NORTHWESTERN UNIVERSITY

Rare Event Simulation for Lightwave Systems Using the Cross-Entropy Method

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Applied Mathematics

By

Graham Mauch Donovan

EVANSTON, ILLINOIS

June 2008

UMI Number: 3303639

UMI Microform 3303639

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

Rare Event Simulation for Lightwave Systems

Using the Cross-Entropy Method

Graham Mauch Donovan

Rare events are studied in an increasing number of areas, ranging from lightwave and optical communication systems, to industrial routing problems, to rogue ocean waves, to financial asset pricing, to the rare failure of something as common as Gaussian elimination, as well as numerous others. In optical communication systems, a per-bit error rate of one part in 10^{10} quickly becomes relevant because of high data rates, currently 40 Gb/s or greater. Minimizing error rates at a reasonable cost and understanding sources of errors are important aspects of these practical engineering problems.

In this thesis a new method is described which allows the use of multiple importance-sampled Monte Carlo simulations for these systems. The key step is to provide a numerical algorithm for the determination of the biasing distributions, modified probability distributions under which rare events in the system are no longer rare. This new method makes use of a stochastic optimization scheme known as the cross-entropy (CE) method to solve an

optimization problem for these distributions, and the singular value decomposition (SVD), linear operator decomposition, to efficiently compute the important directions in the system, the system modes. The details of the SVD-CE-IS method are presented in a more general context (it is not constrained to optical systems), and then it is applied to specific optical communication systems.

A further application of this new SVD-CE-IS method is also demonstrated: it can be used as a performance probe, targeting the behavior of the simulated system in regions of specific interest. In these regions, system characteristics can be examined in greater detail to help explain the reasons for the type of performance being considered. For example, simulation trials which generate particular types of errors can be examined to determine the underlying root causes. This capability is another desirable feature of the new method.

Acknowledgements

Bill Kath, for advising this thesis. Elaine Spiller, for many useful discussions and helping me to get started in this area in the first place. Jinglai Li, for his insights.

Personally, the people who made life fun even when work was not: Claire Postlethwaite, Bryan & Jane Smith. My officemates and roommates: Vilas Menon, Joe Hibdon, Joanna Bieri, Nick Poole.

To my parents.

Table of Contents

List of Figures

CHAPTER 1

Introduction

Rare events are studied mathematically in many widely varying contexts, from lightwave and optical communication systems [1–7], to industrial routing problems [8], to rogue ocean waves [9], to financial asset pricing [10], and even the rare failure of something as common as Gaussian elimination [11]. Many different methods have been deployed in the search for rare events. The most straightforward simulation approach, known as *Monte Carlo simulation*, is computationally infeasible for rare events [12], simply because of the number of trials required to generate even one such event. For example, accurately simulating a rare event which occurs with probability 10^{-10} requires more than 10^{10} simple Monte Carlo trials, and in most cases this is impractical.

A number of more sophisticated techniques known as *variance reduction techniques* have been developed and used to alleviate this difficulty [13]. These techniques decrease the simulation variance so that the number of trials required to simulate rare events is greatly reduced, and thus doing so becomes computationally feasible. Within the spectrum of available variance reduction techniques, there are many techniques with various benefits and drawbacks for different types of problems. Generally speaking, *dimensionality* — that is, the number of noise dimensions — is a key complicating factor [14]. Simple techniques which work well on low-dimensional problems are ineffective in higher dimensions, and effective variance reduction schemes for high-dimensional problems often must be much more sophisticated.

To deal with some of the problems associated with high-dimensional systems, here a new method for rare event simulation is proposed that combines Monte Carlo simulations and importance sampling [15] with the application of the singular value decomposition

(SVD) [16] and the cross-entropy (CE) method [17–19]. It is the latter two elements that provide the main benefit here. Because the SVD is designed to determine the perturbations that produce the biggest outputs, the SVD allows one to reduce the effective dimension of the problem. The CE method assesses the probabilistic significance of each of these perturbations. This new technique has the benefit that it places few restrictions on the problems to which it can be applied, while it is still capable of providing physical insight into the inner workings of the simulated system and the nature of the rare events under investigation. In particular, we will apply this technique to the simulation of several lightwave communication systems. This technique does not make use of any particular mathematical structure of the lightwave systems involved, and thus the technique should be easily generalized to a much wider class of problems.

In the past, importance sampling (IS) has been used as a variance reduction technique with great effect to speed simulations of communication systems [14]. However, the application of IS has generally been restricted to problems where sufficient mathematical structure was available to allow essential quantities known as biasing distributions to be estimated via a reduced problem [1, 5, 20, 21]. We propose a new method which can generate the biasing distributions numerically without need for recourse to an analytically reduced problem. In particular, we use the cross-entropy method along with the singular value decomposition to solve an optimization problem for the biasing distributions. With these biasing distributions determined, importance sampling can then be employed to speed up Monte Carlo simulations and provide computationally efficient simulation of rare events in lightwave systems. We will demonstrate this method for two sample problems: one for

which rare-event simulations have been previously obtained with analytically-determined biasing distributions so that we can validate our results, and another which appears too difficult to simulate by such methods.

This dissertation is organized as follows. First, the mathematical details are described, including the Monte Carlo method, importance sampling, multiple importance sampling, and the cross entropy method. Although not strictly speaking a simulation technique, the details of the singular value decomposition are contained here as well, as it is essential to the new method we will be presenting.

We then discuss the background information about optical communications systems required for this work: modeling with the nonlinear Schrödinger equation, the improvement of system performance using dispersion management, the degrading effects of amplifier noise, and the special mathematical structure of optical pulses known as solitons. Because much of the work we will be discussing has no analytical solution and must be performed numerically, we devote a chapter to the relevant numerical methods and issues involved in accurately and efficiently simulating these optical systems numerically, including issues associated with introducing numerical noise.

With this background information in place, we proceed to lay out the proposed new method and demonstrate it for example optical communications systems. The details of the new method are presented first for a general case, one not necessarily an optical communication system. This scheme is then made specific to a general lightwave communication system, and then finally to the particular example systems for which we demonstrate results.

The results of rare-event simulation for the two example systems are then presented, along with comparison with the previously published results for one of the problems, which is approachable via an alternative technique. The second example problem is a modification of the first for which the previous technique is inapplicable. We then demonstrate the utility of our new method as a probe to determine the underlying differences between these two systems. Thus, this new method has utility as a tool to determine the underlying causes of rare events affecting system performance.

Finally, we discuss the potential application of this new method to rare event simulation problems in areas beyond lightwave communication systems. Though demonstration of the technique to these proposed areas is beyond the scope of this dissertation, we believe it can potentially be applied to problems in many other areas.

CHAPTER 2

Mathematics of Simulation

2.1. The Monte Carlo method & importance sampling

Monte Carlo methods are a broad class of computational methods which evaluate an unknown quantity using random sampling. The value of the unknown quantity is evaluated with random trials within a specified sample space, and by collecting statistics on the results of these random samples the underlying unknown quantity can be estimated. A simple unknown quantity to consider for conceptual purposes is nothing more than a one-dimensional definite integral, $\int_V \Phi(x)dx$ on some domain V. The domain is unimportant for purposes of our discussion here, but the quantity being evaluated remains a definite integral.

Consider first the problem of estimating some quantity

$$\ell = \int_V \phi(x) f(x) dx \tag{2.1.1}$$

where the integrand is now written as the product of two functions. In this form, $\phi(x)$ is the function whose expectation with respect to some probability distribution function (PDF) $f(x)$ is to be evaluated. Traditional Monte Carlo integration would determine this quantity by using the estimator

$$\ell_N = \mathbb{E}_f[\ell] = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) \tag{2.1.2}$$

where the x_i are random samples drawn from the PDF $f(\cdot)$. The drawback to traditional Monte Carlo simulation is that if $\phi(x)$ is large when $f(x)$ is small, the variance of the estimate can be unacceptably large, when compared with the number of samples required,

and hence the computational cost is high. While it is true that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \phi(x_i) = \int_V \phi(x) f(x) dx, \tag{2.1.3}$$

i.e., the estimator converges to the desired value by the law of large numbers, it is also true that the variance of our estimator is given by

$$\mathrm{var}[\ell_N] = \frac{1}{N}\mathrm{var}[\phi]. \tag{2.1.4}$$

The convergence rate of the standard deviation of ℓ_N is thus $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$. Consider Chebyshev's inequality [22]

$$\mathbb{P}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\phi(x_i) - \langle\phi(x)\rangle\right)^2 \geq \frac{\delta^{-1}}{N}\mathrm{var}[\phi(x)]\right] \leq \delta$$

which bounds the error between the estimate $\ell_N = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)$ and the exact solution $\ell = \langle\phi(x)\rangle$ at $\frac{\delta^{-1}}{N}\mathrm{var}\{\phi(x)\}$ with probability $1-\delta$. This is the result previously mentioned; to simulate rare events which occur with probability 10^{-10}, we require an accuracy (at least) better than 10^{-10}. Thus we require

$$N \geq \frac{\delta^{-1}\mathrm{var}[\phi(x)]}{(\mathrm{error})^2} = \frac{\left(\frac{1}{100}\right)^{-1}\mathrm{var}[\phi(x)]}{(10^{-10})^2} = \mathrm{var}[\phi(x)]\times 10^{22}$$

trials for a 99% confidence. For many problems, this convergence is simply too slow to achieve good results within a reasonable time. For example, if we are interested in a

binomial random variable, then $\mathrm{var}[\ell_N] \approx \frac{\ell}{N}$ and the relative standard error is

$$\frac{\sqrt{\mathrm{var}[\ell_N]}}{\ell} \approx \frac{1}{\sqrt{N\ell}}.$$

Then to obtain an acceptable error, we require $N \gg 1/\ell$ samples [23].

Importance sampling (IS) is a variance reduction technique that can be employed along with Monte Carlo simulations to gain a dramatic performance improvement over standard Monte Carlo simulations [15]. The idea behind IS is to concentrate the random samples in the areas of sample space that are most important for the quantity being estimated, and then make the appropriate corrections to the statistics. To reduce the variance of this estimator, we introduce the *biasing distribution* $f^*(\cdot)$ and estimate the value of Eq. (2.1.1) using

$$\ell_N = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)\frac{f(x_i)}{f^*(x_i)} = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)w(x_i) = \mathbb{E}_{f^*}[\ell] \tag{2.1.5}$$

so that now we are evaluating the quantity $\phi(x)w(x)$ with samples drawn from $f^*(\cdot)$. Here

$$w(x) = \frac{f(x)}{f^*(x)} \tag{2.1.6}$$

is known as the *likelihood ratio* [12]. The likelihood ratio can be viewed as a correction factor, appropriately weighting a sample trial *as if it had been drawn from the original distribution*, despite having been drawn from a biased distribution.

In principle, the optimal biasing distribution $f^*(\cdot)$ can be calculated by minimizing the variance of our estimator, given by

$$\text{var}\left[\phi(x)\frac{f(x)}{f^*(x)}\right] = \int_V \left(\phi(x)\frac{f(x)}{f^*(x)} - \ell\right)^2 f^*(x)dx. \tag{2.1.7}$$

Clearly, variance minimization occurs when

$$\begin{aligned} \phi(x)\frac{f(x)}{f^*(x)} - \ell &= 0 \\ \rightarrow f^*(x) &= \frac{\phi(x)f(x)}{\ell}. \end{aligned} \tag{2.1.8}$$

In fact, if $f^*(\cdot)$ is given by Eq. (2.1.8), then the variance of our estimator is zero! This is, of course, infeasible, as Eq. (2.1.8) depends on the quantity being estimated, ℓ. The goal of importance sampling, then, is to find $f^*(\cdot)$ such that our estimator is as near as possible to the zero-variance estimator, without having to know the answer *a priori*. There are a number of restrictions and rules of thumb concerning good choices of biasing distributions [24], and in principal choosing a poor biasing distribution can in fact make the simulation results much worse than the case of unbiased trials. In this work we primarily restrict ourselves to mean-shifted distributions but otherwise identical to those of the underlying model, which alleviates many of these difficulties [20].

For simplicity, up to this point we have considered only one distribution and one random variable. If, instead, our problem contains n independent random variables given by x_i, $i = 1 \ldots n$, drawn from distributions $f_i(\cdot)$ and with corresponding biasing distributions

$f_i^*(\cdot)$, then the likelihood ratio is given by

$$w(\vec{x}) = \frac{f(\vec{x})}{f^*(\vec{x})} = \prod_{i=1}^{n} \frac{f_i(x_i)}{f_i^*(x_i)}. \tag{2.1.9}$$

2.1.1. Example of Importance Sampled Monte Carlo Integration

An extremely simple example of importance sampling is to consider 1D Monte Carlo integration. Clearly there are better ways of solving this problem, but it serves to visually demonstrate the action of importance sampling. Suppose we wish to evaluate

$$\ell = \int_{-100}^{100} e^{-x^6} f(x) dx \tag{2.1.10}$$

where we take $f(x)$ to be a uniform distribution on $[-100, 100]$. Thus

$$f(x) = \begin{cases} \frac{1}{200}, & |x| \leq 100 \\ 0, & |x| > 100 \end{cases} \tag{2.1.11}$$

and

$$\ell = \frac{1}{200} \int_{-100}^{100} e^{-x^6} dx \approx 0.00928 \tag{2.1.12}$$

Knowing as we do the structure of e^{-x^6}, we can take as our biasing distribution, for example, the normal distribution with mean 0 and variance 1:

$$f^*(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}. \tag{2.1.13}$$

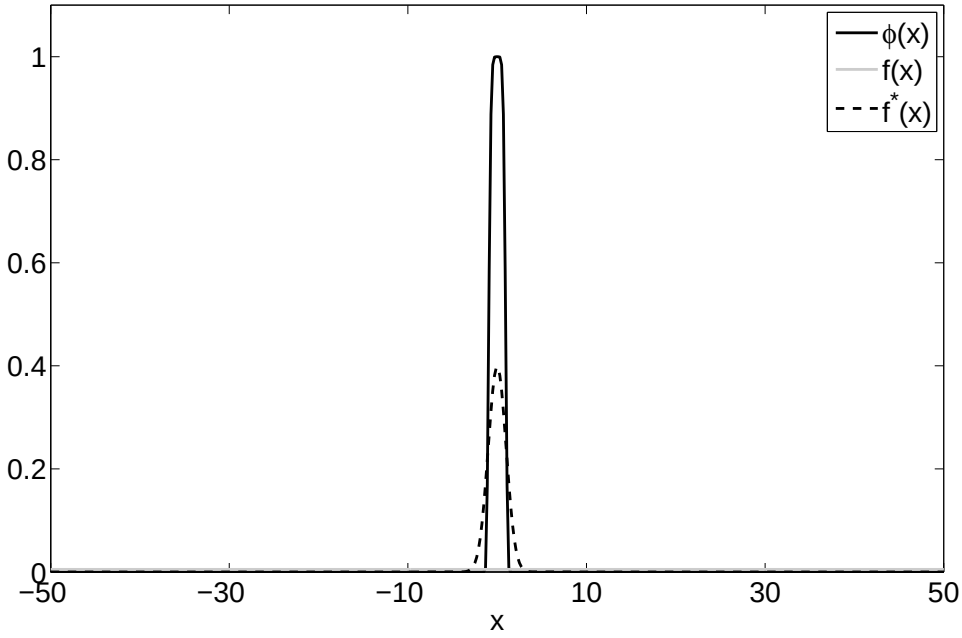

Figure 2.1. **Function for Monte Carlo integration with sampling distributions**: The function to be integrated via Monte Carlo integration $\phi(x)$, along with the original distribution $f(x)$ and the biasing distribution $f^*(x)$

It is easy to see from Fig. 2.1 how the biasing distribution better represents the portion of e^{-x^6} which contributes to the integral than the initial, uniform distribution $f(x)$.

Of course, there are much better biasing distributions that could be used, but even this very rough approximation produces a dramatic increase in simulation performance. In Fig. 2.2, the mean and variance of each estimator is plotted against the number of samples. It is easy to see that the importance-sampled estimator converges to the known solution much more quickly than the crude Monte Carlo estimator. For any given variance level, several orders of magnitude fewer samples are required when using the IS estimator.

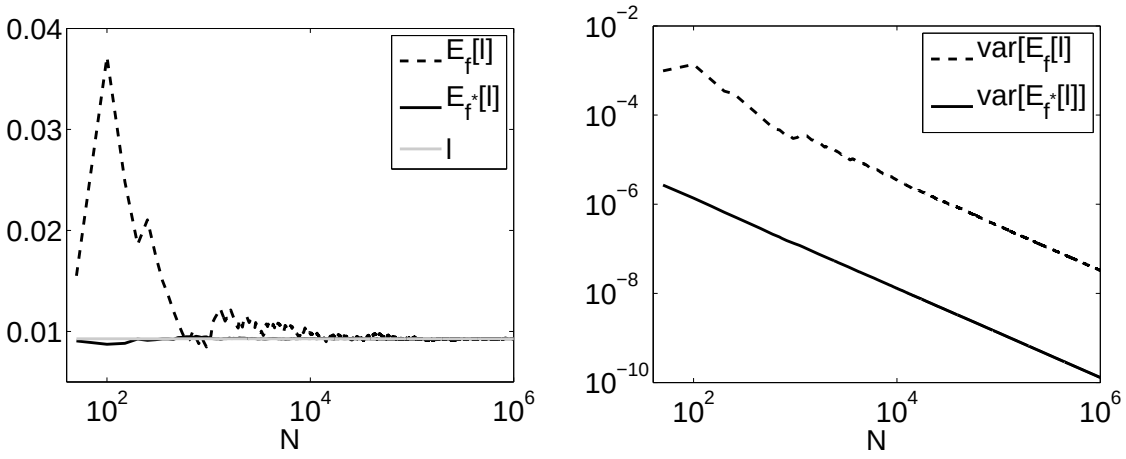

Figure 2.2. **Performance increase in Monte Carlo integration obtained with importance sampling**: Left, convergence of estimated values to true solution as a function of sample size. Right, variance of estimate vs. sample size.

2.1.2. Multiple importance sampling

For some problems, a single biasing distribution may be insufficient to capture all of the desired information. Consider, somewhat trivially, simply performing Monte Carlo integration of a function with disjoint but important regions which contribute to the integral, for example

$$\phi(x) = (x - 100,000)^4 + (x + 100,000)^4. \tag{2.1.14}$$

This is a highly exaggerated, simple problem, but it illustrates the dilemma nicely. One could perform importance-sampled Monte Carlo integration of this problem using a multimodal distribution which captures the character of $\phi(x)$, but in general sampling from

an arbitrary distribution may be difficult. It can be easier to use two simple, uni-modal distributions from which samples are easily generated, centered at each of the relevant regions, and combine the results.

The problems we will be considering can have similar character, though in much higher dimensional spaces. In such cases, we make use of multiple importance sampling. Multiple importance sampling allows the use of several biasing distributions or biasing targets. The Monte Carlo trials require only trivial modification - trials are drawn in prescribed numbers from each biasing distribution. Appropriately weighting and combining the results from multiple biasing distributions requires a bit more work. If we now have M biasing distributions, our estimator then becomes

$$\ell = \sum_{j=1}^{M} \frac{1}{n_j} \sum_{i=1}^{n_j} W_j(\vec{x}_{j,i}) \phi(\vec{x}_{j,i}) w(\vec{x}_{j,i}) \tag{2.1.15}$$

where n_j trials are drawn from the j-th biasing distribution, W_j is a weighting function, and w is given by Eq. (2.1.9). The difficulty lies in determining the appropriate weighting function to be used for combining the results of the different biasing distributions. In what follows, we will use what is known as the *balance heuristic*.

The balance heuristic, developed in [25], uses the weighting function

$$W_j(\vec{x}_{j,i}) = \frac{n_j f_j^*(\vec{x}_{j,i})}{\sum_{k=1}^{M} n_k f_k^*(\vec{x}_{j,i})}. \tag{2.1.16}$$

The balance heuristic is so named because weights samples from distributions according to the number of samples expected to be produced by them. Hence the sample values are

balanced across all biasing distributions. Samples are weighted according to the number that each biasing distribution is expected to generate at each point; distributions that are expected to generate more samples are weighted more heavily. When it is used as the weighting function, the sample value in our estimator does not depend on j or the underlying distribution, for a given $\vec{x}$. By substituting Eq.(2.1.16) into Eq. (2.1.15) and recalling Eq. (2.1.6), one can see that $\frac{1}{n_j} W_j(\vec{x}_{j,i}) \phi(\vec{x}_{j,i}) w(\vec{x}_{j,i})$ becomes independent of j for fixed $\vec{x}$.

In fact, it can be shown that no other weighting function performs significantly better than the balance heuristic, with performance measured in terms of the variance of the estimator. The "variance gap" between the estimator with optimal weighting functions and the estimator with balance heuristic weighting functions is bounded by

$$\left(1 - \frac{1}{M}\right) \frac{1}{n} \ell^2 \tag{2.1.17}$$

for M biasing distributions with n samples drawn from each, with the gap $\to 0$ as $n \to \infty$. The scope of the proof is beyond this dissertation, but the details are contained in [26].

2.1.3. Coefficient of variation and quality of simulated PDFs

We have now provided the necessary detail to simulate PDFs via multiple importance-sampled Monte Carlo simulations, but we have not discussed any method of measuring the quality of these PDFs. While good importance sampling distributions provide variance reduction and rapid convergence to a low-variance estimate, poor biasing distributions can

result in few samples being generated in the region of interest and thus a high-variance estimator. To evaluate the quality of our biasing distributions and the resulting variance of our PDF estimators, we make use of the *coefficient of variation*

$$c_v = \frac{\sigma}{\mu} \tag{2.1.18}$$

which is a dimensionless measure of the standard deviation relative to the mean. We do not use the coefficient of variation as a measure of the simulated distribution, which has no bearing on the quality of the simulation (that is, we are not interested in $\sqrt{\text{var}[\phi(x)]}/\langle\phi(x)\rangle$), but rather on the *bin-by-bin* sample distribution of the histogram. Each Monte Carlo sample can be expressed as performance and estimated probability pairs

$$[\phi(\vec{x}_{i,j}), W(\vec{x}_{i,j})w(\vec{x}_{i,j})]_i \,. \tag{2.1.19}$$

and, these samples are binned into a histogram by the value of ϕ. The k^{th} histogram bin contains performances confined to the interval $\phi_k \pm \frac{\delta\phi}{2}$, while the $W(\vec{x}_{i,j})w(\vec{x}_{i,j})$ values are unconstrained. The coefficient of variation we are interested in is the c_v of this distribution, the estimated probabilities within each bin. A high c_v then indicates wide intra-bin variation relative to the estimated probabilities, and poor convergence of that bin (and thus the PDF). Conversely, a low c_v indicates a tightly clustered intra-bin distribution, relative to the estimated probabilities, and thus good convergence of the histogram in that bin. Coefficients of variation are measured in this way throughout this work to assess

the quality of the biasing distributions. Collecting the coefficient of variation in multiply importance sampled simulations requires some modifications; see [20] for details.

2.2. The cross-entropy method

The cross-entropy (CE) method [17] is an optimization algorithm which we will use to solve an optimization problem for the biasing distributions. In fact, the derivation of the CE method proceeds directly from the premise of finding the optimal importance sampling biasing distribution. Here we follow the derivation of [19]. Consider a random vector $\vec{x} = [x_1, \ldots, x_N]$ drawn from a PDF $f(\vec{x})$ and a real-valued *performance function* $p(\vec{x})$. Then we define

$$\ell = \mathbb{P}[p(\vec{x}) \geq a] = \mathbb{E}_f[I_{\{p(\vec{x})\geq a\}}]$$

as the probability that the performance is greater than some set value a, where $I_{\{p(\vec{x})\geq a\}}$ is the indicator function given by

$$I_{\{p(\vec{x})\geq a\}} = \begin{cases} 1, & p(x) \geq a \\ 0, & p(x) < a \end{cases}$$

and $\mathbb{E}_f$ indicates that the $\vec{x}$ are drawn from the PDF $f(\vec{x})$. If ℓ is sufficiently small, we consider $p(\vec{x}) \geq a$ to be a *rare event*. Of course, as before, one might consider a Monte Carlo simulation to estimate ℓ. The crude Monte Carlo estimator is given by

$$\frac{1}{N}\sum_{i=1}^{N} I_{\{p(\vec{x}_i)\geq a\}}$$

with $\vec{x}_i$, $i = 1 \ldots N$ drawn from $f(x)$. When $p(x) \geq a$ is a rare event, however, this is likely to be a poor estimator. In fact, when

$$N \ll 1/\ell$$

it is likely that the rare event does not occur at all and our estimate is zero, and to obtain a low-variance estimate a very large number of samples is likely required. We can dramatically improve the quality of the estimator by introducing a biasing distribution $g(x)$ and using the importance sampled estimator

$$\frac{1}{N}\sum_{i=1}^{N} I_{\{p(\vec{x}_i)\geq a\}} \frac{f(\vec{x}_i)}{g(\vec{x}_i)} \tag{2.2.1}$$

where the x_i are now drawn from the biasing distribution $g(x)$. Of course, as before, the optimal, but impractical, biasing distribution is given by

$$g^*(x) = \frac{I_{\{p(\vec{x})\geq a\}} f(x)}{\ell} \tag{2.2.2}$$

and results in a zero-variance estimator – only $N = 1$ sample is required in Eq. (2.2.1) to return the true result, ℓ. Of course, we cannot actually use $g^*(x)$ because it depends on the quantity we are trying to estimate, ℓ.

The central idea behind the cross-entropy method is to make use of the *Kullback–Leibler* (KL) distance [19] to estimate this optimal biasing distribution $g^*(x)$. The KL

distance between two distributions $h_1(x)$ and $h_2(x)$ is given by

$$\mathcal{D}(h_1, h_2) = \mathbb{E}_{h_1}\left[\ln \frac{h_1(x)}{h_2(x)}\right] = \int h_1(x) \ln h_1(x) dx - \int h_1(x) \ln h_2(x) dx. \tag{2.2.3}$$

Note that the term *distance* in KL distance is a slight misnomer in that $\mathcal{D}$ is not a true distance in the formal sense. For one thing,

$$\mathcal{D}(h_1, h_2) \neq \mathcal{D}(h_2, h_1).$$

Regardless, the KL distance is a very useful quantity for our purposes. Consider the KL distance between our optimal biasing distribution $g^*(x)$ and another, as of yet unknown distribution $\bar{g}(x) \in \mathcal{G}$

$$\mathcal{D}(g^*, \bar{g}) = \int g^*(x) \ln g^*(x) dx - \int g^*(x) \ln \bar{g}(x) dx. \tag{2.2.4}$$

Assumptions about the family of distributions $\mathcal{G}$ will be important and discussed shorty. Consider first the minimization problem

$$\min_{\bar{g} \in \mathcal{G}} \mathcal{D}(g^*, \bar{g}). \tag{2.2.5}$$

Because the first term on the right hand side of Eq. (2.2.4) does not depend on $\bar{g}(x)$, Eq. (2.2.5) is equivalent to

$$\min_{\bar{g} \in \mathcal{G}} \left(- \int \ln \bar{g}(x) g^*(x) dx \right) \tag{2.2.6}$$

which is trivially equivalent to

$$\max_{\bar{g}\in\mathcal{G}} \int \ln \bar{g}(x) g^*(x) dx. \tag{2.2.7}$$

Substituting for $g^*(x)$, we obtain

$$\max_{\bar{g}\in\mathcal{G}} \int \frac{I_{\{p(\vec{x})\geq a\}} f(x)}{\ell} \ln \bar{g}(x) dx \tag{2.2.8}$$

and by passing to the probabilistic equivalent we obtain

$$\max_{\bar{g}\in\mathcal{G}} \mathbb{E}_f \left[I_{\{p(x)\geq a\}} \ln \bar{g}(x) \right]. \tag{2.2.9}$$

Although this is in some sense already an importance sampled Monte Carlo estimator, there is nothing preventing us from applying importance sampling to it a second time, and obtaining

$$\max_{\bar{g}\in\mathcal{G}} \mathbb{E}_h \left[I_{\{p(x)\geq a\}} w(x) \ln \bar{g}(x) \right] \tag{2.2.10}$$

where $w(x)$ is the likelihood ratio

$$w(x) = \frac{f(x)}{h(x)} \tag{2.2.11}$$

and samples are now being drawn from the distribution $h(x)$. The solution to this program for $\bar{g}(x) \in \mathcal{G}$ is then given by

$$\bar{g}^*(x) = \underset{\bar{g}\in\mathcal{G}}{\operatorname{argmax}} \left\{ \mathbb{E}_h \left[I_{\{p(x)\geq a\}} w(x) \ln \bar{g}(x) \right] \right\} \tag{2.2.12}$$

where argmax denotes the argument of the maximum, i.e. the $\bar{g}$ within $\mathcal{G}$ which maximizes the given expression. The result of this program is the distribution $\bar{g}^*(x)$ within the family of distributions $\mathcal{G}$ which has the minimum KL distance to the optimal importance sampling distribution $g^*(x)$. Unlike obtaining $g^*(x)$, which depends on the quantity we are after, it is feasible to get $\bar{g}^*(x)$.

Under certain assumptions, it may be possible to solve the maximization problem Eq. (2.2.12) analytically. For example, see [18] for solutions when $\mathcal{G}$ is restricted to a natural exponential family. However, we will follow a somewhat more general approach of and assume only that $\mathcal{G}$ is restricted to distributions with *product form*. A distribution $g(x)$ has product form if it can be written $g(x) = \prod_{i=1}^{n} g_i(x_i)$ [19]. Using this mild assumption, we can estimate the moments of the product form distribution which solve the stochastic program Eq. (2.2.12) as follows.

In particular, we can estimate the n^{th} moment of the sample distribution drawn from $\bar{g}^*(x)$ for samples where $p(x) \geq a$ as

$$\begin{aligned}\mathbb{E}_{\bar{g}^*}[(x)^n | p(x) \geq a] &= \int (x)^n I_{\{p(x)\geq a\}} \bar{g}^*(x) dx \\ &= \frac{\mathbb{E}_f\left[(x)^n I_{\{p(x)\geq a\}}\right]}{\mathbb{E}_f\left[I_{\{p(x)\geq a\}}\right]} \\ &= \frac{\mathbb{E}_{\bar{g}^*}\left[(x)^n I_{\{p(x)\geq a\}} \frac{f(x)}{\bar{g}^*(x)}\right]}{\mathbb{E}_{\bar{g}^*}\left[I_{\{p(x)\geq a\}} \frac{f(x)}{\bar{g}^*(x)}\right]}. \end{aligned} \tag{2.2.13}$$

This formula is obtained by introducing conditional probabilities for the indicator functions; for details of the derivation, see [19]. For Gaussian distributions, as we are using, the distribution is fully specified by the first two moments. In fact, we will be using mean-shifted Gaussians (with fixed variance), and in this case only the mean is necessary. For other distributions, more moments may need to be calculated, but this can be done using the same formula.

