Strål säkerhets myndigheten
Swedish Radiation Safety Authority

Author: Arne Sahlberg

Research

2016:26

Ensemble for deterministic sampling with positive weights

Report number: 2016:26 **ISSN:** 2000-0456
Available at www.stralsakerhetsmyndigheten.se

SSM perspective

Background

One of the tasks of SSM is to inspect the Swedish nuclear power industry and make sure that the nuclear power plants operate well within safety margins and, hence, they demand that uncertainties be taken into account in calculations made for safety analysis. SSM requires that these uncertainty estimates be made by either making conservative estimates, i.e., looking at worst reasonable case scenarios, or by making realistic calculations combined with analysis of uncertainty, often known as Best Estimate Plus Uncertainty.

There are several methods for quantifying the uncertainty in a calculated result, and a class of such methods is sampling methods, such as Monte Carlo methods.

A relatively new approach for propagating uncertainty through calculations is Deterministic Sampling (DS), which is a sampling method intended to use as few samples as possible by choosing the samples deterministically, rather than randomizing them, and hence save time in when the computations are heavy. This approach has been used at SSM.

In some particular cases, DS runs into a problem. In the ensemble used in DS each sample has an associated weight, and when some of the weights are negative, the method can fail. This is a problem which has occurred at SSM when DS has been used.

Finding the ensemble for DS with only positive weights is not trivial, but for the method to be useful, this is desired. This motivates why SSM wants to increase the competence in the field of deterministic sampling and uncertainty quantification.

Objective

The aim of this project is to find a reliable method for determining an ensemble for Deterministic Sampling with only positive weights. This report is intended to work as a presentation of the project, as well as to work as a beginner's guide to DS.

Results

In the report several methods for generating DS ensembles are presented, which can encode up to four moments and represent any kind of probability distribution, and encode correlations between parameters. This can be used with many parameters, and the number of samples needed scale linearly with the number of parameters, at best.

DS seems to have potential for uncertainty quantification at SSM, in the nuclear power industry and with heavy calculations done in other field.

Project information

Contact person SSM: Peter Hedberg (KR)
Reference: SSM2015-5407

Strål
säkerhets
myndigheten
Swedish Radiation Safety Authority

Author: Arne Sahlberg
Uppsala Universitet, Uppsala

2016:26

Ensemble for deterministic sampling with positive weights

Date: January 2016
Report number: 2016:26 **ISSN:** 2000-0456
Available at www.stralsakerhetsmyndigheten.se

This report concerns a study which has been conducted for the Swedish Radiation Safety Authority, SSM. The conclusions and viewpoints presented in the report are those of the author/authors and do not necessarily coincide with those of the SSM.

SSM 2016:26

Abstract

Knowing the uncertainty of a calculated result is always important, but especially so when performing calculations for safety analysis. A traditional way of propagating the uncertainty of input parameters is Monte Carlo (MC) methods. A quicker alternative to MC, especially useful when computations are heavy, is Deterministic Sampling (DS).
DS works by hand-picking a small set of samples, rather than randomizing a large set as in MC methods. The samples and its corresponding weights are chosen to represent the uncertainty one wants to propagate by encoding the first few statistical moments of the parameters' distributions.
Finding a suitable ensemble for DS in not easy, however. Given a large enough set of samples, one can always calculate weights to encode the first couple of moments, but there is good reason to want an ensemble with only positive weights. How to choose the ensemble for DS so that all weights are positive is the problem investigated in this project.
Several methods for generating such ensembles have been derived, and an algorithm for calculating weights while forcing them to be positive has been found. The methods and generated ensembles have been tested for use in uncertainty propagation in many different cases and the ensemble sizes have been compared.
In general, encoding two or four moments in an ensemble seems to be enough to get a good result for the propagated mean value and standard deviation. Regarding size, the most favorable case is when the parameters are independent and have symmetrical distributions.
In short, DS can work as a quicker alternative to MC methods in uncertainty propagation as well as in other applications.

Contents

1. Background

Before a nuclear power plant is put in use, or its performance is changed in any way, thorough security assessments have to be made with deterministic methods. Any models and calculations being used for safety analysis and for assessing limits in its construction performance need to be verified and validated as well as take uncertainties into account. Those are dictations from the Swedish Radiation Safety Authority (SRSA) about safety at Swedish nuclear power plants[9].
One of the tasks of the SRSA is to inspect the Swedish nuclear power industry and make sure that the nuclear power plants operate well within safety margins and, hence, they demand that uncertainties be taken into account in calculations made for security analysis. The SRSA recommends that these uncertainty estimates be made by either making conservative estimates, i.e., looking at worst reasonable case scenarios, or by making realistic calculations combined with analysis of uncertainty[9], often known as Best Estimate Plus Uncertainty.
This report is specifically concerned with the Best Estimate Plus Uncertainty approach. In such an approach a calculation is made to the best estimate of the value of used model's input parameters, which gives a result. The question is then what the uncertainty of the result is. Often one has some idea of the uncertainty of the measured input parameters, so the question becomes how these uncertainties affect the uncertainty of the calculated result. This is the problem of uncertainty propagation.
Today, there are several methods of propagating uncertainties through functions; a big class of these methods is sampling methods, of which Random Sampling and Latin-Hypercube Sampling are two examples[12]. Those mentioned are both called Monte Carlo methods, meaning they are based on calculating many randomized samples of the function and analyzing the statistics from these samples.
Random Sampling is a traditional way of propagating uncertainty, but in some cases using such methods is not an option. One such case, which occurs in the modeling of a nuclear reactor, is Computational Fluid Dynamics (CFD) where a single simulation can take several hours. Since Random Sampling can require up to thousands of simulations, depending on what the distribution looks like, those are not useful here due to unreasonably high time cost.
A counterpart to the Monte Carlo methods is Deterministic Sampling, a concept which was introduced in 2013[1] but has its roots in the Unscented Transform from 1995[3], in which a smaller set of samples is determined to fit the statistical distribution of the input parameters. If such an ensemble can be determined, the required number of simulations needed for uncertainty propagation can be decreased significantly.
Propagating error with deterministic sampling is, in short, similar to Monte Carlo methods, but instead, of randomizing thousands of samples and eval-

uating the function at each of these, one uses a small well-chosen ensemble with much fewer samples.
The ensemble used in Deterministic Sampling consists of a small set of samples, called sigma-points, which have a weight attached to them. This weighted set of points should represent the distribution of the parameters as well as possible, which means they should have the same average value, the same variance, the same covariance and preferably the same higher order moments as well. Once such an ensemble is determined the process is the same as with Random Sampling i.e. the function is evaluated at each sigma-point and the mean value and variance of the result is calculated.
Deterministic Sampling has been used for uncertainty propagation in CFD simulations at the SRSA with success, but in some cases, it has shown to give unrealistic values for higher statistical moments, such as even moments which are negative, which is prohibited. This problem seems to arise when the function evaluated is not monotonous, if the ensemble has negative weights. When determining an ensemble with previously used methods, some of the weights have often been negative, which has caused problems.

1.1. Scope

The aim of this project is to find a reliable method for determining an ensemble for Deterministic Sampling with only positive weights.
The main focus is to create ensembles encoding up to four moments which is the number of moments. Some of the methods presented can be used to encode even higher order moments, but usually, in Deterministic Sampling, no greater than two or four moments are used.

2. Theory

2.1. Statistical concepts

This section consists of a short primer on some concepts from the field of statistics and probability theory which is used throughout this report. The reader well versed in those fields can skip to the next section.

Probability Distribution Function (PDF)

A PDF is a function $f(X)$ which shows the probability distribution of a continuous random variable X. The probability that X will be within an interval $[a,b]$ is calculated as

$$P(X \in [a,b]) = \int_a^b f(X)\mathrm{d}X \tag{2.1}$$

Expectation value $\langle X \rangle$

The expectation value is a way of measuring the average value of a random variable, meaning if we sample the random variable enough times, this value will be the average of the samples. For a continuous random variable, it is calculated as

$$\langle X \rangle = \int_{-\infty}^{\infty} X f(X)\mathrm{d}X \tag{2.2a}$$

and for a discrete random variable which can take only certain values, $X \in \{x_1, x_2 \ldots x_n\}$ as

$$\langle X \rangle = \frac{1}{n}\sum_{i=1}^{n} x_i P(X = x_i). \tag{2.2b}$$

Variance $\langle \delta^2 X \rangle$

The variance of a random variable measures how much the variable is expected to differ from its expectation value. The variance is calculated as the average of the square deviation from the expectation value. With δX defined as $X - \langle X \rangle$ and $\delta^2 X$ as $(\delta X)^2$ this becomes simply a matter of taking the average of $\delta^2 X$ with equation (2.2a). For a continuous random variable is calculated as

$$\mathrm{var}(X) = \langle \delta^2 X \rangle = \int_{-\infty}^{\infty} (X - \langle X \rangle)^2 f(X)\mathrm{d}X = \int_{-\infty}^{\infty} \delta^2 X f(X)\mathrm{d}X \tag{2.3a}$$

and for a discrete random variable $X \in \{x_1, x_2 \dots x_n\}$ one has to calculate it with the discrete expressions for the expectation value from equation (2.2b) which becomes

$$\mathrm{var}(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \langle X \rangle)^2 P(X = x_i). \tag{2.3b}$$

Note that this is the variance calculated when the probability distribution is completely known. The variance from a set of samples $\tilde{x} = \{x_1, x_2 \dots x_N\}$, called Sample Variance, is calculated as the average square of the samples deviation from the Sample Mean, $(x_i - \langle \tilde{x} \rangle_{\mathrm{smp}})^2$, since the true mean is not known. This variance would be slightly too small since the sample mean will lie closer to the measured samples than the probability distribution's true mean and to compensate for this one divides by $N-1$ instead of N in Sample Variance. The expression becomes

$$\mathrm{var}(X)_{\mathrm{smp}} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \langle X \rangle)^2. \tag{2.3c}$$

If the number of samples is large enough, though, dividing by $N-1$ does not make much different from dividing by N. Hence, the Population Variance is defined as

$$\mathrm{var}(X)_{\mathrm{pop}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \langle X \rangle)^2. \tag{2.3d}$$

In this report, the Population Variance is the expression for the variance which will be used when calculating the variance for any set of samples.
Closely related to the variance is the Standard Deviation $\sigma = \sqrt{\langle \delta^2 X \rangle}$. This is the actual expectation of how much the random variable deviates from its expectation value, as the variance is the square of this value.

Covariance $\langle \delta X_i \delta X_j \rangle$

Assume we have a set of random variables $X = \{X_1, X_2 \dots X_n\}$. Sometimes random variables are not independent but vary together. This is calculated as the average product of the two random variables deviation from their mean value.

$$\langle \delta X_i \delta X_j \rangle = \int_{\mathbb{R}^n} \delta X_i \delta X_j f(X_1, X_2, \dots X_n) \mathrm{d}X_1 \mathrm{d}X_2 \dots \mathrm{d}X_n \tag{2.4}$$

The covariance between the entire set of random variables can be described by the covariance matrix as

$$\mathrm{cov}(X) = \begin{pmatrix} \langle \delta^2 X_1 \rangle & \langle \delta X_1 \delta X_2 \rangle & \dots & \langle \delta X_1 \delta X_n \rangle \\ \langle \delta X_2 \delta X_1 \rangle & \langle \delta^2 X_2 \rangle & \dots & \langle \delta X_2 \delta X_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \delta X_n \delta X_1 \rangle & \langle \delta X_n \delta X_2 \rangle & \dots & \langle \delta^2 X_n \rangle \end{pmatrix} \tag{2.5}$$

where we note the diagonal is just the variance of each random variable. We can also note that the covariance matrix is symmetric since $\langle \delta X_i \delta X_j \rangle = \langle \delta X_j \delta X_i \rangle$.

The moments of a random distribution

The moments of a probability distribution is an important feature, which can be seen as a class of expectation values[6]. For a random variable X with PDF $f(X)$ and every integer n the raw moment is

$$\langle X^n \rangle = \int_{-\infty}^{\infty} X^n f(X) \mathrm{d}X. \tag{2.6}$$

Note that the first raw moment is the expectation value. The nth central moment is the moment centred around the expectation value and is calculated as

$$\langle \delta^n X \rangle = \langle (X - \langle X \rangle)^n \rangle = \int_{-\infty}^{\infty} (X - \langle X \rangle)^n f(X) \mathrm{d}X. \tag{2.7}$$

We can now note that the first central moment is always be zero and the second central moment is the variance. The third and fourth central moments are related to what is known as skewness and kurtosis.
If the random variable is discrete, we have to use the discrete expression for the expectation value from equation (2.2b).
When calculating the central moment of a set of samples $\tilde{X} = \{x_1, x_2 \dots x_N\}$ one uses the following expression

$$\langle \delta^n \tilde{X} \rangle = \frac{1}{N} \sum_{i=1}^{N} (x_i - \langle \tilde{X} \rangle)^n, \tag{2.8}$$

which is the expression which will be used throughout the rest of this report when calculating the moments of an ensemble.
In this report, when moments are mentioned from now on, it refers to the central moments, except for the first moment which indicates the expectation value.

Mixed moments

In the same way, as the moments are related to the variance, the mixed moments are linked to the covariance. If there are n random variables $\{X_1, X_2 \dots X_n\}$, the mixed moment of order m is

$$\langle \delta X_{i_1} \delta X_{i_2} \dots \delta X_{i_m} \rangle = \int_{\mathbb{R}^n} \delta X_{i_1} \delta X_{i_2} \dots \delta X_{i_m} f(X) \mathrm{d}X_{i_1} \mathrm{d}X_{i_2} \dots \mathrm{d}X_{i_n}, \tag{2.9}$$

where $i_1, i_2 \dots i_m$ can be any index of the variables, even all the same. The second order mixed moment is the covariance (or variance if the indexes

are the same) and higher order mixed moments describe higher orders of interdependence between random variables.
Independent variables have covariance zero. Higher order mixed moments are not necessarily zero for independent variables, however. For example one of the fourth order mixed moments for two random variables X_1 and X_2 is

$$\left\langle \delta^2 X_1 \delta^2 X_2 \right\rangle = \int_{\mathbb{R}^2} (X_1 - \langle X_1 \rangle)^2 (X_2 - \langle X_2 \rangle)^2 f(X_1, X_2) \mathrm{d}X_1 \mathrm{d}X_2, \tag{2.10}$$

which is not zero in general, even for independent variables, since the quantity being integrated is positive or zero.

2.2. Input-parameter uncertainty and the propagation of such

All measured values have some level of uncertainty associated with them. It can come from the finite resolution of the instrument, such a ruler not giving an accurate measurement smaller than the millimeter scale or the digital thermometer only showing one decimal. It can come from the skill of the operator, such as the problem of pressing the button on a stopwatch at the exact right time. It can be a systematic uncertainty, such as the instrument having some error in its calibration. It can even come from the item being measured whose properties are not stable, such as measuring room temperature which will change during the day. In short, when making any kinds measurements, there is always an uncertainty to take into account. If we do not know anything about the uncertainty of a measurement, the measurement is practically worthless.
As an example, assume we have built a steam-boiler, and we want to measure the pressure in it to be sure it is not too high. We put a sensor in it and read its value. The pressure in the steam boiler will not be constant, but is expected to vary slightly during the day. From time to time we take a look at the sensor note the value. After a few days, we have gathered some measurements which have all given slightly different results. The different measurements are shown in a blob plot in Figure 2.1. The average value of the measurements is marked by the dotted line.

