uncertainty in the bus shunt and filter capacitances for ONR IPS. Each location on the plot represents a difference grid in the corresponding element. Recently evaluated difference grids are denoted by + and old difference grids by O. The title of each quadrant is the portion of the integral's domain covered by that element. It also corresponds to the percent variation from the mean of the uncertain couple.	60
5-8	Mean trajectories with standard deviation envelopes of pulse power system states.	62
5-9	Histogram of pulse power system states at $t = 30$ seconds.	63
6-1	Mean absolute error for position, velocity, and x_3 for falling body problem.	70
6-2	Trajectories of the Kraichnan-Orszag system.	71
6-3	Error, variance, and mean number of points propagated for dimension-adaptive filters with different error tolerances on Kraichnan-Orszag system.	72
6-4	Noisy Kraichnan-Orszag trajectories.	73
6-5	Performance of filters on the Kraichnan-Orszag system.	74
6-6	Trajectories of the Lorenz system.	75
6-7	Performance of filters on the Lorenz system.	76
A-1	Schematic for RTDS notional all-electric ship at Florida State University.	81
A-2	Comparison between interpolated and true values for maximum frequency deviation of main generator one and auxiliary generator two for RTDS notional electric ship at Florida State University.	82

Chapter 1

Introduction

Large-scale engineered systems pose a class of design problems for which there is currently no existing comprehensive approach. The space of possible realizations can be immense for such problems due to dimensionality, placing them well outside the computational limits of traditional techniques. The all-electric ship is a representative example of large-scale systems, and its development is of significant interest to the control and simulation communities [9, 23, 35]. Its superiority over current ships includes high efficiency power transfer through the ship, rapid reconfiguration, layout flexibility, and azimuthing thrusters providing enhanced maneuvering and hydrodynamic efficiency; presently, this is only a potential. To find designs and corresponding control strategies that attain this potential, the system must be considered as a whole; from this perspective, it is the size of the problem that defines it.

Very few real systems can be truly described by a mathematical model. Reasons for this include system parameters that can only be known to a certain accuracy, unpredictable interaction with the environment, and dynamics unaccounted for in model equations. Large-scale systems have numerous components and many modes of interaction with the environment, and consequently tend to have many sources of uncertainty.

A standard way to broaden the scope of a model is to represent uncertain aspects of the actual system with random variables or processes, resulting in systems of stochastic differential equations [22, 29]. Each random variable in a model contributes

a dimension to the sample space of the model, which is the space of all possible realizations. Large systems often have high-dimensional sample spaces, and because they are usually analytically intractable, they suffer what is known as 'The Curse of Dimensionality', a term coined by Richard Bellman to describe the computationally difficulties that arise in high-dimensional spaces.

In this work, collocation algorithms for numerically solving stochastic differential equations are explored. Information is extracted from the resulting solutions in the form of statistical moments and probability distribution functions (PDF's), which are respectively computed via numerical integration and interpolation. These tools are applied to electric ship models and unscented Kalman filtering.

This thesis makes three basic contributions. First, a framework for analyzing power system models under uncertainty is developed and applied to three different systems, a notional electric ship model, a pulse power system, and a larger notional electric ship model undergoing a charging event. For each system, sensitivity is inferred from a different perspective. Second, a modification to the the sparse grid, a collocation algorithm, is presented and justified via theoretical bounds and numerical experiments. Third, a dimension-adaptive collocation algorithm is then implemented within the unscented Kalman filter, and improvement is found over other filters on two example systems.

Chapter 2

Uncertainty in Dynamic Systems: Setting the Stage

Uncertainty can be incorporated into dynamical systems such as the electric ship by modeling them with stochastic differential equations. In this chapter, a cursory overview on stochastic differential equations and methods for solving them numerically is given, followed by three examples of dynamical systems with uncertainty, the latter of which is the all-electric ship. From a general perspective, we seek the solution $x : \Omega \times \mathfrak{R}^+ \rightarrow \mathfrak{R}$ to the stochastic differential equation

$$L(t, \omega; x) = f(t, \omega), \quad x_0 = g(\omega), \quad t \in \mathfrak{R}^+, \quad \omega \in \Omega, \quad (2.1)$$

where L is the differential operator and Ω the sample space [45]. Equation (2.1) may contain both random variables and random processes. A random process may be represented by a random variable at each moment in time; this however makes the sample space Ω infinite dimensional, a feature that cannot be directly accommodated computationally. The Karhunen-Loeve expansion [14] offers a means to approximate random processes using a finite number of random variables in a truncated spectral expansion. Unfortunately, accurately representing a random process tends to dramatically increase the dimension of Ω .

For all but the simplest of systems, (2.1) must be solved numerically. There

are multiple approaches, including brute force Monte Carlo simulation, polynomial chaos [42], and collocation, the latter of which we focus on here.

Useful information is generally extracted from a solution $x(t, \omega)$ to (2.1) through integration, i.e. finding mean and variance trajectories, and interpolation, which enables the construction of PDF's.

The expected value of a continuous random variable ω and that of a function of a continuous random variable $g(\omega)$ are given respectively by

$$E[\omega] = \int_{\Omega} \omega p(\omega) d\omega, \quad E[g(\omega)] = \int_{\Omega} g(\omega) p(\omega) d\omega, \quad (2.2)$$

If we view equations in the form of (2.1) as functions of random variables, moments of the trajectories can then be calculated as in (2.2):

$$E[x(t, \omega)] = \int_{\Omega} x(t, \omega) p(\omega) d\omega. \quad (2.3)$$

$x(t, \omega)$ is the solution to an ODE corresponding to a specific realization in Ω . Rarely, $x(t, \omega)$ may be calculated analytically; more often it is obtained through simulation. For large-scale systems, simulations can be computationally very expensive, so it is desirable to employ an efficient discretization of (2.3), i.e. one in which high accuracy is achieved from a minimal number of simulations. In the next three sections examples of dynamical systems with uncertainty are given.

The examples in this chapter illustrate how uncertainty is incorporated into dynamical systems and the information that is gained from their solutions.

2.1 Uncertainty in a First Order System

This example demonstrates simply the calculations carried out on more complicated systems later in this thesis. Consider the first order stochastic ordinary differential equation

$$\dot{x} + R_1 x = 0, \quad x(0) = R_2, \quad (2.4)$$

where R_1 and R_2 are uniformly distributed random variables in the interval $[0\ 1]$. The analytical solution is given by

$$x(t, \mathbf{R}) = R_2 e^{-R_1 t}. \quad (2.5)$$

The mean trajectory can then be computed analytically by integrating (2.5) multiplied by one (because $\mathbf{R}$ is uniform) over the system's random space, $[0\ 1] \times [0\ 1]$:

$$E[x(t, \mathbf{R})] = \int_0^1 \int_0^1 r_2 e^{-r_1 t} dr_1 dr_2 = \frac{1}{2t} (1 - e^{-t}).$$

The variance can be computed similarly, and is

$$\text{var}(x(t, \mathbf{R})) = \int_0^1 \int_0^1 (r_2 e^{-r_1 t} - E[x(t)])^2 dr_1 dr_2 = \frac{1}{6t} (1 - e^{-2t}) - \frac{1}{4t^2} (1 - e^{-t})^2.$$

Notice that because this system has two random parameters, the moment calculations are double integrals.

2.2 Uncertainty in a Double Pendulum

Now consider a double pendulum with uncertainty in the initial displacement of each arm (Fig. 2-1). As in the previous example, this system has a two dimensional random space. However, the double pendulum equations (see [37]) cannot be solved analytically, and so (2.3) must be solved numerically for this system, both in the sense that the solution of the system equations $\mathbf{x}(t, \mathbf{R})$ must be obtained through simulation and that the statistical moment integrals must also be computed by discretizing the random space.

Fig. 2-2 shows the mean trajectories with standard deviation envelopes for this system. The solutions to the system equations were generated using the fourth order Runge-Kutta scheme, and the moments were then computed using techniques described in chapter 3. Notice that at about 2.3 seconds the variance grows very large. This is because the system after that time is sensitive to the prescribed uncertainty,

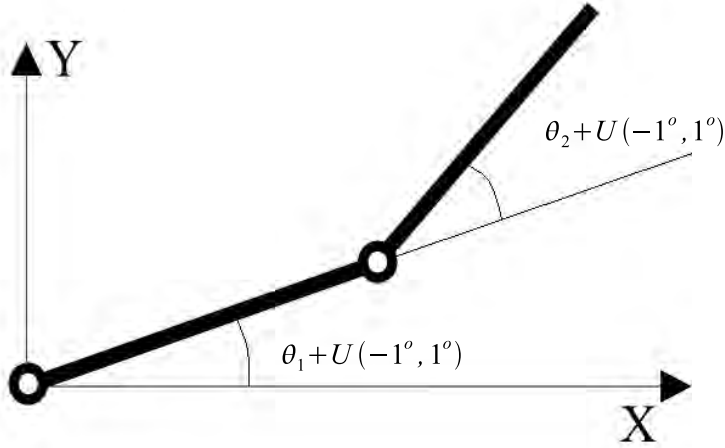


Figure 2-1: Double pendulum with uniform uncertainty in the initial positions.

meaning that it can exhibit a wide range of behavior depending on where in the random space the system is realized. Mathematical definitions of sensitivity will be given in chapter 5, and will be used to quantify the robustness of systems to uncertainty.

2.3 Uncertainty in the All-Electric Ship

Physically, uncertainty in the all-electric ship comes in many forms. Values of physical parameters like capacitances, inertia constants, and flux leakages are only known to manufacturers' specifications, and may change with time and use. There is significant environmental interaction in the form of wind and waves; a large wave can partially or fully expose the propellers, dramatically and suddenly altering the torque load on the induction motors. Lastly, there is a diverse range of operating conditions, with each mode utilizing the ship systems differently. Traveling long distances will place the ship under fairly steady conditions, while in a combat scenario there will be potentially heavy loading from propulsion and radar, irregular pulsed loads from weapon systems, and the ultimate uncertainty, damage. Clearly, modeling the full scope of uncertainty that might be seen by the all-electric ship (or any large-scale system) would take one well outside of the framework presented here.

Simplified analysis is revealing nonetheless, and often simplifying one aspect of

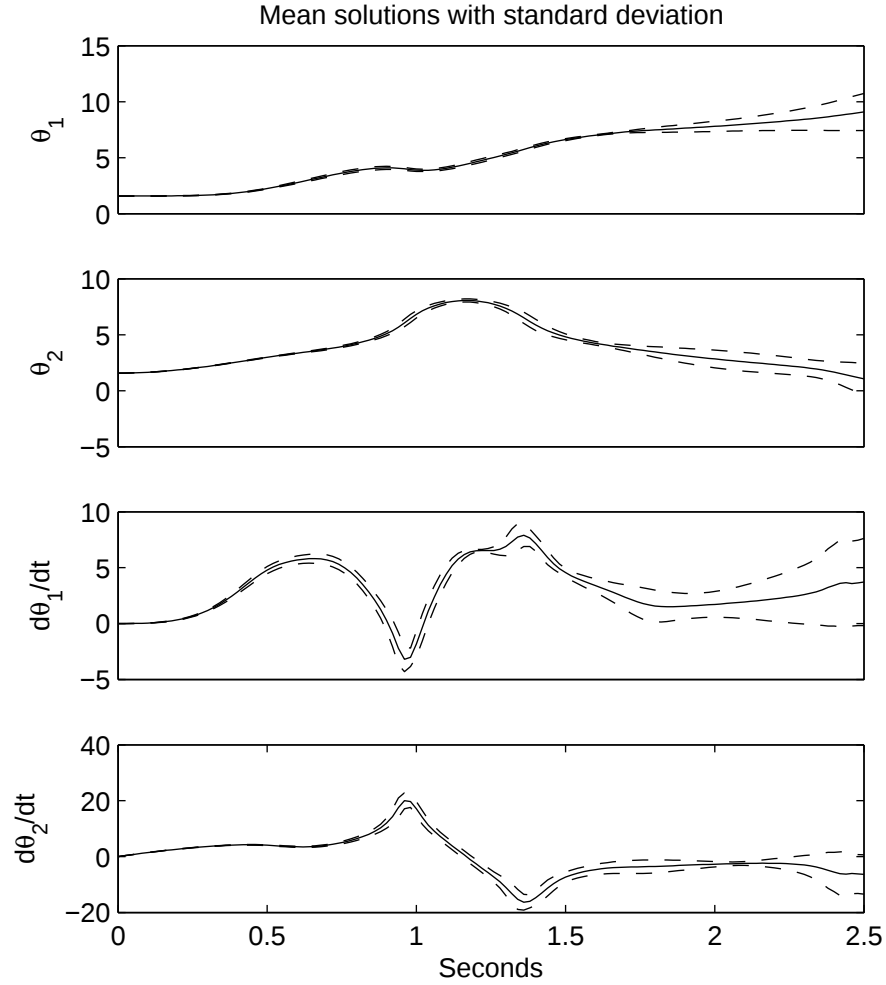


Figure 2-2: Mean solutions with standard deviation envelopes for a double pendulum with uniform initial position uncertainty.

an analysis allows greater depth in another. By restricting the types of uncertainty considered, the size of the system one can study is increased. Plainly, by allowing only parametric and initial condition uncertainty in models of large-scale systems, it becomes possible to study the effect of that uncertainty not on components or portions of the systems, but on the systems as a whole. In chapter 5, such analysis is applied to the all-electric ship.

2.4 Summary

In this chapter, a brief overview of how uncertainty in dynamical systems can be modeled with stochastic differential equations is given. Three illustrative examples were shown.

Chapter 3

Technical Overview

In this chapter, collocation algorithms for solving stochastic differential equations are presented, along with basic elements from which they can be constructed and methods for extracting information from the resulting solutions.

Collocation approximates solutions to (2.1) by calculating solutions to deterministic differential equations corresponding to special points in the random space Ω and then using those solutions to construct a function over Ω at each moment in time. Monte Carlo methods differ only in the choice of points, using pseudo- or quasi-random points rather than collocation points.

At this point, it is sensible to simplify the discussion from stochastic differential equations to functions; a solution to a stochastic differential equation at a fixed time is essentially a function over a random space, and in the case of collocation, finding that solution amounts to approximating a function at each moment in time. Two basic settings in which collocation algorithms are used are numerical integration and interpolation of functions.

3.1 Numerical Integration

Probability says that means, variances, and higher moments of random variables are integrals. These integrals must be computed numerically for large-scale systems, and thus, because of 'The Curse of Dimensionality', become problems of efficiently

sampling a multi-dimensional random space; this is a quintessential aspect of this work.

Numerical integration [6] is essentially a problem of efficiently characterizing a function over a space by sampling it at discrete points. Considered here are algorithms that can be expressed as a sum of function evaluations multiplied by weights, such that

$$\int f(\mathbf{x})d\mathbf{x} \approx \sum_i w_i f(\mathbf{x}_i). \quad (3.1)$$

Later, these algorithms are applied to (2.3) to compute moments of trajectories of dynamical systems.

3.1.1 In One Dimension

Familiar one-dimensional numerical integration schemes (also called rules) are Riemann sums, the trapezoidal rule, and Simpson's rule. These methods produce relatively crude results in comparison with polynomial-based approaches, but have in common that they can all be written in the form of (3.1), and thus, once the weights and function evaluation points (sometimes called nodes) have been computed, are identical in implementation. The set of points and weights comprising the level i rule of a numerical integration scheme will be denoted Φ_i . With this notation, one-dimensional rules can be thought of as sequences. Denoting the exact integral I , (3.1) can be written

$$I[f] \approx \Phi_i[f].$$

The theoretical underpinnings of polynomial methods differ substantially from those of Riemann sums. Polynomials are fit to the integrand, and the resulting polynomial function is then integrated. In general, the nodes are the roots of the fitted polynomials. For this reason, these methods perform exceptionally well on very smooth integrands that can be well approximated by a polynomial, and correspondingly poorly on less smooth, badly behaved functions. Three different polynomial methods are employed here.

Gaussian Quadrature

Gaussian quadrature [6,15,17,36] is perhaps the best known polynomial-based method, famed for its $2n - 1$ degree of polynomial exactness, meaning that with n function evaluations, all integrands which are polynomials of order $2n - 1$ or less will be integrated exactly. Gaussian quadrature utilizes orthogonal polynomials, for example Legendre, Hermite, or Laguerre polynomials. Each set of orthogonal polynomials is orthogonal with respect to a different inner product, the kernel of which is the weighting function of the integral approximated by the associated rule. Legendre polynomials are orthogonal with respect to a constant weighting function:

$$\int_{-1}^1 L_i(x)L_j(x)dx = \begin{cases} 0 & \text{for } i \neq j \\ \frac{2}{2i+1} & \text{for } i = j \end{cases}, \quad \int_a^b f(x)dx \approx \sum_i w_i^L f(x_i^L).$$

Also of importance are the Hermite polynomials, which are orthogonal with respect to the Gaussian weighting function:

$$\int_{-\infty}^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} H_i(x)H_j(x)dx = \begin{cases} 0 & \text{for } i \neq j \\ n!\sqrt{2\pi} & \text{for } i = j \end{cases}, \quad \int_{-\infty}^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} f(x)dx \approx \sum_i w_i^H f(x_i^H).$$

When approximating expectations of functions of random variables (2.2), the choice of orthogonal polynomial should reflect the type of random variable, because a given probability density function will often be identical to some orthogonal polynomial's weighting function. Obviously, Legendre polynomials should be used for uniform random variables, and Hermite polynomials for normal random variables.

Clenshaw-Curtis Quadrature

The nodes of Clenshaw-Curtis [4] quadratures are at the roots of Chebyshev polynomials. In the past, the ease with which they could be computed was considered a virtue of Clenshaw-Curtis quadratures; with modern computing power, this is no longer a significant advantage. Clenshaw-Curtis only achieves n polynomial exactness, but is known to exceed its theoretical performance expectations [41]. A further

advantage is nesting: the set of nodes for each level are contained in the nodes for all successive levels. In section 3.1.2, it will be seen that this is desirable for the sparse and dimension-adaptive grid formulations, where nesting allows function evaluations to be reused.

Gauss-Kronrod-Patterson Quadrature

Gauss-Kronrod-Patterson quadratures [10, 31, 33] are an extension of Gaussian rules first developed by Kronrod and then iterated by Patterson to yield a sequence of nested quadratures. The n_g point Gaussian quadrature base level has a polynomial exactness of $2n_g - 1$, and each successive level with m new points increases the exactness by m . The overall exactness for an n_g point Gaussian quadrature with $\sum m$ Kronrod-Patterson extension points is then $2n_g - 1 + \sum m$.

In section 4.3 we find that, when using Gauss-Kronrod-Patterson quadrature beginning with the three point Gaussian rule, shifting the index by one yields significantly better results than what is achieved with conventional indexing, that is, with the first level corresponding to a single point. We find a similar but smaller improvement when the same shift is applied using Clenshaw-Curtis quadratures.

3.1.2 In Multiple Dimensions

One-dimensional numerical integration schemes are extended to multiple dimensions by tensor products, which, in all implementations considered here, are more plainly Cartesian products. For example, suppose Φ_{10} is the ten point Legendre polynomial Gaussian quadrature. Then $\Phi_{10} \otimes \Phi_{10}$ is the one hundred point two-dimensional integral approximation, and $\Phi_{10} \otimes \Phi_{10} \otimes \Phi_{10}$ is the one thousand point three-dimensional integral approximation. It is through this exponential increase in points necessary to maintain a constant level of discretization with increasing dimension that the Curse of Dimensionality enters numerical integration. This example is a 'full grid' multidimensional scheme. In the coming sections, sparse grids, dimension-adaptivity, and, the alternatives to Cartesian product methods in high dimensions, Monte Carlo and

quasi-Monte Carlo will be discussed.

Full Grids

The d -dimensional, level q full grid collocation integral approximation, which we will denote F_q^d , is the tensor product of d one dimensional, level q quadratures:

$$\begin{aligned} F_q^d[f] &= \Phi_q \otimes \cdots \otimes \Phi_q[f] \\ &= \sum_{i_1=1}^{m_q} \cdots \sum_{i_d=1}^{m_q} w_{i_1, \dots, i_d} f(x_{i_1}, \dots, x_{i_d}) \\ w_{i_1, \dots, i_d} &= w_{i_1} \cdots w_{i_d}, \end{aligned} \tag{3.2}$$

where m_q is the number of points in Φ_q . The full grid formulation is written here as having identical component quadratures in each dimension for concision, but this is not a requirement: any level of any valid rule may be used. This is often advantageous to keep in mind; a function that is poorly behaved in some dimensions and relatively smooth in others can be more efficiently integrated by using high level approximations in the troublesome dimensions and lower level approximations in the others, rather than by using the same high level quadrature in every dimension. The dimension-adaptive introduced in section 3.1.2 does this automatically.