There is one additional difficulty to be resolved still: $p(\vec{x}) \geq a$ is still a rare event under $f(\vec{x})$. Eq. (2.2.13) is thus not yet of any practical use - our estimated values are still likely to be zero, as no rare events will occur. The solution to this problem is to develop a multi-stage algorithm, such that we create a sequence of performance values a^k and biasing distributions $g^k(\cdot)$ for $(k = 1, 2, \dots)$ such that

$$p(\vec{x}) \geq a^k \tag{2.2.14}$$

is not a rare event when drawn from $g^k(\cdot)$. The method of achieving this is the sample quantile. By taking the $(1-\rho)$-quantile performance from performance samples at the k-th level of the algorithm as a^k (where ρ is an algorithm parameter for the size of the performance quantile), the moments of $g^{k+1}(x)$ can be estimated via Eq. (2.2.13). This process then repeats until $a^k \geq a$ [19].

The practical setup then is as follows. We begin by defining the *performance function* $p(\cdot)$ for the system at hand. We define $p(\cdot)$ to be the quantity of interest for our rare events. For example, in a lightwave transmission system in which errors are rare, p might be given by the center voltage after detection at the end of the transmission line. Further, we select some value a such that $p(\cdot) > a$ is our desired rare-event condition.

We allow the CE method to control the distributions from which the random variables in the system are drawn, the $g^k(\cdot)$. The CE method then is posed to return these $g^k(\cdot)$ in optimal fashion iteratively until $p(\cdot) > a$ is no longer a rare event.

The algorithm proceeds as follows [19]: k is the iteration number in this algorithm, N is the sample size (per level), f is the initial distribution, g^k is the distribution at the k^{th} level of the algorithm, a is the target value of the performance metric $p(Z, \rho)$, where Z will be random samples, and ρ is a parameter in the algorithm specifying the quantile size. The $(1-\rho)$ quantile of the performances of the Z's is $p(Z, \rho)$. The algorithm will generate the sequence of distributions $g^1, g^2, \ldots, g^N$ such that the performance function being greater than the target value ($p(Z) > a$) is no longer a rare event when samples are drawn from $g^N(\cdot)$ as it was under $g(\cdot)$.

The algorithm then proceeds as follows:

Cross-Entropy Method Algorithm

(1) $k = 1$, set N, $g^0 = f$: Initialize the algorithm.

(2) Draw random samples $Z^1 \ldots Z^N$ from the PDF g^k: Sample from the current distribution.

(3) Let $\hat{p}^k = \min\{a, p_N(Z, \rho)\}$: Compute the performance quantile of our current sample.

(4) Compute g^{k+1}: Calculate the parameters for the next distribution in the sequence. (See details following algorithm.)

(5) If $\hat{p}_k = a$, stop. Return g^k as the output of the algorithm: Here we check to see if the distribution sequence has reached the target.

(6) Else, $k \leftarrow k + 1$, go to step 2

For simplicity, this is the most basic version of the CE method. There are modifications to step 6 to improve the convergence properties of the algorithm. For details, see [19].

Step 4 requires additional explanation. For continuous distributions, computing the entire PDF is computationally infeasible. Luckily, we do not require the full distribution. We consider instead that $g^k(\cdot)$ must belong to some family of distributions parameterized by a vector $\vec{v}$ such that

$$g^k(\cdot) = g^k(\cdot, \vec{v}). \tag{2.2.15}$$

Computing the PDF g^k now requires computing the parameter vector $\vec{v}$, which in turn requires computing moments of the sample distribution according to Eq. (2.2.13). It is

necessary to derive a formula relating the parameter vector to the moments. For example, if we allow $g^k(\cdot, \vec{v})$ to be a mean-shifted Gaussian distribution, then $\vec{v}$ has length 1 (the mean-shifted Gaussian distributions are a one-parameter family) and we need only calculate the mean to determine the new distribution. Then the single-parameter family is

$$g^k(x, \vec{v}) = g^k(x, v) = g^k(x, \mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

and the formula relating the parameter vector and the moments is trivial: $\vec{v} = v = \mu = \mathbb{E}_{\tilde{g}^*}[(x)^1] = \mathbb{E}_{\tilde{g}^*}[x]$. To fit the parameter vector $\vec{v}$ in this case, we need estimate only the mean of the sample distribution. To obtain the value of this parameter, we apply Eq. (2.2.13) (with $n = 1$) directly to the performance quantile samples, independently for both the numerator and the denominator.

In principle, we could use only the CE method to generate the biasing distributions for any system. Unfortunately, the dimension of the space in which we wish to optimize is simply too large to obtain reasonable results from the CE method. Consider that for a lightwave system, we would require two distributions for each computational Fourier mode (real and imaginary), at every point noise is introduced into the system. The dimensionality of the system considered in this way is simply far too large. In Sec. 2.3 we resolve this problem by employing the singular value decomposition to reduce the dimensionality of the optimization problem.

2.2.1. Coin-flipping with the CE method

A simple problem in rare-event simulation is the coin-flipping problem. Consider that we take 100 fair coins and flip them, and count the number of heads. What is the probability of having 80 or more heads? The answer turns out to be, in fact, startlingly small:

$$p_{\geq 80\%} = \sum_{n=0}^{20} (.5)^{80+n} (.5)^{20-n} \begin{pmatrix} 100 \\ 80+n \end{pmatrix} \sim 5.5795 \times 10^{-10} \tag{2.2.16}$$

If we were to attempt to simulate this result via standard Monte Carlo simulations, we would find that our estimator has an unacceptably high variance for a computationally reasonable number of samples. The solution to this problem in the context of importance sampling is equivalent to using an unfair coin - one that comes up heads 80% of the time - and correct the statistics with the likelihood ratio. We will use the cross-entropy method to demonstrate the same result.

For this problem, the probability distributions we consider are discrete distributions with a single parameter we will call w such that the initial distribution f is given by

$$\begin{aligned} f(0) &= 1 - w \\ f(1) &= w \end{aligned} \tag{2.2.17}$$

where $w = 0.5$ for a fair coin and we take 1 to be heads and 0 to be tails.

Of course, it is only necessary to consider a single distribution for one coin and draw from it 100 times. As a slightly more general result, we will use 100 different distributions, one for each coin. This is instructive in the context of the lightwave systems we will

soon be considering – many distributions must be biased independently. Then the $f_i(\cdot)$, $i = 1 \ldots 100$ are given by Eq. (2.2.17).

We begin then by setting $k = 1$, N =100,000, and $g_i^1 = f_i$. We then draw N =100,000 samples from $g_i^1(\cdot)$ and evaluate the performance of each sample by counting the total number of heads. The histogram of this result is given in Fig. 2.3 along with the elite sample histogram, the histogram of the $(1-\rho)$ quantile.

To evaluate our progress, we check the value of the $(1-\rho)$ quantile - in this case, it is 56. (Again, ρ is an algorithm parameter. Here, $\rho = .1$) That is, the next set of distributions in the CE method sequence will be biased as far as a coins that comes up heads an average of 56% of the time. As $56 < 80$, our target value, the algorithm continues. Using the $(1-\rho)$ quantile samples, we compute the distribution parameters w_i by calculating moments of the performance quantile samples by applying Eq. (2.2.13) to the samples in the performance quantile for $n = 1$. For this problem, the mean estimator involves calculating the sample mean of

$$(x_i)^1 I_{\{p(x_i)\geq a\}} \frac{f(x_i)}{g_i^k(x_i)} = \begin{cases} 0, & x_i = 0 \\ I_{\{\left(\sum_{i=1}^{100} x_i\right)\geq a\}}(w_i^0/w_i^k), & x_i = 1 \end{cases}$$

in the numerator, and the sample mean of

$$I_{\{p(x_i)\geq a\}} \frac{f(x_i)}{g_i^k(x_i)} = \begin{cases} I_{\{\left(\sum_{i=1}^{100} x_i\right)\geq a\}}[(1-w_i^0)/(1-w_i^k)], & x_i = 0 \\ I_{\{\left(\sum_{i=1}^{100} x_i\right)\geq a\}}(w_i^0/w_i^k), & x_i = 1 \end{cases}$$

in the denominator. Note this only uses samples in the performance quantile. Here the superscript on w denotes the CE method level — w_i^0 is the initial (level 0) value for the i^{th} distribution parameter. The result of these calculations is used as the mean of the distribution, w_i^{k+1}, at the next level. Note that due to the construction of Eq. (2.2.17), our distributions belong to a single parameter family, so we only need compute the parameters w_i as the mean for each of the 100 distributions. These parameters are then used to fix the 100 biased distributions for the next iteration, the g_i^2. These calculated means w_i^k are returned for each of the 100 distributions and plotted in Fig. 2.4. The process then repeats, and as it does we obtain Fig. 2.3, displaying the evolution of the sample histograms and performance quantiles. The algorithm proceeds in this fashion until the $(1-\rho)$ quantile level is larger than 80. In this case, this occurs at the fifth iteration.

The results of the algorithm can be summarized by the w_i values at each level of the algorithm. Fig. 2.4 plots these w_i values for each iteration. Observe that the final iteration of the algorithm gives a mean value of 0.819, so that the cross-entropy method is indicating that as a biasing function we should use a biased coin which comes up heads 81.9% of the time on average. This is of course the result obtained in this simple case analytically. However, the coin-flipping example problem serves to give an intuitive grasp of the method. Note that the value of the average mean here is unique to the artificial introduction of 100 distributions where one would do – in general, the parameters of each distribution in the method are independent and their average has little meaning.

One additional study which is possible with this simple example problem is checking the convergence of the CE method itself. While estimating the variance of $\vec{v}^k$ directly from a

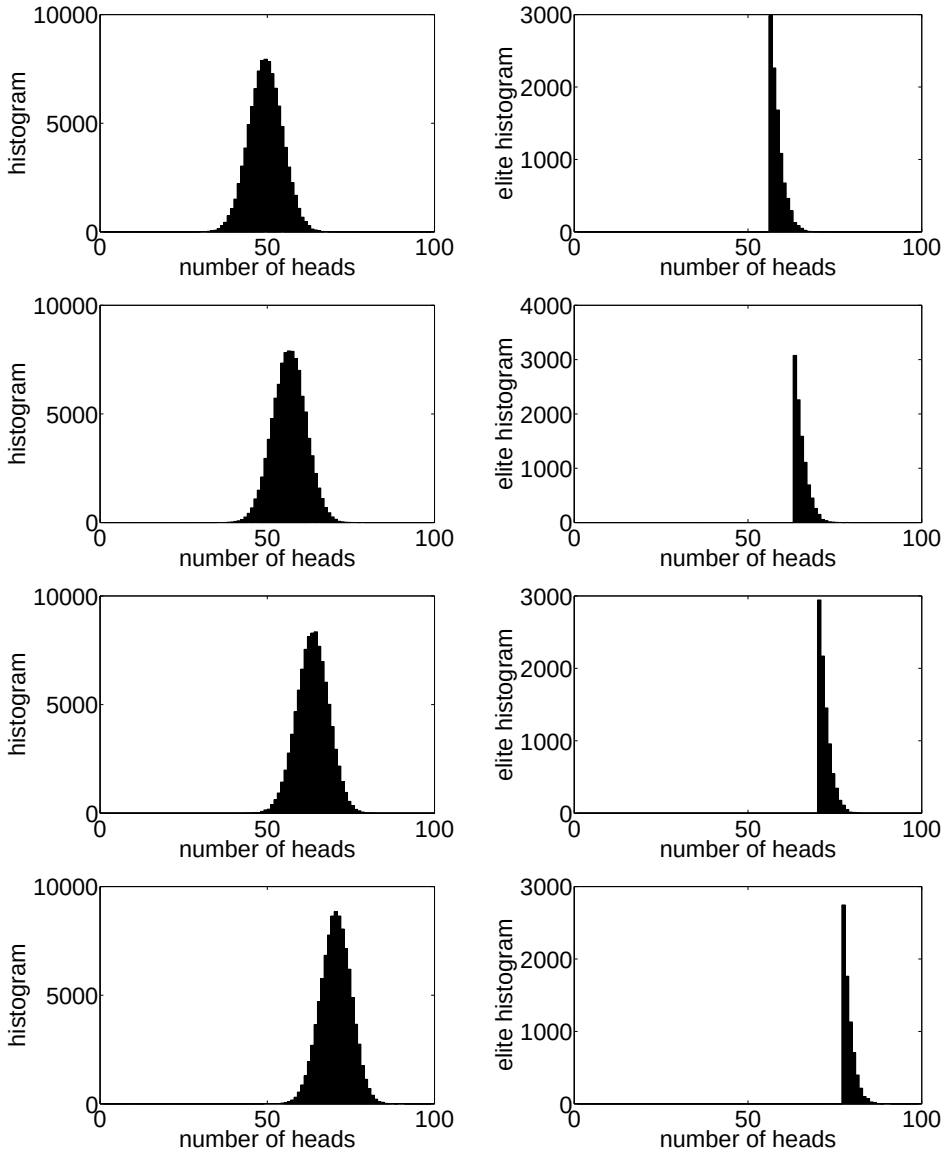

Figure 2.3. **Evolution of histograms for coin flipping problem**: Left column, full sample histograms at each CE method iteration. Right column, corresponding performance quantile sample histograms.

single instance of the CE method is very difficult, for a simple problem like this we have the computational power necessary to perform Monte Carlo simulation directly on the CE method. Thus, we set N =100,000 and run the CE method multiple times, collecting

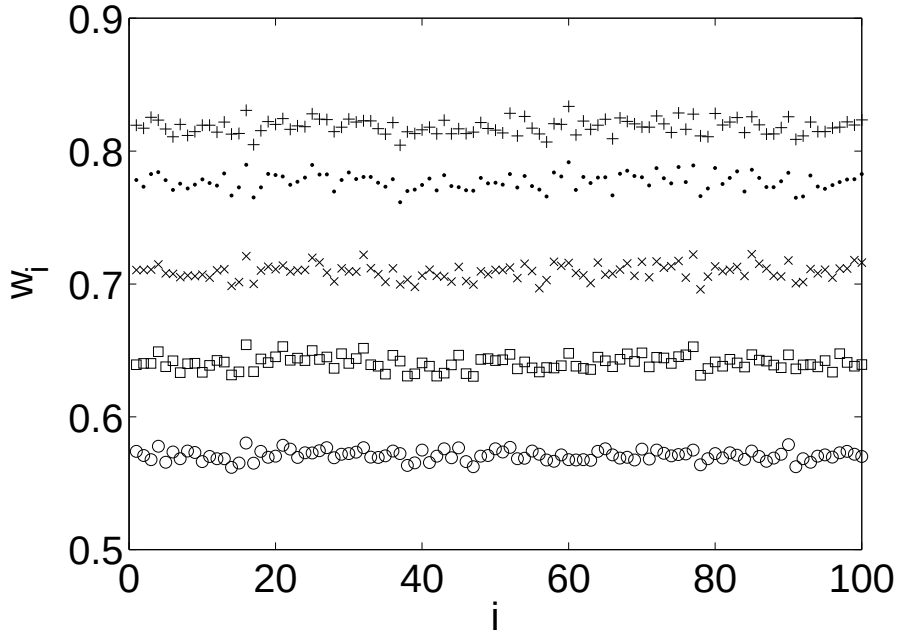

Figure 2.4. **Biasing distribution means for the coin flipping problem**: Output biasing distribution means for each coin, proceeding upward with each cross-entropy level toward the analytical solution at $w \equiv 0.8$.

statistics as we go. Fig. 2.5 shows the mean values for each parameter along with $\pm$ one standard deviation error bounds.

2.2.2. Sum of Gaussians with the cross-entropy method

Another simple rare-event simulation problem is the sum of Gaussians problem. Consider the rare event

$$\sum_{i=1}^{N} x_i > \bar{A} \tag{2.2.18}$$

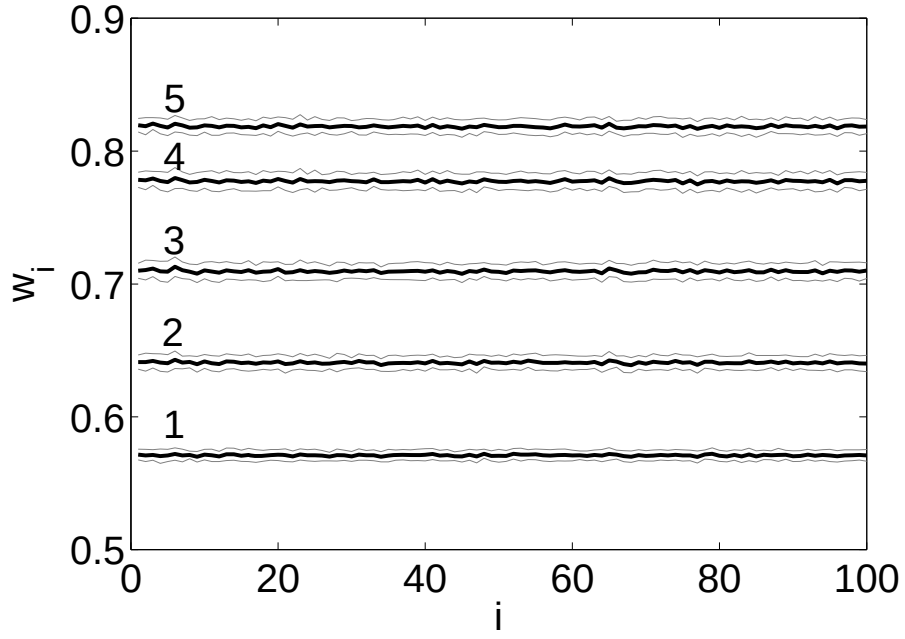

Figure 2.5. **Average biasing distribution means for the coin flipping problem**: Multiple trials of the cross-entropy method, with biasing parameter output averaged and standard error bands applied.

where the x_i are drawn from N Gaussian distributions with PDFs

$$f_i(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu_i)^2/(2\sigma^2)},$$

where we take σ to be equal and fixed for all i. Then the family of distributions we are using is parameterized by only the mean, and we can make use of Eq. (2.2.13) to estimate the mean at each step in the CE method toward the goal that Eq. (2.2.18) is no longer a rare event under the final biasing distributions. We then proceed via the main cross-entropy method algorithm and obtain the sequence of PDFs and quantiles in Fig. 2.6.

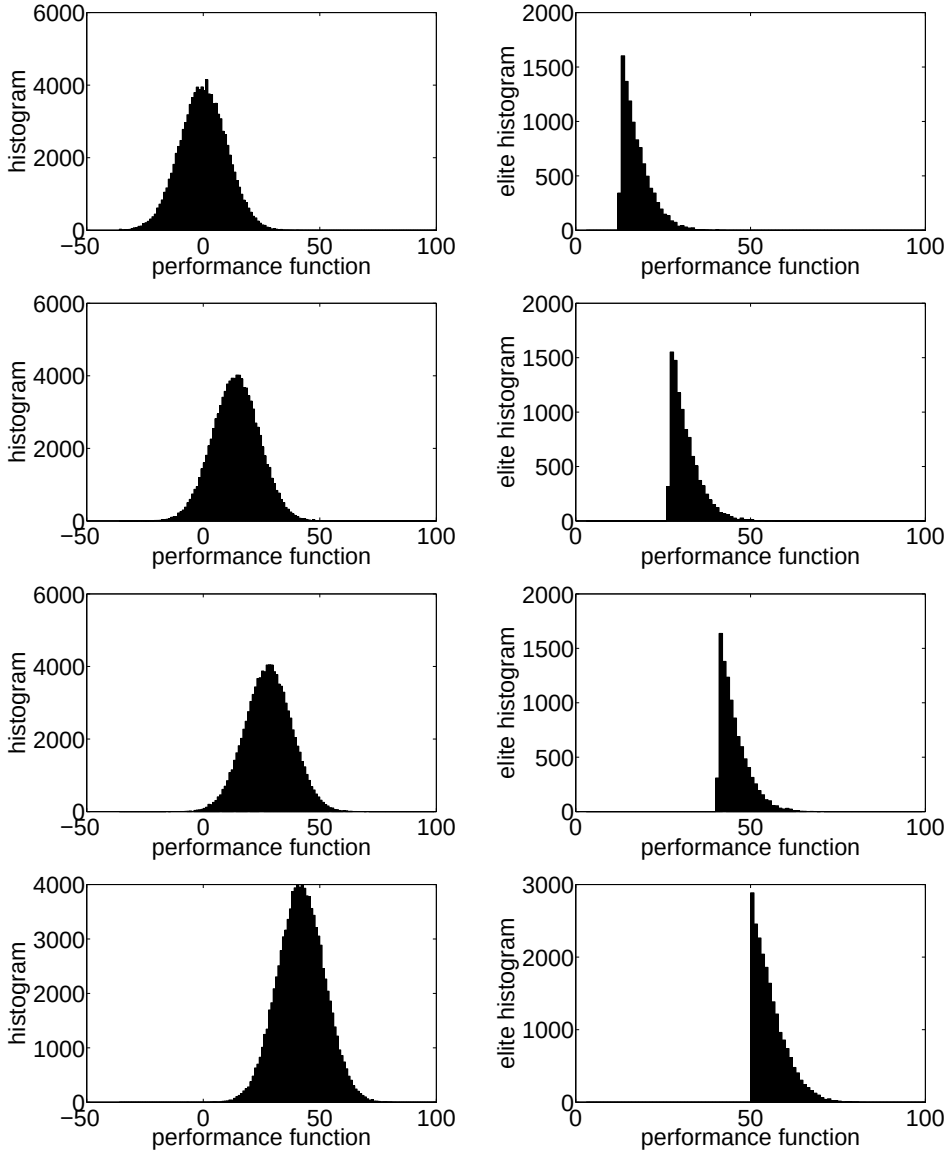

Figure 2.6. **Evolution of histograms for sum-of-Gaussians problem**: Full distribution histograms in the left column, elite sample histograms in the right column, with CE method iterations from top to bottom, with $\sum_{i=1}^{N} x_i$ on the x-axis.

The CE method algorithm, following the sample PDFs and histograms, converges on the optimum distribution parameters to reach our specified target. Recall that we have

parameterized the normal distributions in this problem only by the mean. The converged optimum values of the means are given in Fig. 2.7. We can compare this numerical result to the analytical solution. Consider

$$\max\left\{\sum_{i=1}^{N}\exp\left(-(x-\mu_i)^2\right)\right\} \Longleftrightarrow \min\left\{\sum_{i=1}^{N}\left((x-\mu_i)^2\right)\right\}$$

subject to

$$\sum_{i=1}^{N} x_i = \bar{A}.$$

The solution to this optimization problem is given by

$$\mu_i = \frac{\bar{A}}{N}, \qquad \forall i \tag{2.2.19}$$

and thus the solution to the optimization problem is to shift the mean of each Gaussian distribution by an equal share $\bar{A}/N$ for each of the N distributions. We can see that the numerical solution produced by the cross-entropy method agrees very well.

2.3. The singular value decomposition

We make use of the singular value decomposition (SVD) to reduce the dimensionality of the optimization problem by numerically finding the singular modes of the linearized system [16]. To see the need for this, consider a lightwave transmission system that has N_a amplifiers, and we wish to use N_f numerical Fourier modes to simulate the system. Without the SVD, the optimization space is $2N_aN_f$. For a typical problem, perhaps $N_a \sim 100$ and

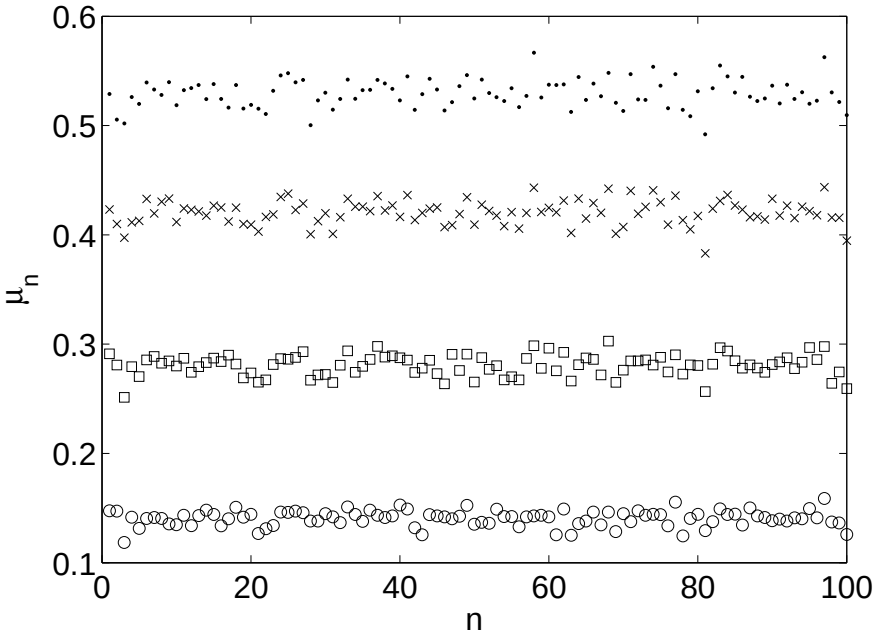

Figure 2.7. **Biasing distribution means for the sum-of-Gaussians problem**: Output biasing distribution means for distribution, proceeding upward with each cross-entropy level toward the analytical solution at $\mu_i = 0.5$.

$N_f \sim 1000$. 10^5 optimization dimensions is simply beyond the capability of the CE method (or any other). By finding the modes of the system, we can reduce the overall dimension from $2N_aN_f$ to perhaps CN_a, where C is a small integer constant – a vastly more tractable number for getting good results from the CE method.

In general, if M is an m-by-n matrix, then there is a factorization of the form

$$M = P\Sigma Q^* \tag{2.3.1}$$

known as the *singular value decomposition* where

- Q is n-by-n and contains the orthonormal input basis vectors

- P is m-by-m and contains the orthonormal output basis vectors
- Σ is a m-by-n diagonal matrix and contains the singular values as entries along the diagonal.

The SVD can be thought of as a generalization of the eigenvalue/eigenvector decomposition. In fact, if M is Hermitian positive semi-definite, then the SVD reduces to the eigenvalue/eigenvector decomposition [27]. There is an additional relationship between the SVD and the eigenvalue decomposition that is useful for our purposes. That is, the squares of the nonzero singular values of M are equal to the nonzero eigenvalues of M^*M and the columns of Q are the eigenvectors of M^*M.

Finding the singular modes in the problem here via the SVD proceeds as follows. Consider some nonlinear propagation operator for our system $\mathcal{N}(z_L; 0)[\cdot]$, for now just a generalized operator, and some initial data $U(0, t)$ such that

$$U(z_L, t) = \mathcal{N}(z_L; 0)[U(0, t)]. \tag{2.3.2}$$

Because of the conventions of the example problem we will soon be considering, we here use the notation with initial data as a function of t and propagation in the z-direction.

We will call $u_0(z, t)$ the *nominal solution* to Eq. (2.3.2) for all z and t. We then linearize $\mathcal{N}(z_L; 0)[\cdot]$ about our nominal solution u_0, and obtain a linear operator Φ such that the solution to the linearized problem can be written as

$$u(z_L, t) = \Phi(z_L; 0)u(0, t). \tag{2.3.3}$$

We now wish to find the modes of this linearized operator, Φ. It may not be possible to form the full matrix Φ, but we do not need to in order to take the SVD. To determine the SVD iteratively, it is only necessary to know the *action* of the direct linear operator as well as that of the adjoint linear operator (Φ^*). Then we use an iterative eigenvalue/eigenvector solver to find the eigenvalues and eigenvectors of $\Phi^*\Phi$, from which it is trivial to calculate the singular values and input vectors of Φ.

In a typical lightwave system, consider then the propagation problem from $z = 0$ to $z = z_L$ with intermediate amplifiers at $z = z_1, z_2, \ldots z_N$. We need to compute the modes *at each amplifier*; recall that we are generating the biasing distributions for importance sampling in a reduced-dimension space. The SVD gives us the most important directions in this state space, but it is different at each amplifier (the noise distributions at each amplifier are independent), and so we must compute the SVD at each amplifier. To do this, we first solve the nonlinear problem from $z = 0$ to $z = z_L$ and store it – we will need the full nonlinear solution for each subsequent application of the SVD. Then, at each z_n, we apply an iterative procedure to compute the eigenvalues and eigenvectors of the action $\Phi^*(z_L, z_n)\Phi(z_L, z_n)$ (and hence the SVD of $\Phi(z_L, z_n)$) , i.e. from the current location to the end of the line. This gives us the singular modes at each amplifier. The relative importance of the output modes from the SVD is given by the corresponding singular values (calculated from the eigenvalues). For the problems we have been studying, the essential behavior is captured by a handful of the largest singular values. Concrete examples from lightwave systems will be presented in following chapters where the systems have been described in the necessary detail.

CHAPTER 3

Optical Communications Systems

3.1. The nonlinear Schrödinger equation

The primary governing equation for the type of lightwave communication system we will be considering in this dissertation is the nonlinear Schrödinger (NLS) equation. Later in the thesis we will merely cite this (or a close relative) as the appropriate model for the system at hand, but here we consider some aspects of the derivation and the limitations of the NLS as a model for optical systems.

Propagation of an optical pulse in fiber, as an electromagnetic phenomenon, is governed by Maxwell's equations [28]. We will not proceed through all of the details of the derivation of the NLS from Maxwell's equations, but will follow the derivation in [29]. By merely some formal manipulations, without assumption, one can begin at Maxwell's equations and arrive at the wave equation

$$\nabla^2 \mathbf{E} - \frac{1}{c^2}\frac{\partial^2 \mathbf{E}}{\partial t^2} = -\mu_0 \frac{\partial^2 \mathbf{P}}{\partial t^2} \tag{3.1.1}$$

where $\mathbf{E}$ is the electric field, $\mathbf{P}$ is the induced polarization, c is the speed of light in a vacuum, μ_0 is the vacuum permeability, and t is our time variable. From this starting point, a number of simplifying assumptions are needed. First, we split the induced polarization into linear and nonlinear parts

$$\mathbf{P} = \mathbf{P}_L + \mathbf{P}_{NL} \tag{3.1.2}$$

so that we may treat the nonlinear part as a small perturbation to the linear part. The optical field is also assumed to be "quasi-monochromatic", that is, the spectral width is

much less than the center frequency ($\Delta\omega \ll \omega_0$). As a practical matter, for the systems we will be considering, this will restrict us to pulses with width ≥ 0.1 ps [29]. We must also assume that the optical field maintains its polarization while propagating through the optical fiber. For our systems, pulses will be launched with a single polarization and we assume that this single polarization is maintained throughout. We are thus justified in using the scalar approximation. While in general an optical fiber can support many propagating modes, we restrict ourselves to single-mode fiber.