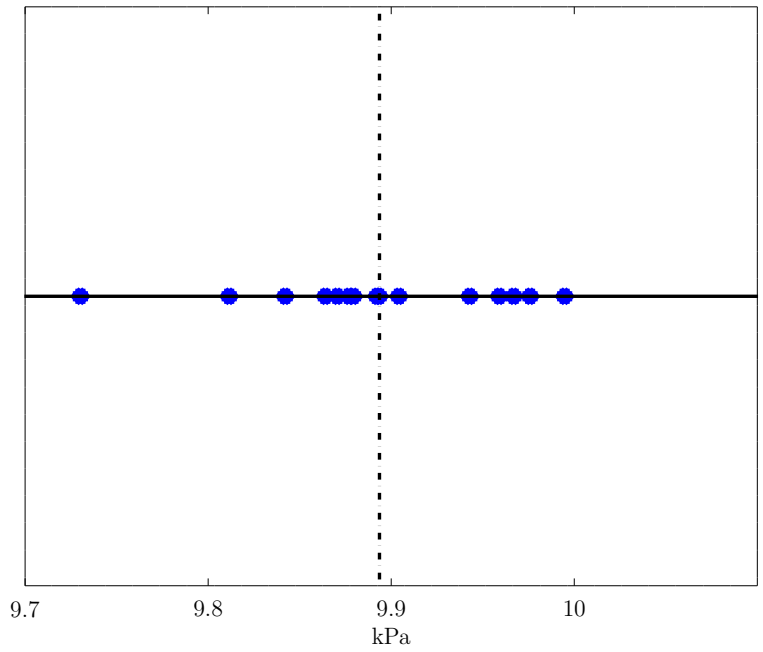


Figure 2.1: A example blob plot showing the distribution of measured values. The average value is marked by a dotted line.

If someone were to ask what the pressure in the boiler is, we could answer with the mean value, around 9.89 kPa, and believe that this is a decent approximation. However, if there were known safety issues with running the boiler with a pressure higher than 10 kPa, we should probably not feel safe in knowing our boiler runs at pressure 9.89 kPa without taking the value's uncertainty into account.
The question of how far from the mean value the actual value is expected to be can be answered with the standard deviation, which represents the average deviation from the mean value. This can be calculated by taking the square root of the sample variance from equation (2.3c), and in this example, we would maybe get a standard deviation of $\sigma \approx 0.07$ kPa. If the measurements are distributed with a Gaussian distribution, which is the most common distribution, it means that based on those measurements there is a probability of 68% that the a measurement lies within one standard deviation or one sigma. The likelihood that the a measurement lies within two sigmas is 95% and that it lies within three sigmas 99.7%.
In this example though we only care about the pressure not being too high, so we just need to look at the one-sided probability. The probability of the value not being greater than one sigma from the mean value is around 84% and that it is bellow two sigma around 98%
This means that in the example with the steam boiler we could say that with 84% confidence the pressure will be bellow 9.96 kPa, which is below the safety limit, but we cannot promise 98% confidence since two sigmas above the mean would be 10.03 kPa which is too large.

2.2.1. Uncertainty propagation

Let's now assume a harder problem. Suppose we have the same steam boiler but instead of knowing a safety limit on the pressure, we now that the walls of the boiler can only take a maximum amount of stress to run safely. We can calculate the stress S on the boiler's walls by knowing the boiler's pressure p along with its dimensions, i.e. radius r and wall thickness t, all of which can be measured, but will have some uncertainty.
The question is now what the uncertainty in $S(p,r,t)$ is and what is the probability that this stress will not be too high, which will depend on the uncertainty of the parameters p, r and t. This is the problem of uncertainty propagation.
In general uncertainty propagation includes some function $f(q_1, q_2, \ldots q_n)$ depending on a number of measured parameters. These parameters have some probability distribution which describes our uncertainty in what their actual value is. Uncertainty propagation is the problem of using the input parameters distribution to find out what the uncertainty in f is. Figure 2.2 illustrates this for two parameters.

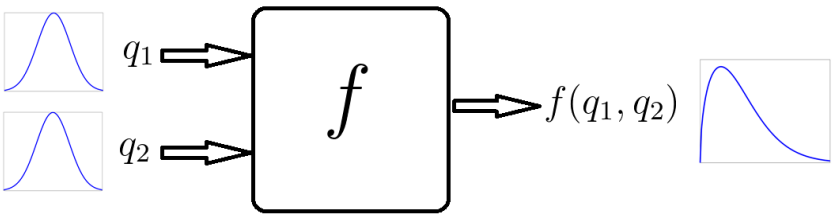


Figure 2.2: Uncertainty propagation through a function of two parameters. The input parameters have some distribution and the distribution of the function will depend on those.

In some simple cases this can be calculated analytically, but often one needs to use numerical methods. There are many numerical methods for uncertainty propagation, a class of such methods commonly used is Monte Carlo-methods. One of the most straightforward Monte Carlo methods is the Random Sampling.

2.3. Random Sampling for uncertainty propagation

Propagating uncertainty with Random Sampling is a very simple brute-force approach. Assume we have a model $f(q)$ which takes a parameter q as input. Assuming the statistical distribution of q is known Random Sampling works by randomizing a large set of samples $\tilde{q} = \{q^{(1)}, q^{(2)} \dots q^{(N)}\}$ from the known (or assumed) distribution of q. The function values $f(q^{(i)})$ are calculated at each point, and the mean value and variance are calculated from the result. If this is done with enough samples, the output values will give an accurate picture of the statistical distribution the model gives based on the distribution of the input parameters.
The problem with RS is, as mentioned in the introduction that it requires many samples. Often this is not an issue, but in some cases where the computational load is heavy, for example in CFD where a single simulation can take hours, or even days[2], the Monte Carlo approach is not an option. This high time-cost is a problem intended to be fixed by Deterministic Sampling.

2.4. Deterministic Sampling

Uncertainty propagation is a problem which can be solved by Monte Carlo methods, but as noted, there can be a problem with high computational time-cost. However, what if we could perform Monte Carlo with just a few samples? Random Sampling works by randomizing a large number of samples which, due to probability, will give a good result if there are enough of them. If we could instead pick a small set of points, which represents the input parameters distribution well, this could give a good result when used to propagate uncertainty.

Deterministic Sampling (DS) as a concept was introduced by Hessling in 2013[1] but has its origin in the Unscented Transform from back in 1995[3]. It works similarly to Random Sampling, but instead of randomizing thousands of points it determines a small ensemble of weighted points which can operate as a quicker alternative to Monte Carlo. An illustration of this is shown in Figure 2.3.

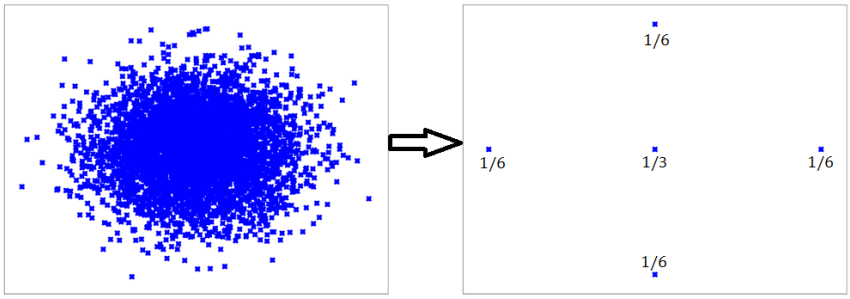


Figure 2.3: An illustration of how a set of samples, randomized from a probability distribution, can be approximated by a small deterministically chosen weighted ensemble.

This section will begin with a mathematical motivation intended to show why the idea of DS makes sense. The DS as a method will be described in more detail before describing how such an ensemble can be determined, and why a weighted ensemble is used rather than keeping all weights equal. It also intends to show why determining an ensemble with only positive weights is not trivial but a goal worth striving for.

2.4.1. Motivation

Assume uncertainty propagation through a function $f(q)$ is being performed by Random Sampling with N samples $\tilde{q} = \{q^{(1)}, q^{(2)} \dots q^{(N)}\}$. The mean and variance, which one is usually looking for, is

$$\langle f(\tilde{q}) \rangle = \frac{1}{N} \sum_{i=1}^{N} f(q^{(i)}) \tag{2.11a}$$

and

$$\left\langle \delta^2 f(\tilde{q}) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \left[f(q^{(i)}) - \langle f(\tilde{q}) \rangle \right]^2 \tag{2.11b}$$

Expressing everything in the deviation from $\mu = \langle \tilde{q} \rangle$, i.e. $q^{(i)} = \delta q^{(i)} + \mu$ and by performing a Taylor expansion around $\langle \tilde{q} \rangle$ one gets

$$f(q^{(i)}) = f(\mu + \delta q^{(i)}) = \sum_{j=0}^{\infty} \frac{1}{j!} \delta^j q^{(i)} \frac{\mathrm{d}^j f}{\mathrm{d}q^j}(\mu) = \tag{2.12}$$

$$= f(\mu) + \delta q^{(i)} \frac{\mathrm{d}f}{\mathrm{d}q}(\mu) + \frac{1}{2} \delta^2 q^{(i)} \frac{\mathrm{d}^2 f}{\mathrm{d}q^2}(\mu) + \dots$$

Let's insert this expression for $f(q^{(i)})$ into equation (2.11a) and hence find a new expression for the propagated mean value. We get

$$\langle f(\tilde{q}) \rangle = \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{j=0}^{\infty} \frac{1}{j!} \delta^j q^{(i)} \frac{\mathrm{d}^j f}{\mathrm{d}q^j}(\mu) \right]$$

which can be rewritten as

$$\langle f(\tilde{q}) \rangle = \sum_{j=0}^{\infty} \frac{1}{j!} \left[\frac{1}{N} \sum_{i=1}^{N} \delta^j q^{(i)} \right] \frac{\mathrm{d}^j f}{\mathrm{d}q^j}(\mu).$$

One can now notice that the sum inside the bracket is the expression for the moment of order j. Hence an expression for the propagated mean value is

$$\langle f(\tilde{q}) \rangle = \sum_{j=0}^{\infty} \frac{1}{j!} \langle \delta^j \tilde{q} \rangle \frac{\mathrm{d}^j f}{\mathrm{d}q^j}(\mu). \tag{2.13a}$$

The same thing can be done for the propagated variance by inserting equation (2.12) into (2.11b). The derivation becomes a bit more complicated but the propagated variance expressed in the moments becomes

$$\langle \delta^2 f(\tilde{q}) \rangle = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{i!j!} \left[\langle \delta^{i+j} \tilde{q} \rangle - \langle \delta^i \tilde{q} \rangle \langle \delta^j \tilde{q} \rangle \right] \frac{\mathrm{d}^i f}{\mathrm{d}q^i} \frac{\mathrm{d}^j f}{\mathrm{d}q^j}. \tag{2.13b}$$

The important thing to notice here is that what matters for the calculated mean and variance of the output is not the number of points in the ensemble, but only the statistical moments of the ensemble. This means there is not necessarily a reason to perform thousands of simulation. If a small ensemble fulfills enough moments, it will give a good approximation of the propagated uncertainty.

2.4.2. Description

Deterministic Sampling is a method for propagating uncertainty through a model using a small set of samples (called sigma-points) which represent the statistical distribution of the model's parameters.
Once the sigma-points are chosen the method works just like Monte-Carlo methods, meaning the function is evaluated at each of the points and the statistical properties of the function-values are calculated.

For example, assume a function $f(q)$, depending on the parameter q which has some known uncertainty with mean value $\langle q \rangle$ and variance $\langle \delta^2 q \rangle$. Also, assume the mean value and variance of $f(q)$ is sought. This can be approximated by selecting an ensemble of two sigma-points, both one standard deviation away, which would be the points

$$\begin{aligned} \tilde{q} &= \{q^{(1)}, q^{(2)}\} \\ q^{(1)} &= \langle q \rangle + \sqrt{\langle \delta^2 q \rangle} \\ q^{(2)} &= \langle q \rangle - \sqrt{\langle \delta^2 q \rangle} \end{aligned} \tag{2.14}$$

Note that this ensemble has the same average value and variance as the the parameter q. Now we can approximate $\langle f(q) \rangle$ and $\langle \delta^2 f(q) \rangle$ by calculating the mean and variance for $f(\tilde{q})$ as

$$\langle f(q) \rangle \approx \langle f(\tilde{q}) \rangle = \frac{f(q^{(1)}) + f(q^{(2)})}{2}$$

and

$$\langle \delta^2 f(q) \rangle \approx \langle \delta^2 f(\tilde{q}) \rangle = \frac{(f(q^{(1)}) - \langle f(\tilde{q}) \rangle)^2 + (f(q^{(2)}) - \langle f(\tilde{q}) \rangle)^2}{2}.$$

If $f(q)$ happens to be a linear function this would be an equality, but for non-linear functions, it is just an approximation.
This idea can be extended to functions of several parameters. A variant of Deterministic Sampling used for propagating the uncertainty of several parameters with covariance is called the unscented Kalman filter[1]. It builds an ensemble which encodes covariance in the following way. Assume n parameters $\{q_1, q_2 \dots q_n\}$ with a dependence described by covariance matrix C. An ensemble of $2n$ sigma-points which encode this covariance is

$$\begin{aligned} q^{(\pm,i)} &= \langle q \rangle \pm \sqrt{n} \Delta(i,:), \\ \Delta \Delta^{\mathsf{T}} &= C, \end{aligned} \tag{2.16}$$

where $\Delta(i,:)$ refers to the ith row of Δ. In this ensemble each sigma-point is now a point in n-space. Note how ensembles (2.4.2) and (2.16) are quite similar. For several parameters $\sqrt{\langle \delta^2 q \rangle}$ has been changed to the rows of the matrix Δ, which is the matrix square root of the covariance matrix, and would in the case $n = 1$ revert to ensemble (2.4.2).
The ensembles mentioned here encodes the first two statistical moments. The equations (2.13) tells us that to make a better approximation, the way to do this is to encode more of the statistical moments into the ensemble.
In principal, this is the only thing one needs to know about deterministic sampling. Select an ensemble which satisfies the statistical moments of the

parameter to a sufficiently high order and evaluates the function at these points to get the propagated uncertainties. This works for functions in several dimensions and with interdependent variables. The entire difficulty when solving the problem of error propagation with Deterministic Sampling is to find the ensemble.