With some information about the function being integrated, the error for full grid collocation with polynomial based component quadratures can be bounded. For functions in C^r (functions with bounded mixed derivatives up to order r), the error of full grid collocation is estimated by

$$|I[f] - F_q^d[f]| = O(n_q^{-r/d}) \tag{3.3}$$

where n_q is the total number of points in the grid. The factor of d in the exponent is a manifestation of 'The Curse of Dimensionality'. As dimension grows, it causes the error order to grow as well at an exponential rate. For low dimensional, smooth integrands, a full grid is often the best choice among non-adaptive methods.

Sparse Grids

In 1963, Smolyak introduced what is now known as Smolyak's formula and is the underlying formulation of all sparse grid methods [12, 21, 28, 34, 38, 43]. Rather than using one high order tensor product, the sparse grid is a sum of lower order Cartesian products, and tends to perform better in slightly higher dimension than full grid approaches. Define the level i difference quadrature

$$\Delta_i = \Phi_i - \Phi_{i-1}, \quad \Delta_1 = \Phi_1.$$

Smolyak's formula can then be written

$$S_q^d[f] = \sum_{|i|_1 < q+d} \Delta_{i_1} \otimes \cdots \otimes \Delta_{i_d}[f]. \quad (3.4)$$

where $|i|_1 = i_1 + \cdots + i_d$.

At this point it becomes clear why nested component quadratures are preferred for sparse grids. Nesting causes points of different grids in the sum to coincide, and the number of common points increases with both the level and dimension of the sparse grid.

The error for a sparse grid with polynomial based component quadratures can be estimated in the same fashion as the full grid case:

$$|I[f] - S_q^d[f]| = O(n_q^{-r} \log(n_q)^{(d-1)(r+1)}). \quad (3.5)$$

As with full grid collocation, the factor d eventually causes sparse grids to fall to 'The Curse' as well, but not as quickly. The penalty incurred by dimension is less severe because of the $\log(n_q)$ term, and consequently sparse grid collocation is often superior in mid-dimensional situations. Fig. 3-1 shows full and sparse grid examples.

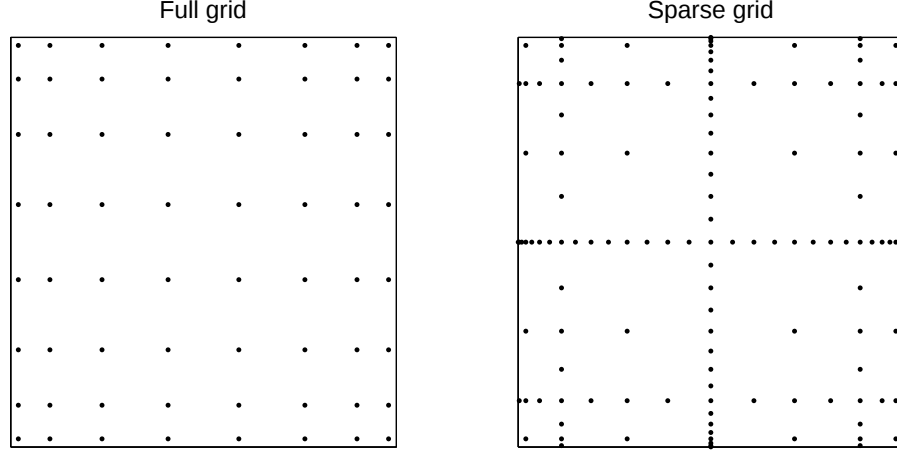


Figure 3-1: Left: Full grid with Legendre polynomial Gaussian quadrature. Right: Sparse grid with Gauss-Kronrod-Patterson quadrature.

Dimension-Adaptivity

Reference [13] presents an adaptive algorithm which places points according to online estimation of its convergence. The basic building block of the dimension-adaptive algorithm is the difference grid Δ defined in the previous section. Let $\mathbf{E}_i[f] = I[f] - \mathbf{Q}_i[f]$ be the error of some quadrature $\mathbf{Q}_i$, be it one or multidimensional. Because the error is converging to zero, $\mathbf{E}_i \gg \mathbf{E}_{i+1}$, and $\mathbf{E}_i \approx \mathbf{E}_i - \mathbf{E}_{i+1}$. Then

$$\mathbf{E}_i \approx \mathbf{E}_i - \mathbf{E}_{i+1} = I - \mathbf{Q}_i - I + \mathbf{Q}_{i+1} = \mathbf{Q}_{i+1} - \mathbf{Q}_i = \Delta_{i+1}.$$

The algorithm builds an approximation from difference grids, and their magnitude, which estimates the convergence of a dimension or combination of dimensions, indicates which difference grids should be evaluated next. For example, suppose that for a two dimensional integrand the evaluated difference grid with the largest magnitude is $\Delta_2 \otimes \Delta_3$. Then, if they are valid difference grids, the next two grids to be evaluated will be $\Delta_3 \otimes \Delta_3$ and $\Delta_2 \otimes \Delta_4$. Fig. 3-2 illustrates the behavior of the algorithm on a test integral from section 4.1. To demonstrate the versatility of the algorithm, the space has been divided into nine identical elements, inside each of which the dimension-adaptive algorithm was executed. Multi-elements [20] is another powerful

approach to numerical integration, but which is not explored here. The function is C^0 along the central axes of the space, and analytic elsewhere. Grids that are of high order in the dimension perpendicular to the C^0 ridges are evaluated, while relatively low order grids are computed where the function is better behaved.

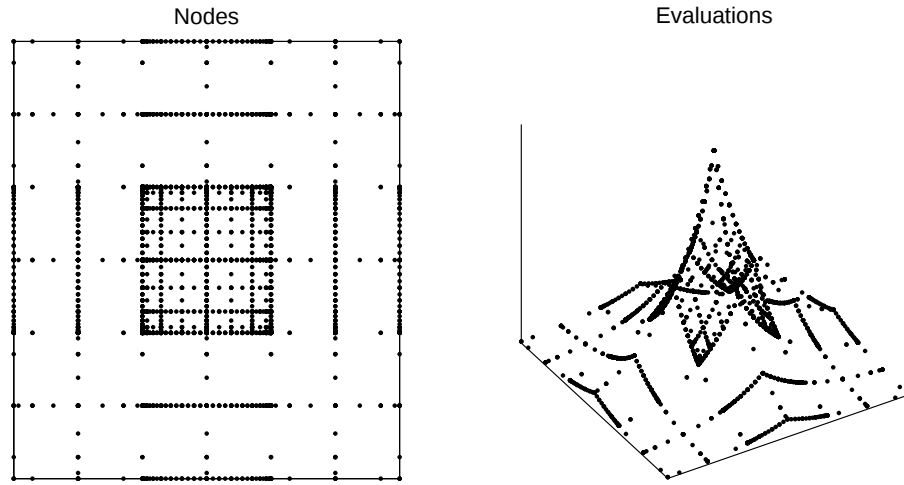


Figure 3-2: Nine element Clenshaw-Curtis dimension-adaptive integration of the two dimensional instantiation of test function five from [11]. The nodes chosen by the algorithm are shown on the left, and the function evaluations at those points on the right.

The algorithm (from [13]) is as follows:

Algorithm:

$DimAdapt(f)$

$\mathbf{i} = [1, \dots, 1]$

$\mathbf{O} = \emptyset$

$\mathbf{A} = \{\mathbf{i}\}$

$r = \Delta_{\mathbf{i}}f$

$\eta = g_{\mathbf{i}}$

while $\eta > tol$

$\mathbf{i}_m = \{\mathbf{i} \in \mathbf{A} : g_{\mathbf{i}} = \max \mathbf{g}\}$

$\mathbf{A} = \mathbf{A} \setminus \mathbf{i}_m$

$\mathbf{O} = \mathbf{O} \cup \mathbf{i}_m$

$\eta = \eta - g_{\mathbf{i}_m}$

for $k = 1 : d$

$\mathbf{j} = \mathbf{i}_m + \mathbf{e}_k$

if $\mathbf{j} - \mathbf{e}_q \in O$ for all $q = 1 : d$

$\mathbf{A} = \mathbf{A} \cup \mathbf{j}$

$r = r + \Delta_{\mathbf{j}}f$

$\eta = \eta + g_{\mathbf{j}}$

end if

end for

end while

return r

Symbols:

$\mathbf{O}$ old index set

$\mathbf{A}$ active index set

$\Delta_{\mathbf{i}}f$ integral increment

$g_{\mathbf{i}}$ local error indicator

η global error estimate

$\mathbf{e}_k$ k^{th} unit vector

tol error tolerance

r computed integral value

Monte Carlo and Quasi-Monte Carlo

For all but trivial functions, Cartesian product based methods will invariably become ineffective in high enough dimension. Although it is not the focus of this work, it is worth discussing the high-dimensional alternative: Monte Carlo methods. Pure

Monte Carlo, or pseudo-randomness, converges according to the central limit theorem at $1/\sqrt{n}$, independently of dimension. Monte Carlo methods also do not depend on the smoothness of the integrand, and for badly behaved functions are sometimes superior even in low dimensions. Quasi-Monte Carlo [3,27,39] methods offer improvement to $1/n$ optimally and $\log(n)^d/n$ in the worst case. None of these are attractive rates, but in high dimensions it is often the best there is. Fig. 3-3 shows pseudo-random and Sobol sequence quasi-random points.

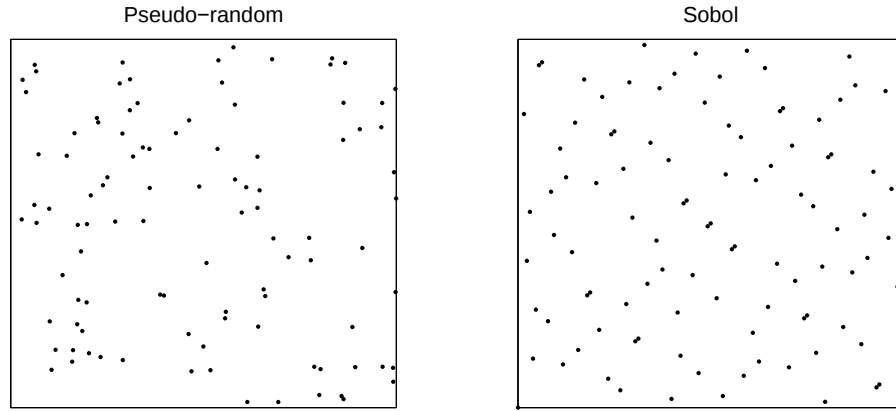


Figure 3-3: Left: 100 pseudo-random points. Right: 100 quasi-random points from the Sobol sequence.

3.2 Interpolation

Interpolation [2,7,21] is closely tied to numerical integration, and often utilizes the same function samples. In fact, with the exception of Monte Carlo, all of the algorithms discussed in this chapter can be used to interpolate as well as integrate functions (Monte Carlo is in a sense a more direct but sometimes less efficient route to what interpolation provides). Essentially, rather than fitting polynomials to a function and integrating the fit, we are now evaluating it.

The one-dimensional barycentric interpolation formula of the second form is prefer-

able for its numerical stability and computationally efficiency. It is given by

$$P_q[f](x) = \frac{\sum_{i=1}^{m_q} \frac{w_i}{x-x_i} f(x_i)}{\sum_{i=1}^{m_q} \frac{w_i}{x-x_i}}, \quad w_i = \prod_{\substack{j=1 \\ j \neq i}}^{m_q} \frac{1}{x_i - x_j}. \quad (3.6)$$

Full, sparse, and dimension-adaptive interpolants formulas are constructed by taking Cartesian products of one dimensional interpolants, as with numerical integration.

The intent is to use interpolation to generate PDF's, i.e. histograms. A direct way to accomplish this without interpolation is with Monte Carlo: evaluate a function or simulate a system at a bunch of random points and sort the outcomes into bins. So why then is interpolation necessary? Rather than direct evaluation, which can be costly, instead apply Monte Carlo to the interpolant, which is always cheap to evaluate, and generate histograms of interpolated function values or system simulations. Again consider the electric ship; one evaluation takes ten minutes, so directly applying Monte Carlo to create a 1,000 point histogram would be time consuming. However, suppose the response can be characterized with fair accuracy using 100 collocation points. The simulation can now be interpolated 1,000 times to produce nearly the same histogram (the similarity increases with the number of collocation points used) in a trivial length of time.

Again consider the double pendulum example of section 2.2. Fig. 3-4 shows interpolated trajectories overlaid on trajectories obtained by direct simulation corresponding to the same realization of the uncertain parameters. Fig. 3-5 shows two sets of histograms generated by the same Monte Carlo points, on the left generated by interpolation and on the right by direct simulation.

There is strong agreement between the simulated and interpolated trajectories; it is only near the end when the variance has grown large that any disparity is noticeable.

The histograms also agree fairly well. All have multiple modes, meaning that each arm can have substantially different positions and velocities depending on where in the random space the system is realized, and the interpolated histograms have captured this feature.

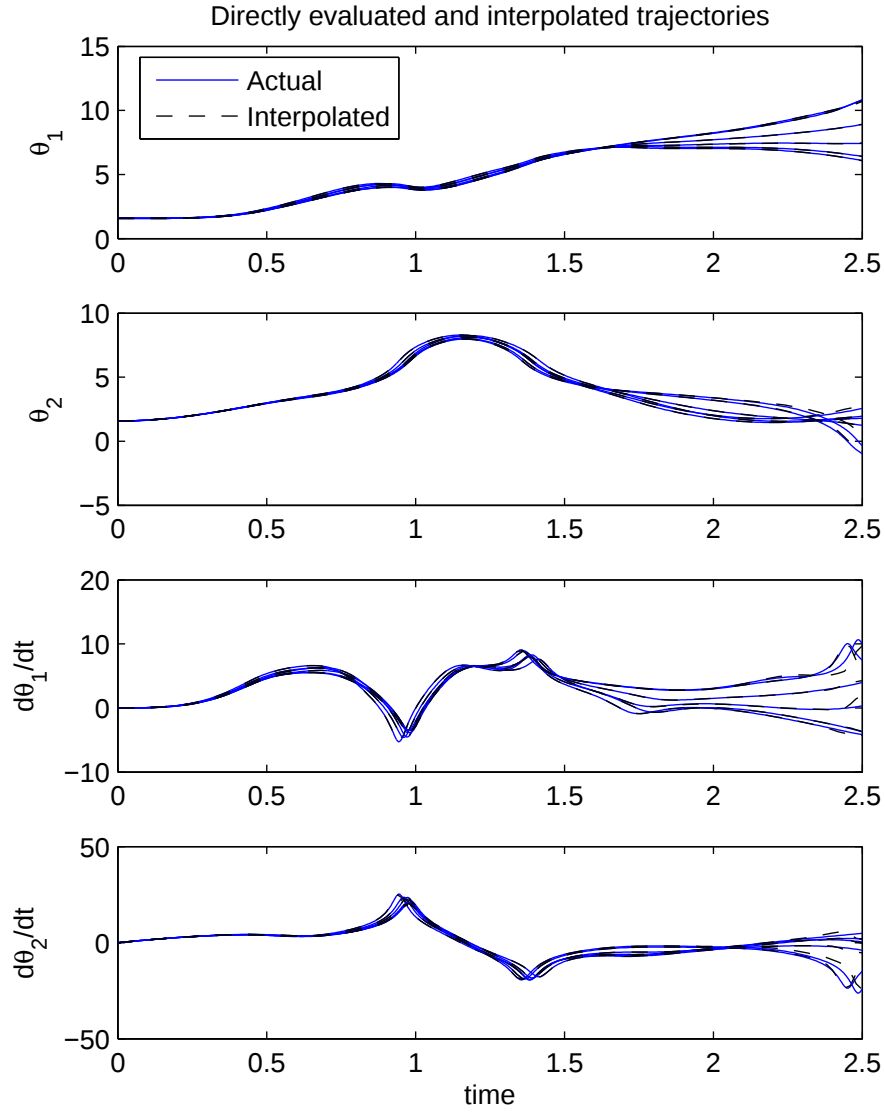


Figure 3-4: Interpolated and directly evaluated trajectories of double pendulum states.

3.3 Summary

An overview of collocation algorithms for numerically solve stochastic differential equations is given, along procedures for extracting statistical moments and PDF's from the resulting solutions. Guidelines for using each algorithm are suggested, specifically that full and sparse grids perform well on smooth functions in low and mid dimensions, and that Monte Carlo or quasi-Monte Carlo will eventually become su-

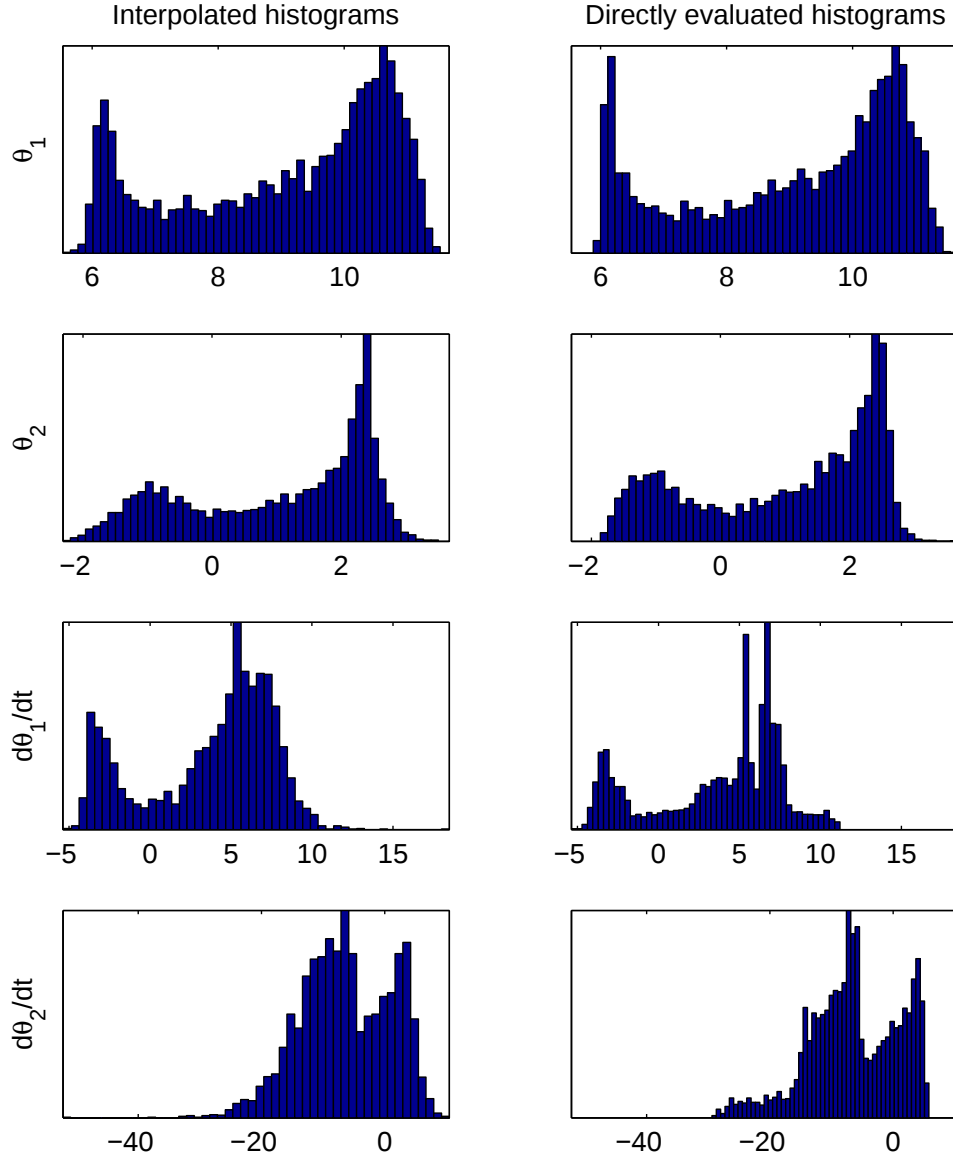


Figure 3-5: Interpolated and directly evaluated histograms of double pendulum states at 2.5 seconds.

prior with increasing dimension, or even in lower dimensions if the function is poorly behaved. The dimension-adaptive algorithm from [13] in many cases will outperform the non-adaptive collocation methods.

Chapter 4

Comparison of Integration Methods

Numerical integration is perhaps the simplest setting in which to compare the performance of the algorithms of the previous chapter. In this chapter, a comparison is given using the scalar test functions from [11]. In section 4.3, numerical results and theoretical justification are presented for a modification to conventional sparse grids yielding improved performance.