We will also assume that the optical field can be described using the slowly-varying envelope approximation. That is, the electric field will be expressed in terms of the fiber mode (which gives the transverse structure), a slowly-varying envelope function, and a rapidly-varying phase term. If we refer to our fiber propagation direction as z, this slowly-varying envelope approximation allows us to neglect the second derivative of the slowly varying envelope with respect to z in the derivation. The signal envelope function is then the critical quantity with which we will be concerned, denoted here by U. Note that U is a function of propagation distance as well as either time ($U = U(z,t)$) or frequency ($U = U(z,\omega)$), where the initial condition is given in either time or frequency and propagated through space. Also note that the time domain and spectral versions are equivalent (under transfer via the Fourier or inverse Fourier transform) and both will be used throughout this work where a switch results in either simplicity or insight.

We will also make an assumption about the form of the nonlinearity involved. We have already assumed that the nonlinear portion of the induced polarization is small compared to the linear part, and under this assumption we justify an expansion of the induced

polarization in terms of the Taylor series. The material comprising the the optical fiber (specifically, fused silica) is centro-symmetric, which means that there is no second order quadratic term, and thus the first non-zero nonlinear term in the induced polarization is cubic.

With these assumptions in hand, we arrive at the nonlinear Schrödinger (NLS) equation

$$\frac{\partial U}{\partial z} = -i\frac{d}{2}\frac{\partial^2 U}{\partial t^2} + i\gamma|U|^2 U - \alpha U \tag{3.1.3}$$

where U is the envelope function, d is the group-velocity dispersion coefficient, γ is the nonlinear coefficient, z is the propagation distance, and t is the retarded time

$$t \leftarrow \tau = t - \frac{1}{v_g z} \tag{3.1.4}$$

where v_g is the group velocity coefficient. Note here that the wavenumber β has been expanded in a Taylor series about the center frequency with coefficients β_n:

$$\beta(\omega) = \beta_0 + (\omega - \omega_0)\beta_1 + \frac{1}{2}(\omega - \omega_0)^2\beta_2 + \ldots \tag{3.1.5}$$

With this definition in mind,

$$\begin{aligned} v_g &= \frac{1}{\beta_1} \\ d &= \beta_2 \end{aligned} \tag{3.1.6}$$

and higher order terms have been neglected under the quasi-monochromatic assumption as previously discussed. The nonlinear coefficient γ is given by

$$\gamma = \frac{2\pi n_2}{\lambda A_{\text{eff}}} \tag{3.1.7}$$

where n_2 is the Kerr coefficient, λ is the light wavelength, and A_{eff} is the effective fiber mode area. The linear transmission loss of the fiber is described by the loss coefficient α.

The dimensional units for this problem are given in the following table.

quantity	t	z	d	γ	α	$\lvert U\rvert^2$
units	ps	km	ps^2/km	1/(W-km)	1/km	W

Several different non-dimensionalizations are commonly used. In the non-dimensionalization that we will use for our discussion, consider the two length scales in the problem, the dispersion length scale and the nonlinear length scale. Accordingly, we define

$$\begin{aligned} L_D &= \frac{T_0^2}{|d|} \\ L_{NL} &= \frac{1}{\gamma P_0} \end{aligned}$$

with a characteristic width T_0 which we will take to be the pulse width and a characteristic power P_0 which we take to be the pulse power. By restricting ourselves to the regime where dispersion and nonlinearity play roughly equal roles, we assume $L_D = L_{NL}$ to obtain the relation

$$T_0 = \sqrt{\frac{|d|}{\gamma P_0}}. \tag{3.1.8}$$

We then apply the following change of variables

$$\begin{aligned} U &\leftarrow \frac{U}{\sqrt{P_0}} \\ z &\leftarrow \frac{z}{L_D} \\ t &\leftarrow \frac{t}{T_0} \end{aligned} \tag{3.1.9}$$

to obtain the dimensionless nonlinear Schrödinger equation

$$\frac{\partial U}{\partial z} - \frac{i}{2}\frac{\partial^2 U}{\partial t^2} - i|U|^2 U = 0 \tag{3.1.10}$$

Here the scalings have been specifically chosen to give unity coefficients, for convenience.

3.2. Solitons

In the anomalous dispersion regime ($d < 0$), the nonlinear Schrödinger equation gives rise to the famous soliton solution. The soliton is a "solitary wave" which maintains its shape and travels at a constant speed due to the balancing of the dispersive and nonlinear effects. Solitons were first observed in water waves in 1834 by John Scott Russell, who referred to the soliton as a "wave of translation" [30]. The existence of optical solitons in fiber was proposed in 1973 [31] and demonstrated in 1980 [32], and much work has since been based on the idea.

For solitons in the nonlinear Schrödinger equation, for convenience we will work with the dimensionless version, Eq. (3.1.10). The soliton solution for this equation is given by

$$U(z,t) = A\,\mathrm{sech}\left(A\left[t - T - \Omega z\right]\right)\exp\left(i\Omega t + \frac{i}{2}(A^2 - \Omega^2)z + i\phi\right), \tag{3.2.1}$$

where A,T,Ω, and ϕ are known as the *soliton parameters*. They correspond, accordingly, to the amplitude, position, frequency, and phase of the pulse. The reader should be aware of potential ambiguity in the terminology with z as the direction of propagation: "position" T refers to the location of the pulse in retarded time t, and "frequency" Ω refers to the offset from the carrier frequency.

The theory of solitons has contributed heavily to the construction of this new rare event simulation method. In a soliton system, perturbation theory can be applied to find analytically the modes of the system corresponding to the soliton parameters; i.e. for frequency, the frequency mode is the direction of perturbation which produces the largest change in the soliton frequency parameter Ω. These modes are given by [33]:

$$\begin{aligned} U_A &= \frac{1}{A}\left(t - A(t-T)\tanh\left(A(t-T)\right)\right)U_s, \\ U_T &= A\tanh(A(t-T))U_s, \\ U_\Omega &= -i(t-T)U_s, \\ U_\phi &= iU_s, \end{aligned} \tag{3.2.2}$$

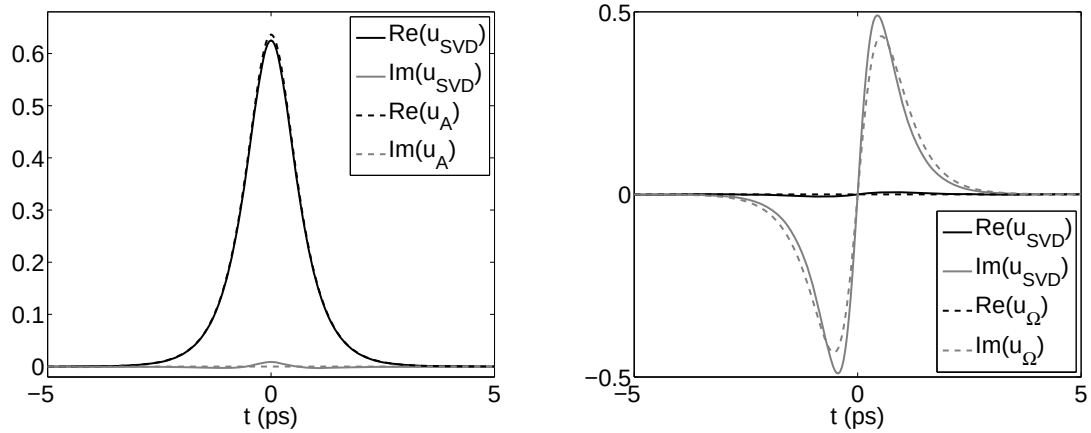

Figure 3.1. **Soliton modes found with the SVD**: Amplitude and frequency modes for a soliton pulse found with the SVD, compared with the analytic solution. Left – amplitude mode, right – frequency mode. Black – real part, gray – imaginary part. Solid – SVD mode, dashed – analytic mode.

where U_s is the soliton solution given by Eq. (3.2.1). In such a system, arbitrary pulse perturbations can be described in terms of the projection of the perturbation onto the modes, and a remaining radiative component, where the perturbation energy which is not in the direction of a mode will be shed by the soliton as radiated energy. That is, for a perturbation ΔU,

$$\Delta U = U_A \Delta A + U_T \Delta T + U_\Omega \Delta\Omega + U_\phi \Delta\phi + R. \tag{3.2.3}$$

where ΔA, ΔT, $\Delta\Omega$ and $\Delta\phi$ are the induced changes in the pulse parameters and R is the radiative component. For this case, where the modes are available analytically, we can test the SVD method of determining the system modes against the analytic solution. The results of the comparison are displayed in Fig. 3.1.

Within this framework, a reduced problem (in the asymptotic, perturbation sense) can be posed and solved to generate the biasing distributions required for importance sampling. Our method strives to be more general than this previous technique which requires the mathematical structure of soliton perturbation theory, but this theory is instructive in guiding the new method. Rather than having an asymptotic, analytical solution for the modes, they will be computed numerically using the singular value decomposition. Without an analytical form for the modes and evolution equations for the pulse parameters derived from soliton perturbation theory, there is no reduced problem to solve for the biasing distributions. However, we can still expect to find the important directions for the system by using the SVD. The invariances of the NLS lead to integral conserved quantities by Noether's theorem; in the case of soliton pulses, this gives the soliton parameters. Even in the absence of soliton pulses, we have these conserved quantities, but we have no analytical form for the linearized modes. We seek to find the modes then through the use of the SVD. The biasing distributions will have to be found by posing an optimization problem which depends upon these numerically determined modes; the optimization problem can then be solved by the cross-entropy method.

3.3. Dispersion management

No practical commercial systems operate in the soliton regime as described in the previous section. This is because in practice *dispersion management* is used, whereby the the dispersion profile of the transmission line is tailored to enhance system performance. Generally, the *dispersion map*, i.e., the arrangement of dispersion in the transmission line,

is a periodic alternation of fiber segments with normal and anomalous dispersion, such that a pulse launched into the transmission line "breathes" as it propagates: broadening and then narrowing, according to the dispersion of the current section of fiber. The dispersion, then, as a function of propagation distance z, is periodic and piecewise-constant. The piecewise-constant and periodic nature of the dispersion map, as well as the corresponding pulse breathing, is illustrated in Fig. 3.2. In the upper portion of the figure, contours of the propagating, breathing pulse are plotted for three dispersion map periods; the lower plot is of the corresponding dispersion map.

A well-designed dispersion managed system yields a performance increase over a constant-dispersion system. For example, dispersion managed systems have been designed to minimize channel cross-talk as well as four-wave mixing [29, 34], nonlinear effects which commonly degrade system performance.

For our purposes, we will generally consider dispersion maps in terms of an average dispersion term and a zero-mean, varying term, namely

$$d(z/z_a) = d_{\text{avg}} + \frac{1}{z_a}\Delta D(z/z_a) \tag{3.3.1}$$

where z_a is the dispersion map period, here set equal to the amplifier spacing, and we require that

$$\int_0^1 \Delta D(z')dz' = 0. \tag{3.3.2}$$

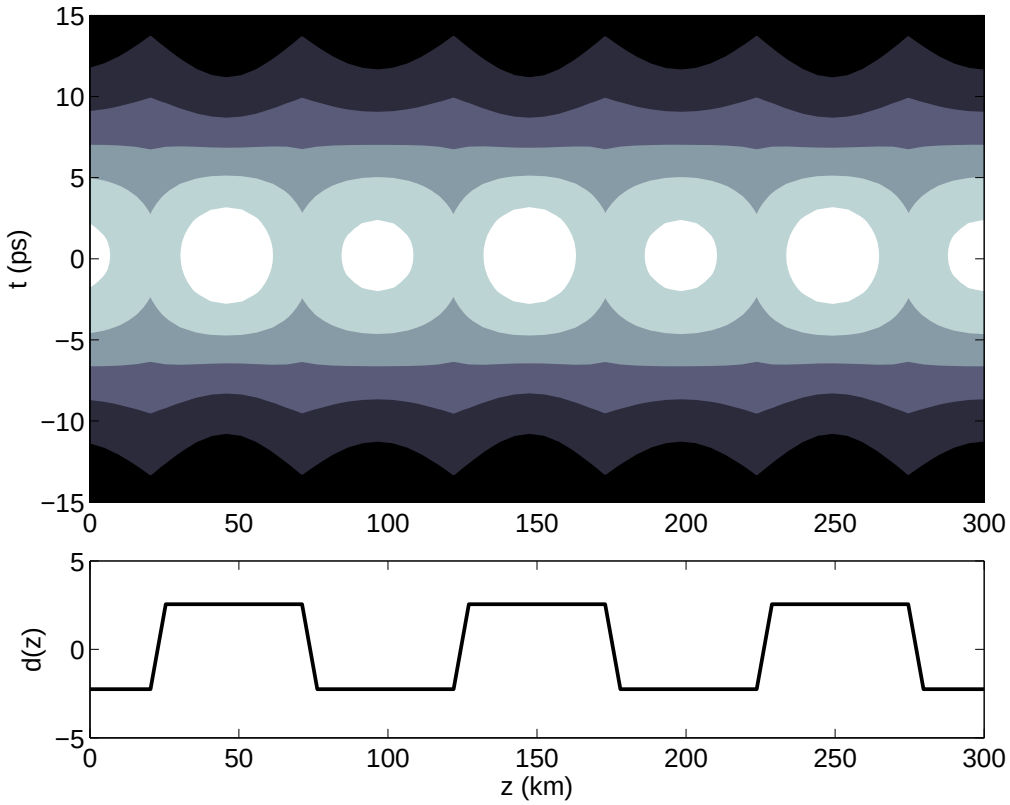

Figure 3.2. **Breathing pulse and dispersion map**: Top, contour plot of breathing pulse propagating through a dispersion managed fiber line, through three dispersion map periods. Bottom, the corresponding dispersion map of the system.

With this definition of the dispersion map, a key quantity for determining the effect of the map on the system is the *map strength*, s, given by

$$s = \frac{1}{4}\int_0^1 |\Delta D(z')| dz' \tag{3.3.3}$$

which measures the total variation away from the mean dispersion in the map. When $s = 0$, the dispersion managed system reduces to a constant dispersion system; however,

the simple sech()-shaped solitons are not supported in the dispersion managed regime ($s \neq 0$).

3.4. Dispersion managed solitons

Although simple solitons are not supported in dispersion managed systems, a similar set of solutions exists, known as dispersion managed solitons. They maintain many of the same properties as their cousins from constant-dispersion systems, but in general the soliton shapes cannot be found analytically and must be computed numerically. Unlike the simple soliton solution these dispersion managed solitons do not have a stationary energy profile throughout propagation, but are stroboscopically stationary when observed with the period of the dispersion map. There are a number of methods available for finding these pulse shapes [35], but we use an iterative correction method originally due to [36] and modified to preserve pulse energy [37, 38].

The central idea is that launching a pulse that is "near" the desired dispersion managed soliton shape (for example a Gaussian or sech() shaped pulse with the proper pulse energy and desired width) into the dispersion-managed fiber produces a pulse evolution with slow, secondary oscillations. If one averages out these slow, secondary oscillations, the underlying true dispersion managed soliton pulse shape can be extracted. The idea then is to begin with this "near" DM soliton pulse, $u_0(t)$, and propagate numerically through the transmission line. As the pulse is propagated through the periodic dispersion maps, the full-width, half-max pulse width (T_{fwhm}) is observed stroboscopically (that is, at the same point in each map). This propagation continues until both a local minima and maxima in T_{fwhm}

are observed, and the pulse profiles corresponding to these extremal widths are aligned in phase at the pulse center and then averaged, and the resulting average pulse is corrected in energy back to the original pulse. Deviations between the launched pulse and the ideal pulse result in slow oscillations during propagation; averaging out these oscillations allows the underlying ideal pulse to emerge. That is,

(1) Launch $u_0(t)$ into the dispersion managed fiber.

(2) Repeat:

  (a) Propagate the pulse through the dispersion map.

  (b) Observe T_{fwhm} at a fixed point (stroboscopically).

  (c) Check for local minima and maxima in T_{fwhm}. Record the pulse profile at these locations as $u_{min}(t)$ and $u_{max}(t)$, respectively. Exit loop when both ($u_{min}(t)$ and $u_{max}(t)$) are found.

(3) Phase-align at pulse-center and average:

$$\alpha = \text{atan}\left(\frac{\Im\left[u_{max}(0)\right]}{\Re\left[u_{max}(0)\right]}\right) - \text{atan}\left(\frac{\Im\left[u_{min}(0)\right]}{\Re\left[u_{min}(0)\right]}\right)$$

$$\hat{u} = \left(u_{max} + \exp\left(i\alpha\right)u_{min}\right)/2$$

(4) Rescale to original pulse energy:

$$u = \hat{u}\sqrt{\frac{\int \left|u_0(t)\right|^2 dt}{\int \left|\hat{u}(t)\right|^2 dt}}$$

(5) Compute the residual

$$\int |u_{max} - \exp(i\alpha) u_{min}| \, dt$$

(6) If the residual is larger than the predefined threshold, goto step 1 with $u_0 \leftarrow u$.

(7) Return $u(t)$ as the converged dispersion-managed soliton pulse shape.

This method is efficient and robust for finding the dispersion managed soliton pulse shapes [36]. There are potential convergence issues when the net dispersion is normal, but in this work we deal exclusively with net anomalous dispersion systems. We will need to compute dispersion managed soliton pulse shapes for the systems we will be studying in detail later. Note, however, that this procedure does not need to be repeated, except when the system parameters are changed. The entire rare-event simulation procedure for a fixed dispersion-managed soliton system requires only that the pulse shape be found once and stored for reference.

3.5. Amplifier noise

Transmission loss due to absorption and scattering in communications systems must be balanced by amplification, or the transmitted signal would be lost before reaching the detector [29]. The type of amplifier used is called an *erbium doped fiber amplifier*, or EDFA. Raman amplification is also commonly used. Systems with in-line EDFAs, like those that we will be considering, are subject to a performance penalty due to accumulating noise. Each amplifier adds what is known as *spontaneous emission* noise to the signal as part of the amplification process. These amplifiers are in-line, and the spontaneous emission

noise from one amplifier is then amplified by the next, and so on, leading to *amplified spontaneous emission* (ASE) noise. This noise can build up in such a way that in rare cases the transmitted signal can't be properly detected at the output. It is in fact precisely these cases of noise buildup that we are interested in; they are the rare events in these communication systems that cause transmission errors.

The model for amplified spontaneous emission noise is additive Gaussian white noise. While in reality the noise consists of a discrete number of photons, the noise bandwidth and pulse energy are sufficiently large to allow the use of the (continuous) Gaussian noise model. The mean of the noise distribution is zero, and the variance is set by the *power spectral density*, essentially power per unit bandwidth. The power spectral density of ASE noise at the carrier frequency ω_0 is given by

$$S(\omega_0) = \hbar\omega_0 n_{\mathrm{sp}} \frac{(G-1)^2}{G \log^2 G} \tag{3.5.1}$$

where amplifier gain and transmission loss have been averaged out, $\hbar$ is Planck's constant, n_{sp} is the spontaneous emission factor, and G is the amplifier gain. Thus the noise model is given by

$$\langle N(t) \rangle = 0 \tag{3.5.2}$$

and

$$\begin{aligned} \langle N(t)N^*(t')\rangle &= S(\omega_0)\delta(t-t') \\ &= \hbar\omega_0 n_{\mathrm{sp}}\frac{(G-1)^2}{G\log^2 G}\delta(t-t') \end{aligned} \tag{3.5.3}$$

where the bandwidth of the noise is restricted by our spectral numerical method. Recall that U is a complex quantity; the noise is added as half of the above quantity in each quadrature. In general, the amplifier gain can be specified arbitrarily by system design, but we use the amplifier gain which is specifically tailored to counteract transmission loss. That is, in relation to previously specified parameters, with a loss rate given by α and amplifier spacing z_a, periodic amplifier gain is

$$G = e^{2\alpha z_a}. \tag{3.5.4}$$

Our full model for optical fiber propagation with added noise, then, is given by

$$\frac{\partial U}{\partial z} = -i\frac{d}{2}\frac{\partial^2 U}{\partial t^2} + i\gamma|U|^2 U + \sum_{k=1}^{N_a} N_k(t)\delta(z-kz_a) \tag{3.5.5}$$

where the periodic loss and gain is averaged out [29]. The noise-addition process is illustrated graphically for a sample pulse and noise variance in Fig. 3.3.

We have now fully specified the simulation and optics-related theoretical background material necessary for the development of our new method. In practice, some details of numerical implementation of the idea still remain.

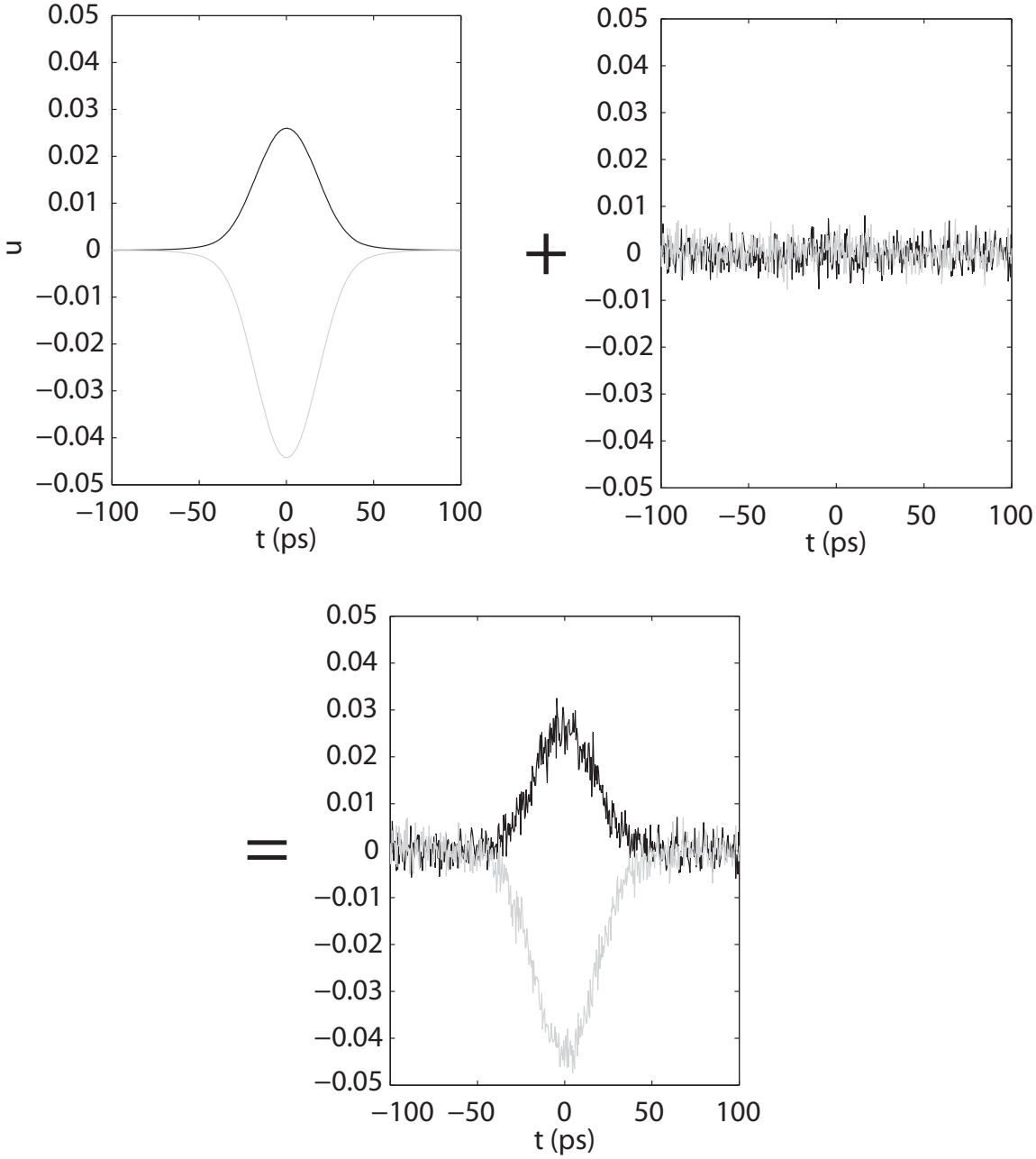

Figure 3.3. **Illustration of noise addition to an optical pulse**: Schematic illustrating a noiseless optical pulse, both real and imaginary parts (black and gray, respectively), and the numerical ASE noise to be added, both real and imaginary, and the resulting noisy pulse.

CHAPTER 4

Numerical Methods

There are several different numerical methods required for the work in this dissertation. For one, the physical systems concerned are governed by the nonlinear Schrödinger equation (NLS) or NLS-type equations. Because we wish to simulate these system directly, we must have numerical schemes for solving these governing equations quickly and accurately. Secondly, these lightwave systems operate in the presence of amplified spontaneous emission (ASE) noise, represented by Gaussian white noise. Our numerical schemes must accommodate this noise as well. Finally, it is necessary to compute the singular value decomposition (SVD) of the linearized propagation operator as part of our rare-event simulation scheme. Efficiently computing the SVD is also a numerical concern.

4.1. Split-step Fourier method

The split-step Fourier method [29] is a popular technique for solving the NLS numerically. As we will see, it has favorable stability properties as well as advantages in terms of introducing numerical noise. In order to illustrate the split-step Fourier method, consider a general partial differential equation with an evolution operator given by

$$\frac{\partial u}{\partial z} = F(u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}, ...) \tag{4.1.1}$$

where we again use the convention of the initial condition in t and propagation in the z-direction.

The split-step Fourier method uses the idea of *operator splitting* and can be applied to problems where Eq. (4.1.1) can be written as

$$\frac{\partial u}{\partial z} = f_1(u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}, ...) + f_2(u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}, ...) \tag{4.1.2}$$

by then alternately solving the sub- or half-problems

$$\frac{\partial u}{\partial z} = f_1(u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}, ...) \tag{4.1.3}$$

and

$$\frac{\partial u}{\partial z} = f_2(u, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}, ...). \tag{4.1.4}$$

There are some accuracy concerns with operator splitting which will be discussed shortly; the primary benefit, however, is that now each half-problem may be solved via the numerical scheme best suited to each half-problem, not necessarily the numerical scheme which is the best compromise for both operators as a whole.

For the NLS, we split the propagation operator

$$\frac{\partial u}{\partial z} = \frac{i}{2} d(z) \frac{\partial^2 u}{\partial t^2} + i\gamma |u|^2 u \tag{4.1.5}$$

into

$$\frac{\partial u}{\partial z} = \frac{i}{2} d(z) \frac{\partial^2 u}{\partial t^2} = \mathcal{D}[u] \tag{4.1.6}$$

and

$$\frac{\partial u}{\partial z} = i\gamma|u|^2 u = \mathcal{N}[u] \tag{4.1.7}$$

and solve the two equations independently. Eq. (4.1.6) is a linear, dispersive, partial differential equation, and Eq. (4.1.7) is merely a nonlinear, first-order ordinary differential equation with t as a parameter.

The dispersive equation is solved via a pseudo-spectral method. By transforming to Fourier space, Eq. (4.1.6) becomes

$$\frac{\partial}{\partial z}\hat{u}(\omega, z) = (i\omega)^2 \frac{i}{2} d(z)\hat{u}(\omega, z) = -\frac{i\omega^2}{2} d(z)\hat{u}(\omega, z). \tag{4.1.8}$$

Discretizing in the Fourier domain using the Fast Fourier Transform (FFT), we obtain ordinary differential equations for each spectral component

$$\sum_j \frac{d}{dz}\hat{u}_{\omega_j}(z)e^{-i\omega_j t} = \sum_j -\frac{i\omega_j^2}{2} d(z)\hat{u}_{\omega_j}(z)e^{-i\omega_j t}. \tag{4.1.9}$$

Note that the ω subscript indicates the Fourier component, not a derivative with respect to ω. We solve these equations independently in the spectral domain for a single time step and obtain

$$\hat{u}_{\omega_j}(z + \Delta z) = \hat{u}_{\omega_j}(z) \exp\left(-\frac{i\omega_j^2}{2} \int_z^{z+\Delta z} d(z')dz'\right). \tag{4.1.10}$$

If $d(z)$ is constant for $z \in [z, z + \Delta z)$, then this reduces to

$$\hat{u}_{\omega_j}(z + \Delta z) = \hat{u}_{\omega_j}(z) \exp\left(-\frac{i\omega_j^2}{2} d(z)\Delta z\right). \tag{4.1.11}$$

We then transform back into the time domain using the inverse Fast Fourier Transform.

The nonlinear subproblem Eq. (4.1.7) is solved in the time domain, taking advantage of the operator splitting to avoid computing the convolution of the nonlinear term required for solution in the spectral domain. We will also use the fact that the pulse intensity is pointwise constant within this subproblem. To see this, multiply Eq. (4.1.7) through by u^* to obtain

$$\frac{\partial u}{\partial z} u^* = i\gamma |u|^4. \tag{4.1.12}$$

It follows that

$$\frac{\partial u}{\partial z} u^* = -\left(\frac{\partial u}{\partial z} u^*\right)^* \tag{4.1.13}$$

and thus

$$\begin{aligned} 0 &= \frac{\partial u}{\partial z} u^* + u \frac{\partial u^*}{\partial z} \\ &= \frac{\partial}{\partial z} |u|^2. \end{aligned} \tag{4.1.14}$$

Thus with both the pulse amplitude and the nonlinear coefficient constant throughout the time step, we obtain

$$u(t, z + \Delta z) = u(t, z) \exp\left(i\gamma |u(t, z)|^2 \Delta z\right). \tag{4.1.15}$$

Alternating the steps as described previously, we obtain our split-step Fourier scheme for the nonlinear Schrödinger equation

$$\tilde{u}^n = \mathcal{F}^{-1}\left[\mathcal{F}\left[u^n\right] \exp\left(-\frac{i\omega^2}{2} d(n\Delta z)\Delta z\right)\right] \tag{4.1.16}$$

$$u^{n+1} = \tilde{u}^n \exp\left(i\gamma |\tilde{u}^n|^2 \Delta z\right). \tag{4.1.17}$$

Note that while the split operators were solved in similar ways, the difference (and thus the operator splitting) is quite important. Solving the nonlinear operator in the spectral domain would require computing a convolution integral, a computationally very costly step. Operator splitting makes this step unnecessary.