2.4.3. Determining the ensemble

The more statistical moments an ensemble fulfills, the more precise the result of the uncertainty propagation will be. Hence, we shall force our set of sigma-points to contain the correct mean value, variance, third moment and so on.
In this section, the problems with determining an ensemble will be presented in a general theoretical manner. A more practical example of this can be found in Appendix E.1.
So the equations our ensemble $\tilde{q} = \{q^{(1)}, q^{(2)} \dots q^{(N)}\}$ should fulfil, with $\delta^k q^{(i)}$ denoting $(q^{(i)} - \langle \tilde{q} \rangle)^k$, are

$$\begin{cases} \langle q \rangle & = \frac{1}{N} \sum_{i=1}^{N} q^{(i)} \\ \langle \delta^2 q \rangle & = \frac{1}{N} \sum_{i=1}^{N} \delta^2 q^{(i)} \\ \langle \delta^3 q \rangle & = \frac{1}{N} \sum_{i=1}^{N} \delta^3 q^{(i)} \\ \vdots & \quad \vdots \\ \langle \delta^m q \rangle & = \frac{1}{N} \sum_{i=1}^{N} \delta^m q^{(i)} \end{cases} \tag{2.17}$$

Note that the expression used to calculate the variance, as well as higher order moments, is the Population Variance from equation (2.3d) and not the Sample Variance from equation (2.3c), even though we have a small set of samples. This is because the set is chosen to encode the exact mean value hence the correction gained from dividing by $N-1$ instead of N is not needed and would, in fact, give an incorrect value.
So if the ensemble should encode m moments, a system of m non-linear equations need to be solved. This system has a finite set of solutions if $m = N$, which seems splendid at first sight. There are two problems here, however. One is that equation (2.17) is messy to work with since it is a system of non-linear equations. There are algorithms which solve these equations numerically, so this may be manageable, but the other problem is that while the system is guaranteed to have solutions, they are not guaranteed to be real-valued, which can be a problem.
If there are several parameters $q_1, q_2 \dots q_n$ we get the same set of equations for each parameter q_i. We may also care about the mixed moments, meaning the parameters q_{i_1} and q_{i_2} may or may not be dependent on each other and

we may want to encode this into our ensemble. The system now becomes even more complicated.

$$\begin{cases} \langle q_{i_1} \rangle & = \frac{1}{N} \sum_{i=1}^{N} q_{i_1}^{(i)} \\ \langle \delta q_{i_1} \delta q_{i_2} \rangle & = \frac{1}{N} \sum_{i=1}^{N} \delta q_{i_1}^{(i)} \delta q_{i_2}^{(i)} \\ \langle \delta q_{i_1} \delta q_{i_2} \delta q_{i_3} \rangle & = \frac{1}{N} \sum_{i=1}^{N} \delta q_{i_1}^{(i)} \delta q_{i_2}^{(i)} \delta q_{i_3}^{(i)} \\ \vdots & \vdots \end{cases} \tag{2.18}$$

Here there is one equation for each moment and each combination of $i_1, i_2, i_3, i_4 \ldots \in \{1, 2, 3 \ldots n\}$.
So, there is a solution, but it may be hard to find, and it may not be real valued. We may hence not always be able to find an ensemble to fulfill the moments we want by solving (2.17). However, there is a way of making this system of equations easier to solve, and making sure the solution is real, by associating a weight to each sigma-point in the ensemble[2].

2.4.4. A weighted ensemble

This section describes the weighted ensemble in a general and theoretical manner. For a clarifying example see Appendix E.2.
Instead of doing the tough job of solving the system of nonlinear equations, with potentially complex solutions, from equation (2.17) and (2.18), we could associate a weight $w^{(i)}$ to each sigma-point $q^{(i)}$. The mean value and statistical moments are then calculated as

$$\langle \tilde{q} \rangle = \sum_{i=1}^{N} w^{(i)} q^{(i)} \tag{2.19a}$$

and

$$\langle \delta^m \tilde{q} \rangle = \sum_{i=1}^{N} w^{(i)} (q^{(i)} - \langle \tilde{q} \rangle)^m \tag{2.19b}$$

for the ensemble and the propagated mean value and moments become

$$\langle f(\tilde{q}) \rangle = \sum_{i=1}^{N} w^{(i)} f(q^{(i)}) \tag{2.20a}$$

and

$$\langle \delta^m f(\tilde{q}) \rangle = \sum_{i=1}^{N} w^{(i)} (f(q^{(i)}) - \langle f(\tilde{q}) \rangle)^m \tag{2.20b}$$

given that $\sum w^{(i)} = 1$, otherwise one would have to divide by the sum. Note that setting $w^{(i)} = \frac{1}{N}$ the non-weighted state is returned.
The equation to solve for the weighted ensemble would now be

$$\begin{cases} 1 & = \sum_{i=1}^{N} w^{(i)} \\ \langle q \rangle & = \sum_{i=1}^{N} w^{(i)} q^{(i)} \\ \langle \delta^2 q \rangle & = \sum_{i=1}^{N} w^{(i)} (q^{(i)} - \langle q \rangle)^2 \\ \langle \delta^3 q \rangle & = \sum_{i=1}^{N} w^{(i)} (q^{(i)} - \langle q \rangle)^3 \\ \quad \vdots & \\ \langle \delta^m q \rangle & = \sum_{i=1}^{N} w^{(i)} (q^{(i)} - \langle q \rangle)^m \end{cases} \tag{2.21}$$

So instead of solving equation (2.17) for the values of q_i one can now set q_i to any reasonable value and solve equation (2.21) for the weights w_i. Now there are $m+1$ equations for an ensemble with m encoded moments, and need hence set $N = m+1$ for the system to have a unique solution. This system is linear when solving for w_i. Hence, it is easy to solve and will always have a real-valued solution.

This means we can always find a weighted ensemble with any amount of statistical moments encoded. This can be extended to several parameters, with correlations, yet one problem remains. There is no guarantee that the weights are positive. In fact, they may be any real number, depending on what the sigma-points are. In some cases this is not a problem, the ensemble still fulfills the requirements and can be used to propagate uncertainty in some cases.

However, using negative weights is not a good idea. A weighted ensemble is intended to work as a discrete approximation of a continuous probability distribution. An ensemble with negative weights is, in principle, a poor approximation since a probability can not be negative. Even if the first moments of the ensemble are correct, the higher order moments can be way off, even get unrealistic values. The propagated results from the use of such an ensemble can also be wrong and even give prohibited values, such as negative even moments[2]. An example of this is shown in Appendix E.2.

It is then clear that for deterministic sampling to be useful in the general case a way of finding an ensemble with positive weights is needed.

2.5. Linear optimization with the Simplex Method

A significant tool for finding an ensemble with positive weights has shown to be the linear optimization method known as the Simplex Method and for this reason, the concept will be described briefly. For a more detailed explanation see Appendix B.1.

Assume we want to maximize or minimize some quantity z which depends linearly on some parameters,

$$z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n,$$

and who's parameters are subject to some linear constraints

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m. \end{aligned}$$

The Simplex Method is an algorithm which will optimize the quantity z subject to the defined constraints. It does so with two important features which are the reason this method is useful here.

1. None of the parameters x_i will get a negative value.
2. If the optimization can be done with some parameters x_i equal to zero, they will be set to zero.

It is for these two reasons the Simplex Method is used in this project, rather than for optimization.

3. Methodology

The problem of finding a reliable method of generating an ensemble of sigma-points representing an arbitrary distribution is approached by a few different strategies. Since an important part of the projects aim is to find ensemble with positive weights, how to calculate weights of an ensemble has here given extra thought.
For computations, the interpretive language GNU Octave, which is very similar to MATLAB, has been used.
This chapter will begin with a description notation utilized in this report. Ensembles and their weights are here represented by matrices and vectors, and this representation is described in Section 3.1.
Next up, in Section 3.2, is a description of Simplex Reduction, a new method based on the Simplex Method for Linear optimization. Simplex Reduction, which is used for calculating ensemble weights and reducing ensemble size, is very central to many of the other new approaches presented in this report.
Then there is a short primer on how to create an ensemble for parameters with a Gaussian distribution in Section 3.3. This leads into the presentation of the Block-Diagonal Gaussian ensemble in Section C.1 which can represent any number of independent Gaussian parameters.
After this, the Shotgun Algorithm is presented in Section 3.4, which is a new method for generating a one-dimensional ensemble for any parameter with any statistical distribution. It uses a combination of randomization and Simplex Reduction.
This is followed by a by a description of how to combine ensembles and build a multidimensional ensemble with any distribution, which is presented in Section 3.5.
Then the problem of encoding correlation into an ensemble is addressed. A solution is shown in two cases, both for Gaussian parameters, in Section 3.6.2 and for parameters with general distributions in Section 3.6.1.
In Section 3.7.1 methods for creating an ensemble for symmetrical distributions, in general, is described. The Heavy Middle ensemble, which can represent four moments for any symmetric distribution is presented in Section C.1.1.
Finally, in Section 3.9, there will be a short description of how the methods have been tested for their use in uncertainty propagation with deterministic sampling.

3.1. Notation used for representing an ensemble

This section describes how an ensemble for Deterministic Sampling will be represented throughout this report.
Here an ensemble, meaning a set of sigma-points representing a random variable q or a set of random variables $\{q_1, q_2, \ldots q_n\}$, is denoted $\tilde{q}$. The ensemble's sigma-points are represented with an upper index, i.e. $\tilde{q} =$

$\{q^{(1)}, q^{(2)} \dots q^{(N)}\}$.
For convenience, an ensemble is usually be represented in matrix form where each row is a sigma-point, and each column represents the components for each parameter. The corresponding set of weights $\tilde{w}$ is represented by a column vector. In the general case with an ensemble for n parameters with N sigma-points is

$$\tilde{q} = \begin{pmatrix} q_1^{(1)} & q_2^{(1)} & \dots & q_n^{(1)} \\ q_1^{(2)} & q_2^{(2)} & \dots & q_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ q_1^{(N)} & q_2^{(N)} & \dots & q_n^{(N)} \end{pmatrix}_{N \times n} \qquad \tilde{w} = \begin{pmatrix} w^{(1)} \\ w^{(2)} \\ \vdots \\ w^{(N)} \end{pmatrix}_{N \times 1} . \tag{3.1}$$

This way of writing the ensemble turns out to be useful for two reasons. For one, the ensemble in matrix form can be scaled, translated, skewed by well-known mathematical operations. Those can also be used to calculate the statistical moments of the ensemble. Secondly, this fits very well when implementing DS in MATLAB or Octave.
In this notation the mean value of the ensemble is calculated as

$$\langle \tilde{q} \rangle^{\mathsf{T}} = \tilde{q}^{\mathsf{T}} \tilde{w} \tag{3.2a}$$

and the moment of order m is calculated as

$$\langle \delta^m \tilde{q} \rangle^{\mathsf{T}} = \left(\tilde{q}^{\mathsf{T}} - 1_{1 \times N} \otimes \tilde{q}^{\mathsf{T}} \tilde{w} \right)^{\circ m} \tilde{w}. \tag{3.2b}$$

Here $1_{1 \times N}$ refers to a matrix of ones with dimension $1 \times N$, the symbol $\otimes$ is the Kronecker product and the notation $A^{\circ m}$ is the Hadamard power m if A, i.e. denotes that each element of A is taken to the power of m.
The mean value and moments are in this definition row vectors containing the values for each parameter, i.e.

$$\langle \tilde{q} \rangle = \begin{pmatrix} \langle q_1 \rangle & \langle q_1 \rangle & \dots & \langle q_n \rangle \end{pmatrix} \tag{3.3a}$$

and

$$\langle \delta^m \tilde{q} \rangle = \begin{pmatrix} \langle \delta^m q_1 \rangle & \langle \delta^m q_2 \rangle & \dots & \langle \delta^m q_n \rangle \end{pmatrix}. \tag{3.3b}$$

Exemple

As an example, assume two parameters q_1 and q_2 with a Gaussian distribution. Let them both have mean value $\langle q_i \rangle = 0$ and variance $\langle \delta^2 q_i \rangle = 1$. Their third moment would then be $\langle \delta^3 q_i \rangle = 0$ and their fourth moment $\langle \delta^4 q_i \rangle = 3$.
One ensemble which encodes these first four moments is the five points

$$\tilde{q} = \{(\sqrt{3}, \sqrt{3}), (\sqrt{3}, -\sqrt{3}), (-\sqrt{3}, \sqrt{3}), (-\sqrt{3}, -\sqrt{3}), (0,0)\} \tag{3.4a}$$

with weights

$$\tilde{w} = \left\{ \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{2}{3} \right\}. \tag{3.4b}$$

How such an ensemble can be found is described in Section 3.7.1.
This ensemble is here represented in matrix form as

$$\tilde{q} = \sqrt{3}\begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \\ 0 & 0 \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} 1/12 \\ 1/12 \\ 1/12 \\ 1/12 \\ 2/3 \end{pmatrix}. \tag{3.5}$$

The mean value of the ensemble is, when calculated from equation (3.2a),

$$\langle \tilde{q} \rangle^{\mathsf{T}} = \tilde{q}^{\mathsf{T}}\tilde{w} = \sqrt{3}\begin{pmatrix} 1 & 1 & -1 & -1 & 0 \\ 1 & -1 & 1 & -1 & 0 \end{pmatrix}\begin{pmatrix} 1/12 \\ 1/12 \\ 1/12 \\ 1/12 \\ 2/3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.6}$$

Its variance, calculated from equation (3.2b), becomes

$$\left\langle \delta^2 \tilde{q} \right\rangle^{\mathsf{T}} = \left(\tilde{q}^{\mathsf{T}} - 1_{1\times 5} \otimes \tilde{q}^{\mathsf{T}}\tilde{w} \right)^{\circ 2} \tilde{w} =$$

$$= \left[\sqrt{3}\begin{pmatrix} 1 & 1 & -1 & -1 & 0 \\ 1 & -1 & 1 & -1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right]^{\circ 2} \begin{pmatrix} 1/12 \\ 1/12 \\ 1/12 \\ 1/12 \\ 2/3 \end{pmatrix} = \tag{3.7}$$

$$= \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Performing the same calculation for the third and fourth moments would also produce the correct results.

3.2. Calculating positive weights and reducing ensemble size with Simplex Reduction

The aim of this project is to find a way to create ensembles for Deterministic Sampling with positive weights. This is difficult since, for general distributions, there is no known way of finding a mathematical expression to calculate where the sigma-points should be to get positive weights.
What is known is that the weighted set of points need to fulfill equation (2.21). This could be done by either letting all weights be equal and then solving for the sigma-points $q^{(i)}$, which will usually give complex solutions. Alternatively, one could pick some reasonable sigma-points $q^{(i)}$ and then solving for the weights, which will usually give negative weights.
The method called Simplex Reduction is a new approach to this problem and can be used to find an ensemble with positive weights. It is based on

the Simplex Method for Linear optimization, described in Section 2.5 and Appendix B.1, hence its name and will seek positive solutions for the weights of equation (2.21) and, if possible, give unnecessary points the weight zero. This means that if there are more points than needed Simplex Reduction can find a solution with positive weights and reduce the ensemble size.
This method is the central part of the Shotgun Algorithm described in Section 3.4 and can be used to reduce the size of most other ensembles presented in this report. It is also used to recalculate weights to encode covariance in Gaussian ensembles, as described in Section 3.6.2.
In this section, all that will be said about Simplex Reduction is that it is an algorithm that calculates weights to a set of sigma-points given a set of moments that are to be encoded. If there are more sigma-points than needed these get the weight zero and hence these can be removed to reduce the ensemble size. A more detailed explanation of the algorithm can be found in Appendix B.2.
Note that it is not claimed that the reduced ensemble gained is the of the minimum size, just that it is a working set of positive weights which may reduce ensemble size.

3.3. Creating Gaussian ensembles

An ensemble for parameters with Gaussian distributions can be found by the general technique described in chapters 3.4, a feature of the Gaussian distribution is that it is, in this case, easier to find a general expression for the ensemble while the non-symmetric distributions will some element of randomizing when searching for an ensemble.
The Gaussian ensemble presented here will turn out to be a lot quicker to create, and be much smaller than the ensemble for general distributions and since the Gaussian distribution is the most common its special treatment is well motivated.
This chapter begins with a short primer on the Gaussian ensemble and then continue to derive a general expression for a Gaussian ensemble encoding four moments without dependence. The ensemble presented is called the Block-Diagonal Gaussian ensemble.