4.1 Test functions

The six test functions from [11] are:

OSCILLATORY	$f(\mathbf{x}) = \cos(2\pi u_1 + \sum_{i=1}^d a_i x_i)$
PRODUCT PEAK	$f(\mathbf{x}) = \prod_{i=1}^d (a_i^{-2} + (x_i - u_i)^2)^{-1}$
CORNER PEAK	$f(\mathbf{x}) = (1 + \sum_{i=1}^d a_i x_i)^{-(d+1)}$
GAUSSIAN	$f(\mathbf{x}) = \exp(-\sum_{i=1}^d a_i^2 (x_i - u_i)^2)$
C^0	$f(\mathbf{x}) = \exp(-\sum_{i=1}^d a_i x_i - u_i)$
DISCONTINUOUS	$f(\mathbf{x}) = \begin{cases} 0 & x_1 > u_1 \text{ or } x_2 > u_2 \\ \exp(\sum_{i=1}^d a_i x_i) & \text{otherwise} \end{cases}$

where $\mathbf{a}$ and $\mathbf{u}$ are tunable parameters which respectively determine the difficulty of the functions to integrate and the location of their features in the region of integration. Plots of the two dimensional instantiation of each function are shown in Fig. 4.1. In

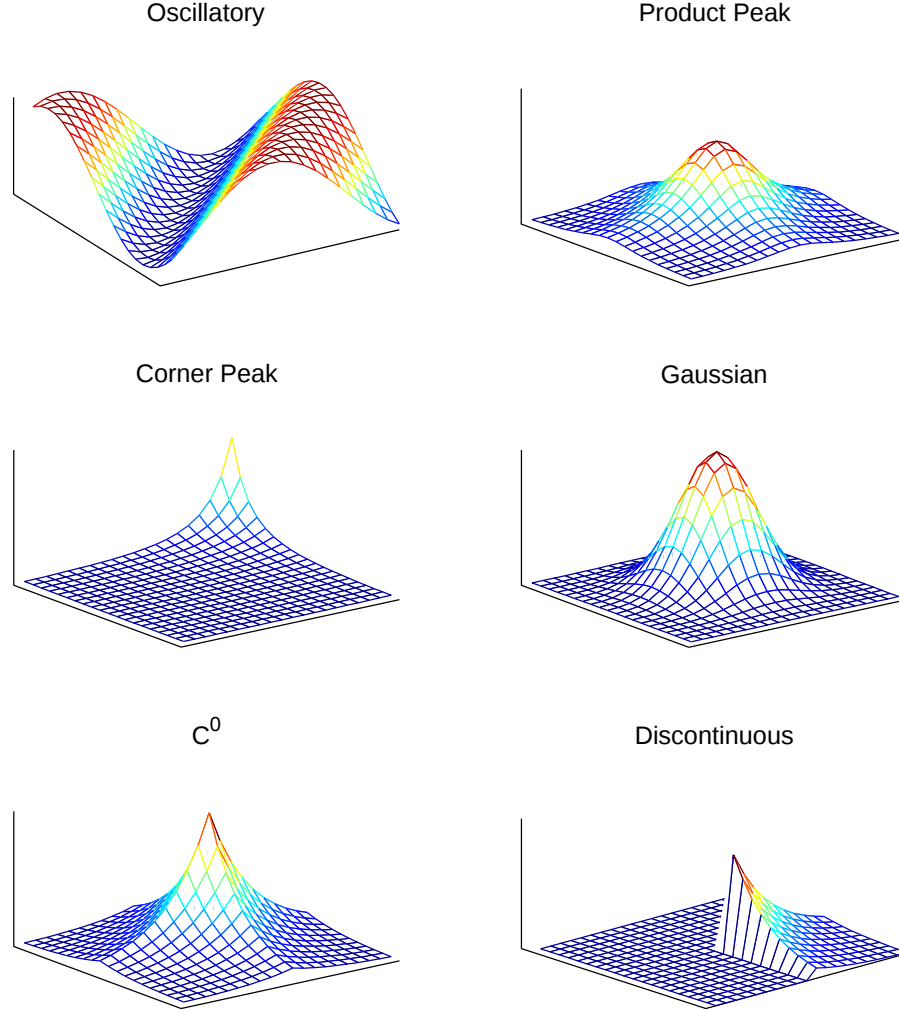


Figure 4-1: Two dimensional test functions

the numerical experiments of the next section, $\mathbf{a}$ was chosen to make each integral roughly the same difficulty, and $\mathbf{u}$ was generated randomly over a set of trials.

4.2 Convergence of Methods

The theoretical error estimates from section 3.1.2 say that full grids will perform best on smooth integrands in low dimensions, sparse grids on smooth integrals in mid-dimensional ranges, and Monte Carlo or quasi-Monte Carlo on high dimensional integrands and even non-smooth integrands in low dimensions. Note that the comparison given here is summary, but adequate; it is not necessary to show the performance of every combination of one dimensional quadrature and multidimensional algorithm on each test function in a full range of dimensions to capture the essence of when each method is appropriate.

4.2.1 Non-adaptive methods

Fig. 4.2.1 shows convergence plots comparing full grid with Legendre polynomial Gaussian Quadrature, sparse grid with Clenshaw-Curtis, Monte Carlo, and quasi-Monte Carlo on the oscillatory, product peak, and corner peak test functions. Fig. 4.2.1 shows the same for the Gaussian, C^0 , and discontinuous test functions.

The oscillatory function is analytic, and as would be expected the collocation algorithms are superior to Monte Carlo and quasi-Monte Carlo even into ten dimensions, although the performance gap is decreasing. The full grid is superior in two dimensions, roughly equivalent to the sparse grid in six dimensions, and inferior in ten, also what is expected from the theoretical bounds.

The product peak, corner peak, and Gaussian functions have similar stories: full grids are best in low dimensions, sparse grids in mid dimensions, and Monte Carlo and quasi-Monte Carlo, if not the favorite by dimension ten, soon will be. Notice the corner peak plot for ten dimensions: Monte Carlo is dramatically more efficient than quasi-Monte Carlo. Quasi-Monte Carlo has $1/N$ convergence in the optimal case, but depends on dimension in the worst case, as is clearly happening here.

The continuous and discontinuous functions are considerably more difficult than the rest. Both have very low smoothnesses, and are therefore difficult to fit polynomials to. Monte Carlo and quasi-Monte Carlo are arguably for the continuous

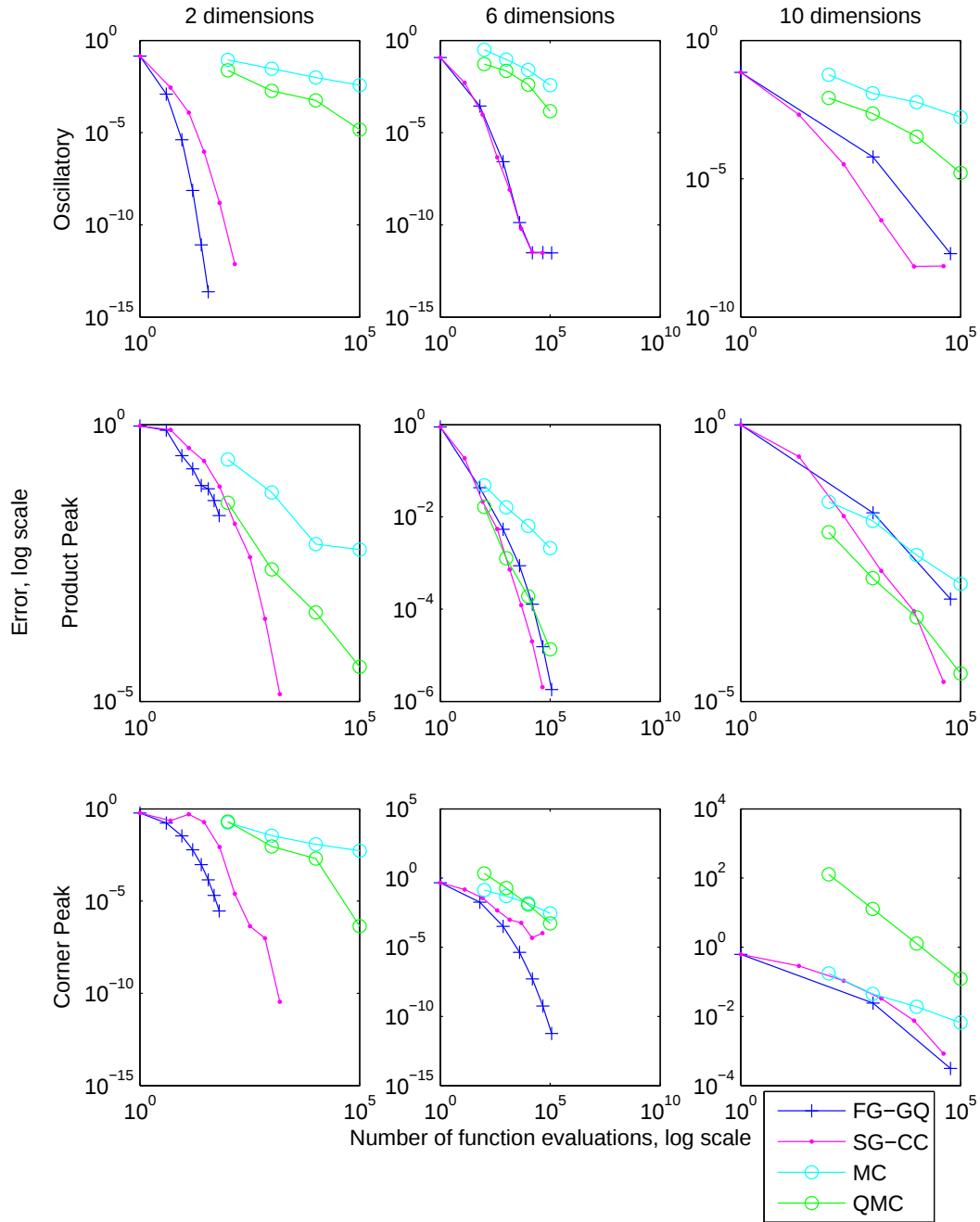


Figure 4-2: Log-log plots of error versus number of function evaluations for product peak, oscillatory, and corner peak test functions. Legend entries correspond to: FG-GQ - Full grid with Gaussian Quadrature, SG-CC - Sparse grid with Clenshaw-Curtis, MC - Monte Carlo, QMC - Quasi-Monte Carlo.

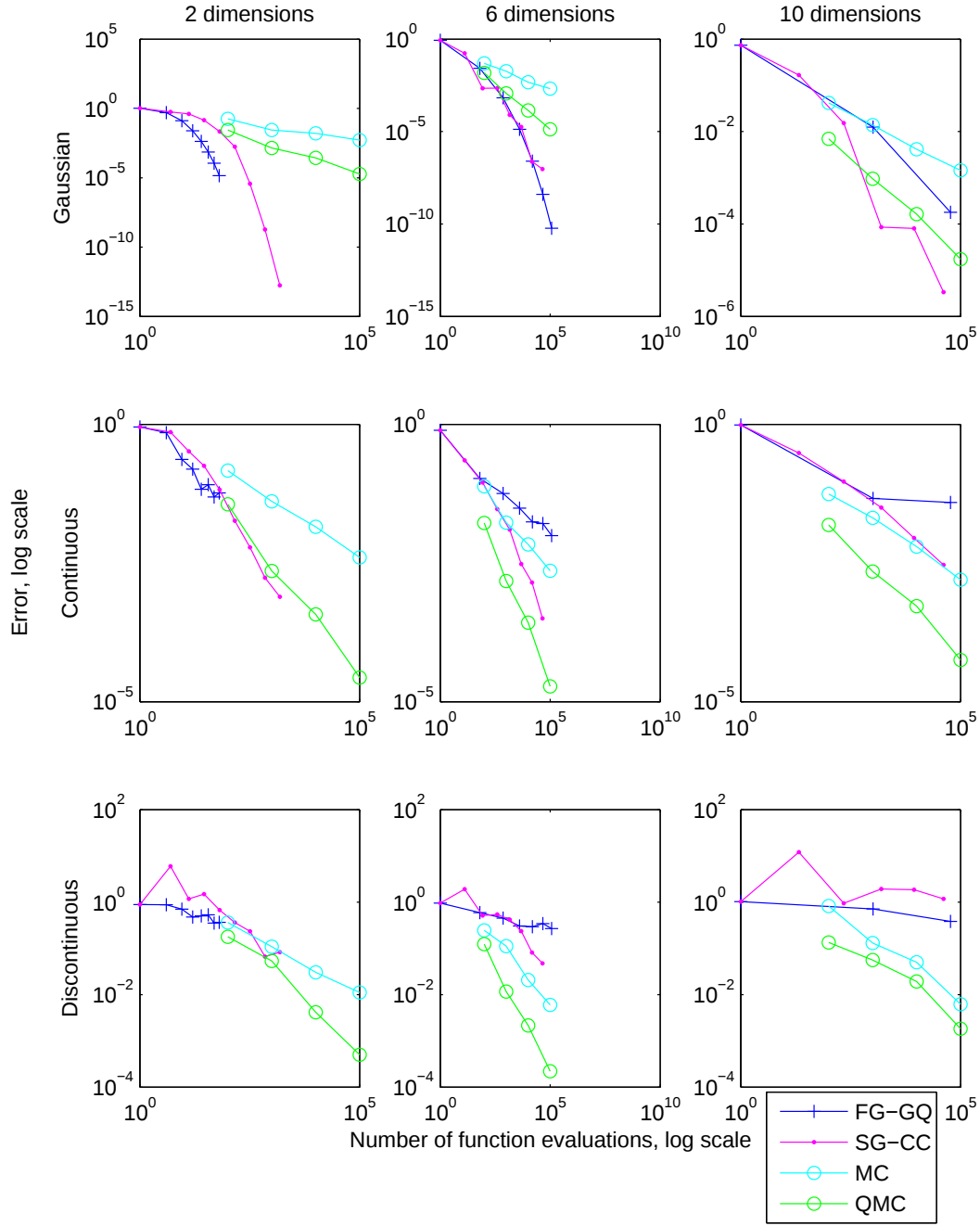


Figure 4-3: Log-log plots of error versus number of function evaluations for Gaussian, C^0 , and discontinuous test functions.

function and clearly for the discontinuous function the best choice even as low as two dimensions.

4.2.2 Dimension-adaptivity

Figures 4.2.2 and 4.2.2 compare full and sparse grids with the dimension-adaptive algorithm discussed in section 3.1.2, [13]. The dimension-adaptive algorithm is as good as or superior to the best of the other methods in nearly all cases, with the adaptivity 'running astray' only on the discontinuous function, for which a Monte Carlo would be superior.

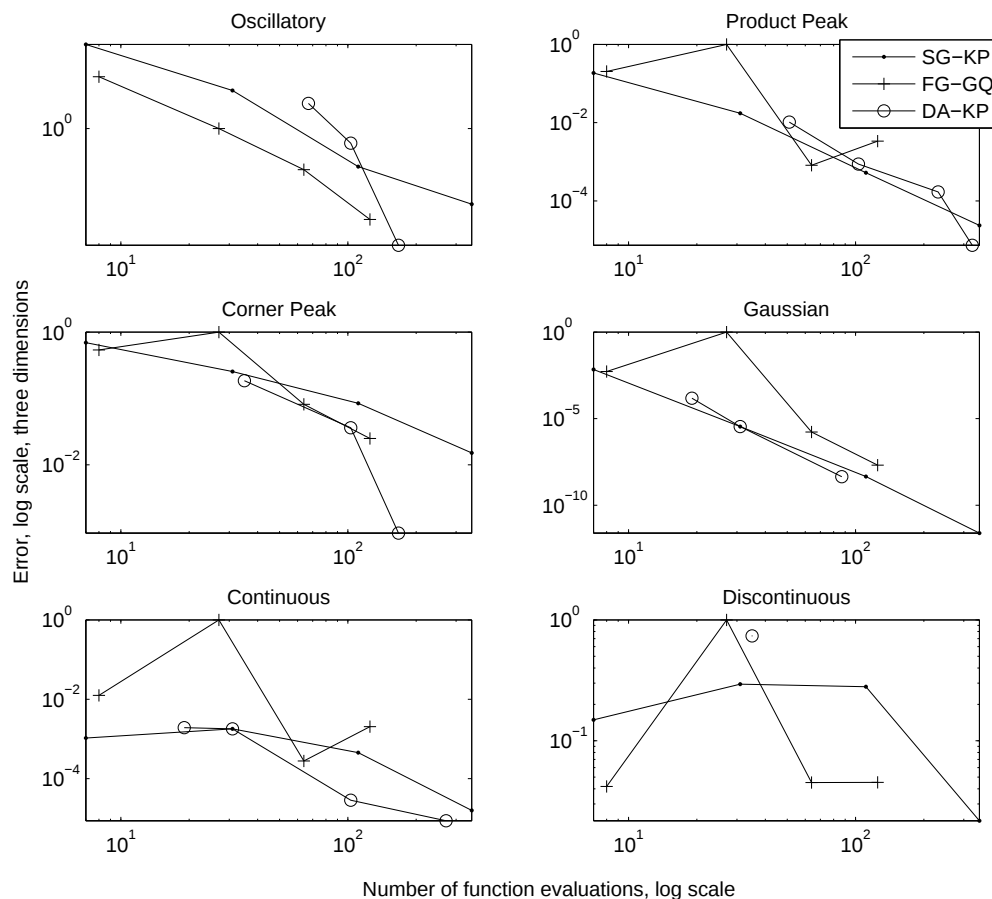


Figure 4-4: Log-log plots of error versus number of function evaluations. Sparse grid with Gauss-Kronrod-Patterson (SG-KP), full grid with Gaussian Quadrature (FG-GQ), and dimension-adaptivity with Gauss-Kronrod-Patterson (DA-KP) in three dimensions are shown.

The performance of each method varies significantly across the different test functions. No one method is universally superior, but prior knowledge of the integrand,

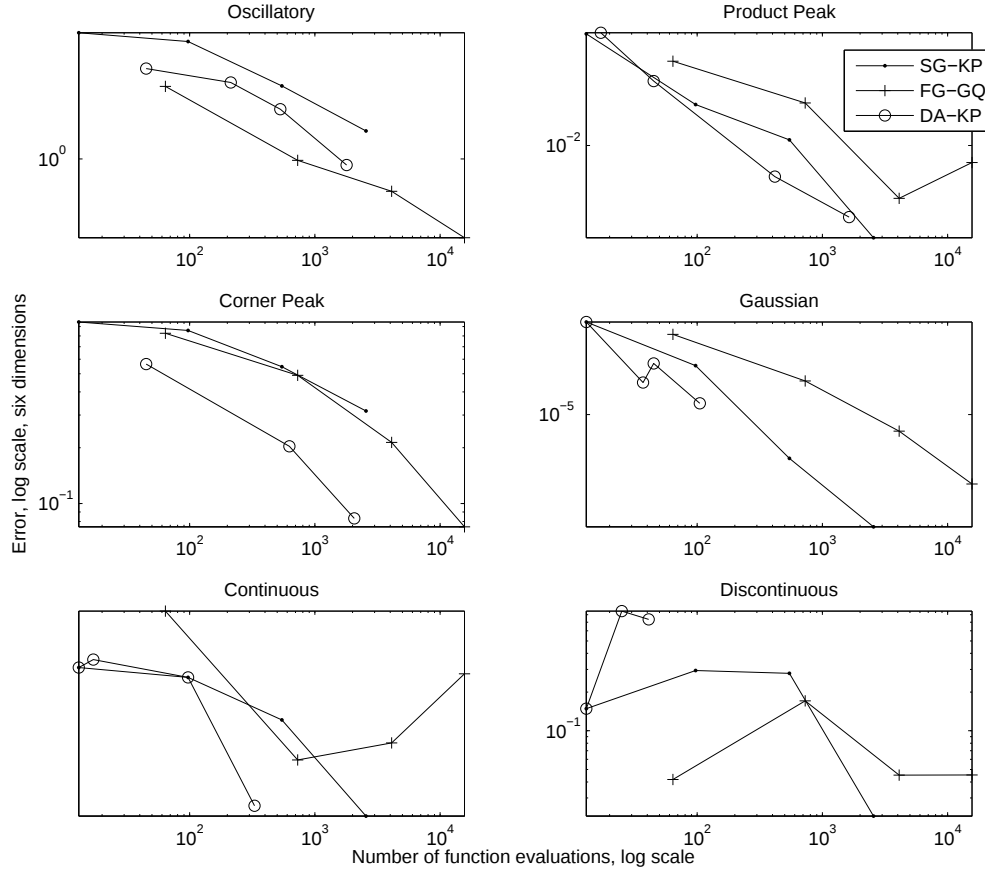


Figure 4-5: Log-log plots of error versus number of function evaluations for algorithms in six dimensions are shown.

such as smoothness and dimension, can provide information that is highly relevant to choosing an approach. It is observed that among the non-adaptive methods, full grids are superior in low dimensions, sparse grids in mid dimensions, and Monte Carlo and quasi-Monte Carlo in high dimensions or for non-smooth integrands. The dimension-adaptive algorithm in most cases outperforms the other grid methods, but should be used cautiously, particularly on badly behaved functions.