The error associated with operator splitting can be easily assessed. We begin with the exact solution to Eq. (4.1.5) given formally as

$$u(t, z + \Delta z) = \exp\left(\Delta z(\mathcal{D} + \mathcal{N})\right) u(t, z) \tag{4.1.18}$$

and, by expanding using the Baker–Campbell–Hausdorff formula [39], we see that

(4.1.19)

$$\exp\left(\Delta z(\mathcal{D} + \mathcal{N})\right) = \exp\left(\Delta z\mathcal{D} + \Delta z\mathcal{N} + \frac{1}{2}\Delta z^2[\mathcal{D}, \mathcal{N}] + \frac{1}{12}\Delta z^3[\mathcal{D} - \mathcal{N}, [\mathcal{D}, \mathcal{N}]] + \dots\right)$$

for non-commuting operators $\mathcal{D}$ and $\mathcal{N}$ where $[f,g] = fg - gf$. By employing the scheme given in Eq. (4.1.16), we are then left with $\frac{1}{2}\Delta z^2[\mathcal{D},\mathcal{N}]$, a truncation term of $\mathcal{O}(\Delta z^2)$.

We can improve upon this scheme. Consider instead applying the operator splitting as

$$u(t, z+\Delta z) = \exp\left(\frac{\Delta z}{2}\mathcal{D}\right)\exp\left(\Delta z\mathcal{N}\right)\exp\left(\frac{\Delta z}{2}\mathcal{D}\right)u(t,z) \tag{4.1.20}$$

by applying Eq. (4.1.19) twice, we see that the $\mathcal{O}(\Delta z^2)$ truncation term becomes

$$\begin{aligned}\frac{1}{2}\Delta h^2\left[\frac{\mathcal{D}}{2},\mathcal{N}\right] + \frac{1}{2}\Delta h^2\left[\mathcal{N},\frac{\mathcal{D}}{2}\right] + \mathcal{O}(\Delta z^3) &= \frac{1}{4}\Delta h^2\left([\mathcal{D},\mathcal{N}] + [\mathcal{N},\mathcal{D}]\right) + \mathcal{O}(\Delta z^3)\\ &= \frac{1}{4}\Delta h^2\left(\mathcal{D}\mathcal{N} - \mathcal{N}\mathcal{D} + \mathcal{N}\mathcal{D} - \mathcal{D}\mathcal{N}\right) + \mathcal{O}(\Delta z^3)\\ &= \mathcal{O}(\Delta z^3).\end{aligned}$$

By applying this scheme, called *symmetric operator splitting*, to each step and combining adjacent like half-steps

$$\begin{aligned}u(t, z+2\Delta z) &= \exp\left(\frac{\Delta z}{2}\mathcal{D}\right)\exp\left(\Delta z\mathcal{N}\right)\exp\left(\frac{\Delta z}{2}\mathcal{D}\right)\\ &\times \exp\left(\frac{\Delta z}{2}\mathcal{D}\right)\exp\left(\Delta z\mathcal{N}\right)\exp\left(\frac{\Delta z}{2}\mathcal{D}\right)u(t,z)\\ &= \exp\left(\frac{\Delta z}{2}\mathcal{D}\right)\exp\left(\Delta z\mathcal{N}\right)\exp\left(\Delta z\mathcal{D}\right)\exp\left(\Delta z\mathcal{N}\right)\exp\left(\frac{\Delta z}{2}\mathcal{D}\right)u(t,z)\end{aligned} \tag{4.1.21}$$

we obtain a higher-order scheme by beginning and ending the computation with a half-step application of the dispersive operator. Eq. (4.1.21) can of course be applied in general

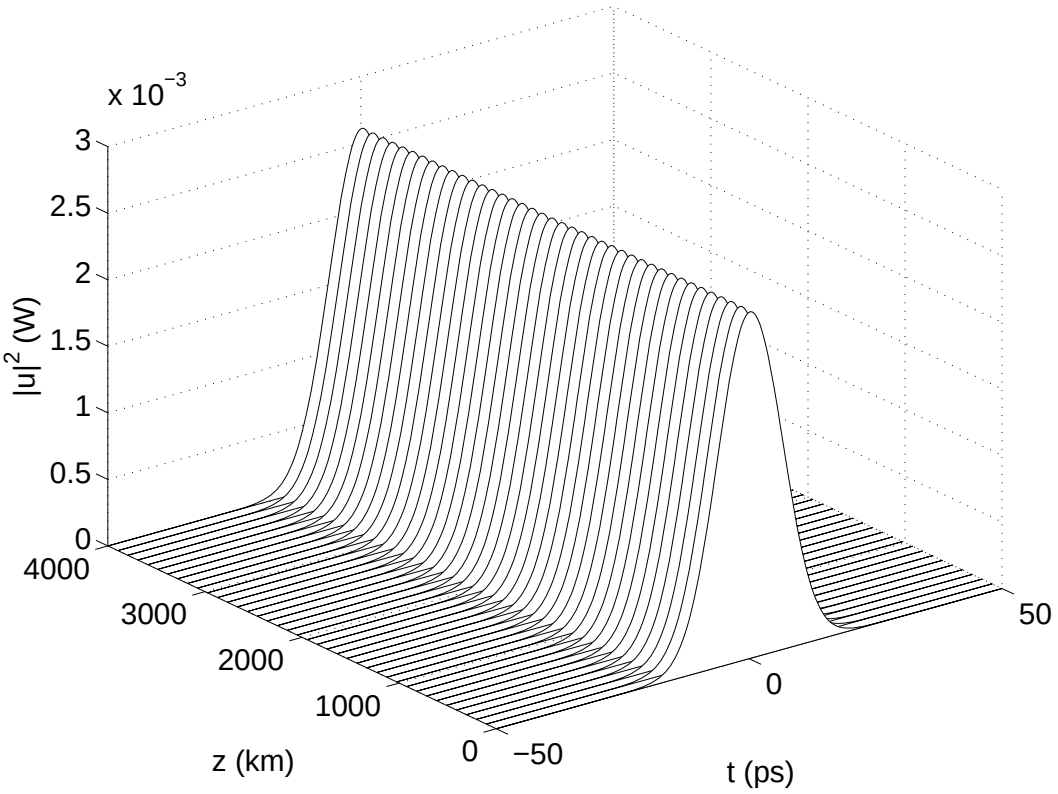

Figure 4.1. **Numerical solution of NLS for a dispersion managed pulse**: The solution to the nonlinear Schrödinger equation with a periodic dispersion map with anomalous average dispersion, observed stroboscopically, found numerically using the split-step Fourier method.

for an arbitrary number of steps. This pseudo-spectral method is also all-orders in Δt, and thus the overall accuracy of the scheme is $[(\Delta z)^2, (\Delta t)^p]$. Solving a representative, noiseless problem for a dispersion-managed pulse, we obtain Fig. 4.1. The error for this numerical solution, with $\Delta z = 5$km and $\Delta t = 0.39$ps, is given in Fig. 4.2. The convergence of the error for this problem in space in time is given in Fig. 4.3.

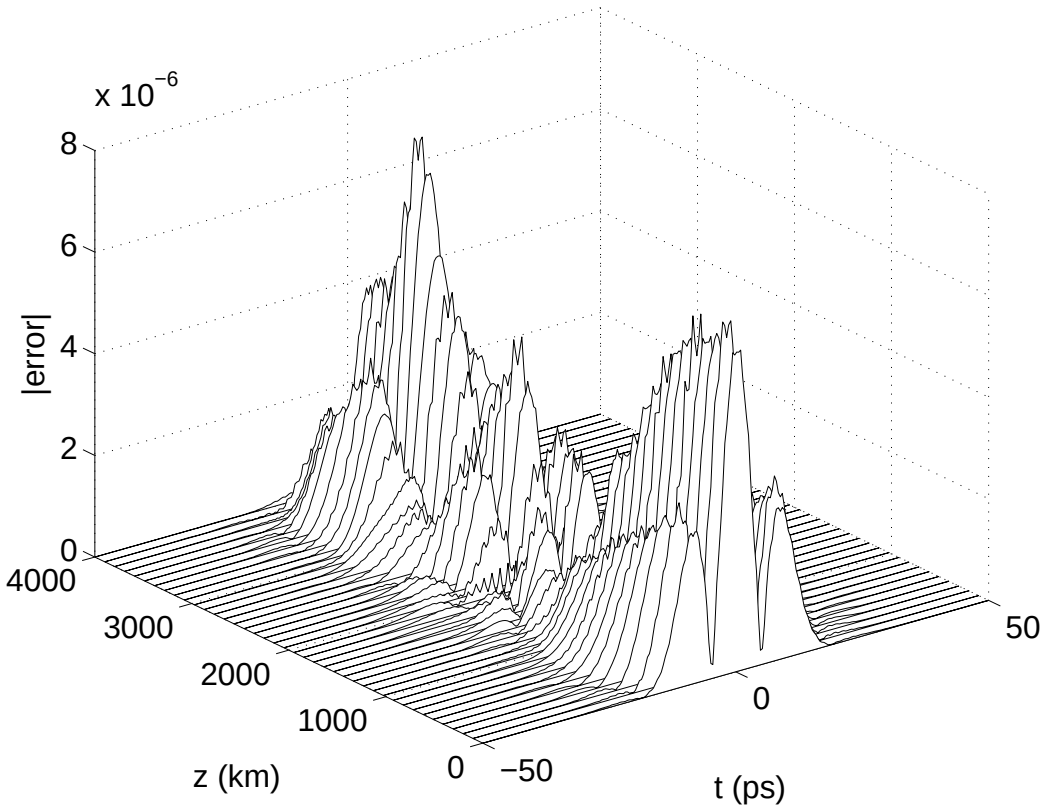

Figure 4.2. **Error for numerical solution of NLS for a dispersion managed pulse**: The calculated numerical error corresponding to Fig. 4.1, split-step Fourier method with $\Delta z = 5$km and $\Delta t = 0.39$ps.

The split-step Fourier method for solving the nonlinear Schrödinger equation numerically has several additional benefits for our purposes. First, the split-step Fourier method has been shown to be the most computationally efficient method for solving the NLS owing to the "exact" nature of numerical solution to the nonlinear step [40]. The method is also ideal for adding numerical noise to the solution, as we will be required to do. The benefit here is three-fold; first, by using a spectral method instead of a finite difference method,

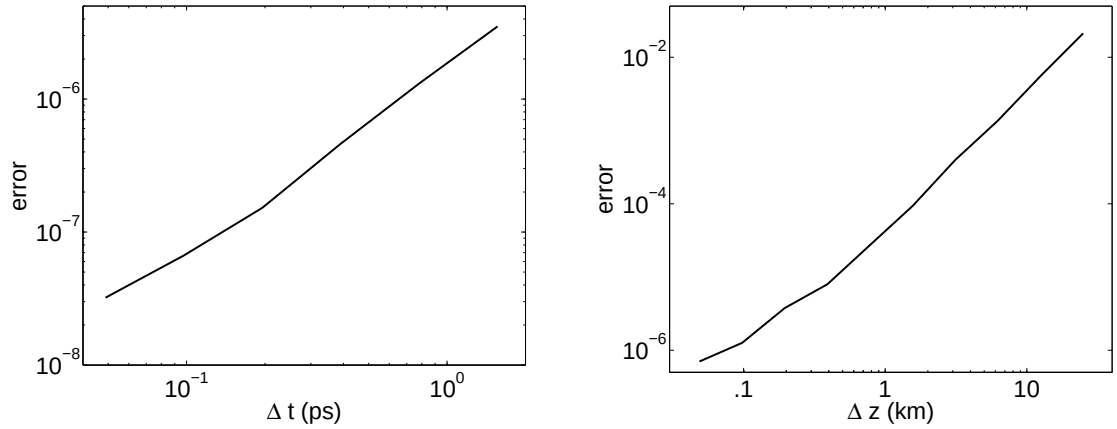

Figure 4.3. **Convergence tests for numerical solution of DM soliton**: Split-step Fourier method numerical solver. Left – convergence in t, right – convergence in z.

any concerns about taking finite difference derivatives of noise are alleviated. Second, the error of the split-step Fourier method is not dissipative, and thus it will not artificially dampen the noise. Finally, the method is unconditionally stable [41].

4.2. A higher-order numerical scheme

An alternate numerical scheme to the split-step Fourier method can be employed to simulate the nonlinear Schrödinger equation when the Δz accuracy is of more concern. For certain applications, it may be more beneficial to employ a modified scheme which still employs an exact solution to the linear operator $\mathcal{D}$ in Fourier space, but uses a fourth order Runge–Kutta "time"-stepping scheme (in our case, remember, we are advancing in space, z) to advance both the integrating factor exact solution to $\mathcal{D}$ and the numerical approximation to $\mathcal{N}$. We make use of this numerical method to solve the linearized equations (direct and adjoint) that are required to compute the SVD; see Sec. 2.3.

The fourth order Runge–Kutta (RK4) [42] method for ordinary differential equation integration uses a weighted average of different numerically calculated slopes to obtain a higher order method than merely approximating the slope with a single approximation. In the forward Euler method, for example, the slope is merely calculated at the beginning of the time-step interval; conversely, in backward Euler the slope is only calculated at the endpoint of the interval. The RK4 method uses four different slope calculations: 1) at the beginning of the interval, 2) at the midpoint of the interval, with the midpoint value calculated using slope from 1, 3) at the midpoint of the interval, with the midpoint value calculated using the slope from 2, and 4) at the endpoint of the interval, with the endpoint calculated using the slope from 3.

If the slope from the n^{th} point is referred to as k_n, employed in solving the ordinary differential equation

$$\frac{dy}{dz} = f(z, y) \tag{4.2.1}$$

then

$$\begin{aligned} k_1 &= f(z_i, y_i) \\ k_2 &= f\left(z_i + \frac{\Delta z}{2}, y_i + \frac{\Delta z}{2} k_1\right) \\ k_3 &= f\left(z_i + \frac{\Delta z}{2}, y_i + \frac{\Delta z}{2} k_2\right) \\ k_4 &= f\left(z_i + \Delta z, y_i + \Delta z k_3\right) \end{aligned} \tag{4.2.2}$$

and the integrating scheme is advanced using the weighted average

$$\frac{y_{i+1}-y_i}{2}=\frac{1}{6}\left(k_1+2k_2+2k_3+k_4\right). \tag{4.2.3}$$

If we return to solving Eq. (4.1.5) numerically, and adapt this integrating scheme along with the integrating factor method from the split-step Fourier technique. This scheme, then, called the *integrating factor Runge–Kutta* 4^{th} *order method* (IFRK4) [43] is given as

(1) $\tilde{u} \leftarrow \mathcal{F}[u(z_0)]$

(2) for $j = 1 \ldots N$

(a) $\bar{u} \leftarrow \mathcal{F}^{-1}[\tilde{u}]$

(b) $s_1 \leftarrow \mathcal{F}(i\gamma|\bar{u}|^2\bar{u})\Delta z$

(c) $\hat{u} \leftarrow \exp\left(-\frac{i}{4}d(z_j)\omega^2\Delta z\right)\tilde{u}$

(d) $\bar{u} \leftarrow \mathcal{F}^{-1}\left[\hat{u}+\frac{1}{2}\exp\left(-\frac{i}{4}d(z_j)\omega^2\Delta z\right)s_1\right]$

(e) $s_2 \leftarrow \mathcal{F}(i\gamma|\bar{u}|^2\bar{u})\Delta z$

(f) $\bar{u} \leftarrow \mathcal{F}^{-1}\left[\hat{u}+\frac{1}{2}s_2\right]$

(g) $s_3 \leftarrow \mathcal{F}(i\gamma|\bar{u}|^2\bar{u})\Delta z$

(h) $\hat{u} \leftarrow \exp\left(-\frac{i}{4}d(z_j)\omega^2\Delta z\right)\hat{u}$

(i) $\bar{u} \leftarrow \mathcal{F}^{-1}\left[\hat{u}+\exp\left(-\frac{i}{4}d(z_j)\omega^2\Delta z\right)s_3\right]$

(j) $s_4 \leftarrow \mathcal{F}(i\gamma|\bar{u}|^2\bar{u})\Delta z$

(k) $\tilde{u} \leftarrow \hat{u}+\frac{1}{6}\left(\exp\left(-\frac{i}{2}d(z_j)\omega^2\Delta z\right)s_1+2\exp\left(-\frac{i}{4}d(z_j)\omega^2\Delta z\right)(s_2+s_3)+s_4\right)$

(l) optional: $u(z_{j+1}) \leftarrow \mathcal{F}^{-1}[\tilde{u}]$

(3) $u(z_N) \leftarrow \mathcal{F}^{-1}[\tilde{u}]$

Here we assume γ is constant, and we have merely written $d(z_j)$ to avoid confusing the algorithm hopelessly. If $d(z)$ is not constant, the $d(z_j)$ terms above should be replaced with $\frac{1}{\Delta z}\int_{z_j}^{z_{j+1}} d(z')dz'$. Clearly this is a much more complicated and computationally costly scheme per step, as compared with the split-step Fourier method. However, with the additional complexity comes additional accuracy. Recall that the accuracy order of the split-step Fourier method was either $\mathcal{O}(\Delta z)$ or $\mathcal{O}(\Delta z^2)$, depending on the step ordering scheme. The IFRK4 scheme has accuracy $\mathcal{O}(\Delta z^4)$.

Of course, we have stated the IFRK4 scheme for solving the nominal problem, Eq. (4.1.5), where it can be applied if necessary. We use the SSFM for the nominal problem and apply the IFRK4 scheme only to the linearized operators, which requires a few simple modifications. The dispersive term is still solved in the same way — it is linear in the nominal problem, and hence unchanged in the linearized problems. The nonlinear terms are now linear, but are still ODEs after the operator splitting and can still be solved as such using the RK4 method.

4.3. Noise in simulations

There are several practical numerical issues pertaining to adding numerical noise to simulations. We have already selected a numerical scheme which does not artificially dampen the noise, the split-step Fourier method. An additional concern is the inherently finite bandwidth of the numerical approximation. True Gaussian white noise has an infinite bandwidth; that is, the power spectrum is constant across all frequency components.

While it is of course impossible to provide infinite noise bandwidth numerically, we do impose the condition that the numerical noise bandwidth sufficiently exceeds the pulse bandwidth for accurate representation of the noise. This argument is aided by the fact that amplified spontaneous emission noise from Erbium-doped fiber amplifiers does not have an infinite noise bandwidth, unlike the Gaussian model. ASE noise from EDFAs typically has bandwidth in the range of 2.5 THz [29], still significantly wider than the pulse bandwidth of roughly 50 GHz. Numerically, we impose a similar condition, that is that the numerical noise bandwidth significantly exceeds the pulse bandwidth. Generally, 64 to 128 Fourier modes per pulse provide sufficient numerical noise bandwidth.

Gaussian random variables are generated numerically by way of a linear congruential generator with a shuffle table (to reduce sequential correlation) [22] which generates a high-quality uniform distribution. The uniform distribution generator can then be adapted to a normal distribution generator with the *Box-Muller* method, which generates the normal distribution

$$p(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y-\mu)^2}{2\sigma^2}\right). \tag{4.3.1}$$

The Box-Muller method [22] takes a pair of uniform random variables $[x_1, x_2]$ and returns a pair of normal random variables $[y_1, y_2]$ via

$$\begin{aligned} y_1 &= \sqrt{-2\ln(x_1)}\sin(2\pi x_2)\sigma + \mu \\ y_2 &= \sqrt{-2\ln(x_1)}\cos(2\pi x_2)\sigma + \mu. \end{aligned} \tag{4.3.2}$$

This method is used to generate normal random variables of any mean and variance for both the cross-entropy method as well as generating full-bandwidth numerical noise for the simulated system itself.

4.4. Computing the SVD numerically

We have previously discussed the general formulation of the singular value decomposition. However, there are some issues concerned with efficiently computing the SVD of our propagation operator, which was presented in general in Sec. 2.3 and will be presented specifically for systems we will demonstrate in Sec. 5.4.1. As mentioned previously, we will take advantage of the fact that if we are concerned with the SVD of a linear operator Φ, the squares of the singular values of Φ are given by the eigenvalues of $\Phi^*\Phi$, and the corresponding orthonormal input vectors are given by the eigenvectors of $\Phi^*\Phi$. We then concern ourselves with the estimation of the largest few eigenvalues and corresponding eigenvectors of $\Phi^*\Phi$, as these contain the essential modes for our purposes.

To efficiently estimate these eigenvalues and eigenvectors, we employ the *implicitly restarted Arnoldi method* as implemented in the freely available software package ARPACK [44]. The full details of this complex algorithm are beyond the scope of this dissertation, but the essential idea is as follows. To solve the eigenvalue problem for the matrix $\Phi^*\Phi$, starting with an initial random vector $\vec{v}$, we form the Krylov matrix

$$K_n \;=\; \left[\vec{v} \;\; \Phi^*\Phi\vec{v} \;\; (\Phi^*\Phi)^2\vec{v} \;\; (\Phi^*\Phi)^3\vec{v} \;\; \ldots \;\; (\Phi^*\Phi)^{n-1}\vec{v}\right].$$

By extracting the orthonormal basis vectors from K_n, we obtain approximations for the eigenvectors corresponding to the n largest eigenvalues of $\Phi^*\Phi$. The span of these vectors is known as the Krylov subspace, denoted $\mathbb{K}_n$. The key to the Arnoldi method is to extract these basis vectors in a numerically stable way. Thus we use the Arnoldi method to execute the similarity transformation $\Phi^*\Phi = QHQ^*$ or, for our purposes of extracting only a few eigenvectors and eigenvalues,

$$(\Phi^*\Phi)Q_n = Q_{n+1}H_n$$

where

$$Q_n \;=\; \begin{bmatrix} q_1 & q_2 & q_3 & \dots & q_n \end{bmatrix}$$

contains the orthonormal basis vectors q_k spanning $\mathbb{K}_n$, and H_n is an upper Hessenberg matrix whose eigenvalues approximate the n largest eigenvalues of $\Phi^*\Phi$. The basic algorithm then is

(1) $\vec{v} =$ random, $q_1 = \vec{v}/||\vec{v}||$

(2) for $k = 1, 2, 3, \dots$

  (a) $b = \Phi^*\Phi q_k$

  (b) for $j = 1 \dots k$

    (i) $h_{j,k} = q_j^* b$

    (ii) $b = b - h_{j,k} q_j$

  (c) $h_{k+1,k} = ||b||$

  (d) $q_{k+1} = b/||b||$

Note that the algorithm makes use of of stabilized Gram–Schmidt orthonormalization. The algorithm outputs the vectors q_k which are the eigenvectors of $\Phi^*\Phi$ and thus the singular input vectors of Φ, corresponding to the singular values which are contained as the square roots of the eigenvalues of the upper Hessenberg matrix H_n, the components of which are also output by the algorithm. ARPACK includes documentation [44] of the many refinements that are used in practice to this basic algorithm, including the crucial implicit restarting procedure. However, this basic overview should serve to give an intuitive grasp of the method.

CHAPTER 5

The SVD-CE-IS Method for Rare-Event Simulation

The previous sections of this dissertation have presented the theoretical underpinnings of a new rare-event simulation method. We now present the details of the SVD-CE-IS method and demonstrations of its effectiveness for simulating rare events in problems in optical communication systems. This new technique makes use of the cross-entropy method, the singular value decomposition and multiple importance sampling along with the Monte Carlo method combined into a new method applicable to a wide class of rare-event simulation problems in optical communications systems and beyond, without need of recourse to intricate mathematical structure of the underlying problem.

The idea of the SVD-CE-IS technique is to simulate rare events via multiple importance sampling, a well-established technique. The difficulty with the application of multiple importance sampling to Monte Carlo simulations is generally the determination of the biasing distributions; this is especially true in complex systems with very large noise dimensionality (see Sec. 2.1.2), and the benefit of this new technique is in the generation of these biasing targets. The technique we propose determines the biasing distributions using the cross-entropy method, a stochastic optimization algorithm outlined in Sec. 2.2. While the cross-entropy method is a powerful optimization tool, it is generally incapable of handling the number of noise dimensions found in the lightwave systems that prompted this work. If the cross-entropy method were capable of handling our full system dimensionality, say N random variables, we could simulate rare events with an algorithm represented schematically by Fig. 5.1.

In each iteration, the cross-entropy method would provide N random variables for our system simulation, and receive in return a performance function value from the simulated

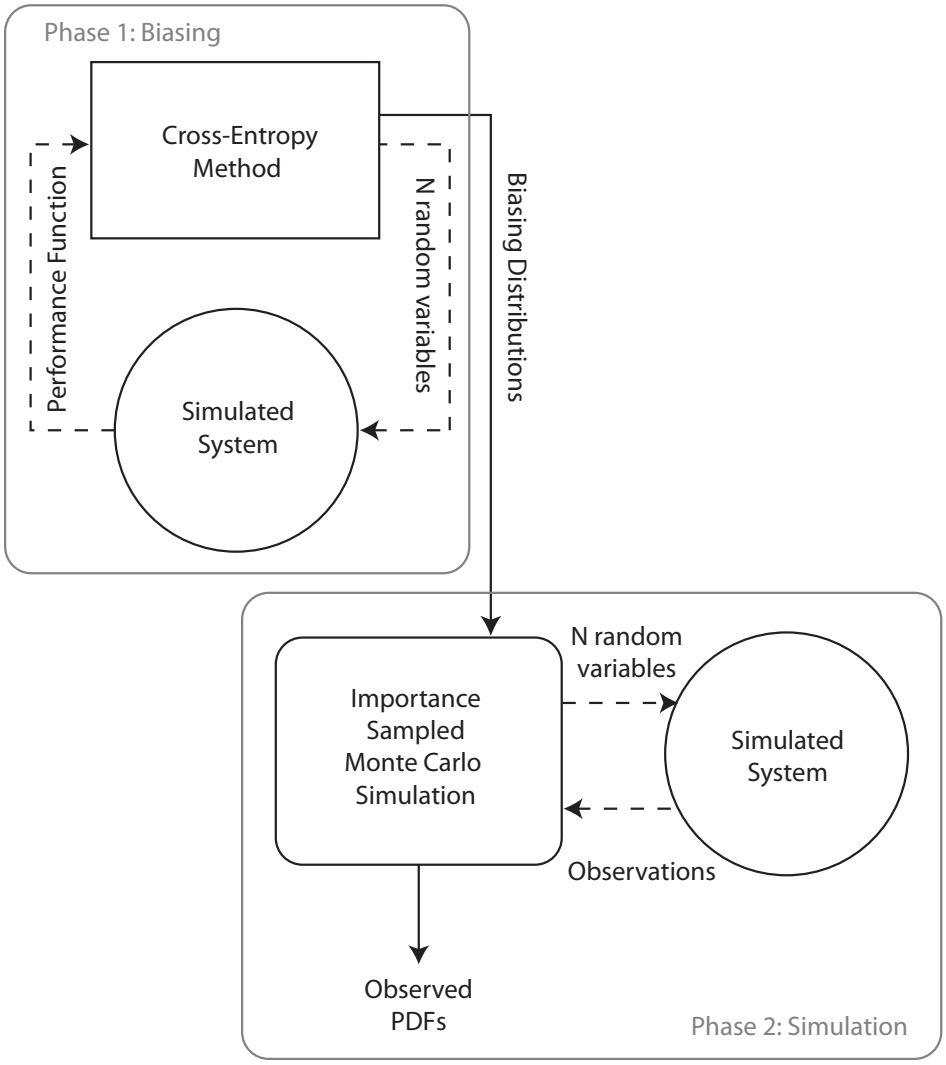

Figure 5.1. **Schematic of CEM based simulation method without the SVD**: Dashed lines represent stochastic elements repeated many times. Solid lines represent elements executed only once. The upper portion of the diagram represents the biasing phase, and the lower portion the simulation phase. The method illustrated in this schematic lacks the dimensionality-reducing aspect of the singular value decomposition, and would be generally infeasible for the problems we wish to consider.

system. This process would be repeated the necessary number of times (dashed lines) until the cross-entropy method obtained the biasing distributions. The biasing distributions

would then be fed into importance sampled Monte Carlo simulations, which again provide N random variables to the simulated system (now the random variables are biased according to the provided biasing distributions) many times until the desired observed histograms are obtained with the prescribed accuracy. However desirable in theory, this algorithm is computationally infeasible for the problems under consideration; $N \approx N_a N_f$ for the amplified lightwave systems under consideration, where N_a is the number of amplifiers in the system (≈ 40) and N_f is the number of computational Fourier modes used, ($\approx 256, 512, 1024$ or more), and the resulting N is simply too large to obtain reasonable results with currently available computing power.

To reduce the dimensionality of the underlying problem, we apply the singular value decomposition to reduce the biasing problem from an infinite (or computationally large) number (N) of Fourier modes to merely a handful ($n \ll N$) of "modes" of the system [16], an approach inspired by soliton perturbation theory-based approaches to similar problems [1, 5, 21]. Having reduced the number of optimization dimensions so dramatically, the cross-entropy method provides optimal biasing distributions for multiple importance sampling in our Monte Carlo simulations in a computationally feasible manner. Note that the singular value decomposition must be called upon again during the importance sampled Monte Carlo phase of the algorithm, because the biasing distributions generated under this scheme depend upon the computed modes of the system, and thus the modes must be recalculated for full noise trials as well. This improved algorithm with the SVD included, the SVD-CE-IS method, is represented schematically in Fig. 5.2. In both figures, the dashed lines represent connections or processes which are executed multiple times to

collect statistics, due to their stochastic nature. The solid lines represent connections or processes which are only made once, based upon the aggregation of the stochastic trials.

This method requires little of the system under consideration. Of course, it must be possible to accurately simulate the system numerically, but this is a requirement held over from any type of Monte Carlo method. The additional requirements imposed by the variance reduction scheme are those of applying the singular value decomposition, which are modest and will be discussed at length later in this chapter.