3.3.1. A primer on Gaussian ensembles

Assume a random parameter q distributed by a Gaussian distribution with mean value μ and standard deviation σ. Since a Gaussian ensemble can be translated and scaled to have any mean value and standard deviation we can, without loss of generality, assume than $\mu = 0$ and $\sigma = 1$ for all Gaussian distributions here, but any reasoning done is valid in the general case.

Two moments

To encode two moments in an ensemble is trivial and what it looks like has already been presented in section 2.4.2, but let's explore more precisely how one can derive it. Assume it was not obvious how to find it, but we wanted a weighted ensemble encoding mean value and variance.
Our ensemble should fulfill two moments, and we want the weights to have the sum one. This means there should be three equations fulfilled. The ensemble is written

$$\tilde{q} = \begin{pmatrix} q^{(1)} \\ q^{(2)} \end{pmatrix} \quad w = \begin{pmatrix} w^{(1)} \\ w^{(2)} \end{pmatrix}, \tag{3.8}$$

which is four unknowns with only three equations. This problem is avoided by realizing that since our distribution is symmetric our ensemble could be symmetric as well. Hence we choose to look for a symmetric ensemble and thus get the relation $q^{(1)} = -q^{(2)} = q'$. If so, the weights must be equal as well, which means $w^{(1)} = w^{(2)} = w'$. Now there are two unknowns but the mean value is automatically fulfilled and the three equations become

$$\begin{aligned} w' + w' &= 1 \\ w'q' - w'q' &= 0 \quad \text{Automatically fulfilled} \\ w'(q'-0)^2 + w'(-q'-0)^2 &= 1 \end{aligned} \tag{3.9}$$

The system is easily solved by $w' = \frac{1}{2}$ and $q' = 1$. So the ensemble becomes

$$\tilde{q}_{\text{2moms}} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}. \tag{3.10}$$

Four moments

To encode four moments in a weighted ensemble one usually needs five points to fulfill a system of five equations. Since all the odd moments are zero in the symmetric distribution, the third moment is encoded by default, if the ensemble is symmetric. So as long as the ensemble is symmetric and centered around zero both the mean value and third moment is already fulfilled. Hence, there are only three equations left not automatically fulfilled. Let's look for an ensemble encoding four moments in the form

$$\tilde{q} = \begin{pmatrix} q^{(\pm)} \\ 0 \\ -q^{(\pm)} \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} w^{(\pm)} \\ w^{(0)} \\ w^{(\pm)} \end{pmatrix} \tag{3.11}$$

i.e. one sample locked in the center and two equally weighted samples $\pm q^{(\pm)}$. This means three unknowns have to be found from the system

$$
\begin{aligned}
w^{(\pm)} + w^{(0)} + w^{(\pm)} &= 1 \\
w^{(\pm)} q^{(\pm)} + w^{(0)} 0 - w^{(\pm)} q^{(\pm)} &= 0 \quad \text{Automatically fulfilled} \\
w^{(\pm)} \left(q^{(\pm)}\right)^2 + w^{(0)} 0^2 + w^{(\pm)} \left(q^{(\pm)}\right)^2 &= 1 \\
w^{(\pm)} \left(q^{(\pm)}\right)^3 + w^{(0)} 0^3 - w^{(\pm)} \left(q^{(\pm)}\right)^3 &= 0 \quad \text{Automatically fulfilled} \\
w^{(\pm)} \left(q^{(\pm)}\right)^4 + w^{(0)} 0^4 + w^{(\pm)} \left(q^{(\pm)}\right)^4 &= 3,
\end{aligned}
\tag{3.12}
$$

since the fourth moment of the standard Gaussian distribution is 3.
From the three equations not already fulfilled one can solve $w^{(\pm)} = 1/6$, $w^{(0)} = 2/3$ and $q^{(\pm)} = \sqrt{3}$. So the Gaussian ensemble fulfilling four moments is

$$
\tilde{q}_{4\text{moms}} = \begin{pmatrix} \sqrt{3} \\ 0 \\ -\sqrt{3} \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} 1/6 \\ 2/3 \\ 1/6 \end{pmatrix}. \tag{3.13}
$$

Ensembles with more than four moments are usually not needed and deriving them will not be explored here. They can be found, though, with the Shotgun Algorithm described in section 3.4. The rest of section 3.3 will explore a way to generalize the four-moment Gaussian ensemble for many parameters while keeping all weights positive.

3.3.2. Generalize to higher dimensions - The idea

Assume n independent parameters q_i all distributed with the Gaussian distribution having a standard deviation σ_i. Since a Gaussian distribution always can be scaled to fulfill any other standard deviation, and translated to any expectation value, all of the distributions are assumed, without loss of generality, to have $\sigma_i = 1$ and $\langle q_i \rangle = 0$. The requirements of an ensemble $\tilde{q}$ representing this distribution is then to have the same first four moments as the parameter q, meaning, $\langle \tilde{q} \rangle = \langle q \rangle = 0$, $\langle \delta^2 \tilde{q} \rangle = \langle \delta^2 q \rangle = 1$, $\langle \delta^3 \tilde{q} \rangle = \langle \delta^3 q \rangle = 0$ and $\langle \delta^4 \tilde{q} \rangle = \langle \delta^4 q \rangle = 3$, since those are the moments of a standard Gaussian distribution which can be calculated from equation (2.7). For several parameters, since the parameters are assumed to be independent, the covariance and should be zero as well as most of the higher order mixed moments.
An ensemble $\tilde{q}_n$ fulfilling the first four moments of such a set of parameters can be calculated in the way described below.
The one-dimensional case as has just been shown the ensemble is

$$
\tilde{q}_1 = \sqrt{3}\sigma \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \quad \tilde{w}_1 = \begin{pmatrix} 1/6 \\ 1/6 \\ 2/3 \end{pmatrix} \tag{3.14}
$$

Since σ is set to be one, it will no longer be printed in the following equations.
Ensemble (3.14) can be extended to more dimensions, but not trivially. If we seek an ensemble working in two dimensions one could think it could simply be set to

$$\tilde{q}_2 = \sqrt{3}\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 0 & 0 \end{pmatrix} \quad \tilde{w}_2 = \begin{pmatrix} 1/6 \\ 1/6 \\ 2/3 \end{pmatrix} \tag{3.15}$$

This is indeed a working ensemble for two parameters fulfilling the first four moments, and the same pattern will work for more parameters by just adding more identical columns. It can seem like this is a valid solution, and that one could simply add more columns if one has more parameters. This is wrong though, since the parameters are assumed to be independent, but this ensemble clearly has some covariance, which can be verified with equation (2.5).
So to satisfy two independent parameters with Gaussian distributions one will have to include more sigma-points. This can be done by doing the equivalent of "splitting" the zero-point into the added dimension. This form of adding a dimension is shown in Figure 3.1.

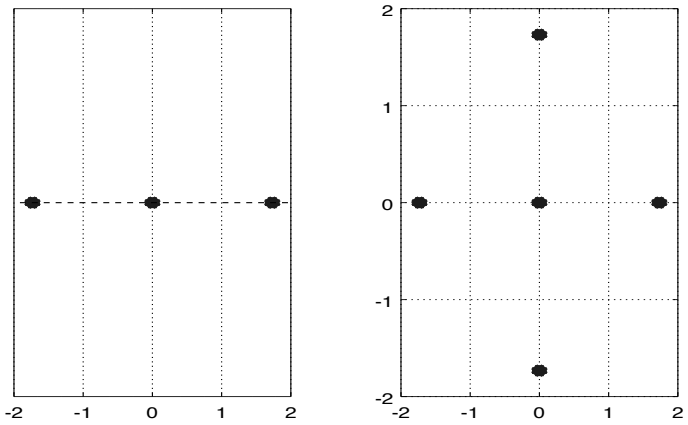


Figure 3.1: Expanding the one dimensional Gaussian ensemble into two dimensions by "splitting" the center sample into the added dimension.

For two dimensions a working ensemble with five sigma-points is

$$\tilde{q}_2 = \sqrt{3}\begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \end{pmatrix} \quad \tilde{w}_2 = \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/3 \end{pmatrix} \tag{3.16}$$

Note how each of the weight of the zero-point has been decreased while both of the parameters will still fulfill equation (2.17) independently and their covariance is zero, as seen from (2.5). One can see it as "splitting" the zero sample, i.e. taking pieces of it and creating new samples for a new parameter. This means decreasing it's weight and adding more points around it. Here both of the parameters still independently have a total weight of $2/3$ at their zeros if counted.
To increase this to three parameters, the same procedure is used, meaning the weight of the middle sample is once decreased when it is split into the next dimension, and a working ensemble is

$$\tilde{q}_3 = \sqrt{3}\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \quad \tilde{w}_3 = \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix}. \tag{3.17}$$

Ensemble (3.17) is an extension in the same way as the two parameter case, but now the zero-point has weight zero and is hence not needed. This splitting is shown in Figure 3.2.

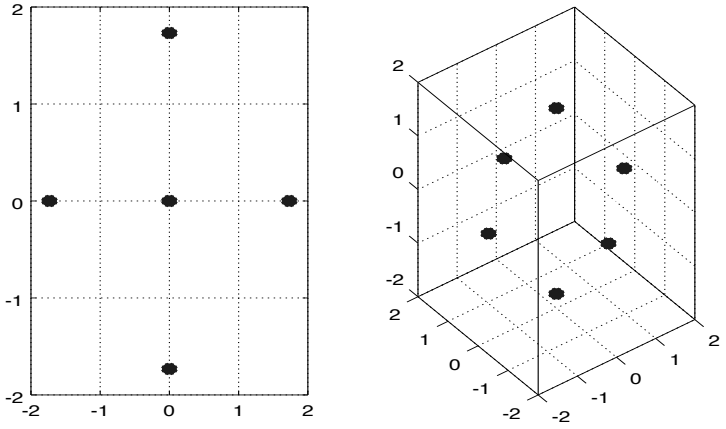


Figure 3.2: Expanding the two dimensional Gaussian ensemble into three dimensions by "splitting" the center sample into the added dimension.

One can verify all moments still are the fulfilled. The problem next is how to advance beyond this point since the central point now has weight zero and cannot be decreased further if all weights are supposed to be positive. Hence adding another parameter by "splitting" the zero sample will not work anymore. So let's split some other point instead.
Due to symmetry one can't just split one point (unless middle point), but

two points in one dimension

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

can be split into four points in two dimensions as

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix} \hookrightarrow \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}, \tag{3.18}$$

which means our three dimensional ensemble is extended to four dimensions as

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \hookrightarrow \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

The reason this works is that the columns still have mean value 0, variance 1 and are orthogonal to each other, which is important since this ensures that the ensemble doesn't have any covariance.
By applying this transformation to the matrix, one can add another parameter. The weights of these points have to be halved now, though, and the expression for the four-parameter ensemble is

$$\tilde{q}_4 = \sqrt{3} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad \tilde{w}_4 = \begin{pmatrix} 1/12 \\ 1/12 \\ 1/12 \\ 1/12 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix}. \tag{3.19}$$

If one were to perform the calculations one could see that this ensemble still fulfills all the moments while being free from dependence.
This should, of course, be generalized to an arbitrary number of parameters and one way of doing this is to view the ensemble as a diagonal-block-matrix where all columns are orthogonal. It can be described, for $n \geq 3$, on the form

$$\tilde{q}_n = \sqrt{3} \begin{pmatrix} C_a & & \\ & C_b & \\ & & C_c \end{pmatrix} \quad \tilde{w}_n = \begin{pmatrix} W_a \\ W_b \\ W_c \end{pmatrix} \tag{3.20}$$

Where the matrices C_k can be any matrix with k columns where all elements $C_k^{(i,j)} \in \{-1, 1\}$ and the columns have mean value zero, variance one and are orthogonal to eliminate covariance. C_k could be any matrix fulfilling these requirement, for example the Extended Hadamard Matrix described in Appendix D.2. W_k is the weights for each of the rows containing C_k.
In the case of 4 parameters as in equation (3.19), the parameters in (3.20) would be

$$C_a = C_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}, \quad C_b = C_c = C_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{3.21}$$

$$W_a = W_2 = \begin{pmatrix} 1/12 \\ 1/12 \\ 1/12 \\ 1/12 \end{pmatrix}, \quad W_b = W_c = W_1 = \begin{pmatrix} 1/6 \\ 1/6 \end{pmatrix}$$

This ensemble for Gaussian parameters, shown in equation (3.20), will due to it's block-diagonal shape be called the Block-Diagonal Gaussian ensemble. The explicit formula for how to generate it in any dimension is shown in Appendix C.1.

3.4. Creating an ensemble with the Shotgun Algorithm

Finding an ensemble for just one parameter is, for parameters with a Gaussian distribution solved for up to four encoded moments. The problem is figuring out such an ensemble for any distribution while keeping all weights positive. For non-symmetric distributions, there is no known general expression for an ensemble with four moments. Instead, randomization has been used to find such an ensemble.
The process of annealing, described by Hedberg and Hessling[2] has shown to work for this. Annealing means an ensemble of five (or as many points as needed to fulfill the moments) sigma-points is randomized, and weights are calculated. This is repeated many times and the best ensemble, which is the one with positive weights and most evenly distributed sigma-points, is saved.
Here, a new way of performing the randomization of an ensemble is described. Instead of randomizing just as many points as needed, it randomizes many times more and then removes the points not needed. This approach can be generalized to several parameters, and it has been named the Shotgun Algorithm.

The Shotgun Algorithm for one parameter

Assuming a parameter q with a known distribution. To create an ensemble fulfilling the m first moments, one can generate a working ensemble by following these steps:

1. Randomise $N \gg m$ points $\{q^{(1)}, q^{(2)} \dots q^{(N)}\}$ from the distribution of q, i.e. many more points than is expected to be required to encode the moments wanted. For $m = 4$, $N = 100$ has been used by the author.

2. Use Simplex Reduction described in Section 3.2 to get a solution where all of the weights which can be set to zero, will be set to zero while fulfilling $w_i \geq 0$. If no solution exists for the randomized samples, steps 1-2 are repeated.

3. All of the sigma-points which have zero-weights are removed. What remains is an ensemble with a small number of points and weights fulfilling the desired moments.

3.5. Combining ensembles, creating a multi-parameter ensembles without correlation

By this point, we have a way of creating an ensemble for a parameter with any distribution by the Shotgun Algorithm, and a way of generating a multi-dimensional ensemble of Gaussian parameters. A method for creating an ensemble for several parameters with an arbitrary distribution has not been presented, however. One could as stated, use the Shotgun Algorithm in several dimensions, which works but will have problems with higher dimensions. The solution presented here is to first generate one-dimensional ensembles for the individual parameters and then combine them with each other to form a multi-dimensional ensemble. The process of combining the ensembles is simple in principal but can cause practical problems with the ensemble size growing too much for many parameters.
Assume one is looking for an ensemble representing the first 4 moments of n uncorrelated parameters $\{q_1, q_2 \dots q_n\}$. Individually ensembles $\{\tilde{q}_1, \tilde{q}_2 \dots \tilde{q}_n\}$ can be found with the Shotgun Algorithm, or some other way. If one wants to propagate the uncertainty through a function of all parameters $f(q_1, q_2 \dots q_n)$, an n-dimensional ensemble is needed, however. This can also be found with the Shotgun Algorithm by randomizing points in n dimension and removing unneeded points with Simplex Reduction. This is not needed though since n one-dimensional ensembles can be combined into an n dimensional ensemble fulfilling their individual moments, and encoding dependence or independence between them. In this section, the simpler process for independent parameters is described. How to get an ensemble with dependence is outlined in section 3.6.1.