4.3 Sparse Grids with Shifted Indexing

A slight modification to the indexing of a sparse grid yields significant gains in convergence in mid dimensional ranges, particularly for smoother functions. Over the next few sections, the modification is described, then theoretically and computationally bounded, and lastly results from numerical experiments are shown.

4.3.1 One-Dimensional Quadratures and Shifting

When using Kronrod-Patterson quadrature beginning with the three point Gaussian rule, shifting the index by one yields significantly better results than what is achieved with conventional indexing, that is, with the first level corresponding to only one point. A similar but smaller improvement is found when the same shift is applied using Clenshaw-Curtis quadratures. A sparse grid with indices shifted α times will be denoted $S_{q,\alpha}^d$ and the corresponding one-dimensional basis sequence by Φ^α .

4.3.2 Error Bounds

Effect of the Shift on Classical Bounds

The error of the shifted grid can be bounded in the same fashion as a conventionally indexed sparse grid, by allowing the shift to propagate through to the final estimate. Following the proof of [43] (see also [28]), the derivation of an error bound for the d -dimensional, once-shifted sparse grid $S_{q,1}^d$ is sketched. Begin by estimating the error of a one-dimensional quadrature as

$$\|I_1 - \Phi_i\| \leq c_r \cdot 2^{-ri}, \quad i \geq 0 \quad (4.1)$$

where $\|\cdot\| = \max |\cdot|$, $\Phi_0[f] \equiv 0$, $\|I_1\| = 2$, and f has bounded mixed derivatives up to order r . The result of removing the lowest one-dimensional quadrature is a shifting of indices; the new error estimate is

$$\|I_1 - \Phi_i^1\| \leq c_r \cdot 2^{-r(i+1)} \quad i \geq 0. \quad (4.2)$$

The magnitudes of the one-dimensional once-shifted difference grids are bounded by

$$\|\Delta_{i_k}^1\| \leq \|I_1 - \Phi_{i_k}^1\| + \|I_1 - \Phi_{i_k-1}^1\| \leq c_r \cdot 2^{-r(i_k+1)} \cdot (1 + 2^r). \quad (4.3)$$

Making use of the property $\|A \otimes B\| = \|A\| \cdot \|B\|$, we have

$$\begin{aligned} \|I_{d+1} - S_{q+1,1}^{d+1}\| &\leq \sum_{|i|_1 \leq q} \|\Delta_{i_1}^1\| \cdot \dots \cdot \|\Delta_{i_d}^1\| \cdot \\ &\quad \|I_1 - \Phi_{q+1-|i|}^1\| + 2 \cdot \|I_d - S_{q,1}^d\|, \text{ so that} \end{aligned} \quad (4.4)$$

$$\sum_{|i|_1 \leq q} \|\Delta_{i_1}^1\| \cdot \dots \cdot \|\Delta_{i_d}^1\| \cdot \|I_1 - \Phi_{q+1-|i|}^1\| \leq \binom{q}{d} \cdot c_r^{d+1} \cdot (1 + 2^r)^d \cdot 2^{-r(q+d+2)}.$$

The result analogous to that in [43] is given by

$$\|I_d[f] - S_{q,1}^d\| \leq c_r \cdot H_r^{d-1} \cdot \binom{q}{d-1} \cdot 2^{-r(q+d)},$$

where $H_r = \max(2^{r+1}, c_r \cdot (1 + 2^r))$. This yields

$$\|I_d[f] - S_{q,1}^d\| = O(q^{d-1} \cdot 2^{-r(q+d)}), \quad (4.5)$$

which is very similar to Smolyak's original estimate:

$$\|I_d[f] - S_q^d\| = O(q^{d-1} \cdot 2^{-rq}). \quad (4.6)$$

The new bound differs by a factor of 2^{-rd} from the original, which suggests an advantage increasing with dimension. If the one-dimensional quadrature indexing is shifted by α rather than 1, a substitution in the above bound gives

$$\|I_d[f] - S_{q,\alpha}^d\| = O(q^{d-1} \cdot 2^{-r(q+\alpha d)}).$$

Next, a point-based bound is obtained for the once-shifted case, by estimating the total number of points in the sparse grid and substituting for q . Assuming the number of points in each one-dimensional quadrature $m_i \leq 2^i$, the total number of points n is approximated by

$$n \leq 2^{q+d} \cdot \binom{q-1}{d-1} = O(q^{d-1} \cdot 2^{q+d}).$$

Substituting this into the level bound above gives

$$\|I_d[f] - S_{q,1}^d\| = O\left(q^{d-1} \cdot \left(\frac{n}{q^{d-1}}\right)^{-r}\right).$$

The inequality $2^{q+d} \leq c_d n$, $q \leq \log(c_d n) - d$ is then substituted to arrive at:

$$\|I_d[f] - S_{q,1}^d\| = O(n^{-r} \cdot (\log(n) - d)^{(d-1)(r+1)}).$$

The point-based bound corresponding to (4.6), the non-shifted estimate, is

$$\|I_d[f] - S_q^d\| = O(n^{-r} \cdot (\log(n))^{(d-1)(r+1)}),$$

which again differs by a term containing d , although here it is in the base instead of the exponent.

The estimates in (4.1) and (4.2) are very crude, and as a result the bound in (4.5) fails to thoroughly capture the behavior of the shifted scheme seen in numerical experiments, namely that there is an improvement in intermediate but not high dimensions. In the next two sections, an exact calculation for the number of points and a more precise formulation of one-dimensional quadrature errors is given, and used them with (4.3) and (4.4) to arrive at a more accurate description of the shifted sparse grid's performance.

The Number of Points

The exact number of points in the nested sparse grid $S_{q,\alpha}^d$ can be directly computed by Theorem 1 in [34]. Define δ_i as the number of points used in Φ_i^α and not Φ_{i-1}^α and set $k = q - d$. The total number of points is then given by

$$n(d+k, d) = \delta_0^d \sum_{j=0}^k c_{j,k} \binom{d}{j}$$

where $\binom{d}{j} \equiv 0$ if $j > d$, and c is defined by

$$c_{j,k} = \sum_{v=1}^{q-j+1} \frac{\delta_v}{\delta_0} c_{j-1,k-v}, \quad j = 1, \dots, k,$$

$$c_{0,k} = 1.$$

One-Dimensional Bounds through Peano's Theorem

The Peano kernel [6] of a quadrature is defined as

$$K_p(t) = \frac{1}{p!} \cdot E_i[(x-t)_+^p],$$

$$(x-t)_+ = \begin{cases} x-t & x \geq t \\ 0 & x < t \end{cases}$$

where p is the polynomial exactness and E_i the error of the quadrature Φ_i ($E_i[g] = 0$ for all polynomials g of order less than or equal to p). The quadrature error can be written as

$$E_i[f] = I[f] - \Phi_i[f] = \int K_p(t) f^{(p+1)}(t) dt.$$

for all f in C^{p+1} . The error can then be approximated by either of

$$E_i[f] \leq \|K_p\| \int |f^{(p+1)}(x)| dx \quad \text{or} \quad E_i[f] \leq \|f^{(p+1)}(t)\| \int |K_p| dx$$

where $\|\cdot\|$ denotes the sup norm. The magnitude of the difference grid (4.3) is estimated with the second inequality above by fixing $\|f^{(p+1)}(x)\|$ at one for all p . This constraint on the function derivatives is a serious simplification, but can be relaxed for cases where some properties of the function are known.

4.3.3 Numerical Results

As shown below, in the non-shifted case Kronrod-Patterson generally performs marginally better than Clenshaw-Curtis based sparse grids. Kronrod-Patterson's polynomial exactness is only slightly superior to that of Clenshaw-Curtis, and furthermore, Clenshaw-Curtis has been known to exceed its theoretical expectations [41]. Differences in polynomial exactness are mostly due to the three-point levels of each sequence: the three point Kronrod-Patterson rule has Gaussian quadrature's polynomial exactness of five, while Clenshaw Curtis only has a polynomial exactness of two.

Computed Error Bounds

Now, using (4.4) and the results of the sections 4.3.2 and 4.3.2, bounds for the conventional and shifted grids can be realistically computed and compared (Fig. 4-6). Firstly, observe that the rate of convergence of both shifted grids is superior in all dimensions, and, according to the crude estimate (4.5), further distancing itself from the non-shifted grid as the dimensions grows. In the Kronrod-Patterson plots for a single shift, a "knee" occurs in the non-shifted curves, after which a similar slope is achieved. Assessment of performance for the two rules with and without the shift is best summarized by noting where the shifted and non-shifted curves cross, i.e. where the accuracy per effort is identical. By this criteria, it is evident that in higher dimensions a desired accuracy may be attainable with a non-shifted grid before the shifted grid has become effective. The Kronrod-Patterson case, however, shows that the benefits of the shift can be realized at much more relaxed tolerances.

Another point worth considering is the performance of sparse grids shifted more

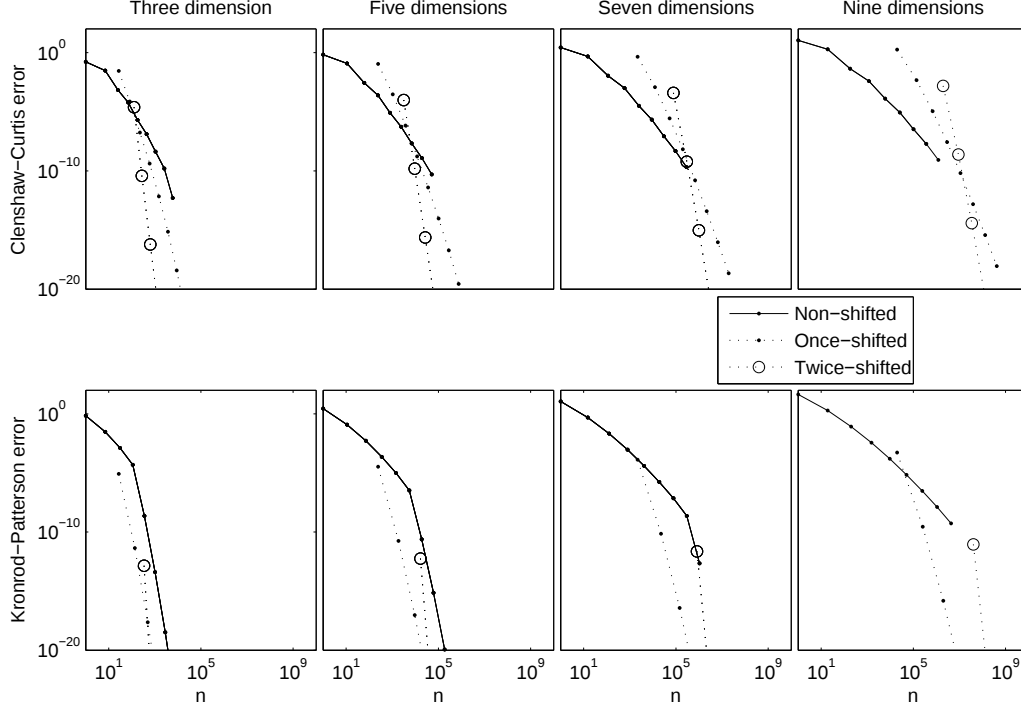


Figure 4-6: Theoretical error convergence of conventional, once-shifted, and twice-shifted sparse grids with Clenshaw-Curtis and Kronrod-Patterson basis quadratures.

than once. Included in Fig. 4-6 are theoretical error curves for twice-shifted sparse grids. The rates of convergence are even higher, and for Clenshaw-Curtis the crossing with the non-shifted curve occurs at approximately the same location as that of the once-shifted curve. However, the dramatic increase in points associated with each shift can make higher order shifted grids unwieldy even in moderate dimensions, and we do not have numerical experiments to verify these predictions.

Numerical Examples

As in section 4.2, the performance of sparse grids with shifted indexing is evaluated on test integrals from [11], specifically the oscillatory, product peak, corner peak, and Gaussian functions (the smooth ones).

Fig. 4-7 shows the performance of regular and shifted sparse grids. For Kronrod-Patterson based grids, significant improvement is seen for the shown test integrals, with the performance improvement due to the shift most noteworthy at dimension

four. As predicted, only a minor improvement is evident with the shift only in low dimensions when using Clenshaw-Curtis; at high dimensions, shifting is clearly unsuitable.

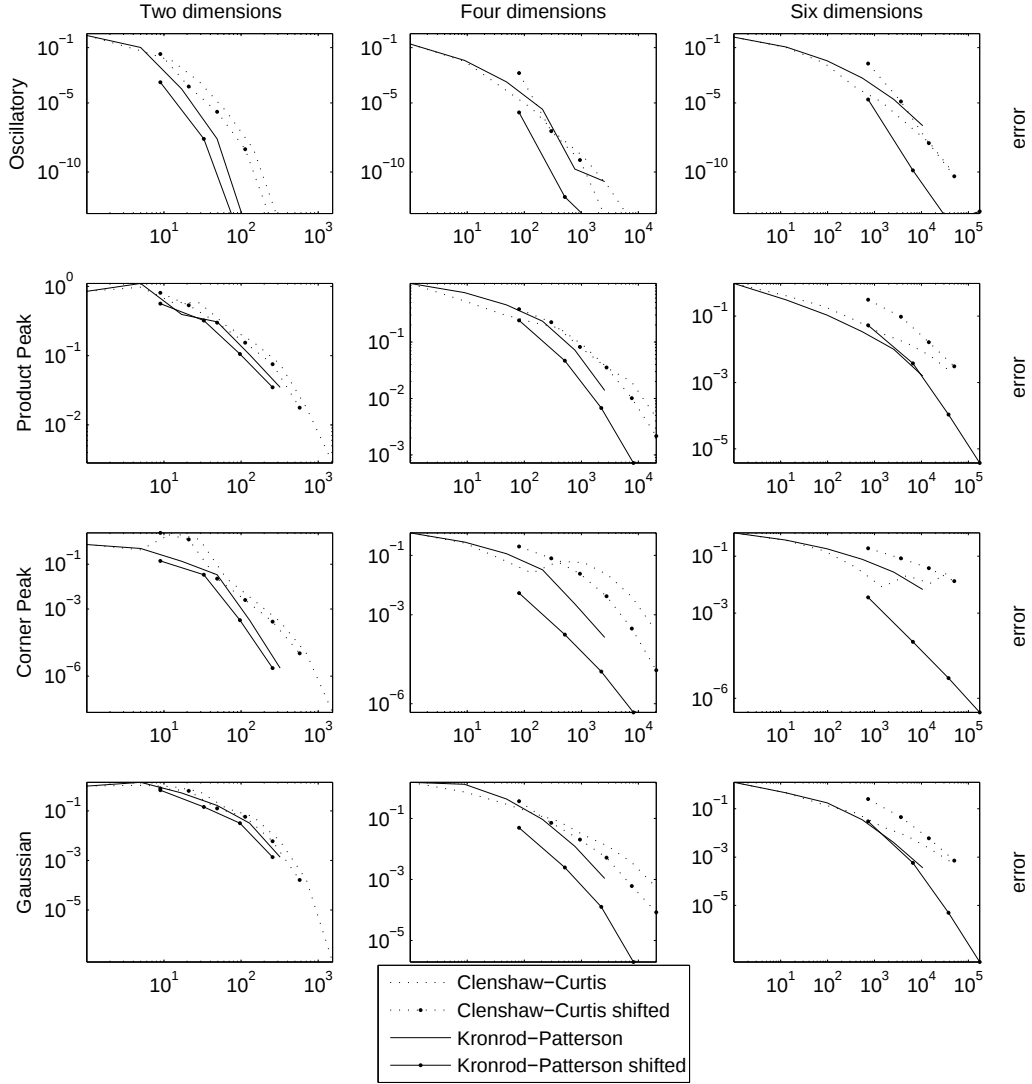


Figure 4-7: Error versus function evaluations for regular and shifted sparse grids using Clenshaw-Curtis and Kronrod-Patterson quadratures

4.3.4 Discussion

It has been shown that for smooth functions, shifting the indexing of a sparse grid yields improved accuracy which sometimes outweighs the associated increase in points.

The modified scheme is more effective with Kronrod-Patterson than with Clenshaw-Curtis sequences, a difference which is suggested by the polynomial exactnesses of the three point levels of each sequence. Intuitively, shifting succeeds because high fidelity on one hyperplane is traded for evenly distributed high fidelity on three hyperplanes. Hence, one would expect inferior performance for functions with weak dimensional coupling.

4.4 Summary

The algorithms of the previous chapter are compared on six standard test functions, and expectations for the performance of each algorithm were confirmed. A modification to the conventional sparse grid algorithm was presented and justified via theoretical bounds and numerical examples.

Chapter 5

Stochastic Simulation of the All-Electric Ship

In this chapter, collocation algorithms are applied to three large-scale models, two describing an electric ship and the other a pulse power system. Sensitivity is inferred for each system to demonstrate how information from stochastic simulation can be interpreted.

5.1 ONR IPS Testbed

5.1.1 Deterministic Simulation

We simulate the Office of Naval Research Integrated Power System testbed (ONR IPS) [23,32] (Fig. 5-1), varying parameters from the port and starboard AC systems, which contains the starboard generator, bus, and propulsion.

First, we examine what happens in the model through deterministic simulation. In zone three, a constant power load, a proportional-integral controller, and two ship service converter modules are turned on, all at time zero. The zone three port voltage, our representative voltage for zone three, rises from the start until about 0.15 seconds when it reaches 400 volts. At time 0.1 seconds, the starboard propulsion current becomes non zero - this is when the motor and motor control turn on. At

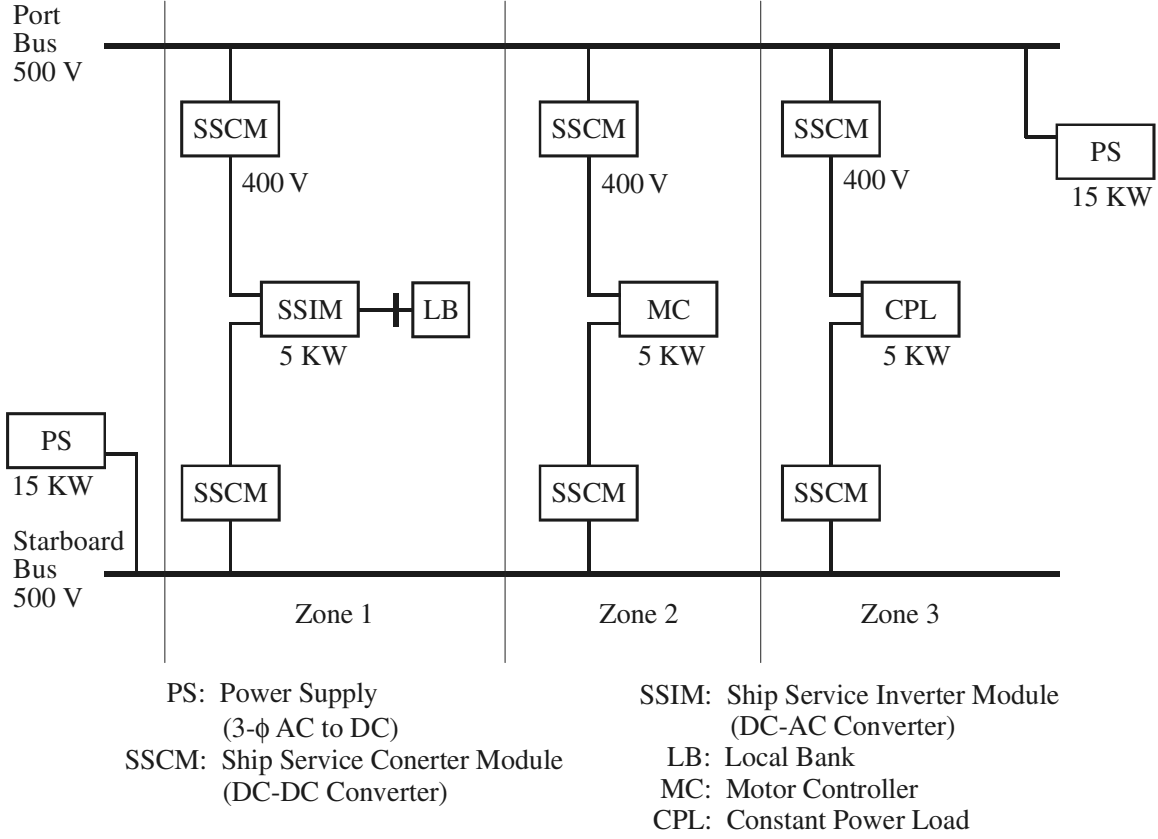


Figure 5-1: Schematic for ONR IPS Testbed.