5.1. Application to a general lightwave system

For a lightwave communication system, rather than a general simulated system, we can qualitatively describe the method of application. The full details of application to a specific system will be presented in a later section. Consider that our general lightwave system is specified by a propagation operator described by

$$U(z_1, t) = \mathcal{N}(z_1; z_0)[U(z_0, t)] \tag{5.1.1}$$

and some initial condition $U(z_0, t) = U_0$, along with a detector operator I to be discussed at length later and a stochastic specification of the noise in the system. We will treat the case of loss compensated by periodic amplification such that noise is added to the system at specified points. Thus Gaussian white noise is added at each amplifier to represent the amplified spontaneous emission noise. Let us suppose for simplicity that they are equally spaced throughout the system, and we call the distance between amplifiers z_a. That is, if there are N_a amplifiers, the system begins at $z = z_0$ and ends at $z = z_0 + N_a z_a$, with the

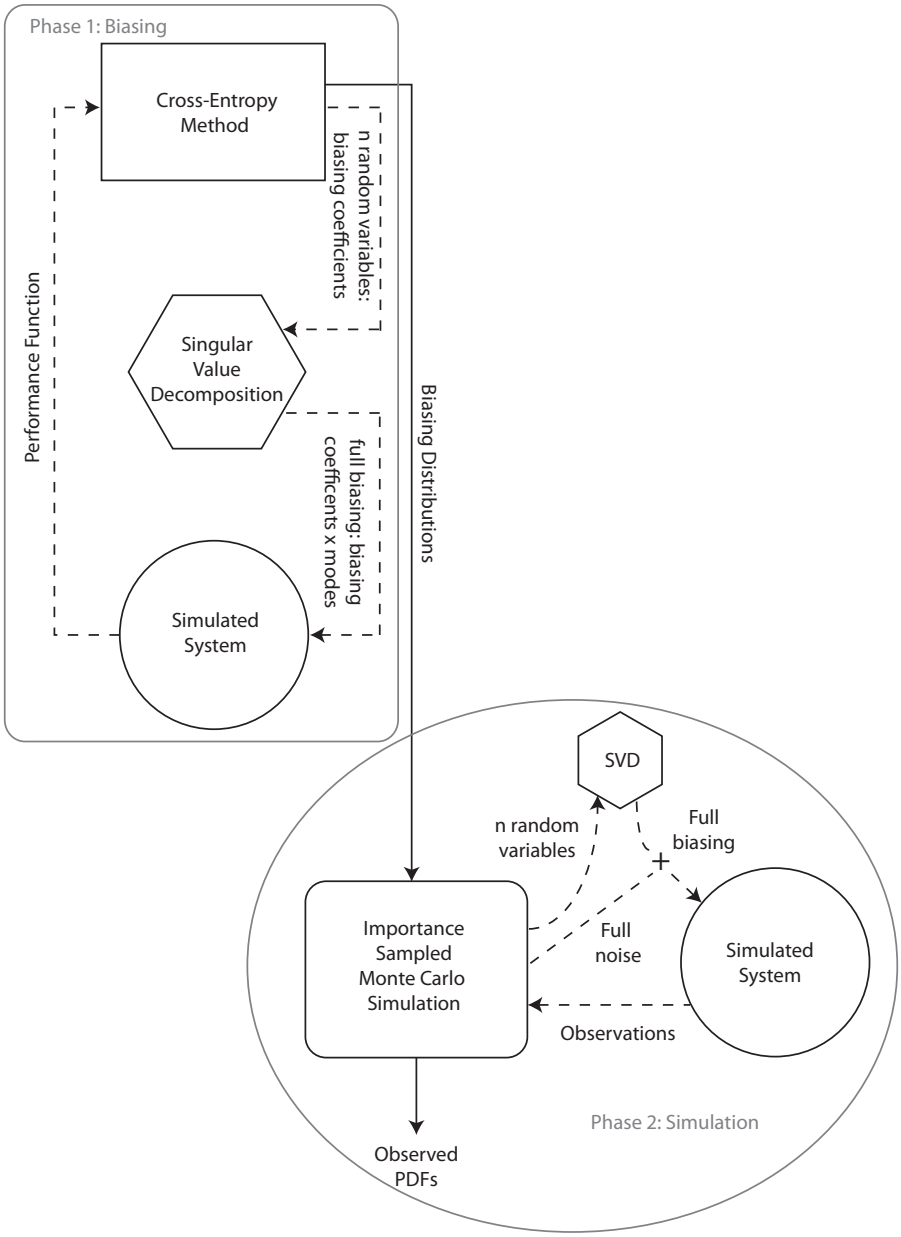

Figure 5.2. **Schematic of the SVD-CE-IS method**: Dashed lines represent stochastic elements repeated many times. Solid lines represent elements executed only once. This schematic represents our new proposed method, with the SVD employed to reduce the noise dimensionality of the simulated problem to a tractable level for the CE method. As before, the upper portion of the diagram represents the biasing phase, and the lower portion the simulation phase.

signal amplified at the intermediate points $z = z_0 + jz_a$. If we are using N_f Fourier modes in our computational scheme, then the simulated system takes in $N = N_a N_f$ random variables for the noise components at each amplifier, via

$$U(z_0 + jz_a, t) \leftarrow U(z_0 + jz_a, t) + \mathcal{F}^{-1}\left[\sum_{\omega=-N_f/2}^{N_f/2} \tilde{r}_{\omega,j} e^{i\omega t}\right] \tag{5.1.2}$$

where the $\tilde{r}_{\omega,j}$ are the N random noise components, full spectrum noise in each Fourier mode. The system then outputs some performance function or observations as a result of propagating the input signal and all the added noise, via the detector operator I. The performance function can be anything at all of concern that can be extracted from the simulated system. If the method of Fig. 5.1 were sound, we would simply supply a 'black-box' function of the system as described above, and simulate rare events.

To use instead the new – and feasible – method of Fig. 5.2, we must discuss how to apply the singular value decomposition to the lightwave system to reduce the noise dimensionality. The idea is to apply the SVD at each amplifier ($z = z_0 + jz_a$ for the j^{th} amplifier). Recall that the SVD provides the mode which produces the largest change in the prescribed output function; thus applying the SVD at the j^{th} amplifier results in the most efficient way to produce the desired change *from that amplifier forward*. Suppose we use $n_m \ll N_f$ singular modes at each amplifier; call these modes $u_{i,j}(t)$ for $i = 1 \ldots n_m$ and $j = 1 \ldots N_a$, i.e. the i^{th} mode at the j^{th} amplifier. (The decision about the number of modes to use depends on both physical knowledge of the system and the spectrum of the singular values.) Then for the purpose of generating the biasing distributions, the top

half of Fig. 5.2, we add noise to the system via

$$U(z_0 + jz_a, t) \leftarrow U(z_0 + jz_a, t) + \sum_{i=1}^{n_m} \hat{r}_{i,j} u_{i,j}(t) \tag{5.1.3}$$

and we have obtained a vast reduction in the noise dimensionality. Now the cross-entropy method need only provide and control $n = n_m N_a$ random variables, a dramatic decrease from $N = N_f N_a$. This reduction makes the problem computationally feasible and we are able to generate the appropriate biasing distributions using this formulation, and it is made possible because the performance of the system is really controlled only by a few singular modes.

During the importance-sampled Monte Carlo phase, the noise addition is somewhat different. While the reduced problem is acceptable for generating the biasing distributions, the full simulations require full noise in each Fourier mode and the biasing distributions are specified with shifted means in terms of the modes. The addition of full-bandwidth noise as well as biased noise thus appears as combination of the two sources, that is

$$U(z_0 + jz_a, t) \leftarrow U(z_0 + jz_a, t) + \sum_{i=1}^{n_m} b_{i,j} u_{i,j}(t) + \mathcal{F}^{-1}\left[\sum_{\omega=-N_f/2}^{N_f/2} \tilde{r}_{\omega,j} e^{i\omega t}\right]. \tag{5.1.4}$$

Where $b_{i,j}$ are the biasing coefficients as determined by the biasing distributions. This can be viewed as a shift in the means of the distributions of the $\tilde{r}_{\omega,j}$. While before the $\tilde{r}_{\omega,j}$ were Gaussian white noise, and thus drawn from distributions with 0 mean, we can rewrite the

noise terms of Eq. (5.1.4) as

$$\begin{aligned}
\sum_{i=1}^{n_m} b_{i,j} u_{i,j}(t) + \mathcal{F}^{-1}\left[\sum_{\omega=-N_f/2}^{N_f/2} \tilde{r}_{\omega,j} e^{i\omega t}\right] &= \mathcal{F}^{-1}\left[\mathcal{F}\left[\sum_{i=1}^{n_m} b_{i,j} u_{i,j}(t)\right] + \sum_{\omega=-N_f/2}^{N_f/2} \tilde{r}_{\omega,j} e^{i\omega t}\right] \\
&= \mathcal{F}^{-1}\left[\sum_{\omega=-N_f/2}^{N_f/2} \left(\tilde{r}_{\omega,j} + \tilde{B}_{\omega,j}\right) e^{i\omega t}\right] \\
&= \mathcal{F}^{-1}\left[\sum_{\omega=-N_f/2}^{N_f/2} \tilde{R}_{\omega,j} e^{i\omega t}\right] \qquad (5.1.5)
\end{aligned}$$

where $\tilde{B}_{\omega,j}$ are the Fourier transform components of $\sum_{i=1}^{n_m} b_{i,j} u_{i,j}(t)$ and the $R_{\omega,j}$ are random normal variables drawn from distributions with the same variance as those of the $\tilde{r}_{\omega,j}$ but now with mean $\tilde{B}_{\omega,j}$ instead of 0. Note also that a time-domain formulation of the noise is also valid, with noise at each time point instead of in each spectral component. The biasing of the modes in the time domain can still be viewed as a mean-shift of the existing normal variables in exactly the same way. Mean-shifted normal distributions allow for a particularly simple likelihood ratio calculation, though in principle much more general distributions could easily be used. All that remains, then, is to collect the desired statistics and apply the correction for the biasing via the likelihood ratio as discussed previously. Recall that by using mean-shifted Gaussian distributions as our biasing distributions, calculating the optimal distribution parameters as part of the cross-entropy method requires simply calculating the value of the single parameter we have used via the first moment, according to Eq. (2.2.13). More precise details of application to a real system are included in the following sections.

5.2. SVD operator formulation including the detector

Up to this point we have assumed that the nonlinear operator $\mathcal{N}$ and, correspondingly, the linearized operator Φ are operations which map $\mathbb{C}^N \to \mathbb{C}^N$. However, in many lightwave systems with a detector at the end of the line, the operator we are interested in might instead map $\mathbb{C}^N \to \mathbb{R}^N$ or $\mathbb{C}^N \to \mathbb{R}$ in some potentially complicated way. For example, the "integrate-and-dump" detector, a common detector, is a $\mathbb{C}^N \to \mathbb{R}$ operator. Defining the action of adjoint operator Φ^* in this case requires a slight formal manipulation.

We have the nonlinear propagation operator which describes the full behavior of our system given by

$$U(z_2, t) = \mathcal{N}(z_2; z_1)U(z_1, t). \tag{5.2.1}$$

We linearize this operator via $U = U + \epsilon u$, and in the case where $\mathcal{N}$ is a differential operator, obtain the linearized operator

$$\frac{d\vec{u}}{dz} = A(z, t)\vec{u} \tag{5.2.2}$$

where we also rewrite the complex scalar quantity u as the real-valued vector quantity $\vec{u} = \begin{bmatrix} \Re(u) & \Im(u) \end{bmatrix}^\top$.

We can write the solution to this linearized problem in terms of the matrix Green's function [45] for Eq. (5.2.2) as $G(z, t; z_0, t_0)$ which propagates the solution $\vec{u}$ from $\vec{u}(z_0, t_0)$

to $\vec{u}(z,t)$ via

$$\vec{u}(z,t) = \int G(z,t;z_0,t_0)\vec{u}(z_0,t_0)dt_0 \tag{5.2.3}$$

where

$$\frac{d}{dz}G(z,t;z_0,t_0) = A(z,t)G(z,t;z_0,t_0) \tag{5.2.4}$$

with

$$G(z_0,t;z_0,t_0) = \mathcal{I}\delta(t-t_0). \tag{5.2.5}$$

We will simplify this notation slightly by suppressing the explicit time dependence. There are two justifications for this:

- we are primarily considering this for applications as a numerical scheme, so we will think of a time discretization and thus vector quantities in the time direction instead of time-dependent quantities, and
- for our applications to this point, A has self-adjoint time derivatives, and so can be treated as a simple matrix instead of a differential operator.

This section is written in the context of a more general formulation; for the specifics of systems governed by the NLS, see the more detailed presentation in Sec. 5.4.1.

Suppressing t-dependence, then, we obtain

$$\vec{u}(z) = G(z; z_0)\vec{u}(z_0) \tag{5.2.6}$$

as the linearized propagation operator where $G(z; z_0)$, using Eq. (5.2.4), solves the linearized differential operator

$$\frac{d}{dz}G(z; z_0) = A(z)G(z; z_0)$$

with

$$G(z_0; z_0) = \mathcal{I}. \tag{5.2.7}$$

Up to this point we have made no modifications to the existing scheme to include the detector, but we have reformulated it slightly to accommodate doing so. We will first consider a simple integrate-and-dump detector given by

$$I = \int_{-T/2}^{T/2} |U(z_L, t)|^2 dt \tag{5.2.8}$$

in the full nonlinear version, where we suppose that the detector is applied at the point $z = z_L$ in the transmission line. Linearizing the detector, again via $U = U + \epsilon u$, we obtain the linearized detector

$$\Delta I = 2 \int_{-T/2}^{T/2} \Re\left(U^* u\right) dt. \tag{5.2.9}$$

If we break this out into real and imaginary parts using

$$U = U_1 + iU_2$$

and

$$u = u_1 + iu_2, \tag{5.2.10}$$

then the linearized detector becomes

$$\begin{aligned}
\Delta I &= 2\int_{-T/2}^{T/2} (U_1 u_1 + U_2 u_2)\, dt \\
&= 2\int_{-T/2}^{T/2} \begin{bmatrix} U_1 & V_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} dt \\
&= 2\int_{-T/2}^{T/2} \vec{U}^\top \vec{u} dt. \qquad (5.2.11)
\end{aligned}$$

This is where the difficulty is introduced by the detector. This is a linear operator that maps $\mathbb{C}^N \to \mathbb{R}$, which is a perfectly sensible element to integrate into our linearized propagation operator Φ, but to apply the Arnoldi method for finding the singular vectors and values numerically we must also define the adjoint operator Φ^*. To make sense of

this, we write the detector as mapping a vector to a vector artificially as

$$\begin{aligned}
\begin{bmatrix} \Delta I \\ \Delta I \end{bmatrix} &= \begin{bmatrix} 2 \\ 2 \end{bmatrix} \int_{-T/2}^{T/2} \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} dt \\
&= \int_{-T/2}^{T/2} \left\{ \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} U_1 & U_2 \end{bmatrix} \right\} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} dt \\
&= \int_{-T/2}^{T/2} \begin{bmatrix} 2U_1(z_1,t) & 2U_2(z_1,t) \\ 2U_1(z_1,t) & 2U_2(z_1,t) \end{bmatrix} \begin{bmatrix} u_1(z_1,t) \\ u_2(z_1,t) \end{bmatrix} dt \\
&= \int_{-T/2}^{T/2} K(t',t;z_1)\vec{u}(z_1,t)dt
\end{aligned}$$

where $K(t',t;z_1)$ is a matrix kernel. This kernel has two special properties, as a result of our construction: $K(t',t;z_1)$ 1) is independent of t', and 2) has two identical rows. Thus the output of applying this matrix kernel is a 2-vector that likewise is independent of t' and has two identical entries. We are now nearly ready to define the adjoint operator containing the detector. First, however, we require an inner product, given by

$$\langle \vec{v}, \vec{u} \rangle = \int_{-T/2}^{T/2} \vec{v}^\top \vec{u} dt. \tag{5.2.12}$$

Using this definition, we have

$$\begin{aligned}\langle \vec{v}, \mathcal{K}\vec{u}\rangle &= \int_{-T/2}^{T/2} \vec{v}^{\top}(t') \int_{-T/2}^{T/2} K(t', t)\vec{u}(t)dt \; dt' \\ &= \int_{-T/2}^{T/2} \left[\int_{-T/2}^{T/2} K^{\top}(t', t)\vec{v}(t')dt'\right]^{\top} \vec{u}(t)dt \\ &= \langle \mathcal{K}^{\dagger}\vec{v}, \vec{u}\rangle\end{aligned}$$

where the † denotes the operator adjoint, so that if we have

$$\mathcal{K}\vec{u} = \int_{-T/2}^{T/2} K(t', t)\vec{u}(t)dt \tag{5.2.13}$$

then

$$\mathcal{K}^{\dagger}\vec{v} = \int_{-T/2}^{T/2} K^{T}(t, t')\vec{v}(t)dt \tag{5.2.14}$$

is the adjoint. If we break out into components, with $\vec{v} = \begin{bmatrix} v_1 \\ \\ v_2 \end{bmatrix}$, we obtain

$$\begin{aligned}\mathcal{K}^{\dagger}\vec{v} &= \begin{bmatrix} U_1(t') \\ \\ U_2(t') \end{bmatrix} \int_{-T/2}^{T/2} \begin{bmatrix} 2 & 2 \end{bmatrix} \begin{bmatrix} v_1(t) \\ \\ v_2(t) \end{bmatrix} dt \\ &= \begin{bmatrix} U_1(t') \\ \\ U_2(t') \end{bmatrix} \int_{-T/2}^{T/2} \left[2v_1(t) + 2v_2(t)\right] dt.\end{aligned}$$

In this case, the adjoint has a particularly simple form; in fact, it is just a simple projection of the "input" (which is the "output" of the direct linearized problem), then multiplied against $\begin{bmatrix} U_1(t') & U_2(t') \end{bmatrix}^\top$, the full nonlinear solution at the output. Note that the solution is truncated outside of $t' \in [-T/2, T/2]$. Other detectors can be handled in very much the same way, though in general the nature of the adjoint as a projection which then multiplies the nonlinear solution does not hold.

If we add the propagation back into the equation, to define the complete operator (and its adjoint), we have

$$\mathcal{K}L\vec{u}(z_0) = \int_{-T/2}^{T/2} K(t', t; z_1) G(z_1; z_0) \vec{u}(z_0) dt \tag{5.2.15}$$

and the adjoint

$$\begin{aligned} (\mathcal{K}\,L)^\dagger \vec{v} &= \mathcal{L}^\dagger\, K^\dagger \vec{v} \\ &= \mathcal{L}^\dagger \begin{bmatrix} U(z_1, t') \\ V(z_1, t') \end{bmatrix} \int_{-T/2}^{T/2} \left[2v_1(z_1, t) + 2v_2(z_1, t)\right] dt. \end{aligned} \tag{5.2.16}$$

Now we have everything required to compute the action $\Phi^*\Phi$ required by the Arnoldi iterations for the singular values and vectors. The action $\Phi^*\Phi$ is given by the action $(\mathcal{K}\mathcal{L})^\dagger \mathcal{K}L = \mathcal{L}^\dagger \mathcal{K}^\dagger \mathcal{K}L$, according to Eq. (5.2.15) and Eq. (5.2.16) along with the linear propagators specified by $\mathcal{L}$ and $\mathcal{L}^\dagger$, which will be specified explicitly for specific problems shortly. In this particular case, we have a bit of luck in that the adjoint linear operator $\mathcal{L}^\dagger$ merely propagates the nonlinear output back to the input, multiplied by a scalar. Thus,

in this case, we need only make the calculation once and carry the scalar obtained by the projection through the linear operator.

5.2.1. Detectors with filtering and the SVD

In the previous section we have demonstrated how to apply a simple integrate-and-dump detector to our problem. Many different types of detectors can be included in much the same way. However, one aspect of common detection schemes which requires a bit more work is filtering. We consider the problem with optical filtering and an integrate-and-dump detector, where we seek the amplitude mode. That is, we want to transform our signal into Fourier space, apply the filter, bring the signal back to the time domain where we apply the integrate-and-dump detector:

$$I = \int_{-T/2}^{T/2} |\mathcal{F}^{-1}[f(\omega)\mathcal{F}[U(t)]]|^2 dt \tag{5.2.17}$$

where $f(\omega)$ is our filtering function. For our problem, T is sufficiently large, relative to the pulse, that we approximate this in the Fourier domain as

$$I = \int_{-\infty}^{\infty} |f(\omega)\tilde{U}(\omega)|^2 d\omega \tag{5.2.18}$$

by Parseval's theorem [46]. Following then as in the previous section, the action of the operator $\Phi^*\Phi$ is given by

$$\mathcal{L}^\dagger\mathcal{K}^\dagger\mathcal{K}\mathcal{L}\vec{\tilde{u}} = \mathcal{L}^\dagger \begin{bmatrix} f^2\tilde{U}_1(z_L,\omega) \\ f^2\tilde{U}_2(z_L,\omega) \end{bmatrix} \int_{\omega_1}^{\omega_2} d\omega' \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\cdot \int_{\omega_1}^{\omega_2} d\omega \begin{bmatrix} 2f^2\tilde{U}_1(z_L,\omega) & 2f^2\tilde{U}_2(z_L,\omega) \\ 2f^2\tilde{U}_1(z_L,\omega) & 2f^2\tilde{U}_2(z_L,\omega) \end{bmatrix} \mathcal{L}\vec{\tilde{u}} \tag{5.2.19}$$

assuming $f(\omega) \in \mathbb{R}$, where the nominal solution in Fourier space is broken out into real and imaginary parts as $\tilde{U}(z,\omega) = \tilde{U}_1(z,\omega) + i\tilde{U}_2(z,\omega)$.

Of course, we might wish to employ a more sophisticated detector employing filtering which is not amenable to this simple change into the frequency domain; perhaps electrical filtering, or some more sophisticated design. Consider a more general detector function acting on the filtered signal $U_f = \mathcal{F}^{-1}[f(\omega)\mathcal{F}[U(t)]]$ given as

$$I = \int_V \mathcal{J}[U_f]\, dt$$

for some detector operator $\mathcal{J}$ on the integration domain V. (The detector we are using is simply this detector with $\mathcal{J}[U] = |U|^2$ and $V\colon\ t \in [-T/2, T/2]$.) Then if we consider the linearization of $\mathcal{J}$ via

$$\mathcal{J}[U + \epsilon u] = \mathcal{J}_0[U] + \epsilon\mathcal{J}_1[u] + \dots$$

then we have the detector operator

$$\begin{aligned} I &= \int_V \mathcal{J}[(U+\epsilon u)_f]\,dt \\ &= \int_V \mathcal{J}[\mathcal{F}^{-1}[f(\omega)\mathcal{F}[U+\epsilon u]]]\,dt \\ &= \int_V \mathcal{J}[\mathcal{F}^{-1}[f(\omega)(\tilde{U}+\epsilon\tilde{u})]]\,dt \\ &= \int_V \mathcal{J}[U_f+\epsilon u_f]\,dt \\ &= \int_V \{\mathcal{J}_0[U_f]+\epsilon\mathcal{J}_1[u_f]+\dots\}\,dt \end{aligned}$$

and the linearized detector operator

$$\Delta I = \int_V \mathcal{J}_1[\mathcal{F}^{-1}[f(\omega)\mathcal{F}[u]]]\,dt$$

where $\mathcal{J}$,$\mathcal{F}$ and $\mathcal{F}^{-1}$ are all linear operators and we may define the necessary operator $\mathcal{K}$ as done previously when the specific form of the detector operator $\mathcal{J}$ is given.

5.3. Mode identification and alignment

Two so far unaddressed issues arise from the identification and alignments of the modes given by the SVD, or the singular vectors. The primary tool for identifying modes is to examine the corresponding singular values; generally there is some (small) number of modes with singular values appreciably greater than 1, and a large number of modes with singular values at or smaller than 1. As the Arnoldi factorization only estimates the

largest magnitude eigenvalues (and eigenvectors) , we are entirely unconcerned with these smaller values. However, suppose that our vector of ordered largest singular values looks like

$$[\ 2.03\ \ 2.0\ \ 1.05\ \ 1.0\ \ 1.0\ \ .9\ ]. \tag{5.3.1}$$

We might expect to easily be able to distinguish at each iteration the modes with singular values around 2 from those with values near 1. But we must also distinguish the two modes with singular values ≈ 2. In a slightly different stochastic realization, on the next iteration, for example, the perceived ordering of these two larger-value modes may no longer be consistent. It is important to keep the modes separate, as they must correspond with assigned biasing strengths that are being adjusted by the cross-entropy method. Convergence of the method will be much more difficult, if not impossible, if the mode being used with a particular biasing is being confused with a different mode.

Generally, for the lightwave systems we are considering, the pulse modes can be distinguished using nothing more than the inner product (or, a pair of inner products). For other potential applications, a more sophisticated system might have to be employed. Suppose we are using the singular value decomposition to determine the modes of a pulse in a lightwave system where the detector is specified such that there are two modes with singular values above one, but otherwise indistinguishable from one another, and in this case they correspond roughly to the adjoint amplitude mode and the frequency mode, u_A and u_Ω respectively. We might find modes that look like noisy versions of Fig. 5.3.

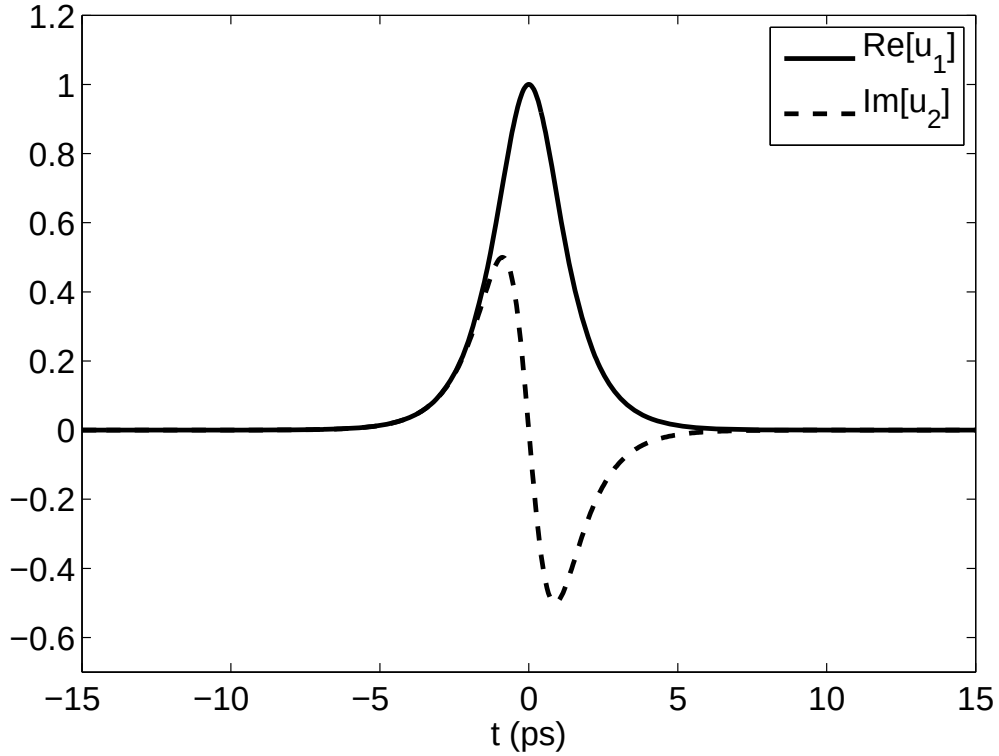

Figure 5.3. **Numerical modes in need of identification**: Two hypothetical modes extracted from the singular value decomposition with similar (and indistinguishable) singular values. The undisplayed portions, Re[u_2] and Im[u_1], are near zero.

However, we are not sure which mode is u_A and which is u_Ω based on the singular values, because in this situation the singular values are close enough to be unreliable identifiers in a system where the singular values vary stochastically and are estimated. We know only that we have two modes, or singular vectors, u_1 and u_2. We need not necessarily know which is the amplitude mode and which is the frequency mode (or, in fact, that they correspond with pulse parameters in this precise way). We do need to know that we are using the proper biasing coefficients; that is, we need to know that u_1 is the same as it

was in the last trial, and will be in the next trial, so that the modes are not mixed up. In this scenario, it is easy to distinguish them, because

$$\langle u_1, u \rangle \approx 2$$

and

$$\langle u_2, u \rangle \approx 0. \tag{5.3.2}$$

where u is the full nonlinear pulse. While these values will also vary stochastically somewhat, they are no longer nearly equal and easy to distinguish for application of proper biasing coefficients.

The second difficulty is mode alignment, which is essentially a sign problem. That is, the modes may come out in different stochastic realizations with opposite signs, which must also be corrected to avoid confusing the cross-entropy method. This outcome is due to the fact that both increases and decreases are large changes, and we have asked for the largest *magnitude* singular vectors. Consider that due to the stochastic nature of the process, we might obtain as our numerical mode either u_1 or u_1' as pictured in Fig. 5.4. These modes both have the same (or very similar) singular values, and so cannot be distinguished in that way, and our inner-product scheme doesn't provide any additional information in this case, either, as

$$\langle u_1, u \rangle \approx 0$$

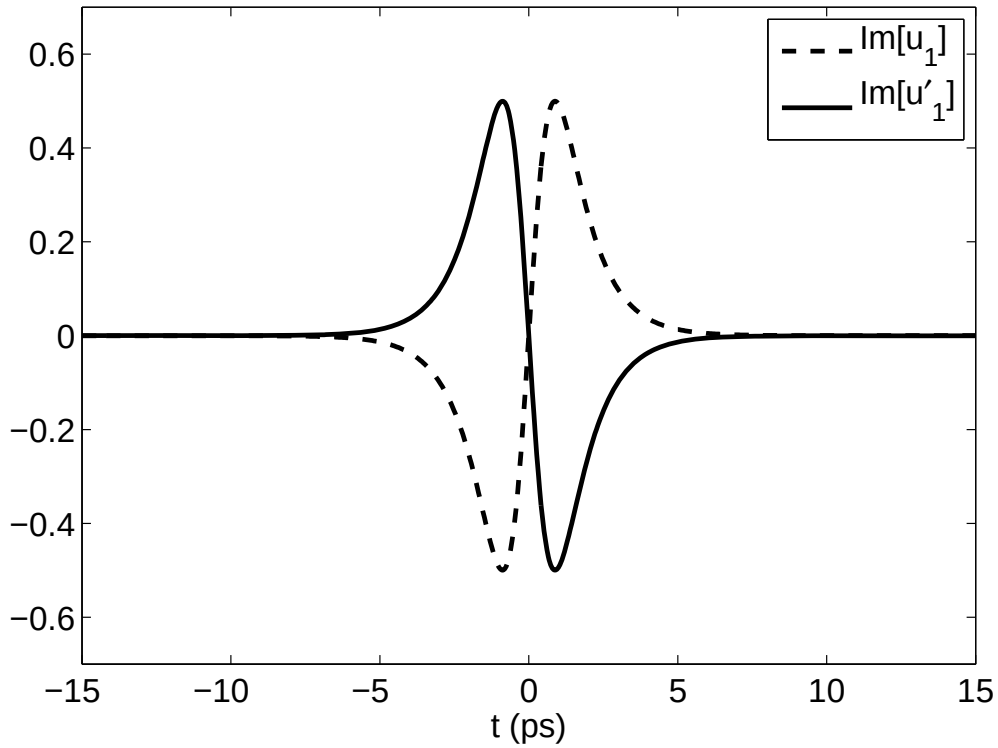

Figure 5.4. **Numerical modes in need of alignment**: Two hypothetical modes extracted from the singular value decomposition with similar (and indistinguishable) singular values as well as indistinguishable inner products with the pulse shape.

and

$$\langle u_1', u\rangle \approx 0. \tag{5.3.3}$$

However, if we use a time-weighted inner product, we can distinguish the sign of these modes for alignment purposes. If we define this second inner product as

$$\langle v_1, v_2\rangle_t = \int (t-T) v_1 v_2^* \, dt, \tag{5.3.4}$$

where T is the pulse position. For the amplitude problem, it is reasonable to assume $T = 0$; for other problems, T can be estimated using the position moment integral on the full nonlinear pulse:

$$T = \frac{\int t|u|^2 dt}{\int |u|^2 dt}$$

Then we can use this to align the modes pictured in Fig. 5.4 by way of

$$\langle u_1, u \rangle_t \approx i$$

and

$$\langle u_1', u \rangle_t \approx -i. \tag{5.3.5}$$

Numerical output modes from the singular value decomposition then can be identified and correctly aligned with the appropriate biasing coefficients using three different measurements: the corresponding singular value, the inner product of the output mode with the pulse shape $\langle u_1, u \rangle = \int u_1 u^* dt$ and the time-weighted inner product of the output mode with the pulse shape $\langle u_1, u \rangle_t = \int (t - T) u_1 u^* dt$. When one is considering noise perturbations of a single pulse, these three measures provide a unique separation of the linear modes into distinct regions of 5-space (the singular values are real, and the two inner products are potentially complex) such that stochastic variations away from the ideal value does not cause the numerical modes to be misidentified, and as such the cross-entropy method and the later importance-sampled Monte Carlo simulations can efficiently

use these modes. Of course, for more complicated situations, it might be necessary to add additional criteria in order to separate the modes.