Combining Ensembles

Two ensembles $\tilde{q}_1$ and $\tilde{q}_w$ can be combined into a higher-dimensional ensemble by simply taking every combination $(q_1^{(i)}, q_2^{(j)})$ and for each such sigma-point multiplying the individual weights together as well, getting the corresponding weight $w_1^{(i)} w_2^{(j)}$. This will give the ensemble

$$\tilde{q} = \begin{pmatrix} q_1^{(1)} & q_2^{(1)} \\ q_1^{(1)} & q_2^{(2)} \\ \vdots & \vdots \\ q_1^{(1)} & q_2^{(N_2)} \\ q_1^{(2)} & q_2^{(1)} \\ \vdots & \vdots \\ q_1^{(N_1)} & q_2^{(N_2-1)} \\ q_1^{(N_1)} & q_2^{(N_2)} \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} w_1^{(1)} w_2^{(1)} \\ w_1^{(1)} w_2^{(2)} \\ \vdots \\ w_1^{(1)} w_2^{(N_2)} \\ w_1^{(2)} w_2^{(1)} \\ \vdots \\ w_1^{(N_1)} w_2^{(N_2-1)} \\ w_1^{(N_1)} w_2^{(N_2)} \end{pmatrix}, \tag{3.22}$$

assuming each ensemble $\tilde{q}_i$ has N_i sigma points.

From now in, when this report mentions that two ensembles are combined, this is what is meant. Note that there is no assumption about the dimension of the ensembles $\tilde{q}_1$ and $\tilde{q}_2$ made. In other words the sigma-points $q_k^{(i)}$ could have several components, the expression for their combined ensemble would be the same. Hence, this method can be used to combine and create ensembles for any number of parameters.

This new ensemble will, according to Theorem A.1 fulfill the individual moments of the distributions and according to Theorem A.2 have the mixed moments representing independent parameters, both of which are presented and proven in Appendix A.1. It is also easy to verify that the weights will sum up to zero.

An example of two one-dimensional ensembles being combined into a two-dimensional ensemble is visualized in Figure 3.3

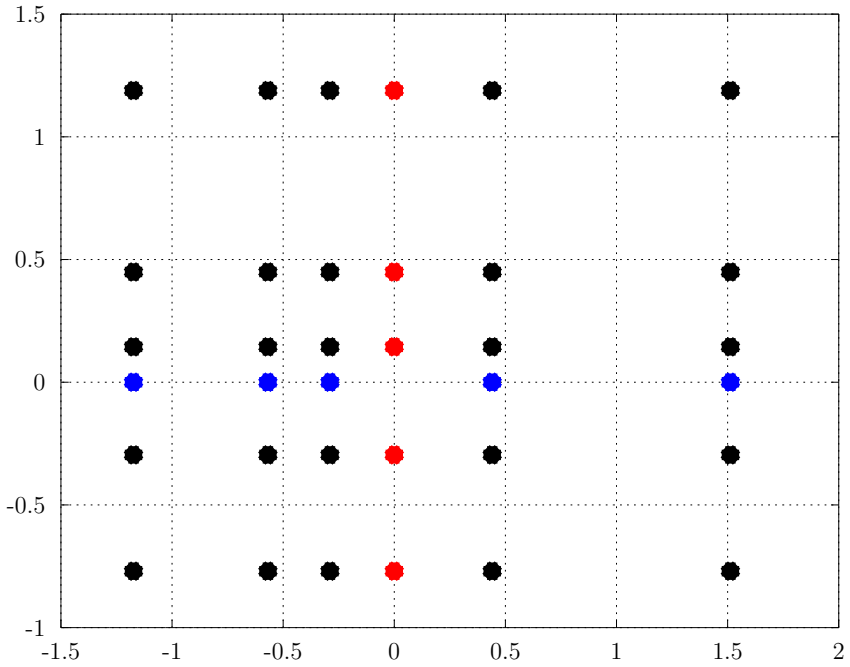


Figure 3.3: The initial ensembles for two parameters in blue and red. Their combined ensemble in black.

Hence, the problem is solved, in principle. One problem with this solution, though, is that the number of sigma-points is now $N_1 N_2$, i.e. as several ensembles are combined the ensemble's size grows exponentially. This growth is a problem since one of the aims of Deterministic Sampling is to keep the number of samples needed low.
The number of sigma points can be significantly reduced, though, by applying Simplex Reduction to the problem, as described in section 3.2, and it turns out many of the points can be removed if one alters the weights.
The final result in the example with two dimensions is shown in Figure 3.4.

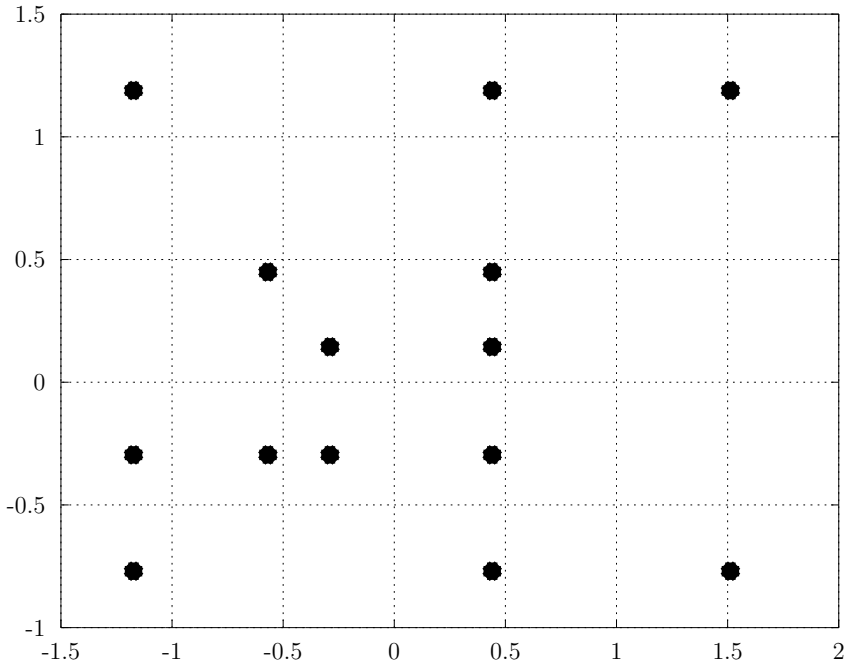


Figure 3.4: The combined ensemble for the two-dimensional example-ensemble after excessive weights have been removed with Simplex Reduction.

This method can in principle be used for any number independent parameters, although creating them can become a practical problem due to finite computer memory.
In this report an ensemble built like this will be called a combined ensemble and when it is mentioned that ensembles have been combined, this is what is meant.

3.6. Covariant ensembles

By now we have working methods for creating ensembles for any distribution and, in principle, an arbitrary number of parameters. What is yet to explore though is how to encode covariance into an ensemble.
What would an ensemble with covariance look like? If we use Random Sampling and sample from two covariant variables the ensemble will look slightly ”squished”, or ”skewed” in some direction. This difference is shown in Figure 3.5.

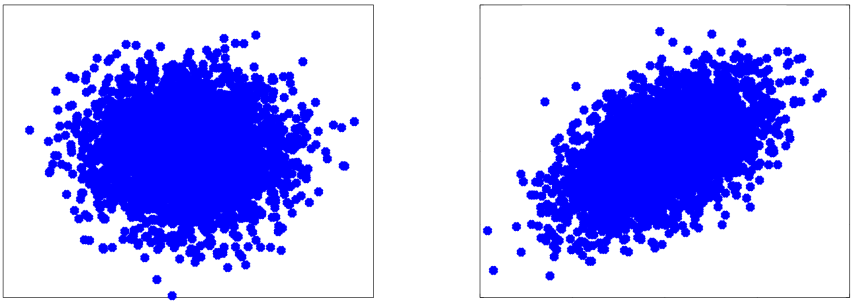


Figure 3.5: Left: RS ensemble without covariance. Right: RS Ensemble with positive covariance.

This section's approach for encoding covariance into a deterministic ensemble works by replicating the skewing shown here. The question is of course how to skew the ensemble in the correct way. The method described here uses the covariance matrix to construct a new basis for the sigma-points, and then moves all the points over to the new basis. This keeps the mean value and variance as well as encodes the covariance. It might alter the higher order moments though, this can be reclaimed by recalculating the weights with Simplex Reduction as long as the ensemble has not been reduced beforehand. This Skewing Method will be described in Section 3.6.1. Another approach which works for Gaussian ensembles, and symmetric distributions in general, is presented in Section 3.6.2. It works not by altering the covariance by moving the sigma-points around, but by placing the sigma-points in a favourable place and from there any covariance can be encoded simply by changing the weights.

3.6.1. Covariant ensembles with the Skewing-method

When creating an ensemble for general distributions, with covariance encoded, one working method is to use a variant of the method for combining ensembles described in section 3.5. When creating a covariant ensemble, the individual ensembles are here combined in a slightly different way which skews the ensemble to encode covariance. After all parameters are combined Simplex Reduction is used to decrease the number of parameters and make sure all marginal moments are correct.
Skewing the ensemble will be accomplished in a similar way to how the unscented Kalman filter[1] encodes covariance in an ensemble, as shown in equation (2.16), with the difference that this can encode higher marginal moments and works for a distribution of any shape.
Assume n parameters $\{q_1, q_2 \dots q_n\}$, with known distributions and a known interdependence described by a covariance matrix C. This will be shown in figures for two parameters, but the reasoning is done for a general n.
The covariance Matrix $C_{n\times n}$ is used to create a new set of vectors $\{\hat{b}_1, \hat{b}_2 \dots \hat{b}_n\}$

which can work as a new basis when skewing the ensemble. This is done by finding a matrix $\Delta_{n\times n}$ s.t.

$$\Delta\Delta^{\mathsf{T}} = C. \tag{3.23}$$

The matrix square root Δ can be found in several ways, for example by diagonalizing $C = SDS^{\mathsf{T}}$ and then taking $\Delta = S\sqrt{D}S^{\mathsf{T}}$.
A new set of basis vectors $\{\hat{b}_1, \hat{b}_2 \dots \hat{b}_n\}$ is built up from

$$\hat{b}_i = \frac{\Delta(i,:)}{||\Delta(i,:)||}, \tag{3.24}$$

where $\Delta(i,:)$ refers to the ith row of Δ, meaning each row of Δ normalized becomes a new basis vector.
Now assuming every parameter q_i has an individual ensemble of N_i sigma-points

$$\tilde{q}_i = \{q_i^{(1)} q_i^{(2)} \dots q_i^{(N_i)}\},$$

a new grid can be constructed in a similar way to how it is done in section 3.5. Every sample of every distribution is combined in every possible combination, but this time the sigma points are moved out into the new base, encoding the covariance, meaning the new sigma points will be

$$q^{(k)} = q_1^{(i_1)}\hat{b}_1 + q_2^{(i_2)}\hat{b}_2 + \ldots + q_n^{(i_n)}\hat{b}_n \tag{3.25a}$$

with every possible combination of $i_1, i_2, \dots, i_n$. The weights are multiplied together like before

$$w^{(k)} = w_1^{(i_1)} w_2^{(i_2)} \dots w_n^{(i_n)}. \tag{3.25b}$$

In matrix form the ensemble would be

$$\tilde{q} = \begin{pmatrix} q_1^{(1)}\hat{b}_1 + q_2^{(1)}\hat{b}_2 + \ldots + q_n^{(1)}\hat{b}_n \\ q_1^{(1)}\hat{b}_1 + q_2^{(1)}\hat{b}_2 + \ldots + q_n^{(2)}\hat{b}_n \\ q_1^{(1)}\hat{b}_1 + q_2^{(1)}\hat{b}_2 + \ldots + q_n^{(3)}\hat{b}_n \\ \vdots \\ q_1^{(N_1)}\hat{b}_1 + q_2^{(N_2)}\hat{b}_2 + \ldots + q_n^{(N_n-1)}\hat{b}_n \\ q_1^{(N_1)}\hat{b}_1 + q_2^{(N_2)}\hat{b}_2 + \ldots + q_n^{(N_n)}\hat{b}_n \end{pmatrix} \tag{3.26a}$$

with the corresponding weights

$$\tilde{w} = \begin{pmatrix} w_1^{(1)} w_2^{(1)} \dots w_n^{(1)} \\ w_1^{(1)} w_2^{(1)} \dots w_n^{(2)} \\ \vdots \\ w_1^{(N_1)} w_2^{(N_2)} \dots w_n^{(N_n-1)} \\ w_1^{(N_1)} w_2^{(N_2)} \dots w_n^{(N_n)} \end{pmatrix}. \tag{3.26b}$$

Another way of viewing this would be to see it as a transformation of an ensemble without covariance to one with covariance using the matrix multiplication

$$\tilde{q}_{\text{cov}} = \tilde{q}_{\text{indep}} \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{pmatrix}. \tag{3.26c}$$

If there is no covariance this would result in the same combination as in section 3.5, but if there is interdependence, this will skew the ensemble making it include the covariance wanted. This difference is shown in Figure 3.6.

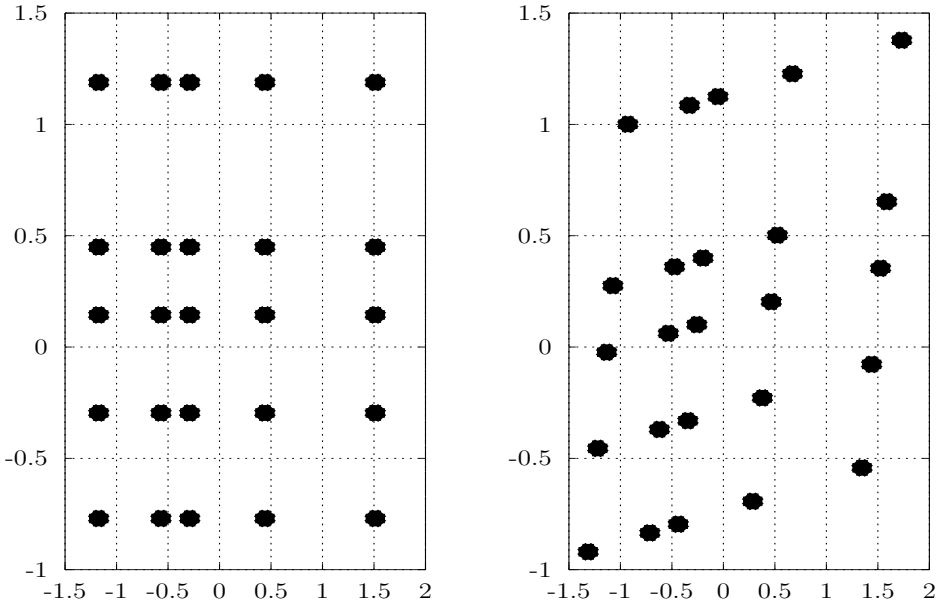


Figure 3.6: The ensembles of two parameters combined without any covariance (left) and skewed to encode a positive covariance (right).