0.4 seconds, the system is loaded from the ship service inverter module in zone one, and the power supply from the starboard AC system increases correspondingly (Fig. 5-2).

5.1.2 Probabilistic Simulation

In addition to determining mean behaviors, multidimensional techniques can be employed for sensitivity analysis, allowing for identification of not only individual sensitive parameters, but also sensitive combinations of parameters. In our simulations, we use both the integral (IS) and maximum (MS) over time of the variance solution as a sensitivity metric.

$$IS = \int var(x(t))dt, MS = \max_t\{var(x(t))\} \quad (5.1)$$

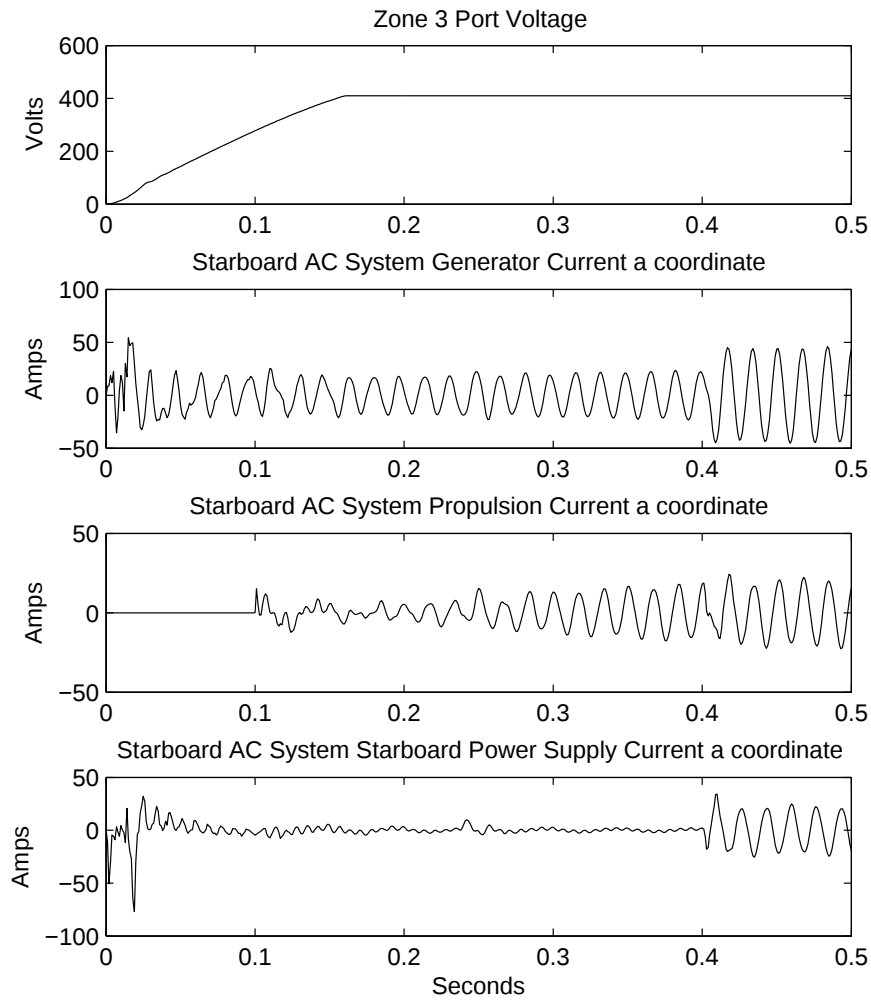


Figure 5-2: Deterministic trajectories for ONR IPS.

The analysis in the next section was done using the integral metric.

Non-Adaptive Simulation

We analyze results from a three dimensional, level four full grid collocation simulation varying each uncertain parameter $\pm 1\%$ from its mean. We considered all triples drawn from the set:

- Bus
 - 1, 7) C - Shunt Capacitance

- 2, 8) C_f - Filter Capacitance
- 3, 9) L_f - Filter Inductance
- Propulsion
 - 4, 10) C - Filter Capacitance
 - 5, 11) L - Filter Inductance
 - 6, 12) Induction Motor Mechanical Load

where parameters 1-6 are starboard and 7-12 port. Fig. 5-3 shows time solutions for the triple $\{1, 2, 3\}$.

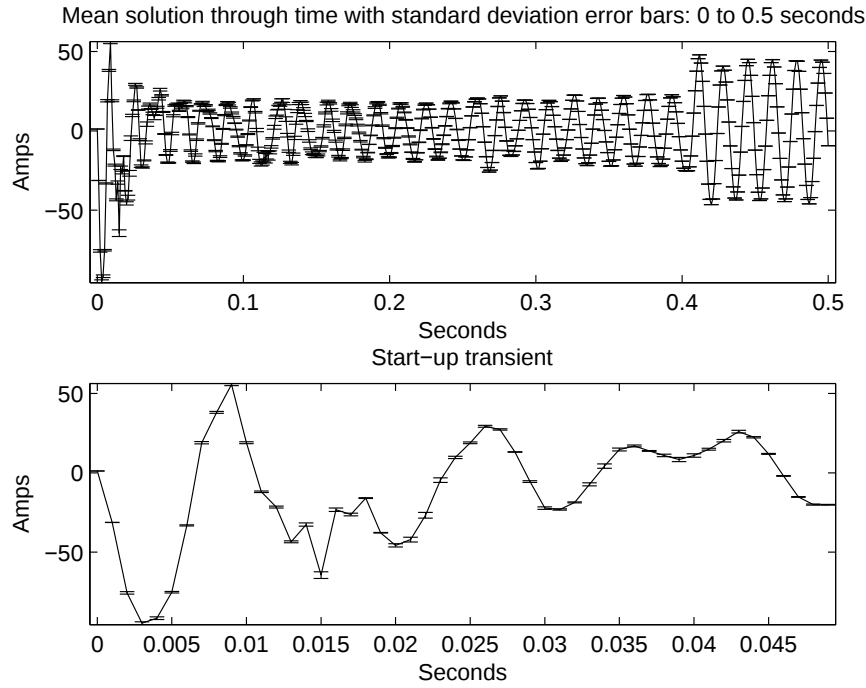


Figure 5-3: Starboard AC system generator abc -reference frame c current mean (curve) and standard deviation (error bars) solutions for the triple containing starboard bus shunt capacitance, filter capacitance and filter inductance for ONR IPS.

There are many ways to structure results from this sort of simulation, for example the most sensitive triple for a specific output: we find that triple $\{1, 2, 3\}$ is in fact the most sensitive triple for the starboard generator currents according to the integrated variance metric, and $\{1, 2, 5\}$ by the maximum metric. We present results that lend themselves easily to graphical analysis, but they are not the only perspectives.

For simulations in n dimensions, the interaction of pairs of uncertain parameters can be viewed by considering all n -tuples containing the same $n - 2$ parameters. Fig. 5-4 shows all triples containing the starboard bus filter inductance (3). It can be seen that the most influential pair on this plot is $\{1, 2\}$, which corresponds to what we previously identified as the most sensitive triple.

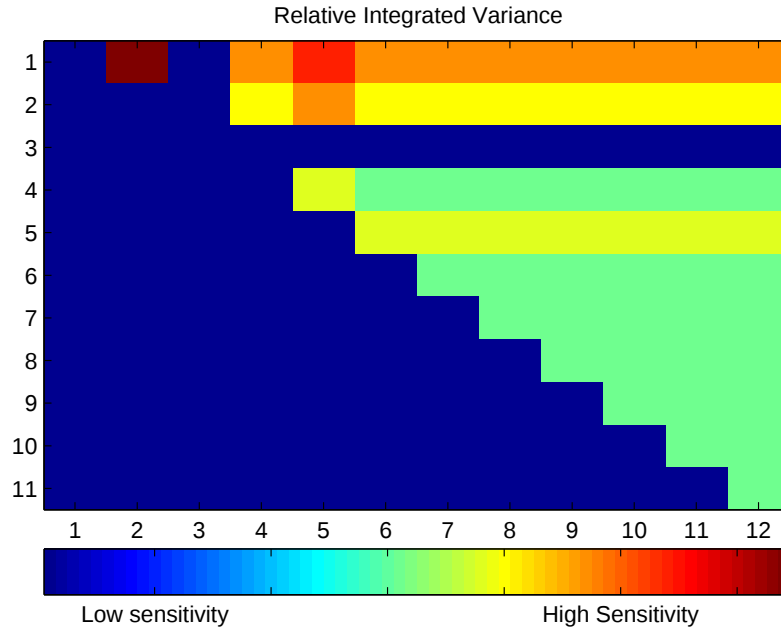


Figure 5-4: Two dimensional integrated variance analysis of the starboard AC system generator abc -reference frame c current: all triples containing the starboard bus filter inductance (3) for ONR IPS.

Taking the most influential triple, we now compare the sensitivity of different output states. Since we are now comparing different outputs with differing units, we normalize the integrated variance by the integral of the square of the deterministic solution. The output states corresponding to each number are listed below.

- Zone 3 DC states
 - 1) Load Voltage
 - 2) Starboard Voltage
 - 3) Starboard Current

- 4) Port Voltage
- 5) Port Current
- Starboard AC system states
 - 6-8) Voltages a, b, c
 - 9-11) Generator Currents a, b, c
 - 12-14) Propulsion Currents a, b, c
 - 15-17) Power Supply Currents a, b, c

The top plot in Fig. 5-5 shows the normalized integrated variances of the output states, and we see that the three most sensitive outputs to the triple $\{1, 2, 3\}$ are the power supply currents. The lower plot is of the multi-dimensional sensitivity (MS): the square root of the normalized variance divided by the sum of the variance of the uncertain parameters in the simulation. This tells the ratio of the standard deviation of the measured states to the standard deviation of the uncertain inputs. There is an eightfold increase in the variation of the power supply currents, which

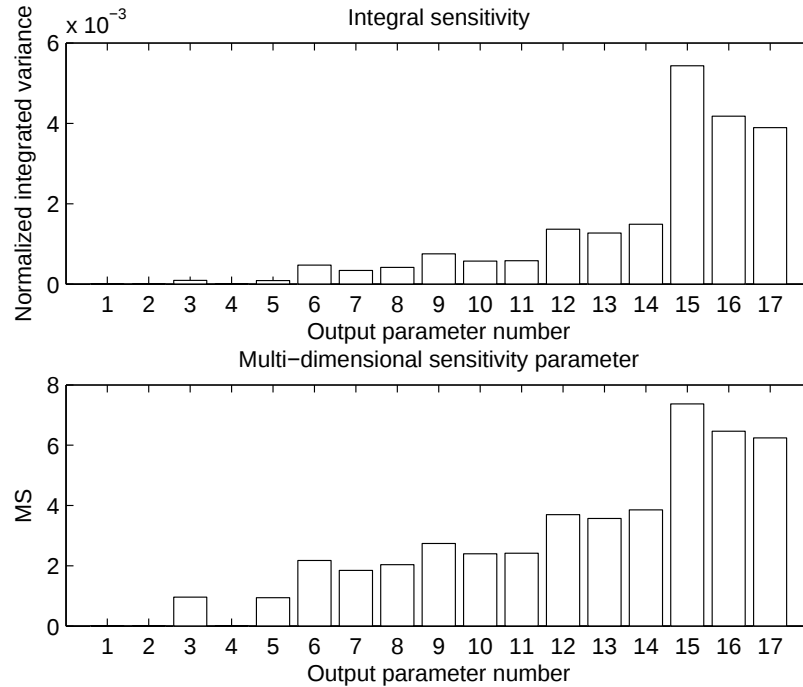


Figure 5-5: Starboard AC system normalized integrated variances and multi-dimensional sensitivity (MS) for ONR IPS.

is not very reassuring. But this is only for $\pm 1\%$ uncertainty. Fig. 5-6 shows the integrated variances (above) and multi-dimensional sensitivities (below) of the power supply currents for $\pm 1 - 17\%$ uncertainty. The near eightfold increase only occurs at $\pm 1\%$ uncertainty, and stays below 2 past $\pm 5\%$ uncertainty (although it does appear to be slightly on the rise past $\pm 15\%$).

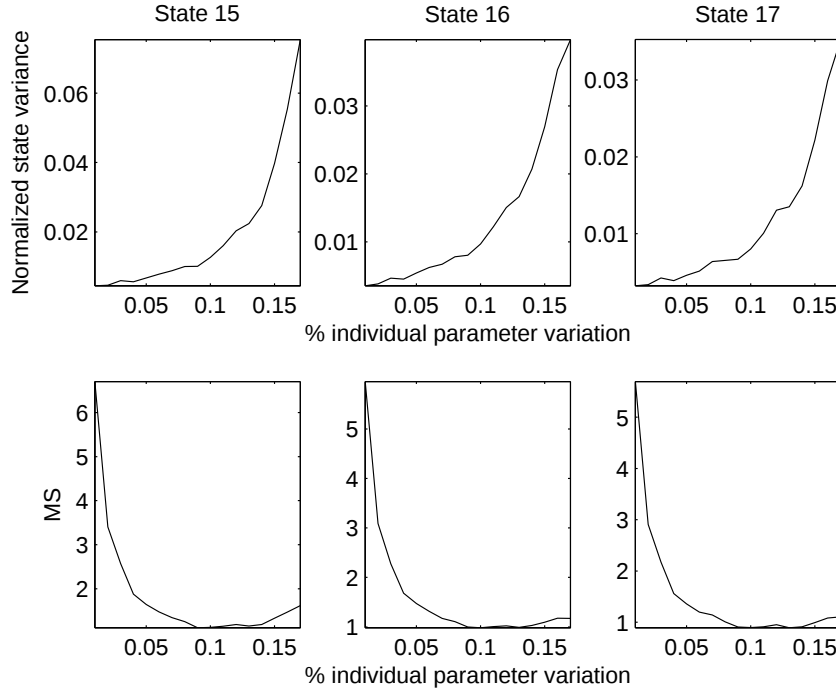


Figure 5-6: Multi-dimensional sensitivity (MS) of starboard power supply currents for $\pm 1 - 17\%$ uncertainty for ONR IPS.

5.1.3 Adaptive Simulation

We now interpret the results of a two dimensional, dimension adaptive, four element simulation of the uncertain pair $\{1, 2\}$, the bus shunt and filter capacitances. Fig. 5-7 shows the behavior of the dimension adaptive algorithm based on the integral convergence of state 15, the a power supply current.

We can see from the small amount of computational effort expended that the model is a relatively well behaved function of the bus shunt capacitance below its mean, but that above a higher resolution is needed for an accurate result (Fig. 5-7).

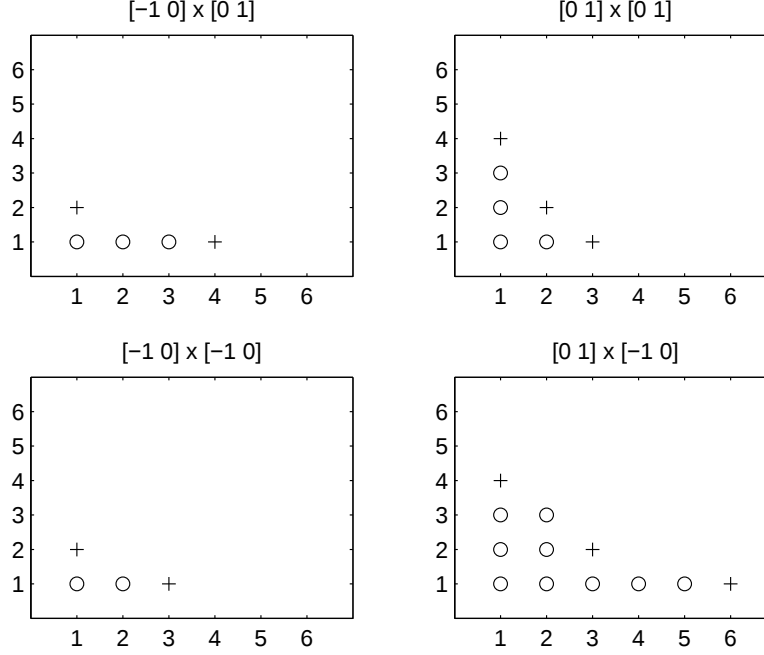


Figure 5-7: Record of dimension adaptive behavior in each element for a two dimensional, four element simulation with uncertainty in the bus shunt and filter capacitances for ONR IPS. Each location on the plot represents a difference grid in the corresponding element. Recently evaluated difference grids are denoted by + and old difference grids by O. The title of each quadrant is the portion of the integral's domain covered by that element. It also corresponds to the percent variation from the mean of the uncertain couple.

Perhaps the most useful observation is the presence of higher order joint difference grids in the $[0\ 1] \times [-1\ 0]$ element, which indicates that there is significant coupled behavior when the shunt capacitance and the filter capacitance are respectively above and below their means. This is not a measure of sensitivity; however, this sort of coupling as a function suggests that high multidimensional sensitivity to uncertainty in a set of parameters may be a result of the interaction between them.

Similar information can be inferred from higher dimensional simulations; we only show a two dimensional result because it is easily presented graphically. The same mean and variance analysis seen in the previous section can be obtained with dimension adaptive simulation, but we omit them here to avoid redundancy.

5.2 Pulse Power System

We analyze uncertainty in a Simulink model describing the operation of a large pulse load reflecting the power consumption of a rail gun [8]. The alternator is charged by accelerating its rotor to 18,000 rpm, at which point the inverter and charging motor are disconnected from the alternator and a shot is fired.

A six dimensional dimension-adaptive simulation using Clenshaw-Curtis points was run, with $\pm 10\%$ uniform uncertainty in the alternator field winding resistance, primary generator one field winding leakage inductance, inductance of charging motor switches, and charging motor phase resistance, and $\pm 5\%$ uniform uncertainty in the "on" configuration output rectifier diode resistance and charging motor excitation flux.

Fig. 5-8 shows mean trajectories with standard deviations for a few states, and 5-9 shows histograms at $t = 30$ seconds. A total of 53 system evaluations were required to attain an error tolerance of 10^{-3} for the charging motor phase one stator current. In most dimensions, the algorithm expended little effort, the highest order joint grid evaluated being level four in the dimension corresponding to the charging motor excitation flux and level three in that corresponding to the diode resistance, indicating some coupling between those parameters. A level three full grid would have used 729 points, taking much more time to achieve a comparable level of accuracy.

It can be seen from Fig. 5-8 that the generator field winding voltage and generator speed are sensitive to the simulated uncertainty, the first during charging before the shot is fired and the latter afterward. The distributions at 30 seconds appear to mostly be Gaussian, with the exception of the generator speed, which is nearly uniformly.