5.4. Application to a specific dispersion-managed system

As a demonstration and validation of the method we will apply it to a problem in rare-event simulation with previously published results. For this purpose we select sample system (b) from a paper by Li, Spiller and Biondini [21], who consider large-amplitude pulse distortions in a dispersion-managed communication system. There are several aspects of this problem that make it an ideal test case. First, the authors have demonstrated a semi-analytic method for generating the biasing distributions, ideal for comparison with a fully-numerical method. Second, the resulting PDF deviates significantly from a normal distribution. And finally, in some sense this problem is on the cutting-edge of the type of problem that may be solved with this type of semi-analytic method; in the previous work, a complicated semi-analytical procedure was required to determine the biasing for importance-sampled Monte Carlo simulations. Here we reproduce the results precisely as described in [21], except that instead of using the averaged, nonlocal governing equation required for the semi-analytic method, the dispersion-managed nonlinear Schrödinger equation (DMNLS), we do so directly from the NLS equation with a variable dispersion coefficient.

As described previously, dispersion-managed propagation is described by the NLS with a variable dispersion coefficient,

$$\frac{\partial U}{\partial z} = \frac{i}{2}\, d(z/z_a)\, \frac{\partial^2 U}{\partial t^2} + i\gamma\, |U|^2 U \tag{5.4.1}$$

where the dispersion $d(z/z_a)$ is a periodic function over the dispersion map length z_a given by the periodic extension of

$$d(z/z_a) = D_{avg} + \begin{cases} 4sD_{avg}, & \left|z/z_a - \frac{1}{2}\right| < \frac{1}{4} \\ -4sD_{avg}, & \left|z/z_a - \frac{1}{2}\right| \geq \frac{1}{4} \end{cases}, z/z_a \in [0,1] \tag{5.4.2}$$

The model calls for amplified spontaneous emission (ASE) noise to be added periodically at evenly spaced amplifiers, with the amplifier spacing corresponding to the dispersion map period, z_a. The detector in the system is given by

$$I = \int_{-T}^{T} |U_f(z_L, t)|^2 dt \tag{5.4.3}$$

where

$$U_f(z_L, t) = \mathcal{F}^{-1}[f(\omega)\mathcal{F}[U(z_L, t)]].$$

where $f(\omega)$ is the filter transfer function. The initial condition for this problem is the *dispersion managed soliton* (DM soliton) with amplitude $\lambda = 2$ computed numerically following Sec. 3.4.

Proceeding as described previously, we use the SVD and the CEM to build a target set of biasing distributions to cover the relevant range of output energies. Using these biasing

distributions, we can make use of multiple importance sampling Monte Carlo simulation to generate the final energy PDF.

We pose the biasing problem in the following way: at each amplifier location, $z = nz_a$ $(n = 1 \ldots N)$, we add to our solution $U(nz_a, t)$ the mode of the system from this location along with a *biasing coefficient* η_n such that

$$U(nz_a, t) \leftarrow U(nz_a, t) + \eta_n u_A(t) \tag{5.4.4}$$

where the η_n will be determined by the CEM and the $u_A(nz_a, t)$ will be determined by the SVD. Note that the η_n play the roll of the $b_{i,j}$ from the method description for a general lightwave system.

The constrained optimization problem we wish to have the CEM solve is then to reach some target in detector value in the most likely way. Observe that $\eta_n u_A(nz_a, t)$ is a noise projection of a Gaussian white noise source; maximizing the probability of this event actually occurring is then equivalent to minimizing the exponent

$$\sum_n \int |\eta_n u_A(nz_a, t)|^2 dt = \sum_n \eta_n^2 \tag{5.4.5}$$

where the mode norm is 1 due to normalization, subject to

$$I = \int_{-T}^{T} |U_f(z_L, t)|^2 dt \geq \hat{I} \tag{5.4.6}$$

where $\hat{I}$ is our target value for the detector output. The performance function for the CE method ($p(Z)$) is then given for this system as

$$p = I = I(\vec{\eta}) \tag{5.4.7}$$

where we write explicitly the dependence on $\vec{\eta} = [\eta_1, \eta_2, \ldots, \eta_N]$. Selecting $\hat{I}$ is an important step in building a set of biasing distributions. In general, it is necessary to use several values of $\hat{I}$ which cover the areas of system state space which one is interested in simulating. If the desired output isn't an entire PDF but merely the probability associated with a specific output condition, it may be possible to use only one biasing target and set the value of $\hat{I}$ to correspond to this specific condition.

For determining the biasing distributions, the system is noiseless aside from the biasing coefficients, which are randomly assigned by the CE method. The performance function therefore only depends upon $\vec{\eta}$. As far as the CE method is concerned, this is a black box procedure - input N randomly drawn biasing coefficients, output detector performance. Assessing the optimal distribution parameter is a simple matter of computing the first moment of the sample distribution by using the moment estimator Eq. (2.2.13). The details of applying the SVD to compute the required mode u_A are an important part of this, described in the next section.

5.4.1. Computing the SVD for the dispersion-managed system

In order to apply our new method, we must be able to compute the modes of the dispersion-managed system at each amplifier location. To that end, we start with the NLS equation, Eq. (5.4.1) and take $U = u_0 + \epsilon \Delta u$ to obtain the linearized equation

$$\frac{\partial \Delta u}{\partial z} = \frac{i}{2} d(z/z_a) \frac{\partial^2 \Delta u}{\partial t^2} + i\gamma(2\Delta u |u_0|^2 + \Delta u^* u_0^2). \tag{5.4.8}$$

It is simplest to split this complex equation into real and imaginary parts and deal with a vector system, as we will need to take the adjoint shortly. We split into real and imaginary parts via $\Delta u = u_1 + i u_2$ and $u_0^2 = \xi + i\upsilon$ to obtain the vector propagation operator

$$\frac{\partial}{\partial z} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -\gamma\upsilon & -\frac{1}{2} d(z/z_a) \frac{\partial^2}{\partial t^2} - 2\gamma |u_0|^2 + \gamma\xi \\ \frac{1}{2} d(z/z_a) \frac{\partial^2}{\partial t^2} + 2\gamma |u_0|^2 + \gamma\xi & \gamma\upsilon \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.4.9}$$

so that now we can define the operator $\mathcal{L}$ in Eq. (5.2.19) as propagation from $z = z_c$ to $z = z_L$ via Eq. (5.4.9) where z_c is the current location. For example, $z_c = j z_a$ if we are at the j^{th} amplifier.

Recall that to define Φ^* we require $\mathcal{L}^\dagger$ as well as $\mathcal{K}^\dagger$. For the detector we are using here, $\mathcal{K}^\dagger$ is given in Eq. (5.2.19). To obtain $\mathcal{L}^\dagger$, we consider the matrix Green's function which solves Eq. (5.4.9) and find $\mathcal{L}^\dagger$ as the propagation operator for the adjoint Green's function. Because the differential operators in the matrix in Eq. (5.4.9) are self-adjoint,

computing the adjoint operator is relatively simple.

$$-\frac{\partial}{\partial z}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -\gamma\upsilon & -\frac{1}{2}d(z/z_a)\frac{\partial^2}{\partial t^2} - 2\gamma|u_0|^2 + \gamma\xi \\ \frac{1}{2}d(z/z_a)\frac{\partial^2}{\partial t^2} + 2\gamma|u_0|^2 + \gamma\xi & \gamma\upsilon \end{bmatrix}^* \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} -\gamma\upsilon & \frac{1}{2}d(z/z_a)\frac{\partial^2}{\partial t^2} + 2\gamma|u_0|^2 + \gamma\xi \\ -\frac{1}{2}d(z/z_a)\frac{\partial^2}{\partial t^2} - 2\gamma|u_0|^2 + \gamma\xi & \gamma\upsilon \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5.4.10)$$

Thus $\mathcal{L}^\dagger$ is given by propagation in the reverse direction from $z = z_L$ to $z = z_c$ via Eq. (5.4.10). Thus, with a suitably defined filter $f(\omega)$ (defined for our dispersion managed system in Sec. 5.6) and with Eq. (5.2.19), we now have the action $\Phi^*\Phi$ required to compute the SVD using the Arnoldi method applied to the associated eigenvalue problem.

Numerically, these linearized propagators are solved using the IFRK4 method, while the solution to the full nonlinear problem is obtained via the split-step Fourier method. It is necessary to have the linear solver z-step size be an integer multiple of the nonlinear solver step size, because at each solution point for the linear solver, the nonlinear solution must be available (ideally, pre-calculated and stored). In practice, the linear solutions require a more accurate numerical solution than the nominal solution, and so using a 1:1 step size ratio with the same numerical method either does not achieve the required accuracy in the linear solver or requires unnecessary computational effort in the nonlinear solver. There are two solutions to this problem: 1) use a lower (e.g., 1:2) step size ratio, or 2), use a higher-order method for the linear solver. We opt for the latter.

As an additional numerical concern, consider the dependence of the linearized problems on the full nonlinear problem. The direct linearized problem, as well as the adjoint problem, depend upon the nonlinear solution (via $u_0^2 = \xi + i\upsilon$), but it is inefficient to compute this nonlinear solution each time it is required by the linearized problems. The linearized problems must be solved many times as part of the iterative application of the operator action in solving the eigenvalue problem. However, the nonlinear problem need only be solved once for each application of the SVD (and thus the eigenvalue problem) and merely stored and referenced by each linear (and adjoint linear) numerical solve.

5.5. A test case for the SVD-CE-IS method

We begin with a test of our method based on the solution from the semi-analytical method. The full numerical method is computationally less efficient, but serves as a good test of the properties of the technique.

We will consider the system as described earlier, with no filter ($f(\omega) = 1$), but with a slight modification. Instead of adding just one mode at each amplifier, as we have described, we will also add the frequency mode $u_\Omega(z, t)$ so that Eq. (5.4.4) becomes

$$U(nz_a, t) \leftarrow U(nz_a, t) + \eta_n u_A(nz_a, t) + \zeta_n u_\Omega(nz_a, t) \tag{5.5.1}$$

and the CEM now controls two sets of biasing coefficients, both the η_n and the ζ_n. We will show that the CEM adjusts η_n properly, but does not change ζ_n since the output of the detector does not depend upon frequency changes.

Of course, our minimization quantity changes so that Eq. (5.4.5) becomes

$$\sum_n \int \left\{ [\eta_n u_A(nz_a, t)]^2 + [\zeta_n u_\Omega(nz_a, t)]^2 \right\} dt \tag{5.5.2}$$

and also the performance function such that Eq. (5.4.7) becomes

$$p = I = I(\vec{\eta}, \vec{\zeta}). \tag{5.5.3}$$

To compute the frequency mode $u_\Omega(z, t)$ for the biasing, we must solve an independent eigenvalue problem (thus a total of two SVD solves at each amplifier) with a different detector and thus a different linearized detector operator, $\mathcal{K}$. Normally, we would only use the one detector which gives the quantity of interest, and this would determine all of the modes upon which the desired output quantity depends. Here we are artificially adding an extra detector so we can include an additional mode that would not normally be present. We define $\mathcal{K}_\Omega$ as the detector for this mode, which is based on the position moment integral

$$T_0 = \frac{\int t|U(N_a z_a, t)|^2 dt}{\int |U(N_a z_a, t)|^2 dt}. \tag{5.5.4}$$

Here one of the connections to soliton perturbation theory and the conserved quantities of the NLS is apparent. If we use as the detector an integral conserved quantity of the NLS (corresponding to a system invariance via Noether's theorem), the detected quantity is the soliton parameter and the largest singular mode is the soliton mode. For the detector Eq. (5.5.4), if a soliton pulse shape is used, the integral conserved quantity is the soliton

position T and the largest singular input mode is the soliton frequency mode u_Ω, since frequency perturbations turn into position perturbations upon propagation. Of course, in this case, we are not using soliton pulses, and the corresponding "pulse parameter" and singular mode must be computed numerically.

Proceeding as in Sec. 5.2, we obtain the linearized operator

$$\mathcal{K}_\Omega = \begin{bmatrix} AU_1(z_1,t) + BtU_1(z_1,t) & AU_2(z_1,t) + BtU_2(z_1,t) \\ AU_1(z_1,t) + BtU_1(z_1,t) & AU_2(z_1,t) + BtU_2(z_1,t) \end{bmatrix} \tag{5.5.5}$$

where for convenience and compactness we have defined

$$A = \frac{-2\int t'\vec{U}^*(z_1,t')\vec{U}(z_1,t')dt'}{\left[\int \vec{U}^*(z_1,t')\vec{U}(z_1,t')dt'\right]^2}$$

and

$$B = \frac{2}{\int \vec{U}^*(z_1,t')\vec{U}(z_1,t')dt'},$$

and the adjoint $\mathcal{K}^\dagger$ is obtained by taking the transpose.

In principle, the addition of an entirely separate second mode should have little or no impact on the solution, other than the computational efficiency. The detector that we use as the performance function for the system as a whole with the CEM is the detector which renders u_A in a natural way, while u_Ω has been introduced artificially. Thus we expect $\vec{\eta}$ to be unchanged aside from a slower convergence rate as a result of the computational inefficiency, and we expect that each component of $\vec{\zeta}$ should converge to zero. The most

efficient perturbation mode should be u_A - this is the definition of what we are asking the singular value decomposition to provide.

Note also that this is not necessarily the appropriate way to treat a system with two (or more) relevant modes. If the detector model is an accurate physical model which outputs the quantities of interest, there is no need to artificially introduce additional detectors to isolate additional modes. In fact, the idea of identifying the numerically computed modes as ones roughly corresponding to the pulse parameters in a manner similar to soliton perturbation theory is not the correct approach. The modes which we are referring to as the amplitude and frequency modes are only rough analogs to those modes in soliton systems. The numerically determined modes, in practice, are linear combinations of these fundamental modes, if they exist, in appropriate combination to produce large changes in the desired output quantity. Identifying the physical meaning of the modes or their number is not necessary; the singular values corresponding to these modes (via the singular vectors) indicate the magnitude of changes, and the overall significance, induced by perturbations represented by that singular vector. The method need only account for the modes with sufficiently large singular values.

Proceeding with our test (with the artificially introduced frequency mode), the algorithm to generate a CE method biasing trial is then

(1) Begin with the DM soliton $U(0, t)$. Set $j = 0$.

(2) Propagate $U(z_j, t)$ via Eq. (5.4.1) from z_j to z_{j+1}.

(3) If $j + 1 = L$, goto 7.

(4) Apply the SVD to $U(z_{j+1}, t)$ from z_{j+1} to z_L to obtain normalized modes ($u_A(z_{j+1}, t)$ & $u_\Omega(z_{j+1}, t)$).

(5) Add modes to U_{j+1} according to random values η_{j+1} and ζ_{j+1} drawn from the appropriate distributions in $\hat{g}^k(\cdot)$.

$$U_{j+1} \leftarrow U_{j+1} + \eta_{j+1} u_A(z_{j+1}, t) + \zeta_{j+1} u_\Omega(z_{j+1}, t)$$

(6) $j \leftarrow j + 1$. Goto 2.

(7) Apply detector with performance metric $p = \int |U(z_L, t)|^2 dt$.

This process is of course repeated many times in the CE method, and the coefficients η_n and ζ_n are updated by calculating the means of the high-performance quantiles, based on the p values of many realizations. When these η_n and ζ_n values are returned by the biasing phase, (i.e. when the CE method has converged), they can be used as the biasing coefficients for importance sampled Monte Carlo simulations.

A sample simulated system realization for the CEM part of the algorithm is given in Fig. 5.5. Recall that the biasing trials do not have full-bandwidth noise, as the noise dimensionality has been reduced to two directions per amplifier by the SVD. In this case, $\hat{I}$ is set for a 50% decrease in pulse energy. To illustrate these modes as SVD output, for the last segment of this trial, the pulse profile is given in Fig. 5.6 and the corresponding modes for the last segment are shown in in Fig. 5.7. One can see that the SVD modes, which are rough analogs of the soliton amplitude and frequency modes, roughly correspond to these roles.

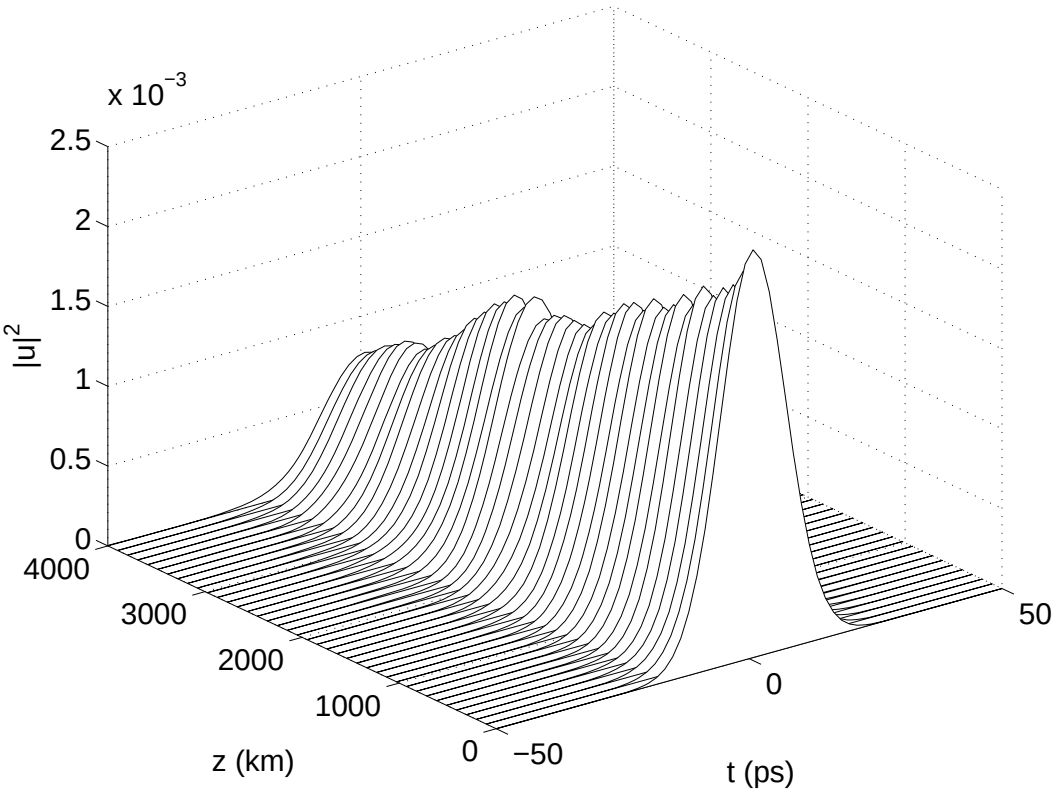

Figure 5.5. **Sample Monte Carlo trial**: Plot of a single Monte Carlo trial, biased in this case toward a decreased energy target.

The results showing the overall convergence of the CE-determined biasing coefficients for this problem are given for $N = 1000$, 2000, 4000 and 8000 in Fig. 5.8. One can see that the results improve as expected as N increases, and agree qualitatively with our expectations for the solution. The frequency mode coefficients ($\vec{\zeta}$) converge to zero as N increases at each level of the cross-entropy method, as this mode has been artificially introduced, and the results bear this out. The nonzero biasing coefficients of the amplitude mode ($\vec{\eta}$) also agree qualitatively with our expectations, though precise quantitative

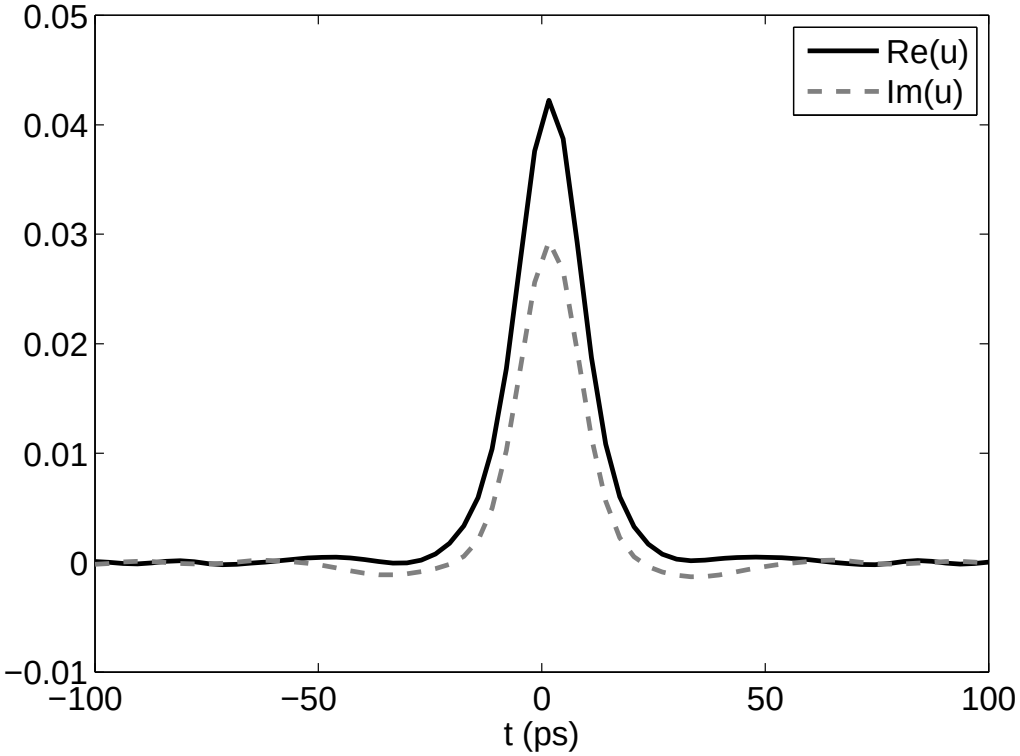

Figure 5.6. **Pulse profile at amplifier**: This is the pulse profile from a sample trial at the last amplifier, where the final set of modes are generated.

comparison between methods would at the least require a very large number of trials, and might still be impossible due to the different assumptions made by the two methods in order to determine the biasing. The method passes these simple tests.

This is only the biasing problem, of course. Because we want to do multiple importance sampling, and combine the results using the balance heuristic, we must use the described method and repeat to generate the biasing for each biasing target. Then these results can be used as the biasing distributions for multiple importance sampled Monte Carlo simulations. Because this test problem is merely designed to test the generation

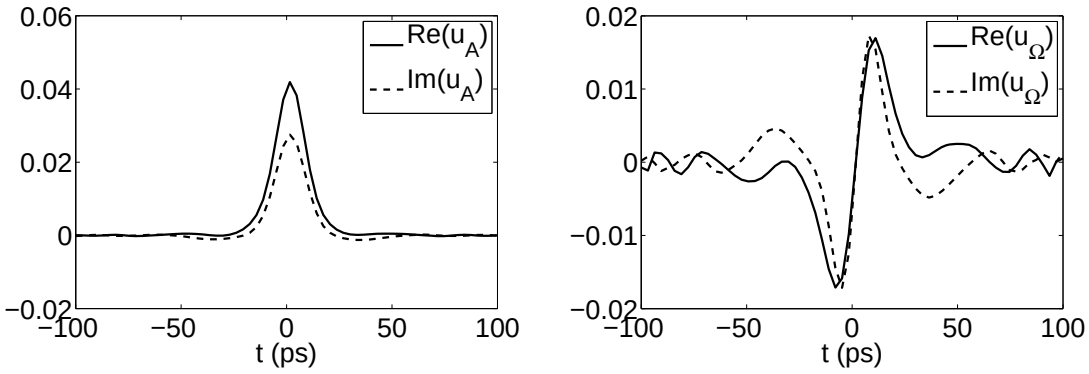

Figure 5.7. **Numerically extracted modes**: The modes extracted numerically for the last segment of a sample trial, from the pulse in Fig. 5.6. Left, the amplitude mode. Right, the frequency mode.

of biasing distributions, we have not yet generated a full set of biasing distributions and run the final simulations. In the next section we will eliminate $\vec{\zeta}$ and u_Ω and only use the modes determined by the SVD from the detector of interest, as originally intended, and demonstrate the full algorithm.

5.6. Full algorithm application to dispersion-managed system

We now apply the full method to our dispersion-managed sample problem, problem (b) from Li *et al* [21]. For this system, the total propagation distance is 4000 km, the average dispersion is 0.15 ps^2/km with a dispersion map strength $s = 4$ and dispersion map period $z_a = 100$km. The nonlinear coefficient is 1.7 (W$-$km)$^{-1}$, the fiber loss is 0.21 dB/km, and the DM soliton amplitude is $\lambda = 2$. The noise spontaneous emission factor is 1.5.

We remove the frequency mode from the algorithm and restore Eq. (5.4.4), so that we regain our computational efficiency, as well as install the filtering function appropriate

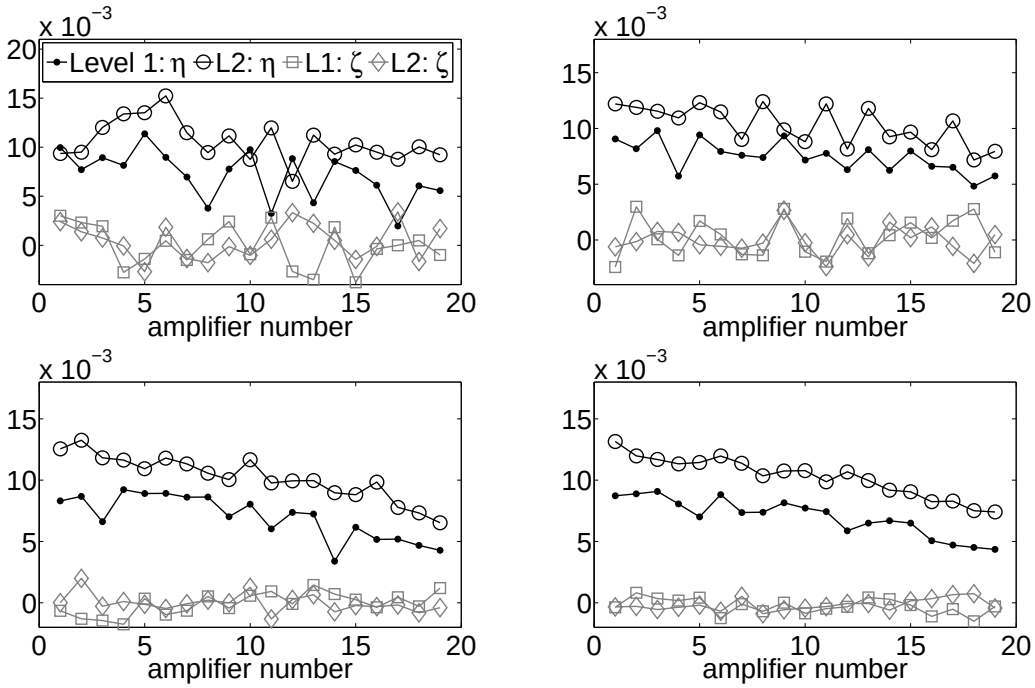

Figure 5.8. **Convergence of biasing coefficients for the amplitude problem**: The results of the convergence test for the SVD-CE-IS method where two modes are biased but only one should impact the detected measure, the pulse amplitude. The biasing for the frequency mode converges to zero, and the biasing for the amplitude mode converges to the amplitude-only biasing solution. In each figure, both sets of biasing coefficients are displayed for two different levels of the CEM. Left to right, top to bottom, the number of trials per CEM level increases from $1,000$ to $2,000$ to $4,000$ to $8,000$.

for a 10 GHz Gaussian optical filter,

$$f(\omega) = \exp\left(-\frac{2\ln(2)\omega^2}{(2\pi 5/T_L)^2}\right), \tag{5.6.1}$$

where T_L is the length of the computational domain, here expressed in picoseconds [47].

Including the filter does affect the shape of the modes, as the singular value decomposition now finds perturbations that maximize the change in the *filtered* detector output. With a very strong, narrow filter one can imagine that the filter could completely dominate the modes. Consider, for example, a filter which passes only a single Fourier mode. Clearly the application of such a filter would drastically alter the computationally-determined mode. With the practical filter we use, the change is more subtle. Fig. 5.9 shows the amplitude mode from a sample trial in the filtered system, along with the (unfiltered) pulse shape at the amplifier where the mode was computed. Both the pulse and the mode have been phase-shifted so that the pulse center-phase is 0. The addition of the filter to the system has changed the "amplitude mode" from a copy of the pulse profile to a different shape, which best reflects the performance increase in the *filtered* detector. Note that this pulse and mode are from a biasing trial, so full-bandwidth noise is not present, only the noise projected in the direction of each mode.