This will create an ensemble of size $N_1 N_2 \ldots N_n$, which will fulfill the covariance. Two problems now arise, however. Firstly, higher moments than the second will change, and secondly, the ensemble size grows exponentially with the number of parameters. Both of these problems are fixed with Simplex Reduction, which can be used to both set weights to that the higher moments are corrected, and the ensemble size can be reduced drastically. The covariant ensemble from Figure 3.6 is shown in Figure 3.3 after Simplex Reduction.

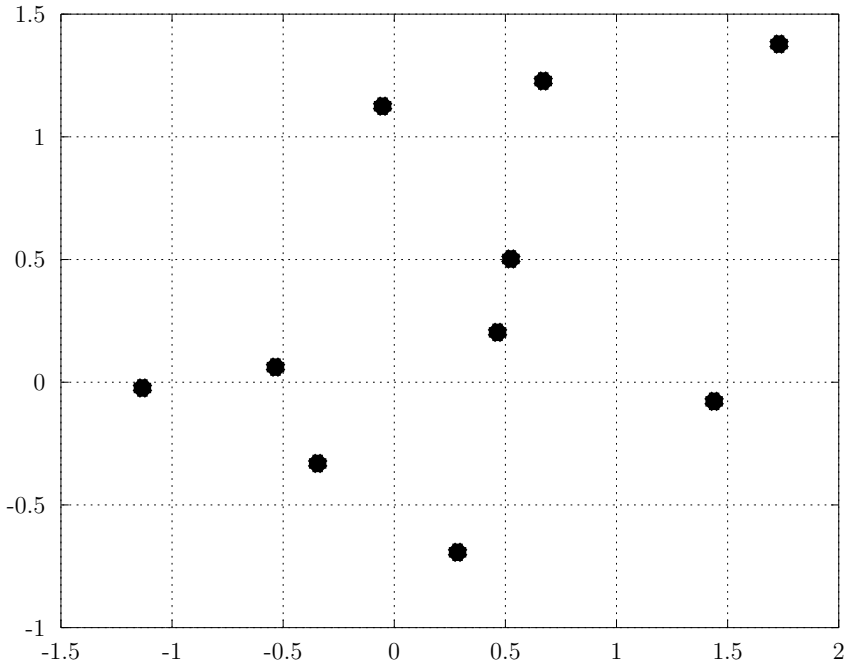


Figure 3.7: The covariant ensemble after Simplex Reduction.

This method will here be called the Skewing-method.

3.6.2. Covariant Gaussian ensembles with the Corner-method

Encoding covariance into an ensemble where all of the parameters have a Gaussian distribution turns out to also be a particular case and because of this, it will also be given its own section. Such ensembles can also be calculated with the technique described in section 3.6.1. Here a shortcut is described for when all parameters have a Gaussian distribution, for an ensemble with four moments encoded.
Assume n parameters $\{q_1, q_2 \dots q_n\}$, with known distributions and a known interdependence described by a covariance matrix C. Without loss of generality, it is once again assumed that each parameter has expectation value zero and variance one.
It is already known that for one parameter a working ensemble is $\tilde{q} = \sqrt{3}\begin{pmatrix}1 & 0 & -1\end{pmatrix}^{\mathsf{T}}$. One way of creating a working ensemble for any number of parameter with Gaussian distribution is to put one sample in each corner of the room $[-\sqrt{3}, \sqrt{3}]^n$ and one sample in the center, and then calculate the weights with Simpelx Reduction. In two dimensions this ensemble would look like in Figure 3.8.

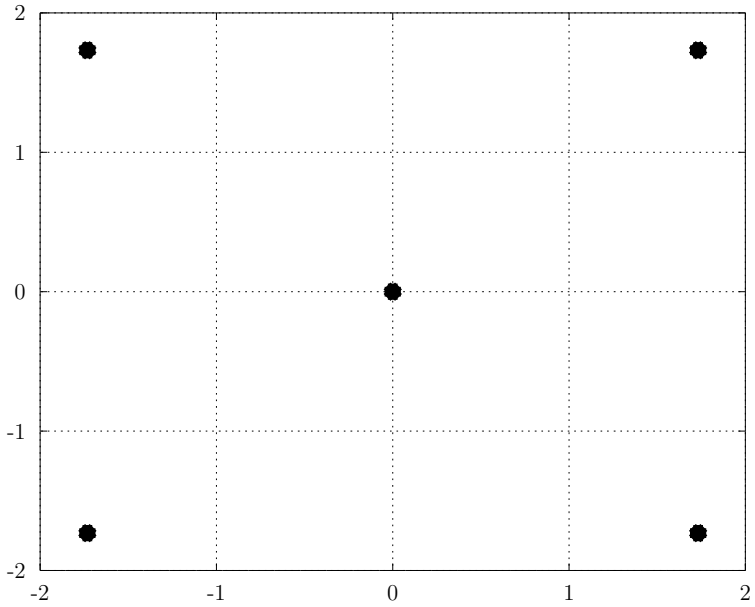


Figure 3.8: The 2D variant of the ensemble which can be weighted to include any covariance.

This ensemble can be adjusted to encode any covariance by changing the weights of the samples. An example of this is shown in Figure 3.9.

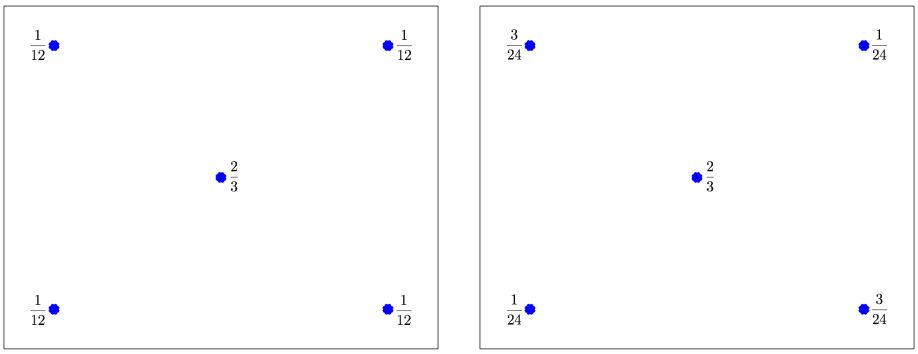


Figure 3.9: Left: Weights set to encode no covariance. Right: Weights set to encode a covariance of -0.5. Both weights encode four marginal moments.

Adjusting the weights is easiest done by Simplex Reduction, which will enforce the covariance wanted and also remove points which are not needed. Removing excessive points becomes important in higher dimensions since every the ensemble grows in size very fast with higher dimensions otherwise. In general such an ensemble can be built with the Corners Matrix defined in Appendix D.1. The ensemble becomes the Corners Matrix with a zero row added, i.e.

$$\tilde{q}_n = \sqrt{3}\begin{pmatrix} C_n \\ 0_{1\times n} \end{pmatrix}, \tag{3.27}$$

with corresponding weights calculated by Simplex Reduction.
This method will here be called the Corner-method.

3.7. An ensemble for Symmetric Distributions in general

The Shotgun algorithm, as described in Section 3.4 can create ensembles for any one-dimensional distribution and by combining such ensembles, as outlined in section 3.5, one can create a multi-dimensional ensemble for any distribution. Still the special cases are relevant to study, such as ensembles for the Gaussian distribution, as described in Section 3.3. This ensemble is much smaller than the ensembles created by combining ensembles. One reason for this is that the Gaussian ensemble is symmetric. It turns out that symmetric ensembles, in general, is a particular case and even though the Gaussian distribution is the most common one, for completeness sake the general case of creating ensembles for symmetric distributions is examined here.
Creating an ensemble for a symmetrically distributed parameter resembles the reasoning done for the Gaussian parameter in Section 3.4, but it will not lead to the same final ensemble. In fact, the Block-Diagonal Gaussian Ensemble will not work in the general case, but this reasoning will lead to an ensemble which does.
Assume a random parameter q with some symmetric distribution. When creating an ensemble with only two moments encoded, i.e. only mean and variance, the ensemble takes the same form for any distribution, so let's look at encoding four moments.

3.7.1. 1D symmetric ensemble with four moments

Let's look for an ensemble with three sigma-points, encoding four moments for a symmetric distribution. Without loss of generality it will be assumed to have mean value 0 and let's let one of the sigma-points be located there, i.e. let $q^{(0)} = 0$. Due to it being symmetric we can now have both of the remaining two sigma-points be mirrored around this point, i.e. the other points will be $q^{(+)} = -q^{(-)} = q^{(\pm)}$. In other words, we look for an ensemble on the form

$$\tilde{q} = \begin{pmatrix} -q^{(\pm)} \\ 0 \\ q^{(\pm)} \end{pmatrix} \quad \tilde{w} = \begin{pmatrix} w^{(\pm)} \\ w^{(0)} \\ w^{(\pm)} \end{pmatrix},$$

which encodes four moments need to fulfil the equations

$$
\begin{aligned}
w^{(\pm)} + w^{(0)} + w^{(\pm)} &= 1 \\
-w^{(\pm)} q^{(\pm)} + 0 + w^{(\pm)} q^{(\pm)} &= 0 \\
w^{(\pm)} \left(-q^{(\pm)}\right)^2 + 0 + w^{(\pm)} \left(q^{(\pm)}\right)^2 &= \left\langle \delta^2 q \right\rangle \\
w^{(\pm)} \left(-q^{(\pm)}\right)^3 + 0 + w^{(\pm)} \left(q^{(\pm)}\right)^3 &= 0 \\
w^{(\pm)} \left(-q^{(\pm)}\right)^4 + 0 + w^{(\pm)} \left(q^{(\pm)}\right)^4 &= \left\langle \delta^4 q \right\rangle
\end{aligned}
\tag{3.28a}
$$

with the third moment being set to zero due to the distribution being symmetric. The second and fourth equations are fulfilled automatically and the three equations that actually need to be solved to get the three parameters $q^{(\pm)}$, $w^{(\pm)}$ and $w^{(0)}$ are

$$
\begin{aligned}
2w^{(\pm)} + w^{(0)} &= 1 \\
2w^{(\pm)} \left(q^{(\pm)}\right)^2 &= \left\langle \delta^2 q \right\rangle \\
2w^{(\pm)} \left(q^{(\pm)}\right)^4 &= \left\langle \delta^4 q \right\rangle .
\end{aligned}
\tag{3.28b}
$$

which is solved by

$$
\begin{aligned}
w^{(\pm)} &= \frac{\left\langle \delta^2 q \right\rangle^2}{2\left\langle \delta^4 q \right\rangle} \\
w^{(0)} &= 1 - \frac{\left\langle \delta^2 q \right\rangle^2}{\left\langle \delta^4 q \right\rangle} \\
q^{(\pm)} &= \sqrt{\frac{\left\langle \delta^4 q \right\rangle}{\left\langle \delta^2 q \right\rangle}}
\end{aligned}
$$

and the ensemble for a one dimensional symmetric distribution with four moments is

$$
\tilde{q} = \sqrt{\frac{\left\langle \delta^4 q \right\rangle}{\left\langle \delta^2 q \right\rangle}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} \frac{\left\langle \delta^2 q \right\rangle^2}{2\left\langle \delta^4 q \right\rangle} \\ \frac{\left\langle \delta^2 q \right\rangle^2}{2\left\langle \delta^4 q \right\rangle} \\ 1 - \frac{\left\langle \delta^2 q \right\rangle^2}{\left\langle \delta^4 q \right\rangle} \end{pmatrix},
\tag{3.28c}
$$

which is easily verified to, in the Gaussian case, be the same ensemble as (3.13).

3.8. Symmetric distributions in higher dimensions

When the Block-Diagonal Gaussian Ensemble was derived in Section 3.3 the one-dimensional ensemble was extended by ”splitting” the point in the

middle and extend it into higher dimensions and after this was no longer possible start "splitting" the edge-points instead.
The same line of thinking could work here, but there is not a reliable way of splitting the midpoint since there is no guarantee that the weight in the middle is large enough. In more precise terms, dividing the midpoint would mean one were to create the new two-dimensional ensemble as

$$\tilde{q}_2 = \sqrt{\frac{\langle \delta^4 q \rangle}{\langle \delta^2 q \rangle}} \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} \frac{\langle \delta^2 q \rangle^2}{2\langle \delta^4 q \rangle} \\ \frac{\langle \delta^2 q \rangle^2}{2\langle \delta^4 q \rangle} \\ \frac{\langle \delta^2 q \rangle^2}{2\langle \delta^4 q \rangle} \\ \frac{\langle \delta^2 q \rangle^2}{2\langle \delta^4 q \rangle} \\ 1 - 2\frac{\langle \delta^2 q \rangle^2}{\langle \delta^4 q \rangle} \end{pmatrix},$$

but there is no guarantee that $1 - 2\frac{\langle \delta^2 q \rangle^2}{\langle \delta^4 q \rangle}$ will always be greater than or equal to zero. In fact, it is easy to find an example when this is not the case. For example if a parameter q is uniformly distributed in $[-1,1]$ it will have $\langle \delta^2 q \rangle = \frac{1}{3}$ and $\langle \delta^4 q \rangle = \frac{1}{5}$. The mid point weight would be, with this split,

$$1 - 2\frac{\langle \delta^2 q \rangle^2}{\langle \delta^4 q \rangle} = -\frac{1}{9}$$

which is prohibited and thus splitting the mid point will not work in the general case.
This is why, for a general symmetric distribution the ensemble will be created by keeping the mid-point's weight constant and splitting the other points. Once again one wants to maintain the covariance to zero, which means each row must be orthogonal to every other. In two dimensions this is accomplished by letting the two-dimensional ensemble $\tilde{q}_2$ be

$$\tilde{q}_2 = \sqrt{\frac{\langle \delta^4 q \rangle}{\langle \delta^2 q \rangle}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \\ 0 & 0 \end{pmatrix}, \quad \tilde{w}_2 = \begin{pmatrix} \frac{\langle \delta^2 q \rangle^2}{4\langle \delta^4 q \rangle} \\ \frac{\langle \delta^2 q \rangle^2}{4\langle \delta^4 q \rangle} \\ \frac{\langle \delta^2 q \rangle^2}{4\langle \delta^4 q \rangle} \\ \frac{\langle \delta^2 q \rangle^2}{4\langle \delta^4 q \rangle} \\ 1 - \frac{\langle \delta^2 q \rangle^2}{\langle \delta^4 q \rangle} \end{pmatrix}.$$

This ensemble has kept the centre point but doubled the number of points not in the centre. Their weights have therefore been halved. In general, this can be done for any number of dimensions, just keep the zero point the same, add orthogonal columns of -1 and 1 for each parameter and multiply

by $\sqrt{\langle \delta^4 q \rangle / \langle \delta^2 q \rangle}$. The weights of the non-center points are set equal and to have the sum $\langle \delta^2 q \rangle^2 / \langle \delta^4 q \rangle$.
This ensemble is the final ensemble presented and can be used for any symmetric distribution. Due to it always having the same, often larger, weight in the mid-point while the surrounding weights decrease with increasing dimension this ensemble is here called the Heavy Middle Ensemble. Its general expression is presented in Appendix C.1.1.