5.3 RTDS

Uncertainty analysis was performed on a pulse load charging event for a notional all-electric ship at the Center for Advanced Power Systems at Florida State University using the dimension-adaptive algorithm (section 3.1.2, [13]). The system, a schematic

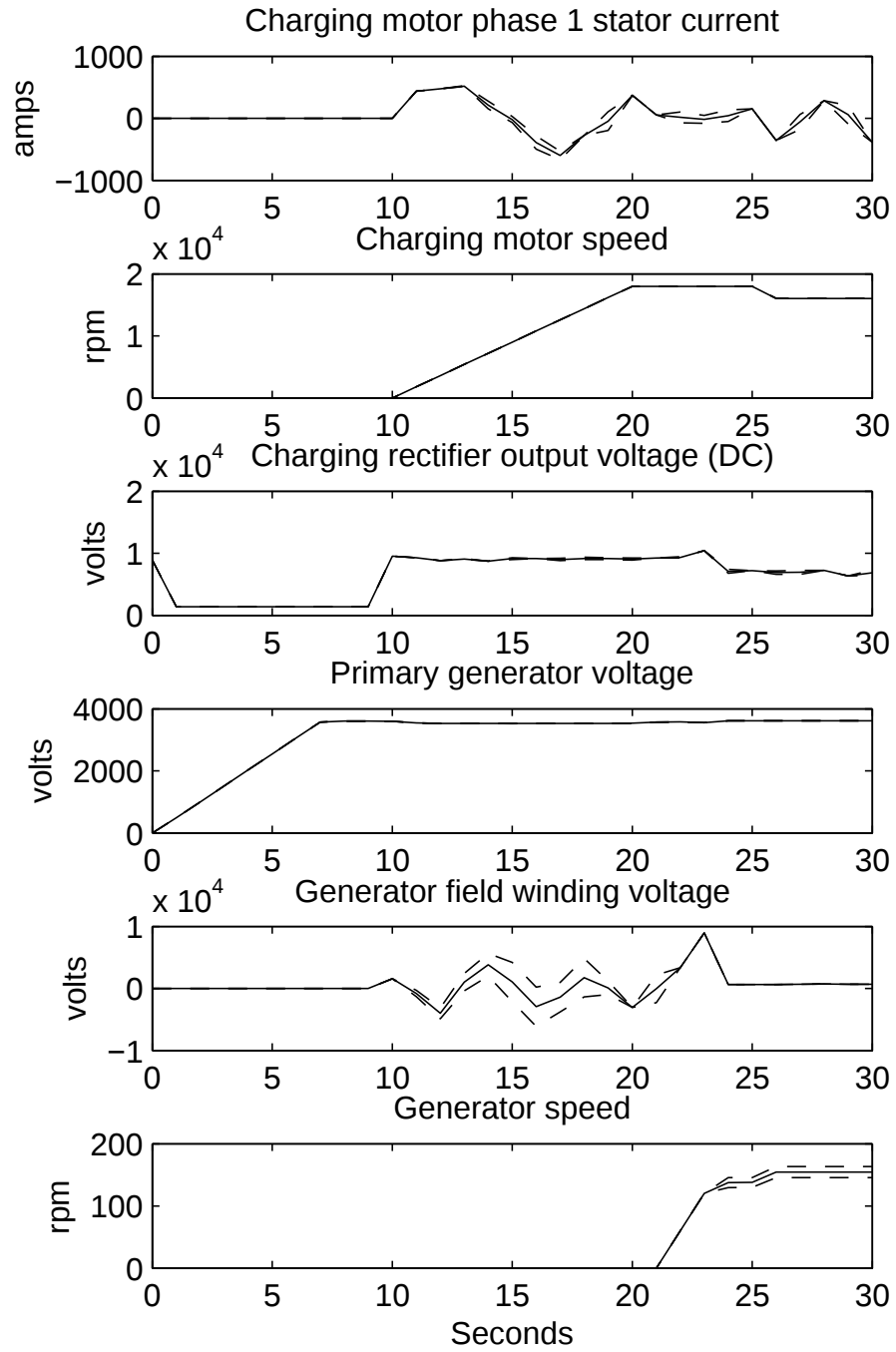


Figure 5-8: Mean trajectories with standard deviation envelopes of pulse power system states.

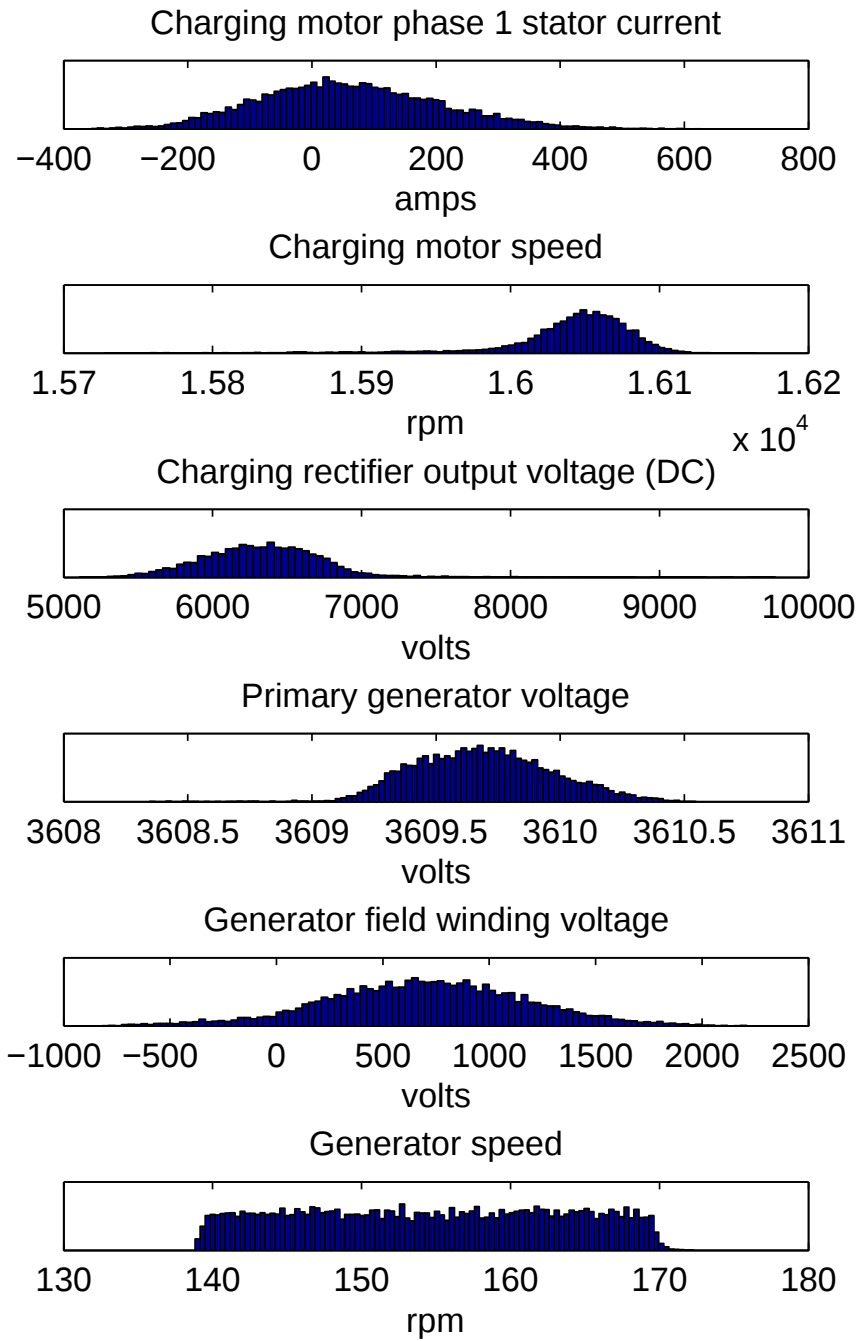


Figure 5-9: Histogram of pulse power system states at $t = 30$ seconds.

of which is given in Fig. A-1, has two 36 MW main generators and two 4 MW auxiliary generators. The load is comprised of two 36.5 MW propulsion motors, and maximum 4 MW ship service loads. A thorough description can be found in [25, 40]. The notional system simulation was run on a real time digital simulator [24], on which large power system simulations can be run in real-time with order $50\mu s$ time steps, and into which physical hardware can be incorporated. The simulations carried out for this analysis were run slower than real-time with a $15\mu s$ time step.

Fig. A-2 in appendix A shows interpolated versus actual values for the maximum deviation from a nominal frequency for main generator one and auxiliary generator two. Detailed results can be found in [26].

5.4 Discussion

Collocation was used to assess the behavior of three systems under mid-dimensional uncertainty. There were significant computational savings over full and sparse grids for simulations in which the dimension-adaptive algorithm was used, as was evidenced by the uneven sampling of those systems' random spaces. Instructions for using a Matlab implementation of the collocation algorithms is given in appendix B.

Chapter 6

Unscented Kalman Filtering with Dimension-Adaptive Numerical Integration

In this chapter, the dimension-adaptive algorithm is implemented in an unscented Kalman filter. The performance of the new filter is evaluated on three example systems from the literature.

6.1 Background

We want to estimate the state $x_k \in \mathbb{R}^n$ of the nonlinear discrete dynamical system with process and measurement noise

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{r}_k,\end{aligned}\tag{6.1}$$

where $\mathbf{w}$ and $\mathbf{r}$ are zero-mean, uncorrelated noise vectors with respective covariances $\mathbf{Q}$ and $\mathbf{R}$.

6.1.1 The Extended Kalman Filter

The Kalman filter produces optimal state estimates based on current and previous measurements by first computing a prediction $\hat{\mathbf{x}}_{k+1|k}$ via a system model and then a corrected prediction $\hat{\mathbf{x}}_{k+1|k+1}$ with a measurement from the actual system. Both the mean and covariance matrix $\mathbf{P}$ of the state are propagated with the filter. When $\mathbf{f}$ and $\mathbf{h}$ in (6.1) are linear with system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$, the predictor portion is calculated by

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= \mathbf{A}\hat{\mathbf{x}}_{k|k} + \mathbf{B}\mathbf{u}_k \\ \mathbf{P}_{k+1|k} &= \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{Q} \\ \hat{\mathbf{z}}_{k+1} &= \mathbf{C}\hat{\mathbf{x}}_{k+1|k} + \mathbf{D}\mathbf{u}_k,\end{aligned}\tag{6.2}$$

with correction

$$\begin{aligned}\mathbf{P}_{vv} &= \mathbf{C}\mathbf{P}_{k+1|k}\mathbf{C}^T, \quad \mathbf{P}_{xz} = \mathbf{P}_{k+1|k}\mathbf{C}^T \\ \mathbf{L} &= \mathbf{P}_{xz}/(\mathbf{P}_{vv} + \mathbf{R}) \\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{L}(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1}) \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{L}\mathbf{P}_{xz}^T.\end{aligned}\tag{6.3}$$

In the nonlinear case, the mean and output can be calculated in the same fashion with the exact system model, but there is no analogous method for the covariance matrix, so an approximation must be used instead. Traditionally, this has been done by linearizing about $\hat{\mathbf{x}}_{k|k}$ and then performing (6.2) and (6.3) with the system and output Jacobians $\mathbf{F}$ and $\mathbf{H}$. This is a shortcoming of the extended Kalman Filter: important characteristics of the model are lost in the linearization, and the Jacobian must be reevaluated at every time step, a tedious and computationally expensive procedure.

6.1.2 The Unscented Kalman Filter

In [19], the necessity for linearization in the extended Kalman filter was circumvented by numerically integrating for $\mathbf{P}_{k|k}$, $\mathbf{P}_{zz}$, and $\mathbf{P}_{xz}$. The original system equations are evaluated at *sigma* points having the same mean and covariance as the state, and the

predicted mean and covariance are then calculated from the transformed points:

$$\begin{aligned}
\sigma_1 &= \hat{\mathbf{x}}_{k|k}, & w_1 &= \kappa/(\eta + \kappa) \\
\sigma_i &= \hat{\mathbf{x}}_{k|k} + \hat{\mathbf{x}}_{k|k}(\sqrt{(\eta + \kappa)\mathbf{P}_{k|k}})_i, & w_i &= 1/2(\eta + \kappa) \\
\sigma_{i+n} &= \hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k}(\sqrt{(\eta + \kappa)\mathbf{P}_{k|k}})_i, & w_{i+\eta} &= 1/2(\eta + \kappa),
\end{aligned} \tag{6.4}$$

where $i = 2, \dots, \eta + 1$. $(\mathbf{A})_i$ denotes the i^{th} row of $\mathbf{A}$, and the matrix $\mathbf{B}$ with square root $\mathbf{A}$ is of the form $\mathbf{A}^T \mathbf{A}$. The mean and covariances are then calculated by

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=1}^{2\eta+1} w_i \mathbf{f}(\sigma_i), \quad \hat{\mathbf{z}}_{k+1} = \sum_{i=1}^{2\eta+1} w_i \mathbf{h}(\sigma_i) \tag{6.5}$$

$$\begin{aligned}
\mathbf{P}_{k+1|k} &= \sum_{i=1}^{2\eta+1} w_i (\mathbf{f}(\sigma_i) - \hat{\mathbf{x}}_{k+1|k})(\mathbf{f}(\sigma_i) - \hat{\mathbf{x}}_{k+1|k})^T \\
\mathbf{P}_{xz} &= \sum_{i=1}^{2\eta+1} w_i (\mathbf{f}(\sigma_i) - \hat{\mathbf{x}}_{k+1|k})(\mathbf{h}(\sigma_i) - \hat{\mathbf{z}}_{k+1})^T \\
\mathbf{P}_{vv} &= \sum_{i=1}^{2\eta+1} w_i (\mathbf{h}(\sigma_i) - \hat{\mathbf{z}}_{k+1})(\mathbf{h}(\sigma_i) - \hat{\mathbf{z}}_{k+1})^T.
\end{aligned} \tag{6.6}$$

6.2 Dimension-Adaptive Unscented Kalman Filtering

The sigma points of [19] are a type numerical integration, specifically one that assumes a Gaussian distribution, as with Hermite polynomial Gaussian quadrature. It is clear that the any valid numerical integration scheme could be used in place of the sigma points, so it is sensible to employ a sophisticated technique. This has already been done to some extent: [18] implemented the unscented filter with Gaussian quadrature, and [16] with quasi-Monte Carlo. We propose a filter that uses the dimension-adaptive integration scheme from [13] in place of the sigma points of the original unscented filter.

6.3 Performance of the New Filter

We now compare unscented Kalman filters with Kronrod-Patterson dimension-adaptivity (DA), Hermite polynomial Gaussian quadrature (GQ), and sigma points (SP) and the extended Kalman filter (E) on the falling body tracking problem from [1], the Kraichnan-Orszag system [30, 42], and a system based on the Lorenz system example in [18]. Gaussian quadrature with Hermite polynomials integrates functions over a Gaussian distribution, and is slightly superior to Legendre polynomial Gaussian quadrature for this application. However, there is no standard nested Gaussian distributed quadrature sequence, which is why we use the uniformly distributed Kronrod-Patterson quadrature in the dimension adaptive filter.

Most of the previous literature present results via the absolute value of the mean error for all trials alongside the error variance. For concision and because the relevant information is retained, we alternatively give the mean of the absolute value of the error in our experiments.

In each experiment 100 Monte Carlo simulations were performed. In this sections figures, EKF stands for the extended Kalman filter, UKF the unscented Kalman filter, GQKF the Gaussian quadrature Kalman filter of [18], and DAQKF the dimension-adaptive quadrature Kalman filter.

6.3.1 Falling Body Problem

The equations of motion for the system are given by

$$\begin{aligned}\dot{x}_1 &= -x_2 + w_1 \\ \dot{x}_2 &= -\exp(-\gamma x_1)x_2^2x_3 + w_2 \\ \dot{x}_3 &= w_3,\end{aligned}\tag{6.7}$$

with measurement

$$z = \sqrt{M^2 + (x_1 - H)^2} + r,\tag{6.8}$$

where $\gamma = 5 \cdot 10^{-5}$, $M = 10^5$, $H = 10^5$, and $\mathbf{w}$ and r are zero-mean uncorrelated noises with covariances given by $\mathbf{Q} = \mathbf{0}$ and $\mathbf{R} = 10^4$, respectively. The process noise is removed for the reason that it can mitigate linearization errors and thus actually improve the the extended Kalman filter's performance. The initial state of the true system is

$$\mathbf{x}(0) = [3 \cdot 10^5 \quad 2 \cdot 10^4 \quad 10^{-3}]^T. \quad (6.9)$$

The initial estimates for the simulated filter were

$$\begin{aligned} \mathbf{x}(0) &= [3 \cdot 10^5 \quad 2 \cdot 10^4 \quad 3 \cdot 10^{-5}]^T \\ \mathbf{P}_{0|0} &= \begin{bmatrix} 10^6 & 0 & 0 \\ 0 & 4 \cdot 10^6 & 0 \\ 0 & 0 & 10^{-4} \end{bmatrix}. \end{aligned} \quad (6.10)$$

With the faulty initial estimate we are assuming the body to be heavier than it actually is. Measurements are made at every second, with 64 fourth-order Runge-Kutta system updates between each measurement.

Little to no improvement is seen over the Gaussian quadrature and sigma point filters (Fig. 6-1).

6.3.2 The Kraichnan-Orszag System

The Kraichnan-Orszag three mode system can be highly oscillatory, depending on initial conditions, and, with infrequent measurements, represents a challenging filtering and stochastic simulation problem. The system equations are

$$\begin{aligned} \dot{x}_1 &= x_2 x_3 + w_1 \\ \dot{x}_2 &= x_1 x_3 + w_2 \\ \dot{x}_3 &= -2x_1 x_2 + w_3. \end{aligned} \quad (6.11)$$

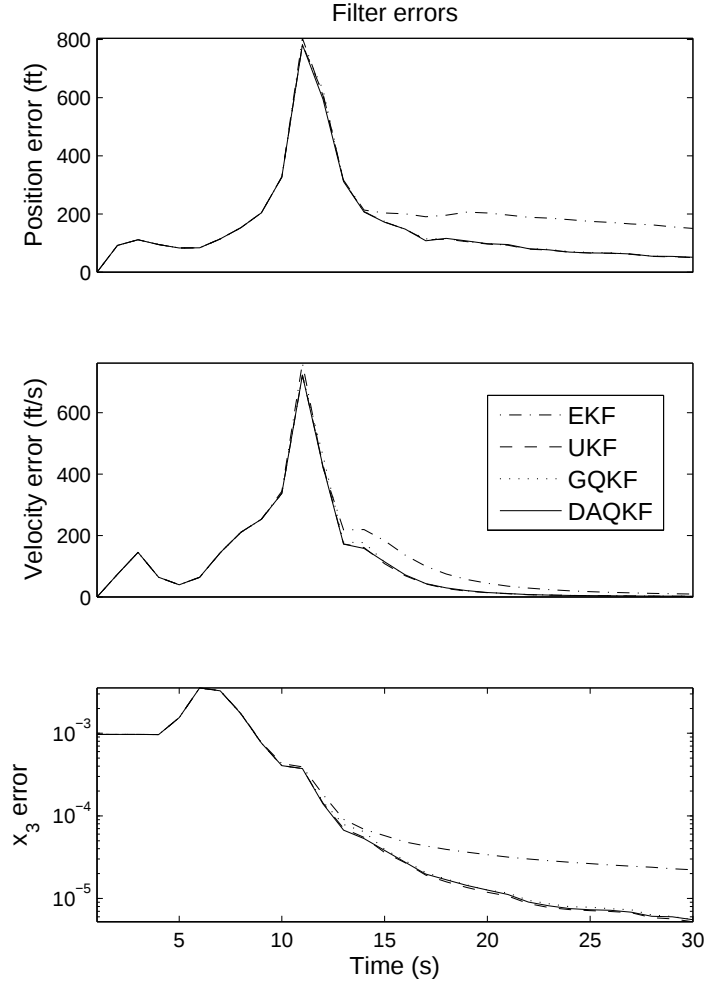


Figure 6-1: Mean absolute error for position, velocity, and x_3 for falling body problem.

As with the falling body problem, the process noise covariance matrix is set $\mathbf{Q} = \mathbf{0}$. For the measurement we use

$$z = \mathbf{x} + \mathbf{r}, \quad (6.12)$$

with measurement noise covariance $\mathbf{R} = 5\mathbf{I}$. The initial state used in both the 'true' and filter simulation of (6.11) is $\mathbf{x}(0) = [0 \ 1 \ 20]^T$, with initial covariance estimate $\mathbf{P}_{0|0} = \mathbf{I}$. A fourth order Runge-Kutta solver with $dt = 0.01$ was used, with measurements taken every $20dt$. Fig. 6-2 shows the corresponding state trajectories.

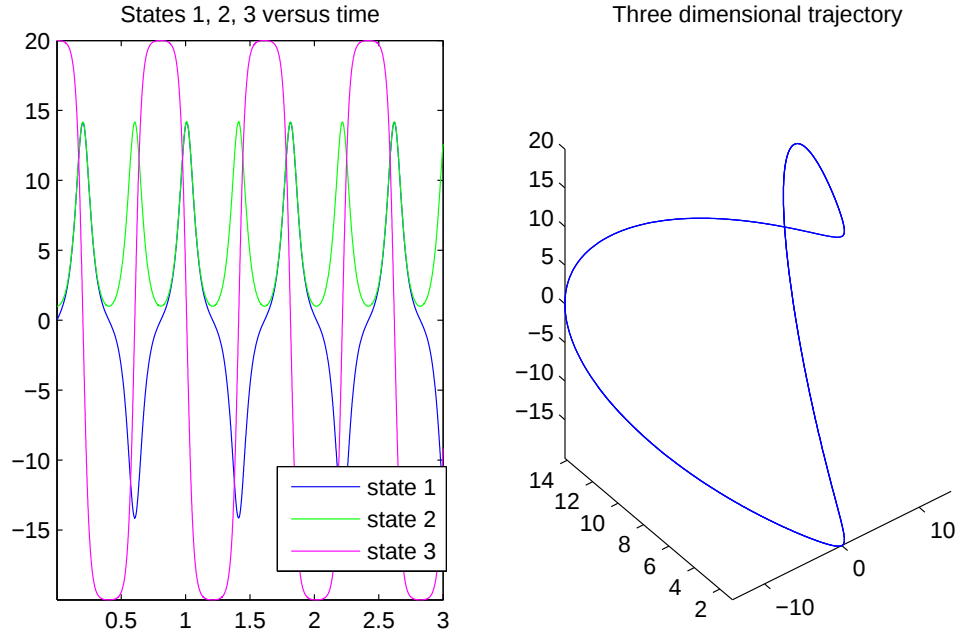


Figure 6-2: Trajectories of the Kraichnan-Orszag system.