Using the cross-entropy method with the singular value decomposition, we generate the biasing distributions for this problem. In principle, each biasing target requires a separate application of the cross-entropy method to determine the biasing distribution, though they can build off one another, in that "further out" biasing targets can use nearer ones as a starting point. For this problem, we will use 6 biasing distributions for the normalized output energy at the end of the system. If the unperturbed value of the normalized output energy is 1 (by virtue of the normalization), we might assign our six biasing targets to correspond to normalized output energies $\hat{I} = 0.25, 0.5, 0.75, 1.25, 1.5, 1.75$ if this is the region of the system state space and PDF with which we are concerned. The spacing and

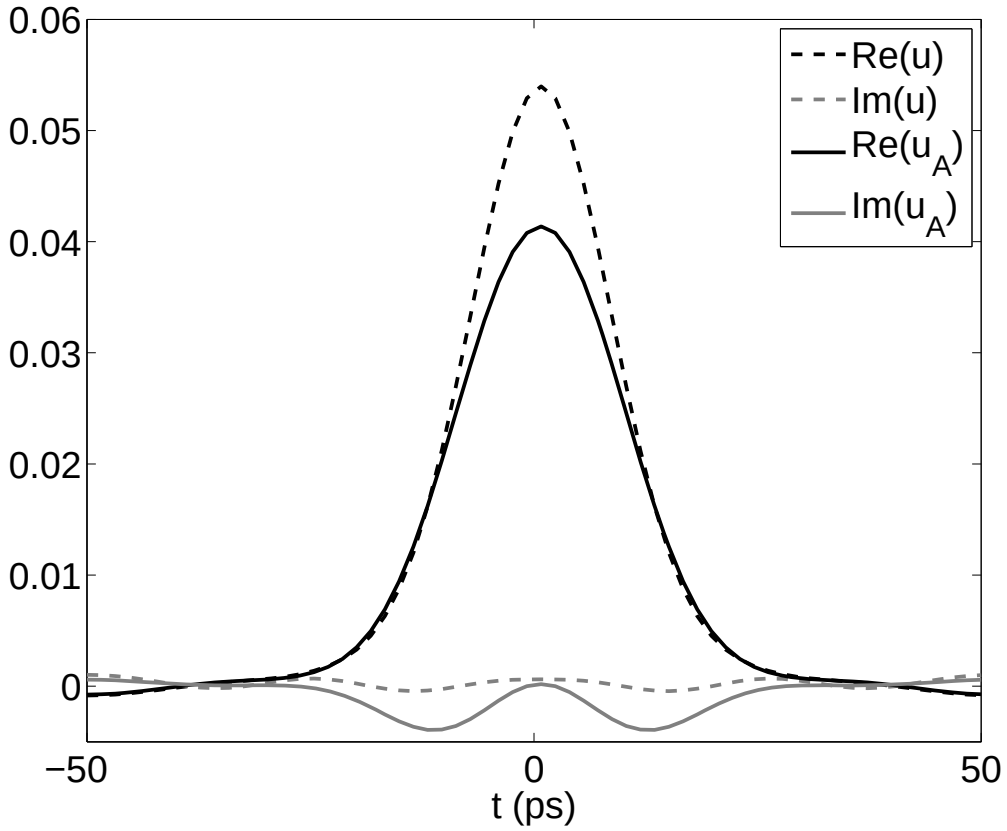

Figure 5.9. **Amplitude mode computed with filtering detector**: The amplitude mode (solid curves) overlaid with the underlying pulse shape (dashed curves) when the mode is extracted from the filtered detector. Both real (black) and imaginary (gray) parts displayed.

density of the biasing targets is problem dependent, as are the number of trials to be used per biasing target. Of minor note is that we have only formulated the CE algorithm for an increasing function. To create the biasing targets for lower energy points, one can simply redefine the performance function with a sign flip. For our example, nothing more than $p \leftarrow -p$ is required.

A useful possible modification is that intermediate iterations of the cross-entropy method, when $p < \hat{l}$, may provide acceptable intermediate biasing targets. Consider that we set $\hat{l} = 1.75$, and the cross entropy method requires 5 iterations to reach this specified value. We can check the performance values of these intermediate iterations to see if there exist distributions with performance values sufficiently close to 1.25 and 1.5 to be used.

Alternatively, we might first bias to $\hat{l} = 1.25$ and obtain that biasing target, and then use this result as the starting place to iterate higher toward 1.5 and 1.75. This approach results in more precise locations for the biasing targets in terms of proximity to the prescribed values.

Using the first approach, where incidental CE method levels are used as the biasing targets, we apply the method twice to generate a full set of biasing targets, reproduced in Fig. 5.10 with the mean of the biasing distribution on the vertical axis and the amplifier number on the horizontal. Note that some data smoothing has been employed to reduce the number of samples required by the CEM. Also displayed is a 7^{th} "biasing" target, the unbiased case. Because this plot is of the distribution means and we are using mean-shifted normal distributions, zero is the unbiased case.

Finally, we can use these biasing targets along with the SVD in multiple importance-sampled Monte Carlo simulations. The complete SVD-CE-IS algorithm, then, has been fully described for this practical problem. First, we compute the biasing distributions required for importance sampling by using the CE method to solve the optimization problem

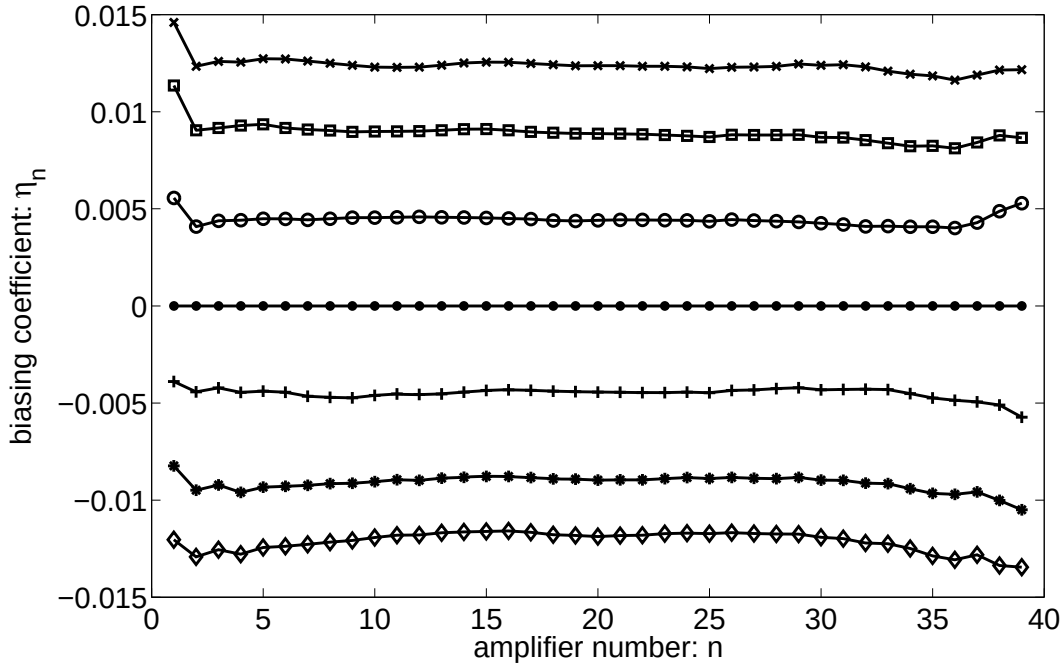

Figure 5.10. **Biasing coefficients for DM soliton system**: The biasing coefficients for the amplitude mode in the dispersion-managed soliton system, for each of the 7 (6 nonzero) biasing targets.

in the SVD-reduced problem. The optimization problem specifies reaching a certain target in the detected quantity ($\hat{I}$) while maximizing probability, i.e. minimizing the biasing strength ($\sum \eta_n^2$). For each biasing trial, we solve the nominal, fully nonlinear problem over the length of the system, and store it to be referenced by each iterative linear and nonlinear application of the action $\Phi^*\Phi$ required by the SVD. The SVD then returns the singular values and modes. These modes are identified and aligned by their singular values, as well as the pair of inner products discussed in Sec. 5.3. The amplitude mode u_A is then multiplied by the biasing strength η_n and added to the propagating solution. These trials

are repeated until the CE method returns the converged biasing distributions, under which $p > \hat{I}$ is no longer a rare event.

These biasing distributions are then used in the second phase of the SVD-CE-IS method, the importance sampled Monte Carlo phase. In this phase, full-bandwidth noise is introduced added to the signal, but it is mean-shifted according to the biasing distributions. The performance values that result are very rare events under the original, mean-zero distributions. We then calculated the likelihood ratio to correct for the biasing, apply the balance heuristic, and collect the statistics of interest (in this case, the filtered output energy histogram).

This filtered output energy histogram we compute is the same observation generated in [21]. The comparison with previous results is shown in Fig. 5.11; and the agreement is nearly perfect. The coefficient of variation is computed as described in Sec. 2.1.3.

Thus, without making use of any of the detailed mathematical structure of dispersion-managed solitons, we have demonstrated the SVD-CE-IS method of simulating rare events in this system and shown that the results agree well with alternative techniques. There are several potential sources of discrepancy between this and previous results. The primary reason is that we use as our governing equation the NLS equation with periodically varying coefficients; in [21], the governing equation used is necessarily the dispersion-managed NLS equation (DMNLS), an averaged equation with a non-local term to account for the averaging of the dispersion management. While the DMNLS does reduce to the NLS as $s \leftarrow 0$, for nonzero s there is a degree of approximation involved. While it would be possible to apply the SVD-CE-IS method to the system described by the DMNLS to eliminate this

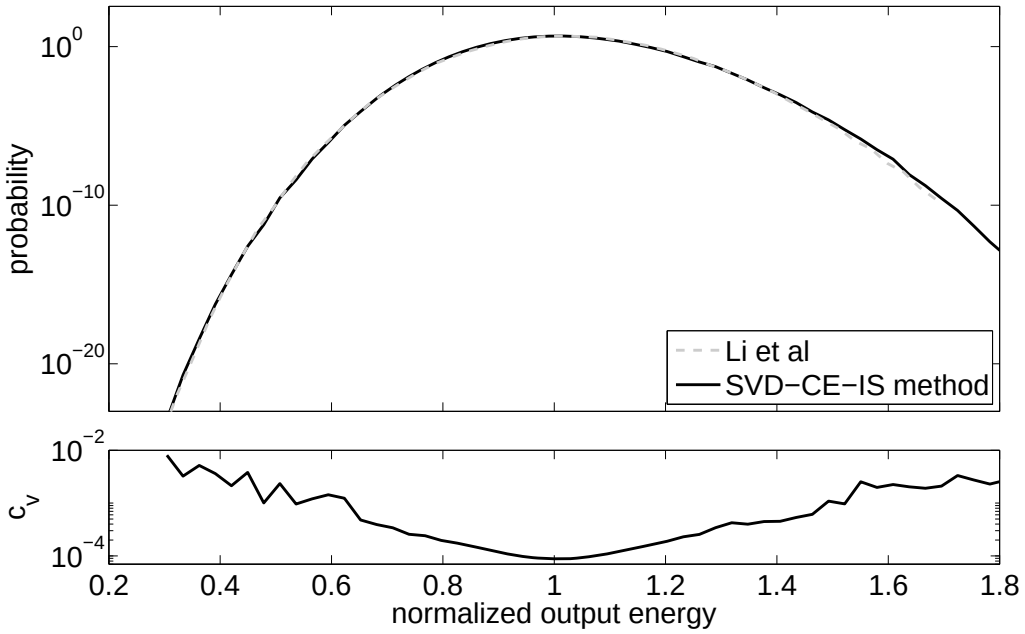

Figure 5.11. **Simulated PDF for DM soliton system**: Top, the full simulated PDF for the DM soliton system (black, solid) compared with the same PDF generated in [21] (gray, dashed). Bottom, the coefficient of variation corresponding to the former.

potential source of discrepancy, this is in some sense counterproductive, since the previous result had to resort to the DMNLS equation as an approximation, because the method used could not be applied to the full equation. Here, the method can be applied directly to the full NLS equation with dispersion management, and thus it is certainly more sensible to do so.

5.7. Beyond analytically tractable problems

We have successfully demonstrated the application of the SVD-CE-IS method to an existing problem for validation. We now move beyond those problems which are tractable

by the existing semi-analytic technique. The dependence of the previous method upon precise mathematical structure is actually quite constraining: to transform the problem into one in which it is still solvable by the new method but not by the old, we only have to change the pulse shape. Instead of the dispersion-managed soliton shape which we used previously, we now use a 50% duty-cycle, raised cosine pulse shape

$$U_0 = \frac{1 + \cos(\pi \sin(\frac{\pi t}{100ps}))}{2c_{DM}} \tag{5.7.1}$$

with the total pulse power matched to that of the dispersion managed soliton via the normalization factor c_{DM}, numerically determined from the previous solution. Of course, we could select any pulse shape of interest for this system – the selected pulse shape is of interest in at least one practical system [48]. The difference in pulse shape is not overly dramatic, and there is no difference in pulse power. The pulse shapes are compared in Fig. 5.12.

In terms of system behavior, however, the change is more noticeable. Without the dispersion-managed soliton initial condition, the nominal solution is no longer stroboscopically stationary, i.e., it no longer returns to the same pulse shape after each dispersion map period. The noiseless numerical solution using this new pulse shape is presented in Fig. 5.13. These longer period oscillations, as compared to the DM soliton evolution shown in Fig. 4.1, will be important in assessing the performance of the system later on.

For this pulse shape there is no other result for comparison, but we can apply our method in exactly the same way as described before, with merely a changed initial condition.

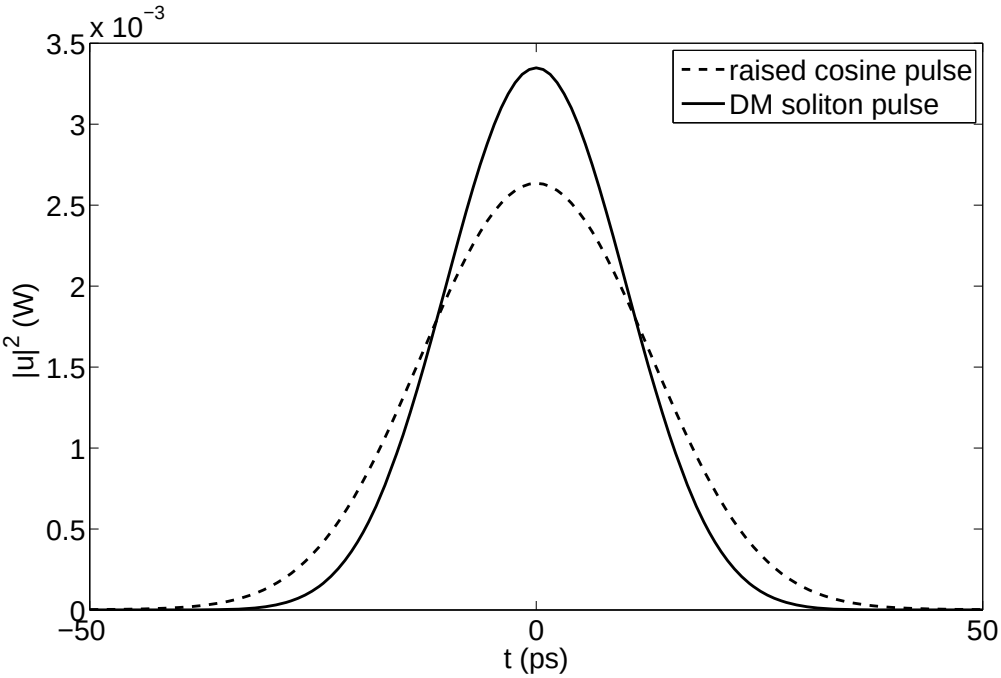

Figure 5.12. **Comparison of pulse shapes**: The two pulse shapes used in the different dispersion managed systems. The dispersion managed soliton shape in the solid curve, and the raised cosine shape in the dashed curve.

There is no dependence upon the initial data elsewhere in the method, unlike in the semi-analytic method where the pulse modes must be known analytically.

We then repeat the method as previously described. The modes can be computed in exactly the same way, though the results differ slightly due to the new pulse shape. A single sample trial from a biasing (reduced noise dimensionality) run is given in Fig. 5.14. In this case, the nominal problem has a more non-stationary character (compare with Fig. 5.13), and this behavior can be seen carried over in the basic behavior of this biasing run. Of course, the system mode, the linearized input vector, (in this case, only one) is computed

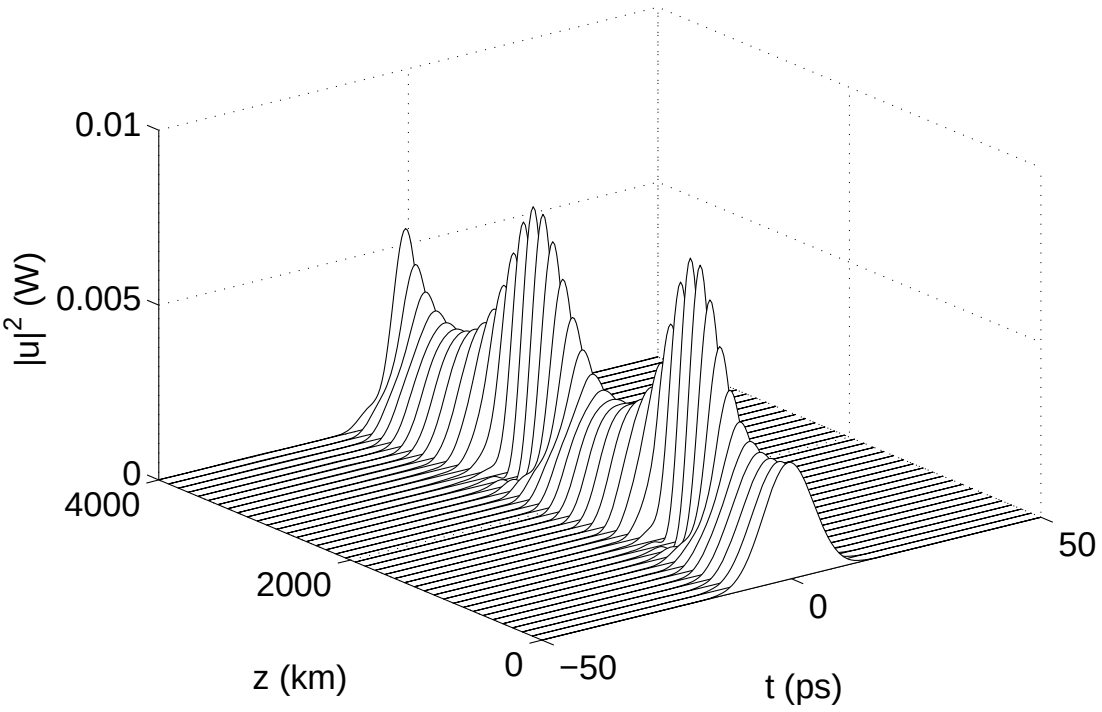

Figure 5.13. **Propagation of the raised-cosine pulse shape**: The noiseless propagation of the raised-cosine pulse shape in the dispersion managed system, solved numerically using the split-step Fourier method, observed stroboscopically.

numerically at each amplifier as previously described, with the result and the underlying pulse shape at the last amplifier given in Fig. 5.15

As before, we generate the biasing distributions and then use them to simulate entire output energy PDF for the new system. With the SVD-CE-IS method, adapting to the new, modified system is straightforward.

Using the generated biasing distributions, we simulate the normalized output energy PDF just as done previously. The biasing targets generated by the cross-entropy method in conjunction with the singular value decomposition are displayed in Fig. 5.16. One can see

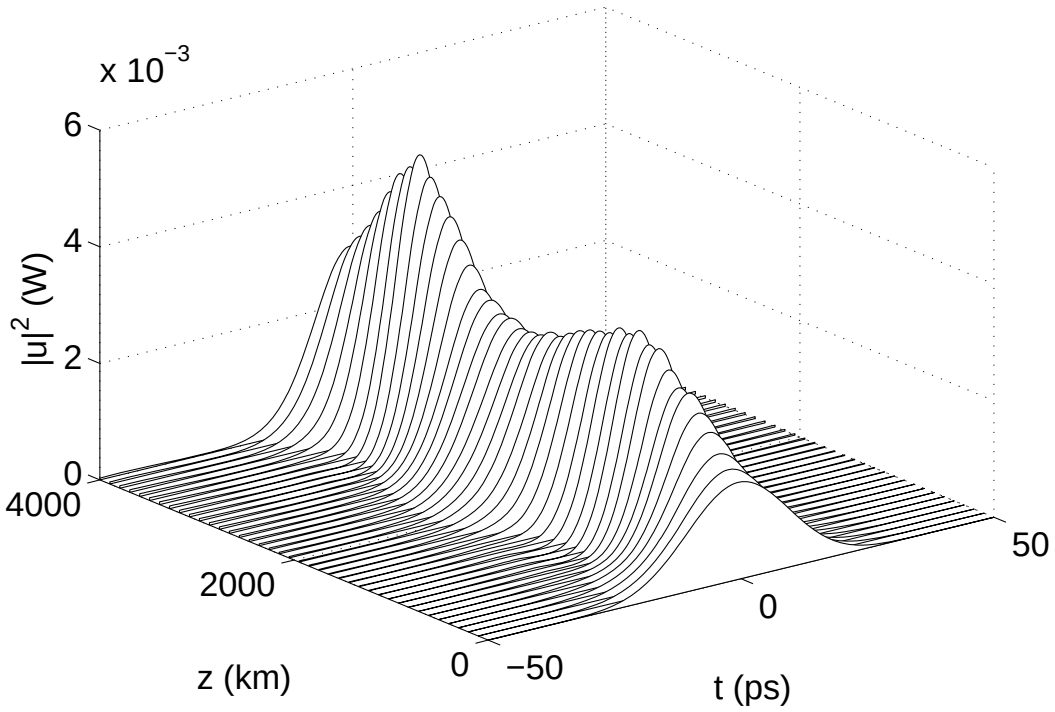

Figure 5.14. **Sample biased Monte Carlo trial for the raised-cosine pulse**: A sample Monte Carlo trial for the raised cosine pulse system with a biasing target generating an increase in pulse energy at the detector.

that in this case the optimal biasing distributions are no longer constant with propagation distance, as they were in the DM soliton case.

The comparison between the simulated PDF for the raised-cosine pulse shape and the previous dispersion-managed soliton pulse shape is illustrated in Fig. 5.17. It is immediately apparent that the simple pulse shape change has had a dramatic impact on the overall system performance. Recall that the input pulse energy has been matched in each case. Despite this, in parts of the PDF the probability of a given normalized output energy outcome differs between the two systems by as much as two orders of magnitude. The

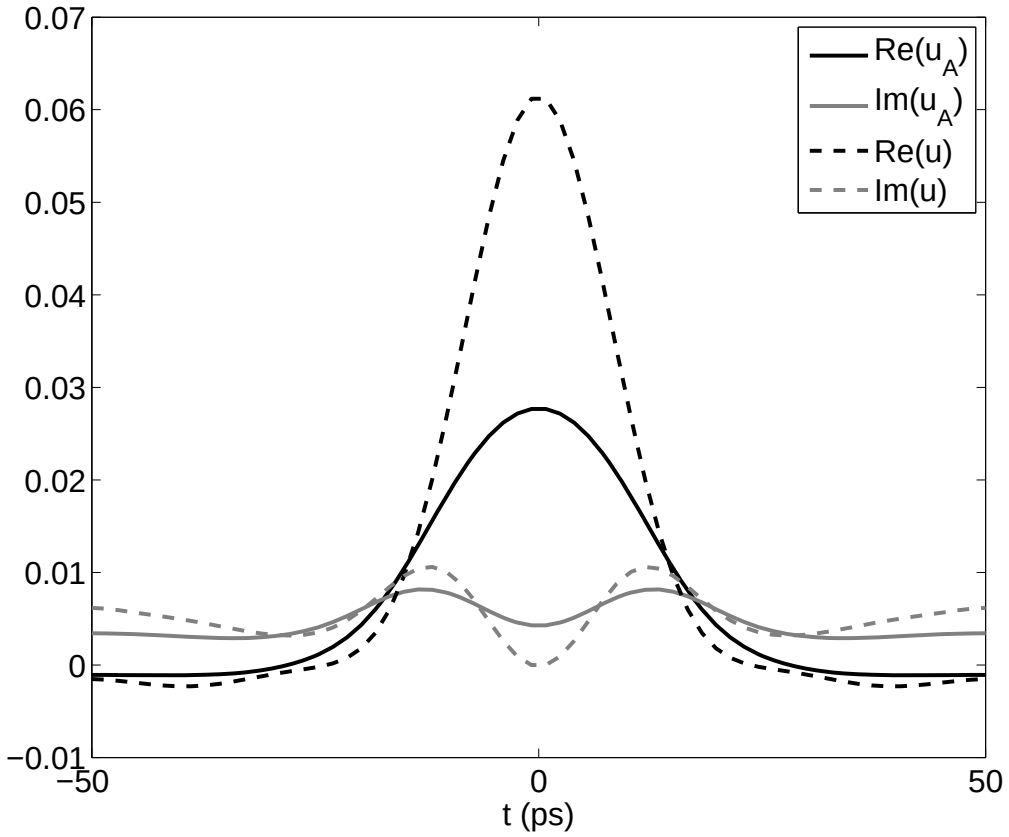

Figure 5.15. **Pulse shape and computed mode from sample trial**: The pulse shape at the final amplifier (dashed) from the sample trial in Fig. 5.14, along with the computed filtered mode at that location (solid). Both real and imaginary parts displayed, black and gray respectively.

ability to easily capture the changes in system performance provoked by changes like this is one of the benefits of the SVD-CE-IS method. Proposed modifications in system design can be tested quickly and with a minimum effort. Furthermore, it is possible to explore the reasons for these differences in system performance using this new method.

If one examines Fig. 5.17, despite the matching of the input pulse powers, the different pulses clearly change the probabilities at the output. However, in the region of roughly 1.1

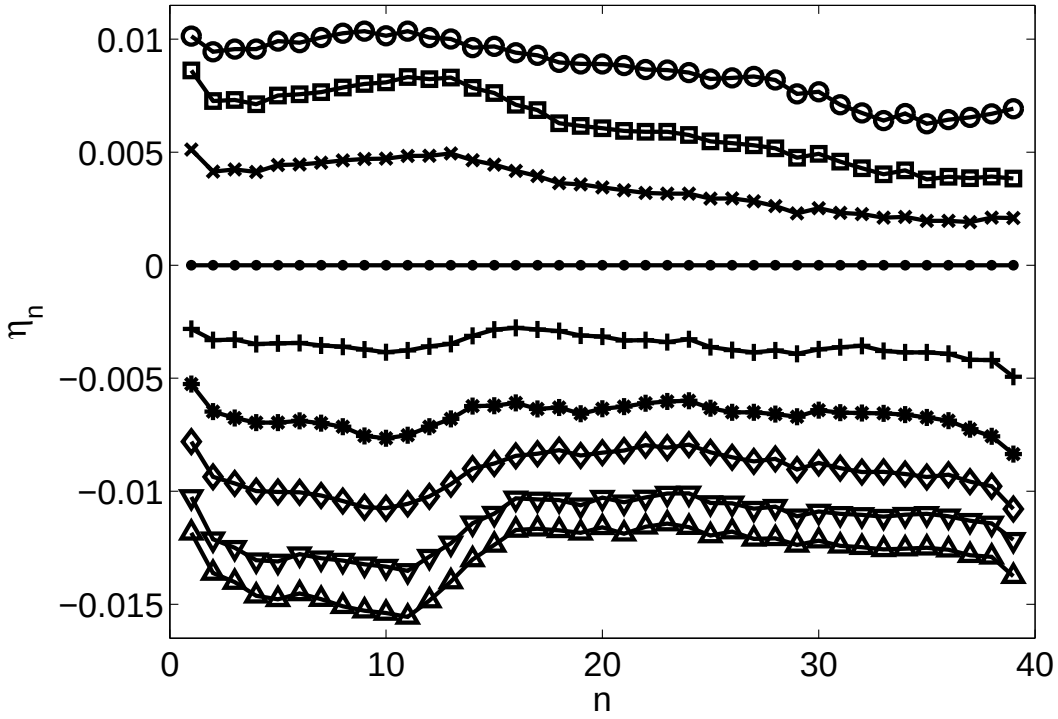

Figure 5.16. **Biasing coefficients for raised-cosine pulse system**: The biasing coefficients for the amplitude mode in the raised-cosine pulse system, for each of the 8 (7 nonzero) biasing targets.

to 1.5 in normalized output energy, the new pulse shape appears to produce significantly lower probabilities than the dispersion-managed soliton, beyond what can be explained by the filter-induced shift. To further explore this region, we can also use the SVD-CE-IS method as a probe of system behavior.

To probe this region, we select a histogram bin around 1.3 in normalized output energy and collect a new set of observations, the filtered pulse energy *path*, from $z = 0$ to 4000 km, as the pulse propagates for each trial. This is simply a new observation from the simulated system, essentially applying the end-of-line detector at each amplifier, and the

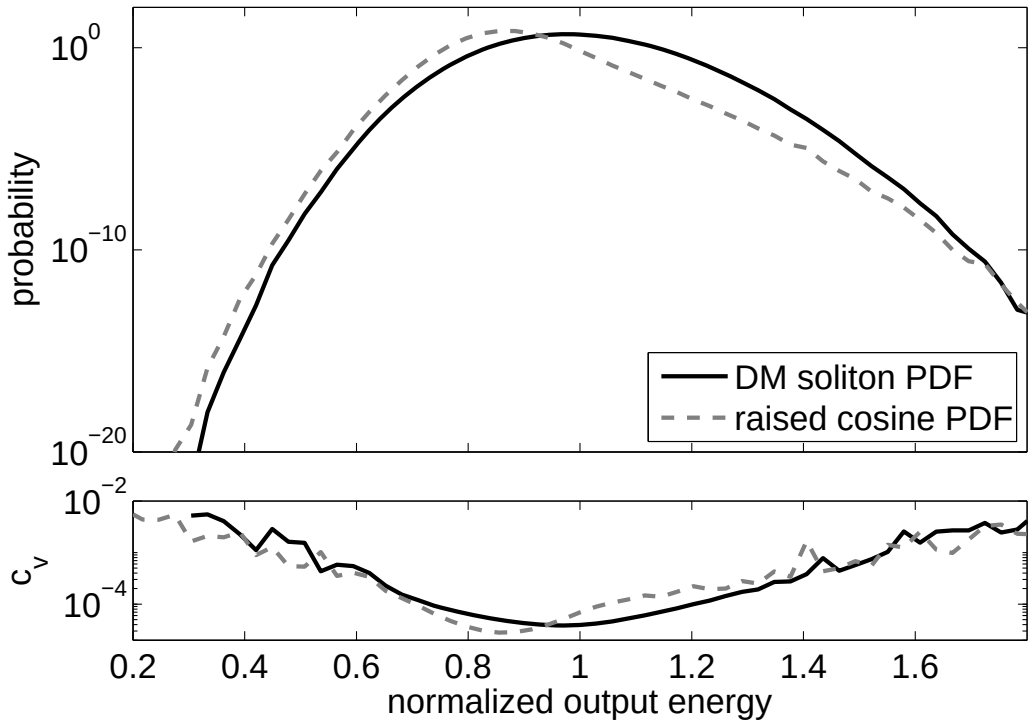

Figure 5.17. **Simulated PDFs for both dispersion managed systems**: Full simulated PDFs generated with the SVD-CE-IS method for the dispersion managed system with both pulse shapes. The DM soliton-based system, solid, black, as previously compared with published results in Fig. 5.11, and the raised-cosine pulse-based system, dashed, gray.

statistics are corrected for the biasing distributions using the likelihood ratio and weighting function just as for the normalized output energy. In this way we collect the mean path for reaching a target output value and can examine this path to understand some of the operational mechanisms responsible for the significant performance difference between the two systems. Doing this for both systems, we obtain the mean energy path for arriving at a designated filtered output energy level (1.3) and can compare them for hints as to the reasons for the difference in system performance.