3.9. Evaluating the methods

The methods for finding an ensemble for Deterministic Sampling described in this chapter have been tested in various cases. They have been evaluated by two criteria, the first being the correctness of the propagated uncertainty. This study looks no further than propagated mean value and variance since those are usually most relevant when looking at a result. The second criteria considered is ensemble size, since keeping the number of samples low is one of the aims of deterministic sampling.
Deterministic Sampling has been tested in both one or more dimensions and with different kinds of distributions. Uncertainty propagation with one parameter has been done with Gaussian distribution with both a broad distribution with $\sigma = 1$ and a more narrow distribution with $\sigma = 0.2$. Similar tests with three parameters have been done with a combination of Weibull and Gaussian parameters without any covariance and three Gaussian parameters with some dependence involved. The three parameter tests have been made with parameters who's distributions have a standard deviation around 0.1.
The classes of functions used to benchmark uncertainty propagation with DS are polynomials, which have a limited number of non-zero derivatives, exponential functions, which have an unlimited number of non-zero derivatives and trigonometric functions, which are periodic. Discontinuous functions have also been tested. Here the Heaviside function

$$\mathrm{H}(x) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

has been used to create a discontinuity in the tested function and the discontinuity is set to occur close to the ensemble's mean value.
For comparison Random Sampling is used.

4. Tests and Results

4.1. Propagation

This section intends to show an overview of the results from performing uncertainty propagation with Deterministic Sampling. The results are presented graphically, and the figures aim to show a pattern rather than give the exact results. A detailed presentation of the results is available in Appendix F.
The propagated uncertainty is presented as error bars with the marked midpoint representing the mean value, and the error bar reaches one standard deviation away. An illustrative example is shown here.
Assume uncertainty propagation has been performed with DS using three ensembles encoding 2, 4 and 6 moments respectively and that RS has been carried out as a reference. The example's propagated mean and standard deviation is shown in Table 4.1.

Table 4.1: An example of the data generated from testing uncertainty propagation with DS and RS.

	Mean	Std
DS 2 moments	0.56	0.35
DS 4 moments	0.67	0.30
DS 6 moments	0.66	0.31
RS	0.66	0.32

This such data will here be presented as in Figure 4.1.

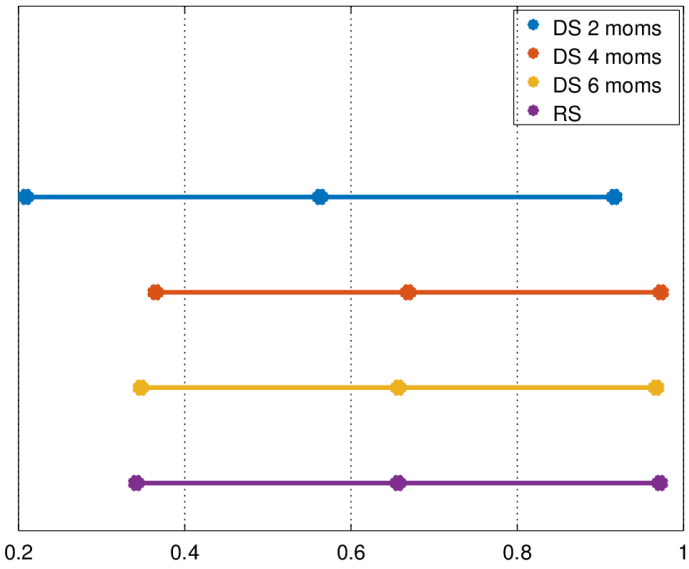


Figure 4.1: The example data from Table 4.1 presented as error bars spanning one standard deviation away from the mean value. The bottom error bar, representing the data gained from RS, is interpreted as the correct uncertainty.

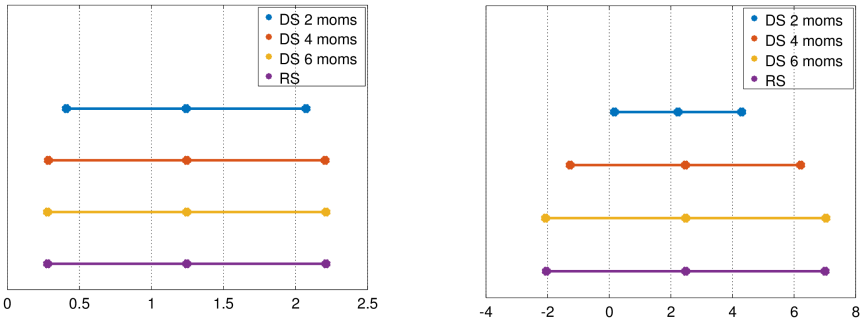
The error bar gained from RS are assume to be the correct uncertainty and hence how much the other error bars agree with the bottom one measures how accurate they are.
In this section, the results of the uncertainty propagation are only presented as error bars. Tables with the data are, as stated, available in Appendix F

4.1.1. One dimensional ensembles

Propagation of uncertainty with Deterministic Sampling has here been tested through functions of one parameter with a Gaussian distribution. Several functions have been used for testing.
The calculation has been done with Deterministic Sampling (DS) with 2, 4 and 6 encoded moments. A Random Sampling (RS) simulation has been done as a reference.
The tests have been done in two cases, one where the parameter had a relatively thin Gaussian distribution ($\sigma = 0.2$) and one where it had a wide Gaussian distribution ($\sigma = 1$). Here the results are shown as figures while Appendix F.1 shown the ensembles used and the tables of the data.

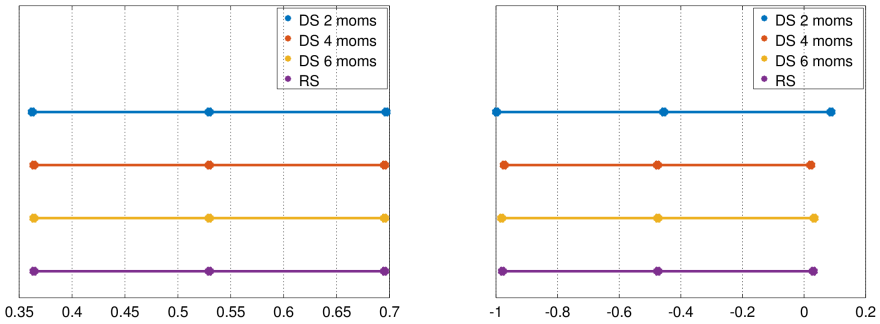
Narrow Gaussian distribution, $\sigma = 0.2$

In the case of a narrow distribution DS has shown to, in most cases, produce a good value with two or four moments. The propagated uncertainties are shown in Figures 4.2 to 4.5.



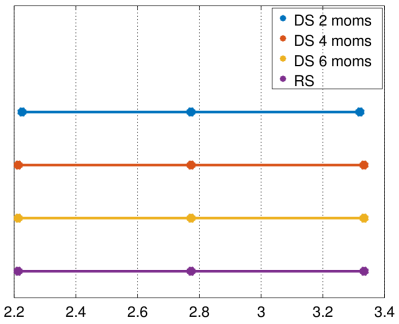
(a) $f(q) = q^4$, $\sigma_q = 0.2$ (b) $f(q) = q^8$, $\sigma_q = 0.2$

Figure 4.2: Uncertainties propagated through polynomials.

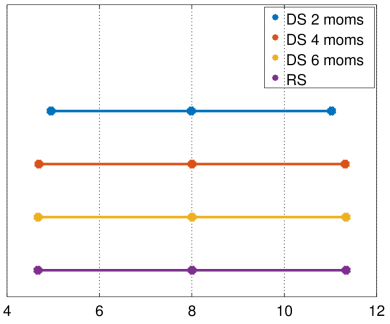


(a) $f(q) = \cos(q)$, $\sigma_q = 0.2$ (b) $f(q) = \cos(4q)$, $\sigma_q = 0.2$

Figure 4.3: Uncertainties propagated through periodic functions.

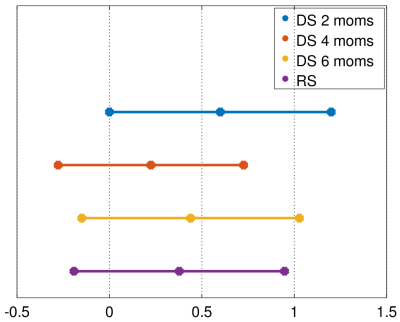


(a) $f(q) = e^q$, $\sigma_q = 0.2$



(b) $f(q) = e^{2q}$, $\sigma_q = 0.2$

Figure 4.4: Uncertainties propagated through exponential functions.

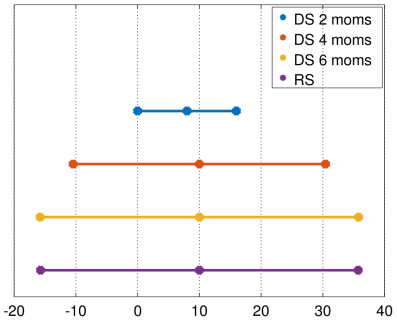


$f(q) = q\mathrm{H}(q-1.1)$, $\sigma_q = 0.2$

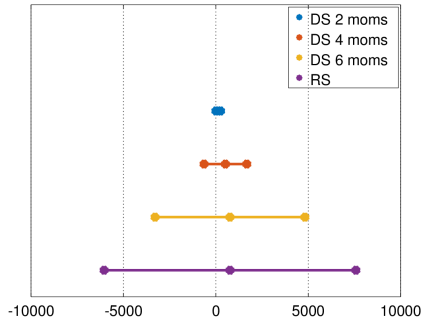
Figure 4.5: Uncertainties propagated through a discontinuous function.

Wide Gaussian distribution, $\sigma = 1$

In the case of a wide distribution DS has shown to have some more problems, depending on what the function looks like. The propagated uncertainties are shown in Figures 4.6 to 4.9.

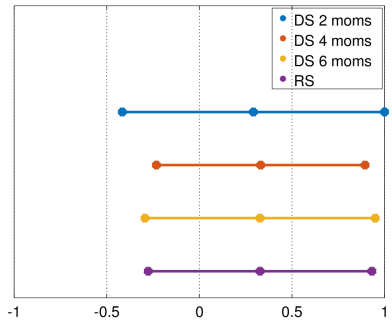


(a) $f(q) = q^4$, $\sigma_q = 1$

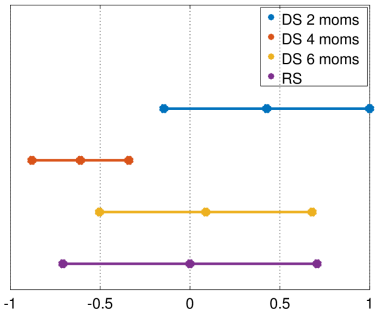


(b) $f(q) = q^8$, $\sigma_q = 1$

Figure 4.6: Uncertainties propagated through polynomials.

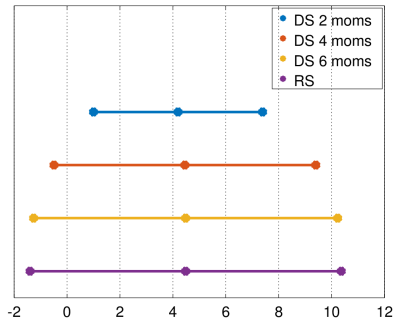


(a) $f(q) = \cos(q)$, $\sigma_q = 1$

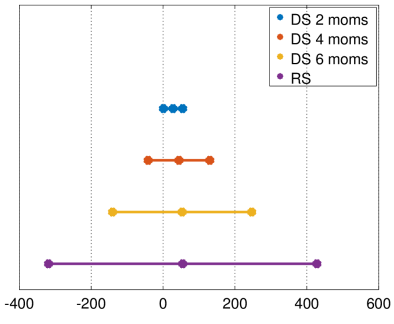


(b) $f(q) = \cos(4q)$, $\sigma_q = 1$

Figure 4.7: Uncertainties propagated through periodic functions.

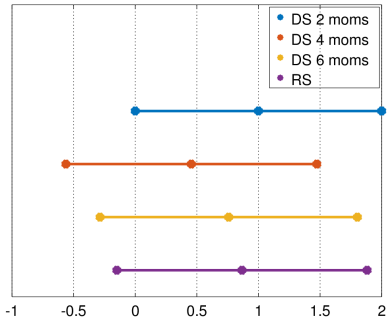


(a) $f(q) = e^q$, $\sigma_q = 1$



(b) $f(q) = e^{2q}$, $\sigma_q = 1$

Figure 4.8: Uncertainties propagated through exponential functions.

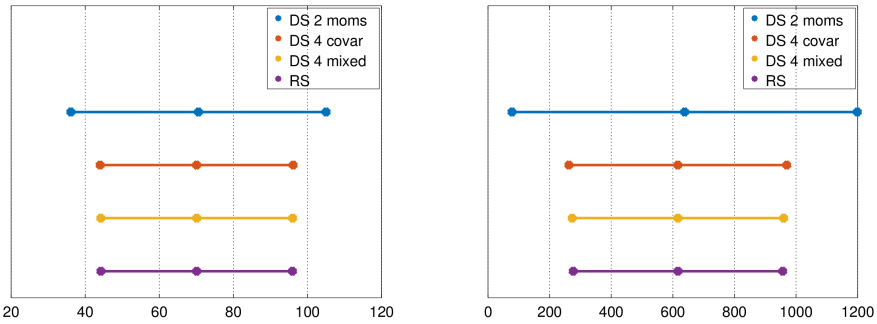


$$f(q) = q\mathrm{H}(q-1.1),\ \sigma_q = 1$$

Figure 4.9: Uncertainties propagated through a discontinuous function.

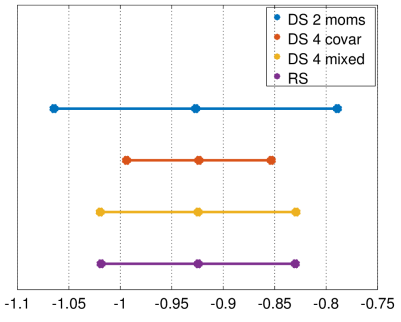
4.1.2. Three-dimensional independent ensembles

Here propagation of uncertainty has been tested with functions of three parameters. The parameters chosen here are of one Gaussian parameter q_1 with $\mu = 1$ and $\sigma = 0.1$ and two Weibull-distributed parameters, q_2 and q_3, with $\alpha = 6$ and $\beta = 1$, giving them a mean value of ≈ 0.93 and a standard deviation of ≈ 0.18.
The testing has been done by Deterministic Sampling with 2 and 4 moments encoded. In the case of 4 moments, the test has been performed in one case with only the covariance set to zero while the higher order mixed moments left unchecked and in one instance with up to the first four mixed moments set to represent independence. RS with 10^7 samples is used as a reference. Once again, the propagated uncertainty is shown as error bars where the midpoint is the gained mean value, and the edges are one standard deviation away. The results are shown in Figures 4.10 to 4.12. A table with the data is avalible in Appendix F.2.

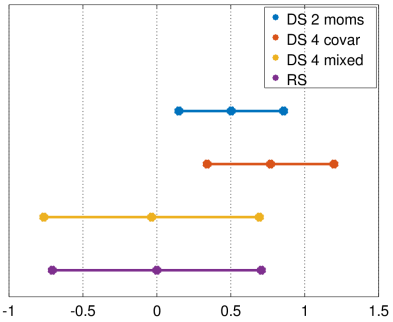


(a) $(q_1+q_2+q_3)^4$ (b) $(q_1+q_2+q_3)^6$

Figure 4.10: Uncertainties propagated through polynomials.



(a) $\cos(q_1+q_2+q_3)$



(b) $\cos\left((q_1+q_2+q_3)^4\right)$

Figure 4.11: Uncertainties propagated through trigonometric functions.