Fig. 6-3 shows the errors of dimension-adaptive filters with different error tolerances. The filters with smaller tolerances outperform those with larger tolerances. Initially, more points are used by the adaptive algorithm to compute more precise means and covariances. As the filters' confidences grow, the covariances shrink, decreasing the size of the region being integrated over. Numerical integrals are easier on smaller domains, and hence fewer points are used by the adaptive filter as the covariance decreases.

These errors are very choppy; the reason for this is in the nature of the Kraichnan-Orszag system. Fig. 6-4 shows a 'true' trajectory in black, and noisy trajectories with noise added. The noisy trajectories sometimes take wrong turns at the corners, causing the error to spike, and then return to the correct path at the next corner, causing the error to spike. Once the variance has decreased sufficiently, the points are close enough together that they all take the same path, and the error smoothens out.

The dimension-adaptive filter outperforms all of the other filters on this example (fig. 6-5). Initially, it expends a significantly larger effort than the others, but

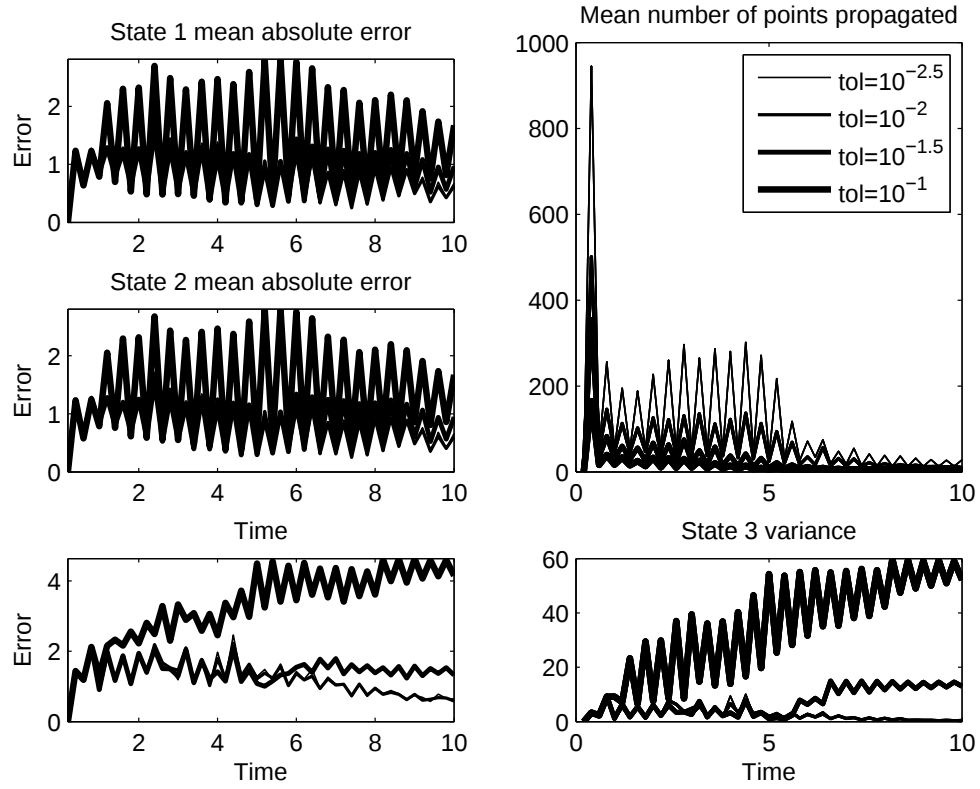


Figure 6-3: Error, variance, and mean number of points propagated for dimension-adaptive filters with different error tolerances on Kraichnan-Orszag system.

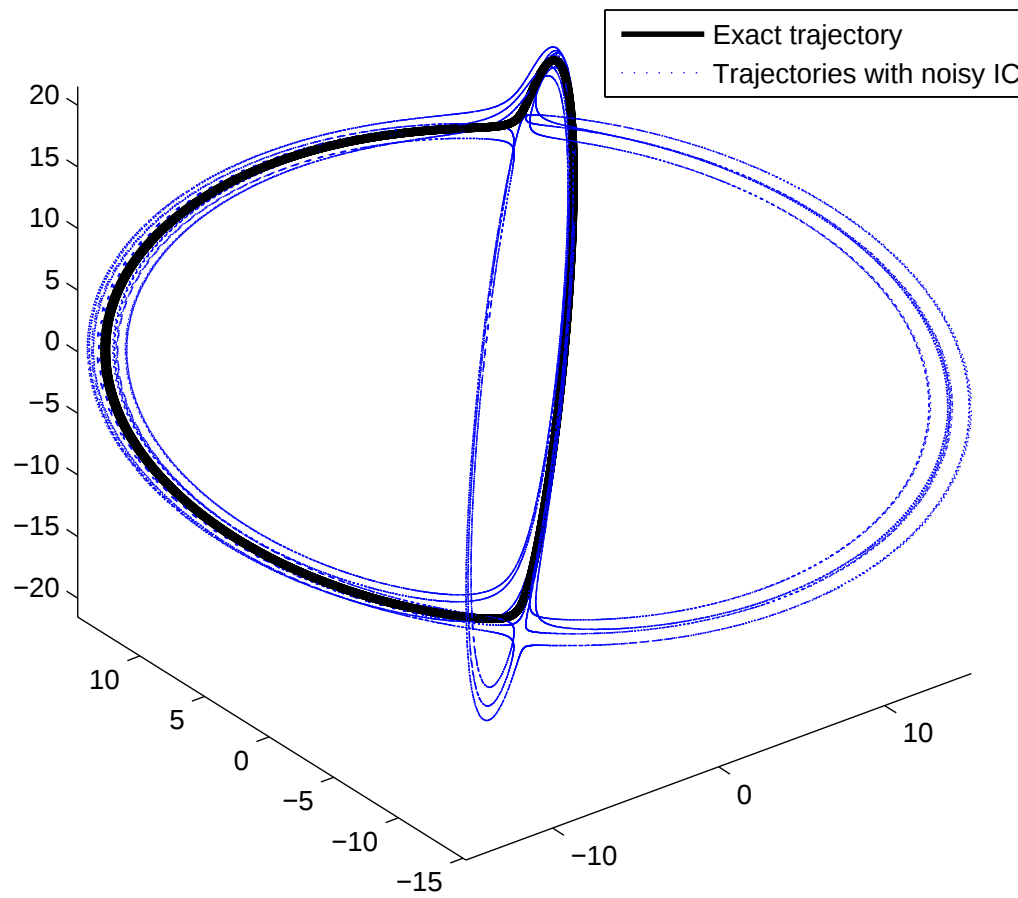


Figure 6-4: Noisy Kraichnan-Orszag trajectories.

eventually only requires about the same effort as the unscented filter to achieve its prescribed error.

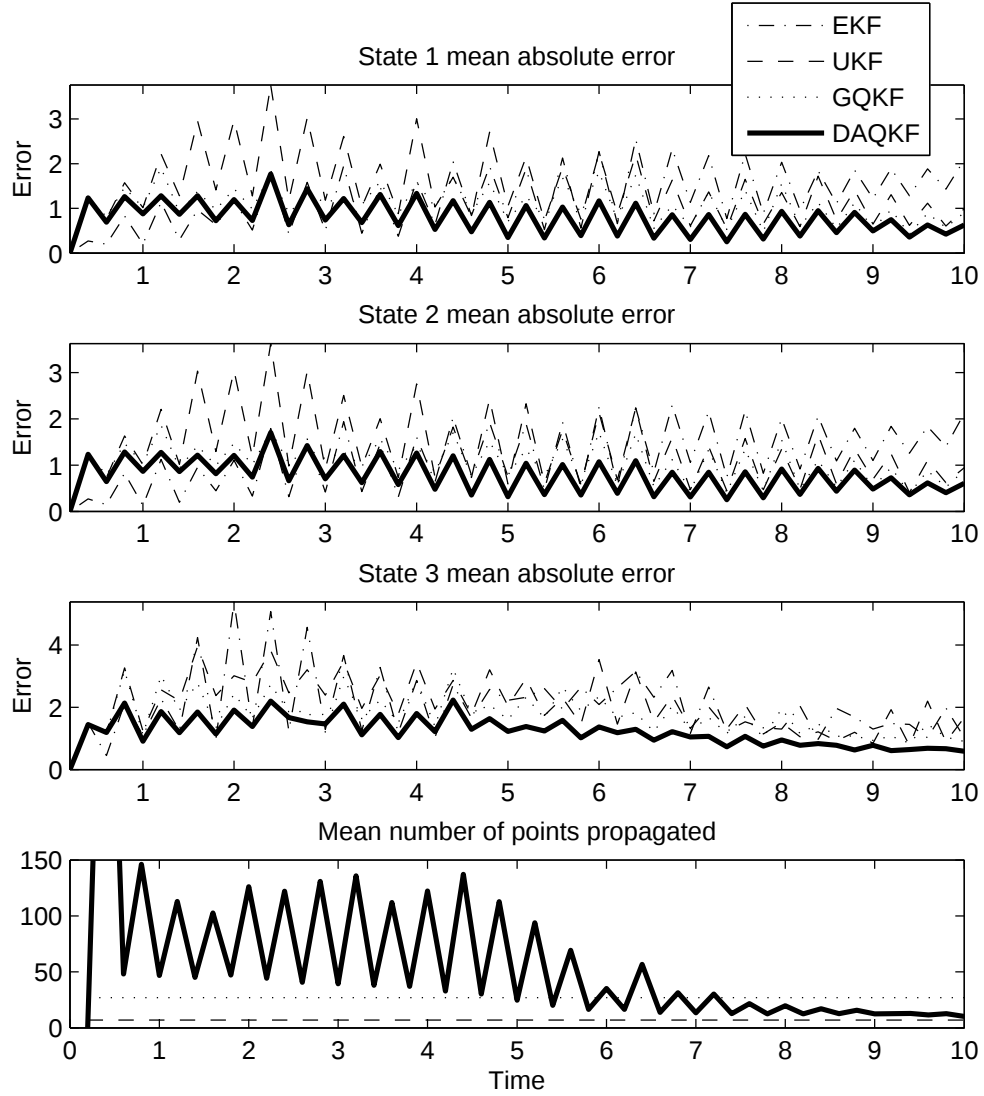


Figure 6-5: Performance of filters on the Kraichnan-Orszag system.

6.3.3 The Lorenz system

Many systems of interest follow trajectories like those of the Lorenz system, e.g. electrical circuits [5] and lasers [44]. A comparison of is shown on an example based

on the Lorenz system from [18]. The system equations are given by

$$\begin{aligned}\dot{x}_1 &= \sigma(-x_1 + x_2) \\ \dot{x}_2 &= \rho x_1 - x_2 - x_1 x_3 \\ \dot{x}_3 &= -\beta x_3 + x_1 x_2 + 0.5w,\end{aligned}\tag{6.13}$$

where $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$, which correspond to the well-known butterfly effect system. Fig. 6-6 show system trajectories for the parameters and initial conditions used. The output is given by

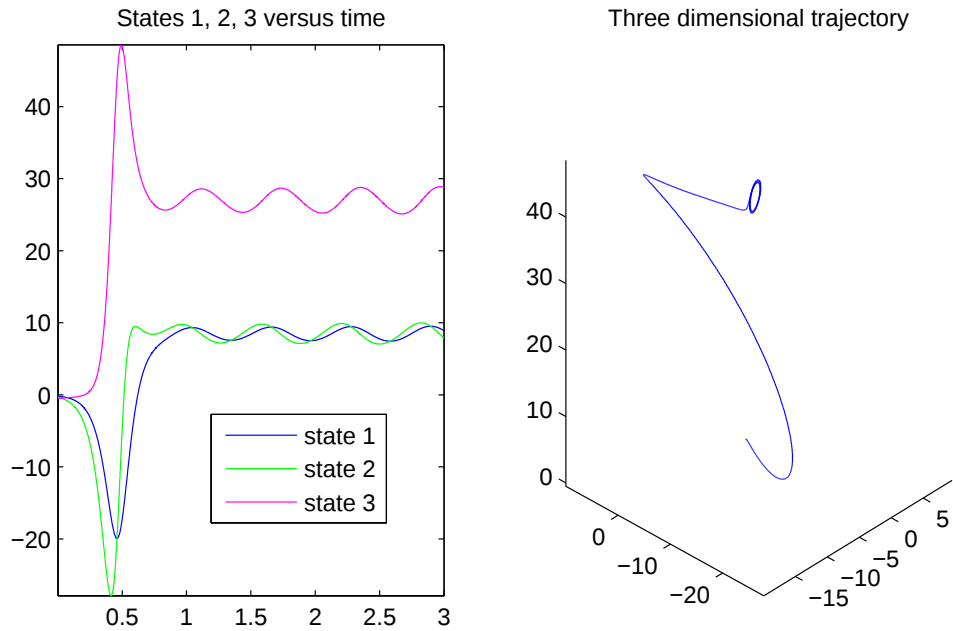


Figure 6-6: Trajectories of the Lorenz system.

$$y = \sqrt{(x_1 - 0.5)^2 + x_2^2 + x_3^2} + 0.065r,\tag{6.14}$$

and the noises w and r each have the same variance $\mathbf{Q} = \mathbf{R} = dt$.

An Euler step with $dt = 0.01$ is used. In [18], measurements were taken at every time step; here instead we take measurements at $10dt$.

The dimension-adaptive filter outperforms the other unscented filters, avoiding the first three error spikes (Fig. 6-7). The number of points used goes above and

below the 27 used by the Gauss-Hermite filter, depending on the number of points needed to achieve accurate integration of the state and covariance. The extended Kalman filter fails on this example, and so it is not shown.

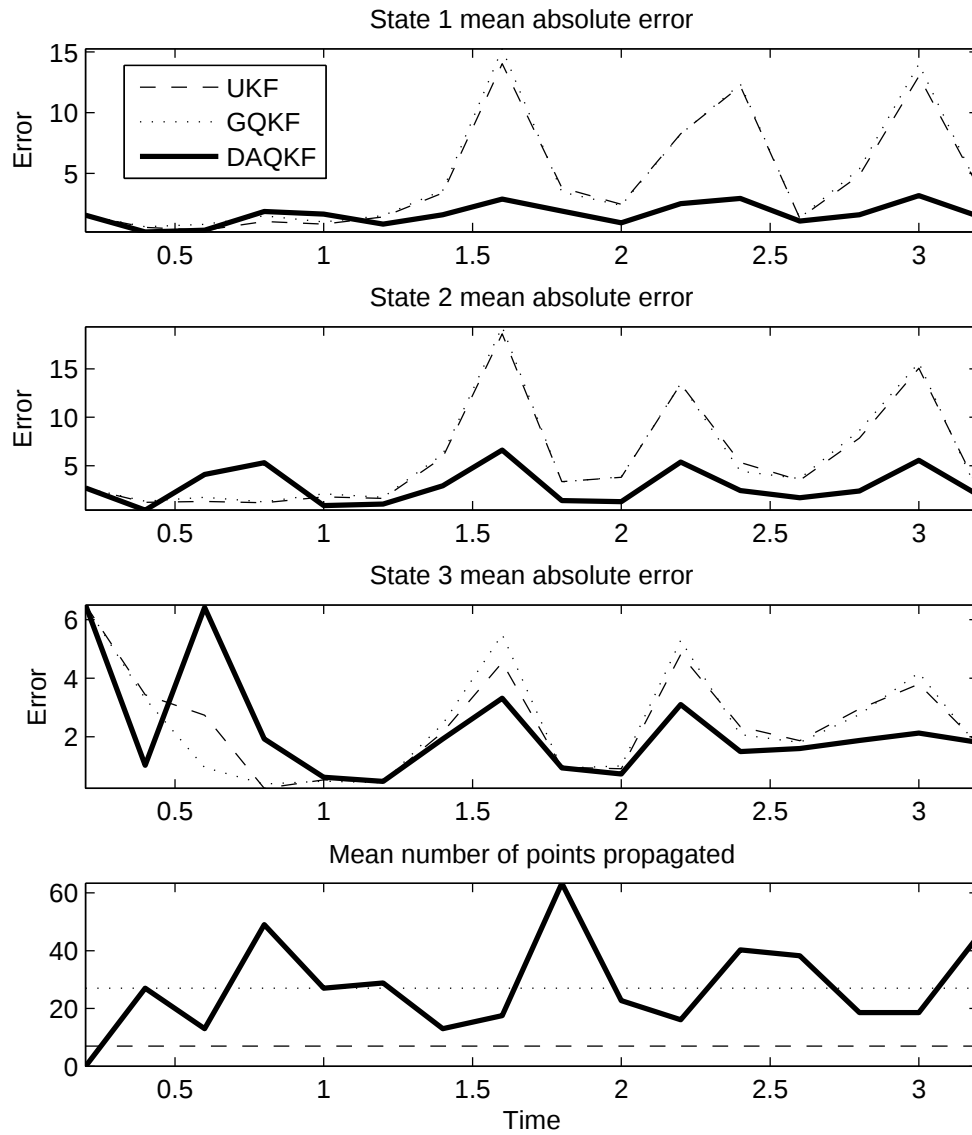


Figure 6-7: Performance of filters on the Lorenz system.

6.4 Summary

A new filter using dimension-adaptive numerical integration within a conventional unscented Kalman filter is presented. Performance gains were seen on two example systems.

Chapter 7

Summary

Stochastic differential equations provide a framework for modeling systems with uncertainty and informing design decisions, but for large-scale systems they are often hard to solve numerically because of high dimensional random spaces. Collocation is an attractive approach for large-scale systems with moderate dimensional ranges because of its 'black-box' implementation and the existence of sophisticated tensor-product algorithms with which it can be used. In this work, full, sparse, and dimension-adaptive [13] grid collocation with various polynomial-based component quadratures were compared on standard test integrals [11] and then applied to compute solutions to stochastic differential equations describing a notional electric ship in standard operation, a pulse power system, and a charging event on a notional electric ship. Statistical moments and PDF's were extracted from these solutions, from which sensitivity was inferred. The capability to assess the behavior of a system under uncertainty is a powerful design tool. It is an indicator of robust design and control strategies; in the case of the electric ship, two such examples are safe loading conditions or ranges of deviation from nominal behavior as a result of imprecise system knowledge. A Matlab toolbox has been created and released to the Electric Ship Research and Development Consortium Partners (appendix B). The results of Chapter 5 were generated with this toolbox; there are also additional users at this writing.

The sparse grid with shifted indexing, a modification to the conventional sparse

grid formulation, was presented, along with justification in the form of theoretical bounds and numerical experiments. The dimension-adaptive grid collocation algorithm was then applied to unscented Kalman filtering, and improvement over the extended and other unscented Kalman filters was seen for two nonlinear filtering problems. An adaptive unscented filter is promising because of its potential for computational savings; in general less computational effort is required as the filter attains greater confidence in its estimates. The gains for the examples shown, however, are modest.

Future work includes collocation using locally adaptive grids, i.e. algorithms that can adapt to regionally rather than dimensionally in the random space. Another direction is the fusion of Monte Carlo and collocation. While at first seemingly counter to what makes Monte Carlo work, which in part is that points are placed directly into the multi-dimensional random space rather than built up from one-dimensional bases, assigning certain 'easier' dimensions to collocation and the rest to an aggregate Monte Carlo dimension may yield faster convergence, and be particularly effective if done adaptively.

Appendix A

Figures from FSU CAPS Collaboration

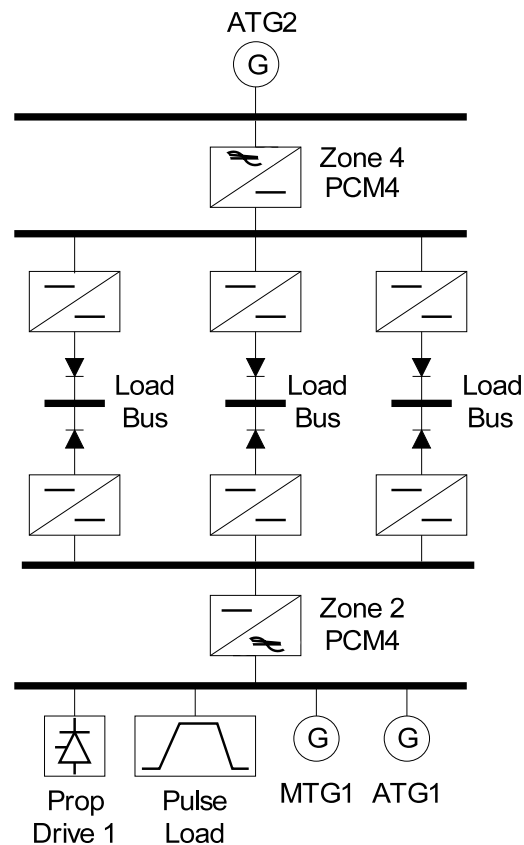


Figure A-1: Schematic for RTDS notional all-electric ship at Florida State University.