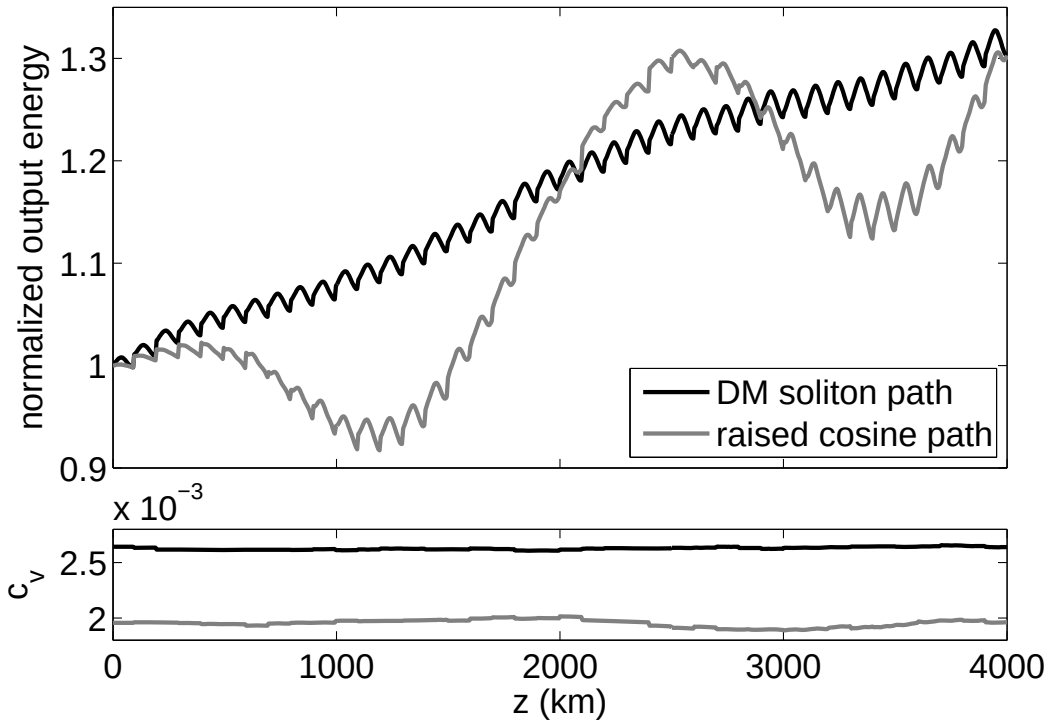

Figure 5.18. **Mean filtered energy paths, energy increase**: The mean filtered energy paths as a function of distance, collected from biased Monte Carlo trials conditioned on the final output detector value falling into a bin at normalized energy = 1.3. Black line, DM soliton path. Grey line, raised-cosine pulse path.

Fig. 5.18 shows the computed mean paths for each system to reach this target value. Recall that as per Fig. 5.17, the raised-cosine pulse system is considerably less likely than the DM soliton pulse to reach this prescribed value. By inspecting the mean paths, we can understand the reasons for the difference in system performance. In each system, the rapid oscillation is a result of the periodic dispersion map; recall that the dispersion map length is $z_a = 100$ km, and so each path in our 4000 km propagation span undergoes 40 of these periodic structures. However, this oscillation is common to both systems; it is the

longer-period oscillation that is different. Not only does the raised cosine pulse have to travel a longer "path" to reach the prescribed value, the longer-period oscillation makes reaching certain target values more difficult. In particular, at the locations corresponding to the "downstroke" of this oscillation, reaching larger amplitudes is more difficult.

We can also probe the amount of directed biasing required by the method to create a particular deflection in state space. By measuring the projection of the biased noise against the total noise introduced at each amplifier, we track this measure of biasing strength. By computing this measure, displayed in Fig. 5.19 along with the paths for easy comparison, both as the local biasing strength at each amplifier and the integrated total biasing strength up to that point, we see that as expected, based on the probability distributions, there is more biasing strength required to deflect the raised cosine pulse shape path to this position for probabilities that are low. As we expected, as described previously, the additional biasing strength for the raised cosine path is required in the section where the long oscillation is working against the desired energy change, e.g., just before $z = 1000$ km.

Similarly, this process can be performed at any point in the system parameter space to evaluate performance measures of interest. The bin around 1.3 was selected merely as a location of large deviation between the performance of the two systems.

Likewise, we can compare the mean filtered energy paths for a target which represents a decrease in pulse energy, perhaps a more physically relevant region for practical systems, even if it is not an area of such large deviation. Thus, we select the histogram bin nearest 0.7 in normalized output energy and collect the statistics for the mean path for each pulse

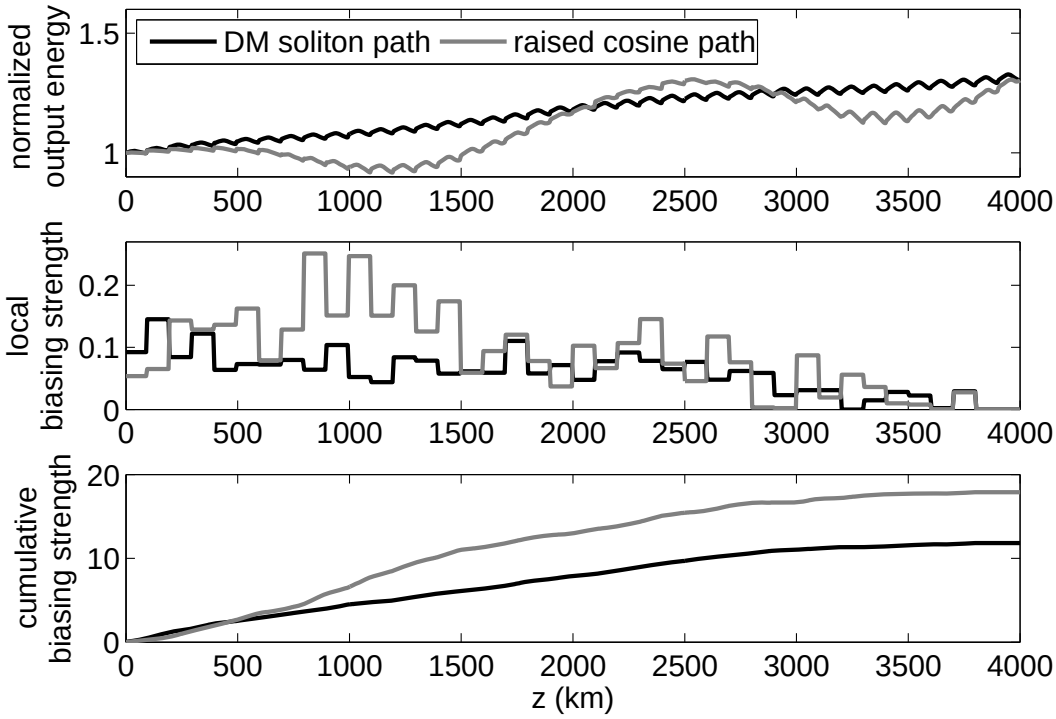

Figure 5.19. **Measured biasing strength, energy increase**: The measured amounts of biasing required, on average, to reach the histogram bin centered at 1.3 in normalized output energy. Top: mean energy paths, for comparison. Middle: local biasing strength, the projection of the biasing noise against total noise at the local amplifier. Bottom: total biasing strength, the net biasing applied to reach this point in the path.

shape, just as described above. In this case, recall that this energy outcome is more likely for the raised-cosine pulse shape than for the dispersion-managed soliton, by slightly less than an order of magnitude. The mean paths in this case are plotted in Fig. 5.20. Again, it appears that the biasing strength required to affect a large change in the filtered energy is very low where the desired downward direction and longer period oscillation are in phase.

We can examine the noise projection measure of the required biasing strength to probe the behavior of the systems in this section of the PDFs as well to explore the reasons

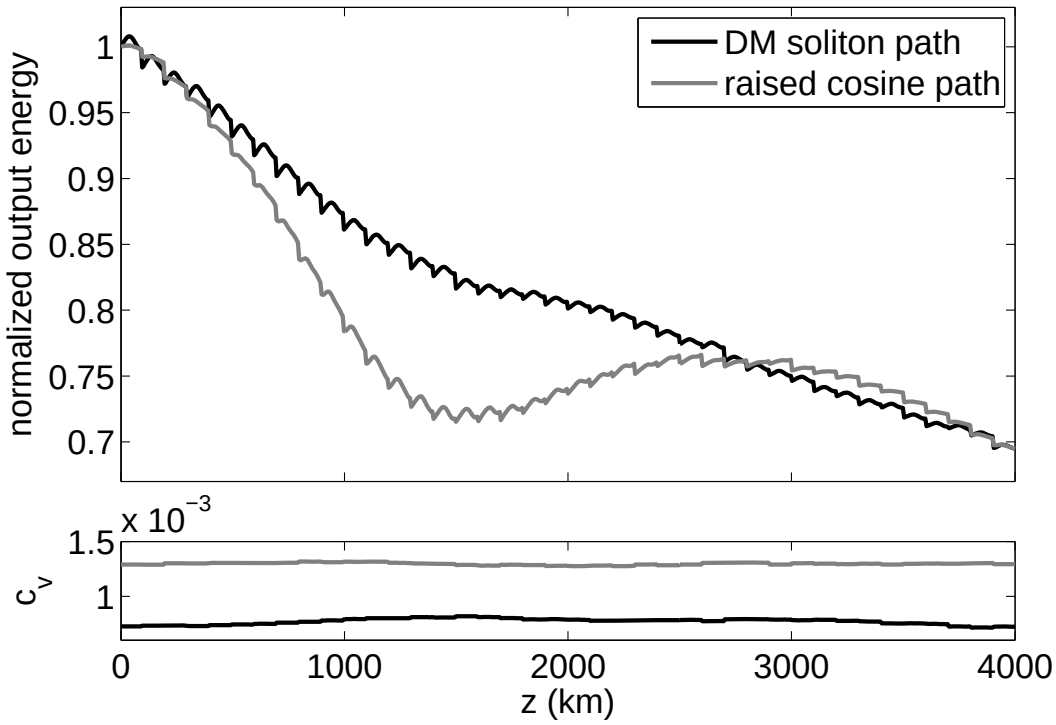

Figure 5.20. **Mean filtered energy paths, energy decrease**: The mean filtered energy paths as a function of distance, collected for biased Monte Carlo trials where the output detector value fell into the bin at normalized energy = 0.7. Black line, DM soliton path. Grey line, raised-cosine pulse path.

for the differences in system performance. The appropriate measurements are presented in Fig. 5.21. In this case, we can readily see that the required local biasing strength for the raised cosine pulse is less than the required strength for the DM soliton at all points, but particularly in those regions where the longer oscillation associated with the raised cosine pulse mean path is traveling downward as part of its normal pattern. In the region from roughly 0 to 1500 km, the raised cosine path decreases very quickly with little biasing strength, in the region of natural energy decrease in the nominal problem. Once the energy

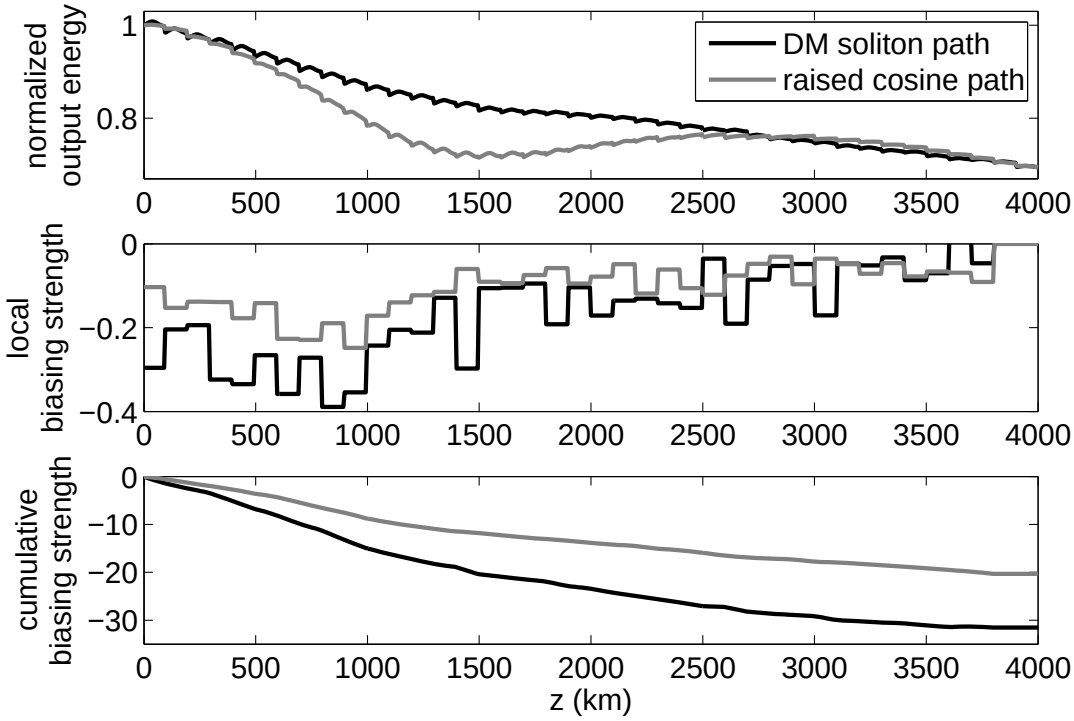

Figure 5.21. **Measured biasing strength, energy decrease**: The measured amounts of biasing required, on average, to reach the histogram bin centered at 0.7 in normalized output energy. Top: mean energy paths, for comparison. Middle: local biasing strength, the projection of the biasing noise against total noise at the local amplifier. Bottom: total biasing strength, the net biasing applied to reach this point in the path.

has been pushed down to this level, it appears to require little biasing to maintain the low level.

We have demonstrated by probing these two regions of system state space with the SVD-CE-IS method that it can be a powerful tool for understanding the behavior and nature or rare events in these systems. Of course, if the system called for a different quantity to be studied for gaining this insight into the system behavior, the method can provide results associated with any observations that can be made from the numerical system simulation.

Probing these regions of system behavior could be a valuable tool in system design. If design requirements were imposed on this system we have been simulating, certainly this information would allow us to tailor the design as required to improve system performance, even far down in the tails of the probability distributions.

5.8. Beyond lightwave systems

Up to this point we have formulated and demonstrated our new rare-event simulation technique in the context of lightwave communication systems, the area of interest that originally inspired the work and the initial test-cases and applications. However, the SVD-CE-IS method is not formulated using any particular elements of a lightwave system model. The requirements placed upon the system to be simulated are much more modest, and the technique should be applicable to much wider-ranging areas. While we have presented details of applying the method to a specific propagation operator, a specific hyperbolic partial differential equation, the nonlinear Schrödinger equation, the method does not depend on any specifics of this equation. Certainly it should be possible to apply the method to the Ginzburg–Landau equation modeling ultra-short optical pulses in a mode-locked laser. But it is also possible to apply it to any linearizable propagation operator, be it a partial differential equation, a system of ordinary differential equations, or virtually any other kind of propagator.

With the current formulation of the method, the only restriction is that the stochastic component be added discreetly, rather than continuously, in the propagation direction, and that the noise also be delta-correlated in this direction. Certainly one can imagine that

applications requiring continuous noise might be accommodated by some discreet noise-discretization approximation in the propagation direction, so even this restriction might be relaxed. The remaining restrictions still allow application to a wide class of problems beyond the scope of lightwave systems that we have so far considered.

The method has also thus far only been formulated for additive noise. While there are no formal results to present, adaptation of the method to other forms of noise have been considered and seem to be possible. Certainly, multiplicative noise appears to present no insurmountable difficulty. Noise which is introduced in a system parameter, rather than as a perturbation to the dependent variable, provides a more substantial hurdle, though it appears that there are possible routes to accommodating this sort of model as well. Certainly we look forward to application of this method not only to different lightwave systems, but areas far beyond lightwave systems.

CHAPTER 6

Discussion

In this thesis we have formulated and presented results for the SVD-CE-IS method for rare-event simulation in optical communication systems. In fact, the method is presented in a more general context, and could easily be applied to areas beyond optical communication systems – the restriction to lightwave systems is only in the choice of sample problems we have selected for this work.

The new method evolved from a previous, semi-analytic method used in many previous works, including some referenced in this dissertation [1, 5, 21, 49]. This method is highly effective for problems where it can be applied, especially those making use of soliton pulse shapes. The authors of [21] managed to extend it to dispersion managed solitons. The impetus for developing the new method was to move beyond these soliton or DM soliton pulse shape restrictions, so that we could study rare events in a much broader class of problems. We have demonstrated that the SVD-CE-IS method we have developed is extremely useful for this task.

The SVD-CE-IS method is structured in such a way that the evolution from the specialized, soliton method is quite apparent – the basic framework is the same: (multiple) importance sampled Monte Carlo simulation. The elements of the previous method which depend upon soliton pulse shapes are 1) the soliton modes, and 2) the soliton parameter evolution equations. The new method we have developed replaces these missing elements, rather than restructuring the entire method. The pulse modes are found in the new method in numerical fashion, by employing the singular value decomposition [16] on the linearized problem to reduce the dimensionality of the overall problem to a tractable level. In the

example problems we present, the dimensionality reduction is from 40,960 to 40. This provides a vast simplification, and enables the replacement of the second missing element.

In the previous method, evolution equations for the pulse parameters were available via soliton perturbation theory. These evolution equations could guide an optimization problem for the biasing distributions. However, while the SVD does provide the modes, it does not provide a replacement for the evolution equations used in the analysis to determine the biasing. For that purpose, we turn to a stochastic optimization algorithm, the cross-entropy method [17]. We use the CEM to solve for the optimal biasing distributions using the SVD to reduce the noise dimensionality to a space on which the CEM can reasonably act. Using this method, we obtain the biasing distributions needed for the importance sampled Monte Carlo simulations.

This method is demonstrated successfully for two lightwave problems. For one, we compare with previously published work [21] and obtain good results. For the other, we take on a problem inaccessible to the previous method. We further demonstrate that the SVD-CE-IS method has a further capability, in that it can be used to probe the differences between these two systems we have studied. The output PDFs indicate significantly different performances between similar systems. By delving into sections of the PDFs and collecting additional data, we are able to asses these differences. This is yet another benefit of the new method.

In total, we have demonstrated an effective new method for rare event simulation in optical systems, which can be applied to problems which may have been impossible or more difficult by existing means. The method further is capable of delving into the details

of system operation to garner physical insights into system performance. We also believe that the method is formulated generally enough to be applied in the future to a much wider class of problems than optical systems.

References

[1] R. O. Moore, G. Biondini, and W. L. Kath. Importance sampling for noise-induced amplitude and timing jitter in soliton transmission systems. *Opt. Lett.*, 28(2):105, 2003.

[2] R. Holzlöhner and C. R. Menyuk. Use of multicanonical monte-carlo simulations to obtain accurate bit error rates in optical communication systems. *Opt. Lett.*, 29:1894–1896, 2003.

[3] Y. Yadin, M. Shtaif, and M. Orenstein. Bit-error rate of optical dpsk in fiber systems by multicanonical monte carlo simulations. *Photonics Technology Letters, IEEE*, 17(6):1355–1357, June 2005.

[4] R. O. Moore, T. Schäfer, and C. K. R. T. Jones. Soliton broadening under random dispersion fluctuations: Importance sampling based on low-dimensional reductions. *Optics Communications*, 256:439–450, December 2005.

[5] R. O. Moore, G. Biondini, and W. L. Kath. A method to compute statistics of large, noise-induced perturbations of nonlinear schr[o-umlaut]dinger solitons. *SIAM Journal on Applied Mathematics*, 67(5):1418–1439, 2007.

[6] E.T. Spiller, W.L. Kath, R.O. Moore, and C.J. McKinstrie. Computing large signal distortions and bit-error ratios in dpsk transmission systems. *Photonics Technology*

Letters, IEEE, 17(5):1022–1024, May 2005.

[7] A. Tonello, S. Wabnitz, I. Gabitov, and R. Indik. Importance sampling of gordon-mollenauer soliton phase noise in optical fibers. *Photonics Technology Letters, IEEE*, 18(7):886–888, April 1, 2006.

[8] Krishna Chepuri and Tito Homem de Mello. Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research*, 134(1):153–181, 2005.

[9] Miguel Onorato, Alfred R. Osborne, Marina Serio, and Serena Bertone. Freak waves in random oceanic sea states. *Phys. Rev. Lett.*, 86(25):5831–5834, Jun 2001.

[10] William J. Morokoff. An importance sampling method for portfolios of credit risky assets. In *WSC '04: Proceedings of the 36th conference on Winter simulation*, pages 1668–1676. Winter Simulation Conference, 2004.

[11] Tobin A. Driscoll and Kara L. Maki. Searching for rare growth factors using multicanonical monte carlo methods. *SIAM Review*, 49(4):673–692, 2007.

[12] J. A. Bucklew. *Introduction to rare event simulation*. Springer-Verlag, New York, 2004.

[13] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley and Sons, Inc, 1981.

[14] Peter J. Smith, Mansoor Shafi, and Hongsheng Gao. Quick simulation: A review of importance sampling techniques in communications systems. *IEEE Journal on Selected Areas in Communications*, 15(4):597–613, 1997.

[15] Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–??, 2000.

[16] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll. Hydrodynamic stability without eigenvalues. *Science*, 261:578–584, July 1993.

[17] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190, 1999.

[18] P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, January 2005.

[19] Tito Homem-de-Mello. A study on the cross-entropy method for rare event probability estimation. *INFORMS Journal on Computing*, 19:381, 2004.

[20] G. Biondini, W. L. Kath, and C. R. Menyuk. Importance Sampling for Polarization-Mode Dispersion: Techniques and Applications. *Journal of Lightwave Technology*, 22:1201–+, April 2004.

[21] J. Li, E. Spiller, and G. Biondini. Noise-induced perturbations of dispersion-managed solitons. *Physical Review A*, 75(5):053818–+, May 2007.

[22] David L. Chopp, 2005. ESAM 495-0 coursenotes, Northwestern University.

[23] R. O. Moore, G. Biondini, and W. L. Kath. A method to compute statistics of large, noise-induced perturbations of nonlinear schrodinger solitons. *SIAM Review*, 50(3):to appear, 2008.

[24] F. James. Monte carlo theory and practice. *Rep. Prog. Phys.*, 43:1145–1190, 1980.

[25] Eric Veach. Robust monte carlo methods for light transport simulation. *PhD thesis, Stanford University*, 1997.

[26] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

[27] Gilbert Strang. *Linear Algebra and its Applications*. Harcourt Brace Jonanovich, San Diego, 3rd edition, 1988.

[28] B. E. A. Saleh and M. C. Teich. Fundamentals of photonics. *NASA STI/Recon Technical Report A*, 92:35987–+, 1991.

[29] Govind P. Agrawal. *Nonlinear Fiber Optics*. Academic Press, 1995.

[30] John Scott Russell. *The Wave of Translation in the Oceans of Water, Air, and Ether*. Ballantyne Press, Edinburgh and London, 1885.

[31] Akira Hasegawa and Frederick Tappert. Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. i. anomalous dispersion. *Applied Physics Letters*, 23(3):142–144, 1973.

[32] L. F. Mollenauer, R. H. Stolen, and J. P. Gordon. Experimental observation of picosecond pulse narrowing and solitons in optical fibers. *Physical Review Letters*, 45:1095–1098, September 1980.

[33] A Hasegawa and Y Kodama. *Solitons in optical communications*. Oxford University Press, New York, 1995.

[34] E. Iannone, F. Matera, A. Mecozzi, and M. Settembre. *Nonlinear Optical Communication Networks*. Wiley, 1998.

[35] V. Cautaerts, A. Maruta, and Y. Kodama. On the dispersion managed soliton. *Chaos*, 10:515–528, September 2000.

[36] J.H.B. Nijhof, N.J. Doran, W. Forysiak, and F.M. Knox. Stable soliton-like propagation in dispersion managed systems with net anomalous, zero and normal dispersion. *Electronics Letters*, 33(20):1726–1727, 25 Sep 1997.

[37] Brian S. Marks. Dispersion management in fiber optic communications. *PhD thesis, Northwestern University*, 2000.

[38] J. H. B. Nijhof, W. Forysiak, and N. J. Doran. The averaging method for finding exactly periodic dispersion-managed solitons. *IEEE J sel top Q elec*, 6(2):330, 2000.

[39] J. A. Oteo. The Baker-Campbell-Hausdorff formula and nested commutator identities. *Journal of Mathematical Physics*, 32:419–424, February 1991.

[40] T. R. Taha and M. J. Ablowitz. Analytical and Numerical Aspects of Certain Nonlinear Evolution Equations. II. Numerical, Nonlinear Schrödinger Equation. *Journal of Computational Physics*, 55:203–+, August 1984.

[41] S. Yu, S. Zhao, and G. W. Wei. Local spectral time splitting method for first- and second-order partial differential equations. *Journal of Computational Physics*, 206:727–780, July 2005.

[42] David L. Chopp, 2004. ESAM 446-1/2 coursenotes, Northwestern University.

[43] Bengt Fornberg and Tobin A. Driscoll. A fast spectral algorithm for nonlinear wave equations with linear dispersion. *J. Comput Phys.*, 155(2):456–467, 1999.

[44] R. B. Lehoucq, D. C. Sorensen, and C. Yang, 1997. ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.

[45] Ivar Stakgold. *Boundary value problems of mathematical physics (vol. 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[46] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan, editors. *Simulation of Communication Systems: Modeling, Methodology and Techniques*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[47] Christie K. Madsen and J.H. Zhao. *Optical Filter Design and Analysis: A Signal Processing Approach*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[48] V. S. Grigoryan, M. Shin, P. Devgan, J. Lasri, and P. Kumar. SOA-Based Regenerative Amplification of Phase-Noise-Degraded DPSK Signals: Dynamic Analysis and Demonstration. *Journal of Lightwave Technology*, 24:135–+, January 2006.

[49] Elaine T. Spiller. Computational studies of rare events in optical transmission systems. *PhD thesis, Northwestern University*, 2005.

[50] W. Gropp. MPICH2 Users' Guide, 2007. (http://tinyurl.com/6eusvj).

APPENDIX

Parallel computing considerations

The method we have presented is very easy to parallelize for efficient computation on a cluster or other parallel computing structure. Simulations for this work have been carried out not only on standard, single processor machines, but a 64-processor distributed-memory beowulf cluster and an 8-processor shared-memory workstation. The key to parallelizing the algorithm is to recognize the computationally time-intensive portions of the process and distribute these sections.

We use a parallel system with a head or control node, which performs the central structuring of the algorithm, and work-assignment and data collection, and a number of work nodes, which do the heavy lifting for the head node. We use a message-passing design appropriate for distributed memory architectures, where data packets containing work assignments are sent from the head node to the work nodes, and the work nodes return data packets with the results of their work assignment to the head node. The head node processes this data and re-assigns work as necessary. We used the message-passing interface MPICH2 [50] to develop this code.

The head node, then, executes the entirety of the cross-entropy method, except for the evaluation of the performance function (as a function of the generated random variables). Each needed performance evaluation is sent to an available work node, which returns the performance function value to the head node for inclusion in the cross-entropy method. This is an appropriate division of labor because the majority of the cross-entropy method steps require relatively little computational power, while each iteration of performance function value from random variables requires the simulation of the nonlinear PDE and

iterations of the linearized PDEs to execute the singular value decomposition to find the modes.

If we refer back to the schematic representation of the method, Fig. 5.2, looking only at the top half of the diagram for the biasing distribution generation phase, recall that the dashed lines represent connections or processes executed many times (stochastic), and the solid lines represent connections made only once. The three-part circulation of dashed lines from the Cross-Entropy Method to the Singular Value Decomposition (via n random variables), to the simulated system (via N random variables) and back to the CEM (via the performance function value) is the portion executed in parallel on the work nodes. Within each level of the cross-entropy method, which is a course division, each trial is independent, and thus can easily be executed in parallel. Only at the end of each level does the work need to be synchronized before beginning the parallel execution phase again. However, for the problems we have been working with, the number of levels is around 6 and the number of trials per level is around $10,000$, so this synchronization phase disrupts the parallel execution infrequently. When the cross-entropy method has reached its target value, the biasing distributions have been determined and can be sent on to the second phase of the method, the Monte Carlo phase.

Likewise, in the second, Monte Carlo phase of the algorithm, a similar parallelization scheme applies. Collection of statistics is handled by the head node, and the execution of individual trials, including the computationally costly PDE simulation and SVD steps, are executed independently by available work nodes. Here there are no cross-entropy method levels to disrupt the parallel execution, and the processes can proceed without interruption.

Again, referring back to Fig. 5.2, now the lower half of the algorithm, the circulation from IS Monte Carlo Simulation to the simulated system (via full noise and reduced noise and the SVD) and back to IS-MC (via observations) is the portion executed independently, in parallel, on the work nodes. After the designated number of trials have been executed on the work nodes, the head node has collected statistics from all the observations and can report the results as PDFs, or as desired.

In pseudo-code, then, the algorithm is represented as

```
// CEM-biasing distribution phase
// head node
while I < Ihat
// CEM levels
      for ii=1:(N/size)
      // N trials at each CEM level, size work nodes
          MPI_DISTRIBUTE(performancefunction())
          //(x size times, distributed to work nodes)
      end for
      //... CEM innards
end while

// pass biasing distributions to
// IS-MC phase
```

```
// head node


for ii=1:(M/size)
// M MC trials
    MPI_DISTRIBUTE(MCperformancefunction())
    // x size times, distributed to work nodes
    //... MC innards
end for
```

The algorithm is very simple to parallelize, which is one of its strengths. For systems where execution of a single trial is a costly endeavor, crude Monte Carlo simulation is particularly infeasible, in terms of computational time, and variance reduction strategies become more important. If the trials were very quick to execute, perhaps it would be possible to merely "brute-force" the rare-event problem by running many, many trials. For complex problems, however, the reduction in variance and hence number of required trials is critical, and the ability to distribute the required trials over parallel architecture simply makes ever more complex problems tractable. Certainly there are many problems, however, where parallelization of this method is unnecessary, as the computational cost of rare event simulation has been reduced so dramatically by the method that it can be handled on a single machine without need of recourse to more sophisticated computational methods (or hardware).