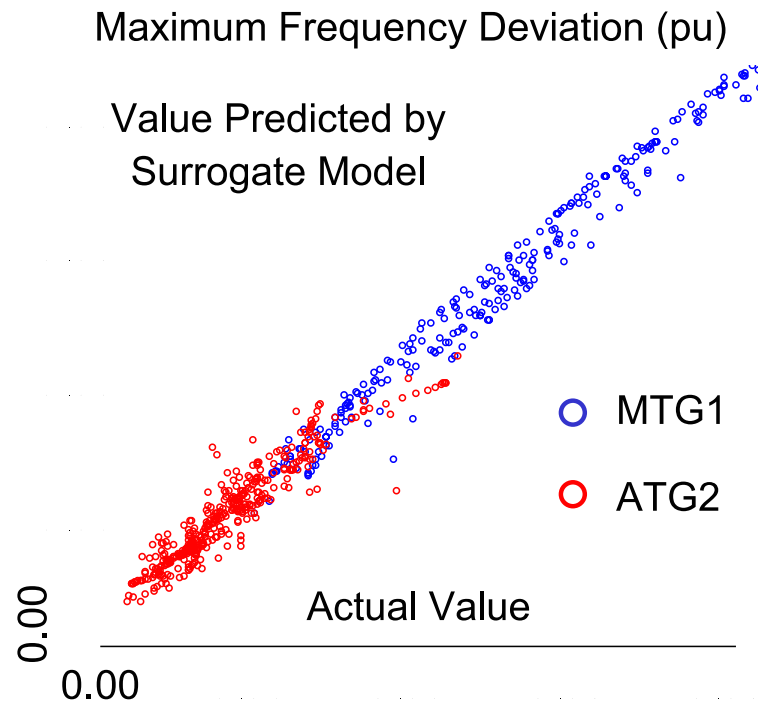


Figure A-2: Comparison between interpolated and true values for maximum frequency deviation of main generator one and auxiliary generator two for RTDS notional electric ship at Florida State University.

Appendix B

Readme for Matlab implementation of Collocation Algorithms

B.1 Overview and introduction

This appendix serves as a guide for the Matlab program *Collocation.m*, a set of routines for numerically integrating and interpolating multi-dimensional functions on the unit cube via tensor products of one dimensional rules. It can be downloaded from <http://web.mit.edu/hovergroup/docs.html>. Monte Carlo and quasi-Monte Carlo techniques are not part of its functionality, although in many situations, namely when dealing with high-dimensional or low-smoothness functions, they are the best choice. The approaches included are:

- Full grid tensor-product cubature
- Sparse grid tensor-product cubature [12, 28, 38]
- Uniform-adaptive full or sparse grid tensor-product cubature
- Dimension-adaptive tensor-product quadrature [13]
- Uniform subspace splitting

Each approach is based on one-dimensional quadratures. Polynomial based rules have the highest polynomial exactness, which is the highest degree polynomial a method will integrate exactly using a fixed number of points. The available one-dimensional basis quadratures are:

- Legendre polynomial Gaussian quadrature [36]
 $2n - 1$ polynomial exactness, not nested, constant integration kernel.
- Hermite polynomial Gaussian quadrature
 $2n - 1$ polynomial exactness, not nested, Gaussian integration kernel.
- Clenshaw-Curtis [4, 41]
 n polynomial exactness, nested, constant integration kernel.
- Kronrod-Patterson [10, 31, 33]
 $2n + m - 1$ polynomial exactness, nested, constant integration kernel. Kronrod-Patterson is a nested sequence of quadratures beginning with Gaussian quadrature. n is the number of Gaussian quadrature points, and m the number of extension points. Specifically, the sequence beginning with the three points Legendre polynomial rule is included. The Kronrod-Patterson rule is not generated by a program, but read from a table that goes up to the seventh level in a sequence. For this reason, it is important when using an adaptive grid to set the maximum grid level below the highest available Kronrod-Patterson rule; otherwise a higher rule will be called, and the program will return an error.

Gaussian quadrature has the highest polynomial exactness, but it is not nested, and for this reason Clenshaw-Curtis and Kronrod-Patterson are superior choices for sparse grids or adaptive integration.

B.2 Using *Collocation.m*

Collocation.m is called by

$$[outputs] = Collocation(q, d, @integrand, options)$$

where q is the level and d the dimension. q is only used as the non-adaptive default level if no adaptive options are specified. `@integrand` is the function handle for the integrand function. The outputs correspond to the solution to the integral plus what is specified in the options, for example, if `...'numval', 1...` is the only input after `integrand`, the output will be $[y, n]$, where y is the computed value of the integral and n the number of function evaluations.

options:

- *grid*
 - CC: Clenshaw-Curtis
 - KPL3: Kronrod-Patterson
 - Leg: Legendre polynomial Gaussian quadrature
 - Her: Hermite polynomial Gaussian quadrature
- *s_f*
 - 0: full grid
 - 1: sparse grid
- *funin*
 - Additional inputs to the integrand function in addition to the evaluation point.
- *timer*
 - 1: Output the time taken.
- *showgrid*
 - 1: In 2 dimensions, display the points and graph of the function as they are evaluated. In the dimension adaptive case, the old and active index sets for the current element are also shown. In 3 dimensions, the evaluation points are plotted.

- *variance*
 - 1: Calculate the variance.
- *mea* (Multi-Element Adaptivity)
 - Inputs are specified as an array:
 1. Number of partitions (1 if none)
 2. Desired resolution for each grid element. For non-adaptive, input an integer. That number will be the level in each partition.
 3. Maximum grid level (to prevent the adaptivity from getting out of control)
 4. Adaptivity - 0: non or uniform adaptivity (depending on 2), 1: Dimension adaptivity
 5. 1: Return record of adaptive behavior
 6. Adaptive criterion for time dependence: 0: Integral, 1: Maximum
 7. For integrands with vector ranges, this specifies the output parameter the adaptivity responds to (only for time dependent).
- *numval*
 - 1: Return the number of function evaluations.
- *timedep*
 - If the integrand returns vectors in time, this input should be a two element array in the form $[dt, t_f]$, where dt is the time step the simulation will be splined to and t_f the end time of the simulation.
- *nout*
 - In the time varying case, this is the number of function outputs.
- *interp*

- An n by d matrix, where n is the number of points being interpolated, d the dimension, and each row a point being interpolated. Do not use subspace splitting or Hermite polynomial Gaussian quadrature if using this feature.
- *showprog*
 - 1: Display which grid is being evaluated and the current attained error tolerance (dimension-adaptivity only).

The function f should be of the form

function $y = f(x, funin)$ for scalar functions.
function $[y, t] = f(x, funin)$ for functions that are vectors through time.
 t is the vector of times the solutions is
 splined to, of the form $t_1 : dt : t_2$.

where x is a vector of the coordinates for the point to be evaluated and $funin$ any extra information for the function. If there are no additional inputs to the function, keep an unused second input. It should return the function value at that point, which can be a scalar or a vector(s) through time. A description of the output of f is provided to the program through the options.

Sometimes there are multiple values of interest; this is often the case in large scale dynamic simulation, when there are many states. Because each function evaluation can be costly, it is sensible to be able to integrate each state using one set of simulations. In this case multiple states can be evaluated, and a matrix with rows corresponding to states and columns to times is returned.

When function evaluations are very expensive, it is highly recommended to have *Collocation.m* call an outer function which evaluates and writes to a file the function value if it is a new evaluation point, and reads the function value from an existing file if it has already been evaluated at that point. It may happen that a simulation is stopped before all the grids have been computed, or more similar simulations are run in the future; writing the evaluations to a file as the program goes prevents lost

or repetitive calculation. Below is an example of how such a program might look.

```
function  $y = outer(x, funin)$ 
    if  $f$  has previously been evaluated at  $x$ 
        read  $f(x, funin)$  from file
    else
        evaluate  $f(x, funin)$ 
        write  $f(x, funin)$  to a file
    end
     $y = f(x, funin)$ 
```

B.2.1 A warning about computational costs

Don't set the adaptive tolerance too small or the level of the non-adaptive grid too high. Computational cost can increase exponentially: if a certain level takes an hour, the next level up could take a year. For the adaptive grids, there is an option to impose a maximum grid level. It's easy to get a feel for how long the program takes with the test integrals (which are more difficult to accurately integrate than the ODE).

B.3 Description of each program

- Collocation files
 - *Collocation.m*
Interface program for user. Assembles options (see below) and calls sub-programs.
 - *ColOutput.m*
Calls multi-dimensional integration programs and assembles computed results.
 - *ColPartition.m*
Partitions the region of integration into identical subspaces.

- *ColFG.m*
Generates and evaluates a full grid tensor-product quadrature.
- *ColSG.m*
Generates and evaluates a sparse grid tensor-product quadrature.
- *ColSGIndexSets.m*
Creates the index sets for the sparse grid.
- *ColDiffGrid.m*
Returns abscissas and weights for one-dimensional quadrature routines. If specified, returns one dimensional difference grid.
- *ColSortComb.m*
Combines coincident points in sparse grids.
- *ColAdaptUnif.m*
Uniform adaptivity: evaluates higher order grids until adaptive tolerance is reached.
- *ColAdaptDim.m*
Dimension adaptivity: the dimension adaptive algorithm from [13].
- *ColAdaptDimGrid.m*
Generates the grid for *ColAdaptDim.m*.
- *ColAdaptDimValid.m*
Checks that the next grid evaluated in *ColAdaptDim.m* is valid.
- *ColInterpFG.m*
Interpolates chosen interpolation points if a full grid has been used.
- *ColInterpSGDA.m*
Interpolates chosen interpolation points if a sparse of dimension-adaptive grid has been used.
- *ColInterpDiffb.m*
Generates the one-dimensional difference coefficients for constructing sparse and dimension-adaptive grid interpolants.

- *FindLegendreRoots.m* (from Numerical Recipes [36])
Returns the abscissas and weights for Legendre polynomial Gaussian quadrature.
- *HermQuad.m*
Returns the abscissas and weights for Hermite polynomial Gaussian quadrature.
- *ClenshawCurtis.m*
Returns abscissas and weights for Clenshaw-Curtis quadrature.
- *KPL3.m*
Returns abscissas and weights for Gauss-Kronrod-Patterson quadrature beginning with the three point Legendre polynomial Gaussian quadrature.
- Auxiliary files
 - *TestIntegrand.m*
Test functions, including those in [11].
 - *TestIntegrandSol.m*
Exact solutions to test functions in *TestIntegrand.m*.
 - *Plots_D2.m*
Plots two-dimensional instantiations of each test function.
 - *dirinit.m*
Adds the directory to the Matlab path, so *Collocation.* can be called from any location.
 - *CollocationDemo.m*
Runs *Collocation.m* on test functions and a first order ODE.

B.4 Test functions

The ten test functions in *TestIntegrals.m* are shown below. The first six are from [11].

OSCILLATORY	$f(\mathbf{x}) = \cos(2\pi u_1 + \sum_{i=1}^d a_i x_i)$
CORNER PEAK	$f(\mathbf{x}) = (1 + \sum_{i=1}^d a_i x_i)^{-(d+1)}$
PRODUCT PEAK	$f(\mathbf{x}) = \prod_{i=1}^d (a_i^{-2} + (x_i - u_i)^2)^{-1}$
GAUSSIAN	$f(\mathbf{x}) = \exp(-\sum_{i=1}^d a_i^2 (x_i - u_i)^2)$
C^0	$f(\mathbf{x}) = \exp(-\sum_{i=1}^d a_i x_i - u_i)$
DISCONTINUOUS	$f(\mathbf{x}) = \begin{cases} 0 & x_1 > u_1 \text{ or } x_2 > u_2 \\ \exp(\sum_{i=1}^d a_i x_i) & \text{otherwise} \end{cases}$
UNCOUPLED	$f(\mathbf{x}) = \sum_{i=1}^d \exp(a_i (x_i - u_i))$
MAX PRODUCT	$f(\mathbf{x}) = \prod_{i=1}^d \max(x_i - u_i + 1, 1)$
ODD C^0	$f(\mathbf{x}) = \exp\left(-\sum_{i \text{ odd}} (a_i x_i - u_i)\right) \exp\left(-\sum_{i \text{ even}} (a_i (x_i - u_i))\right)$
ODD DISCONTINUOUS	$f(\mathbf{x}) = \begin{cases} 0 & \exists i \text{ odd s.t. } x_i > u_i \\ \exp(\sum_{i=1}^d a_i x_i) & \text{otherwise} \end{cases}$

***a** and **u** are tunable parameters.

All entries of **a** are positive and **u** is randomly generated. Run the program *Plots_D2.m* to see the two dimensional instantiation of each function.

B.5 Running *CollocationDemo.m*

The program *CollocationDemo.m* runs the package on some examples. Set *type* = 0 to integrate and interpolate the test functions above. Set *type* = 1 to evaluate the expected value of the solution to the ODE.

$$\dot{x} = \mu_1 x \quad x(0) = \mu_2,$$

where μ_1 and μ_2 are uniformly distributed random variables on the interval $[0, 1]$. The solution to the ODE is $x = \mu_2 e^{-\mu_1 t}$. This is treated as a vector valued function in which each entry is a point in time. Notice that the expected value (what is being calculated)

$$x = \frac{1}{2t}(1 - e^{-t})$$

is not a solution to the ODE. You'll see that the dimension adaptive algorithm requires a much smaller adaptive tolerance to achieve a good error for the variance; this is because the integrand being adapted to is the mean and not the variance.

We evaluate the electric ship in the same fashion, except that rather than having a closed form solution to evaluate, a simulation must be run at each point in the integral's domain.

For sensitivity analysis we look at the variance solution, and either consider the maximum of the variance through time or the integral of the variance through time.

Histograms of interpolated points are generated as well; the usefulness of this feature is more apparent in the time dependent case, where straight Monte Carlo simulation would be an overly expensive means of finding a PDF at later times.

Bibliography

- [1] M. Athans, R. P. Wishner, and A. Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, 13:504–518, October 1968.
- [2] J. Berrut and L. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004.
- [3] Paul Bratley and Bennett L. Fox. Algorithm 659: implementing Sobol’s quasirandom sequence generator. *ACM Trans. Math. Softw.*, 14(1):88–100, 1988.
- [4] C. W. Clenshaw and A. R. Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, (2):197–205, 1960.
- [5] K. M. Cuomo and A. V. Oppenheim. Circuit implementation of synchronized chaos with applications to communications. *Physical Review Letters*, 71:65–68, July 1993.
- [6] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, New York, 1975.
- [7] Philip J. Davis. *Interpolation and approximation*. Dover, New York, 1975.
- [8] L. N. Domaschk, A. Ouroua, R. E. Hebner, O. E. Bowlin, and W. B. Colson. Coordination of large pulsed loads on future electric ships. *Magnetics, IEEE Transactions on*, 43(1):450–455, Jan. 2007.

- [9] Zivi E. and McCoy T.J. Control of a shipboard integrated power system. In *Proceedings of the 33th Annual Conference on Information Sciences and Systems*, March 1999.
- [10] S. Elhay and J. Kautsky. Generalized Kronrod-Patterson type imbedded quadratures. *Aplikace Matematiky*, 37(2):81–103, 1992.
- [11] A. Genz. A package for testing multiple integration subroutines. In *Numerical Integration*, pages 337–340. Kluwer Academic Publishers, 1987.
- [12] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.
- [13] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71(1):65–87, August 2003.
- [14] Roger G. Ghanem and Pol D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag New York, Inc., New York, NY, USA, 1991.
- [15] Gene H. Golub and John H. Welsch. Calculation of gauss quadrature rules. Technical report, Stanford, CA, USA, 1967.
- [16] D. Guo and X. Wang. Quasi-Monte Carlo filtering in nonlinear dynamic systems. *IEEE Transactions on Signal Processing*, 54(6):2087–2098, June 2006.
- [17] J.R. Hockenberry and B.C. Lesieutre. Evaluation of uncertainty in dynamic simulations of power system models: The probabilistic collocation method. *Power Systems, IEEE Transactions on*, 19(3):1483–1491, Aug. 2004.
- [18] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, May 2000.
- [19] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the non-linear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.

- [20] George Em Karniadakis and Spencer J. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, New York, 1999.
- [21] Andreas Klimke. *Uncertainty modeling using fuzzy arithmetic and sparse grids*. PhD thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2006.
- [22] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations (Stochastic Modelling and Applied Probability)*. Springer, November 2000.
- [23] PC Krause and Associates. Power system control development, final report, contract f33615-99-d-2974 for nsf/onr partnership in electric power network efficiency and security (epnes)). Technical report, March 2003.
- [24] R. Kuffel, J. Giesbrecht, T. Maguire, R.P. Wierckx, and P. McLaren. Rtds - a fully digital power system simulator operating in real time. *Digital Power System Simulators, 1995, ICDS '95., First International Conference on*, pages 19–, 5-7 Apr 1995.
- [25] J. Langston, S. Suryanarayanan, M. Steurer, M. Andrus, S. Woodruff, and P.F. Ribeiro. Experiences with the simulation of a notional all-electric ship integrated power system on a large-scale high-speed electromagnetic transient simulator. *Power Engineering Society General Meeting, 2006. IEEE*, pages 5 pp.–, 18-22 June 2006.
- [26] J. Langston, J. Taylor, F. Hover, J. Simpson, M. Steurer, and T. Baldwin. Uncertainty analysis for a large-scale transient simulation of a notional all-electric ship pulse load charging scenario. In *Tenth International Conference on Probabilistic Methods Applied to Power Systems*, Rincon, Puerto Rico, May 2008.
- [27] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

- [28] E. Novak and K. Ritter. High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, (75):79–97, 1996.
- [29] Bernt Oksendal. *Stochastic differential equations (3rd ed.): an introduction with applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.
- [30] Steven A. Orszag and L. R. Bissonnette. Dynamical properties of truncated Wiener-Hermite expansions. *Physics of Fluids*, 10(12):2603–2613, 1967.
- [31] T. N. L. Patterson. The optimum addition of points to quadrature formulae. *Mathematics of Computation*, 22(3):847–856, 1968.
- [32] Krause P.C., Wasynczuk O., and Sudhoff S.D. *Analysis of Electric Machinery and Drive Systems*. IEEE Press and Wiley-Interscience, NY, second edition, 2002.
- [33] K. Petras. gmp_(x)sc, extension of the GNU MP package. http://www-public.tu-bs.de:8080/~petras/software/gmp_xsc/gmp_xsc.html.
- [34] K. Petras. Smolyak cubature of given polynomial degree with few nodes for increasing dimension. *Numerische Mathematik*, (93):729–753, 2003.
- [35] Pradya Prempraneerach. *Uncertainty Analysis in a Shipboard Integrated Power System using Multi-Element Polynomial Chaos*. PhD thesis, Massachusetts Institute of Technology, 2007.
- [36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.
- [37] Jean-Jacques E. Slotine and Haruhiko Asada. *Robot Analysis and Control*. John Wiley & Sons, Inc., New York, NY, USA, 1992.
- [38] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 148:1042–1043, 1963. Russian, Engl. Transl.: Soviet Math. Dokl. 4:240–243, 1963.

- [39] I. M. Sobol. Uniformly distributed sequences with an additional uniform property. *USSR Comput. Math. Math. Phys.*, 16:236–242, 1976.
- [40] M. Steurer, M. Andrus, J. Langston, L. Qi, S. Suryanarayanan, S. Woodruff, and P.F. Ribeiro. Investigating the impact of pulsed power charging demands on shipboard power quality. *Electric Ship Technologies Symposium, 2007. ESTS '07. IEEE*, pages 315–321, 21-23 May 2007.
- [41] L. N. Trefethen. Is Gauss quadrature better than Clenshaw-Curtis? *SIAM Review*, 2007.
- [42] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.*, 209(2):617–642, 2005.
- [43] G. W. Wasilkowski and H. Woźniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11(1):1–56, 1995.
- [44] C. O. Weiss and J. Brock. Evidence for lorenz-type chaos in a laser. *Phys. Rev. Lett.*, 57(22):2804–2806, Dec 1986.
- [45] Dongbin Xiu and Jan S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.