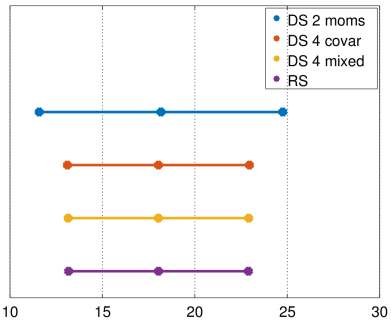


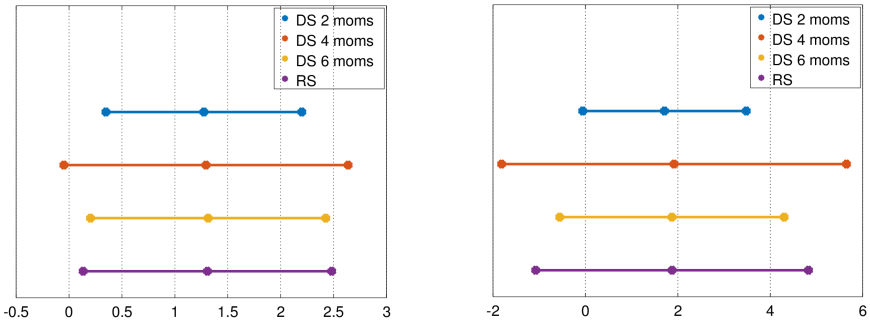
Figure 4.12: Uncertainties propagated through an exponential function.

4.1.3. Three-dimensional ensembles with dependence

Tests have also been done for three parameters with dependence. Here three Gaussian parameters, all with $\sigma = 0.1$ and a covariance matrix

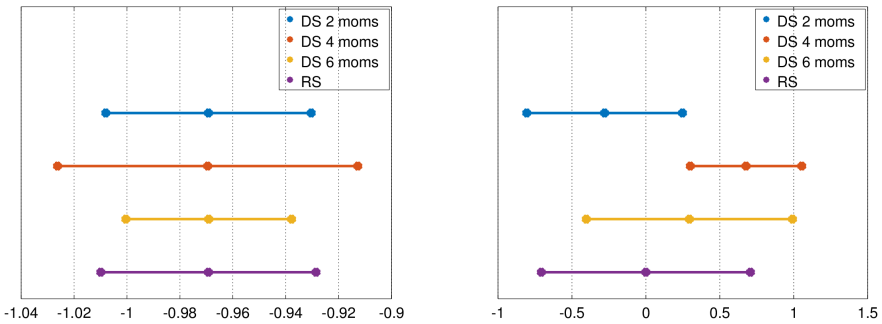
$$C = \sigma^2 \begin{pmatrix} 1 & 0.02 & 0 \\ 0.02 & 1 & 0.6 \\ 0 & 0.6 & 1 \end{pmatrix}$$

have been propagated through several functions with three different ensembles. The ensembles have encoded 2, 4 and 6 marginal moments respectively. They all have the correct covariance encoded but no higher order mixed moments are set.
The results are shown in Figures 4.13 to 4.15. A table with the data is avalible in Appendix F.3.



(a) $(q_1+q_2+q_3)^4$ (b) $(q_1+q_2+q_3)^6$

Figure 4.13: Uncertainties propagated through polynomials.



(a) $\cos(q_1+q_2+q_3)$ (b) $\cos\left((q_1+q_2+q_3)^4\right)$

Figure 4.14: Uncertainties propagated through trigonometric functions.

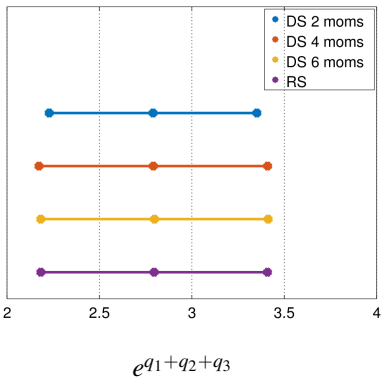


Figure 4.15: Uncertainties propagated through an exponential function.

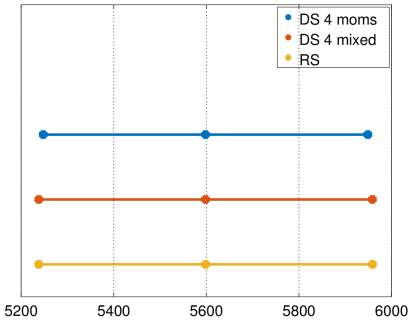
4.1.4. A semi-real-world example

When modeling turbulence in computational fluid dynamics the so-called κ-ε-model[10] is commonly used. It contains five independent uncertain parameters C_μ, C_ε, σ_κ, κ and B. This model was used by Dunn[11] to study uncertainty quantification by Latin Hypercube Sampling and by Hedberg and Hessling[2] to study Deterministic Sampling. Here, the same model will be used with the same parameters as Hedberg and Hessling uses, which are presented in Table 4.2.

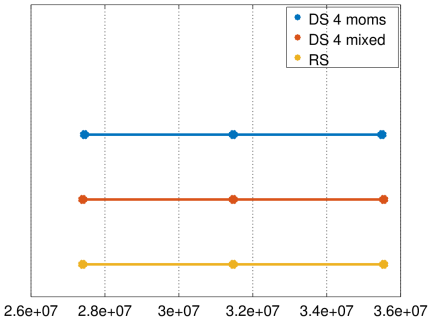
Table 4.2: The uncertain parameters in the κ-ε-model of turbulence.

Parameter	Probability distribution func. $f(x)$		Constants
C_μ	Weibull	$\frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-(x/\beta)^\alpha}$	$\alpha = 45.54$ $\beta = 8.77 \cdot 10^{-2}$
$C_{\varepsilon 2}$	Beta	$\frac{(x-A_1)^{p-1}(A_2-x)^{q-1}}{\int_{A_1}^{A_2}(x-A_1)^{p-1}(A_2-x)^{q-1}\mathrm{d}x}$	$A_1 = 1.8,\ A_2 = 2.2$ $p = 1.2,\ q = 2.0$
σ_κ	Gaussian	$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)}{2\sigma^2}}$	$\mu = 1.0$ $\sigma = 1.67 \cdot 10^{-2}$
κ	Gaussian	$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)}{2\sigma^2}}$	$\mu = 0.41$ $\sigma = 4.89 \cdot 10^{-3}$
B	Gaussian	$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)}{2\sigma^2}}$	$\mu = 5.2$ $\sigma = 0.10$

Two ensembles representing these distributions have been built. Both encode four moments and sets covariance to zero. One of them encodes the higher order mixed moments, though while the other leaves the mixed moments beyond covariance unenforced.
The ensemble was built by first creating a three-dimensional independent Gaussian ensemble for σ_κ, κ and B from the Block-Diagonal Gaussian ensemble. A two-dimensional ensemble for C_μ and $C_{\varepsilon 2}$ has then been created by generating the individual ensembles with the Shotgun Algorithm, combining them and then reducing the size with Simplex Reduction. The 3D and 2D ensembles have then been combined into a full 5D ensemble for the entire model and once again reduced in size again with Simplex Reduction. In one of the cases, only the covariance was forced in the Simplex Reduction, while in the other case up to four mixed moments was forced. The ensemble with only the covariance forced has 20 sigma-points and the ensemble with up to the fourth mixed moment has 54 points.
These ensembles have not been tested on actual CFD-simulations, but evaluated by test-functions. The results are shown in Figures 4.16 to 4.18. The data is shown in more detail in Appendix F.4.

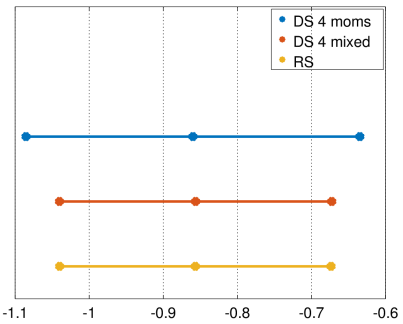


(a) $(\sum q_i)^4$

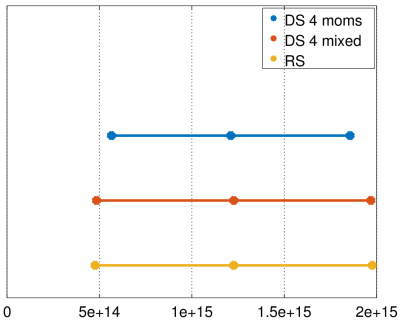


(b) $(\sum q_i)^8$

Figure 4.16: Uncertainties propagated through polynomials.

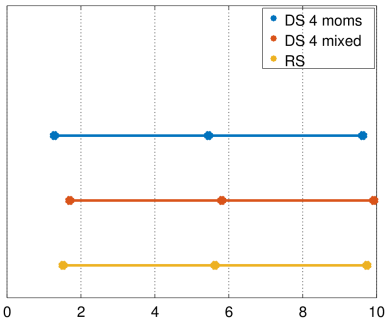


(a) $\cos((4\sum q_i))$



(b) $e^{(\sum q_i)}$

Figure 4.17: Uncertainties propagated through a trigonometric function and an exponential function.



$H(-8.6+\sum q_i)\sum q_i$

Figure 4.18: Uncertainties propagated through a discontinuous function.

4.2. Ensemble size

The size of the ensembles has been tested by constructing ensembles for different numbers of parameters with the methods described. The number of sigma-points has been saved the ensemble sizes gained by the different methods have been compared and graphed.

4.2.1. Block-Diagonal Gaussian Ensemble size

The size of Gaussian ensembles which have been created with the method described in section 3.3 are graphed in Figure 4.19. Along with it is the size of ensembles created with the same method, but reduced in size by Simplex Reduction.

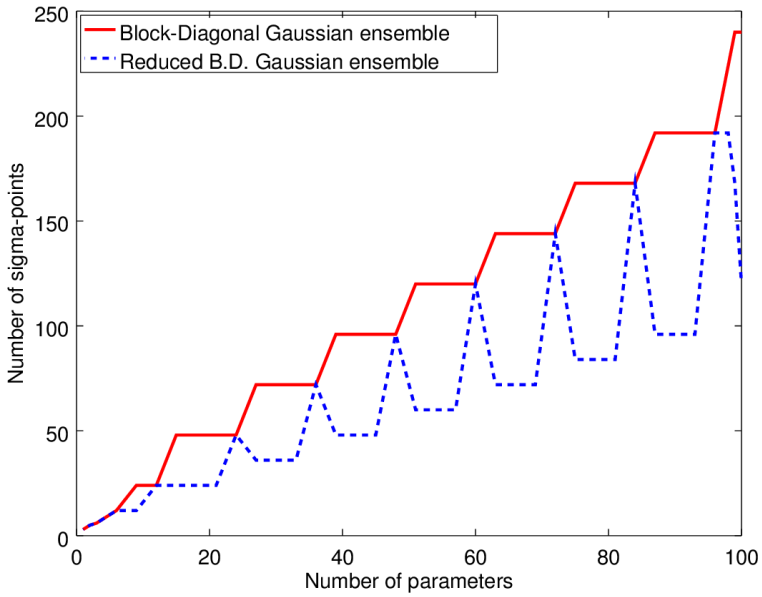


Figure 4.19: The size of the Block-Diagonal Gaussian ensembles and the size of the same ensembles after decreasing its size with Simplex Reduction.

4.3. Heavy Middle Ensemble size

The size of the Heavy Middle Ensemble has been calculated with for up to a hundred parameters and is size graphed in Figure 4.20. The size of the ensemble after it has been reduced by Simplex Reduction is also shown.

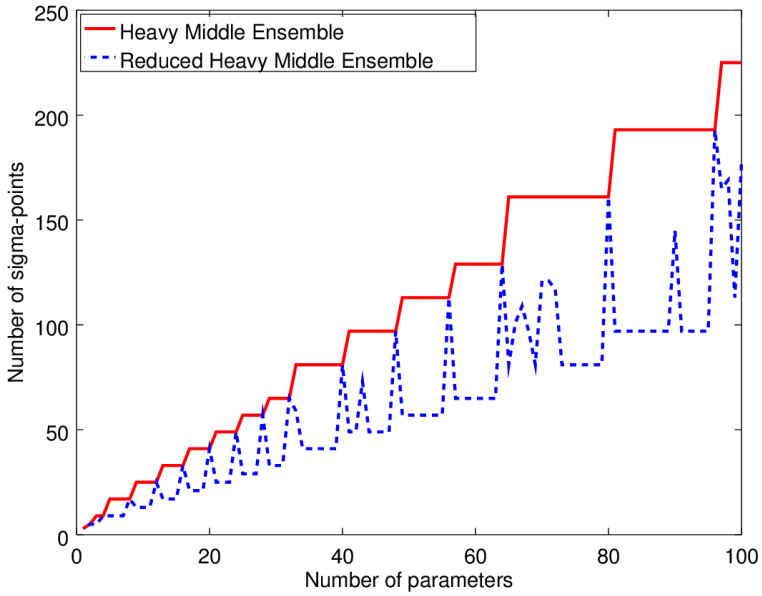


Figure 4.20: The size of the Heavy-Middle ensemble and the size of the same ensemble after decreasing its size with Simplex Reduction.

4.3.1. Combined ensembles

Gaussian ensembles, which are symmetric and thus their ensemble can be smaller, and Weibull ensembles, which are non-symmetric, have been created with an increasing number of parameters. The ensembles have been created by generating a one-dimensional ensemble and combining it with itself several times and at each point decreasing ensemble size with Simplex Reduction enforcing four moments and setting covariance to zero. Their resulting ensemble size has been graphed in Figure 4.21.

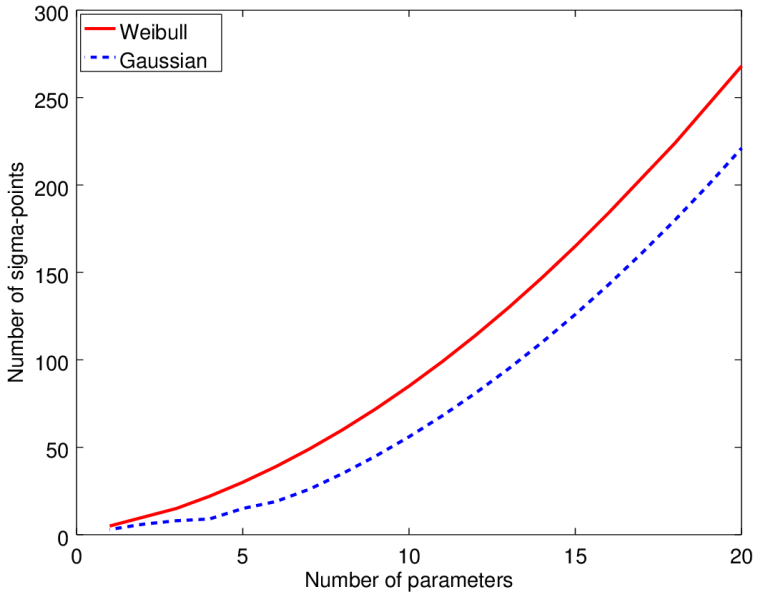


Figure 4.21: The size of ensembles created from combining one dimensional ensembles with increasing number of parameters.

4.3.2. Dependent ensembles

The ensemble size of the covariant Gaussian ensemble generated by the Corner Method is shown in Figure 4.22.

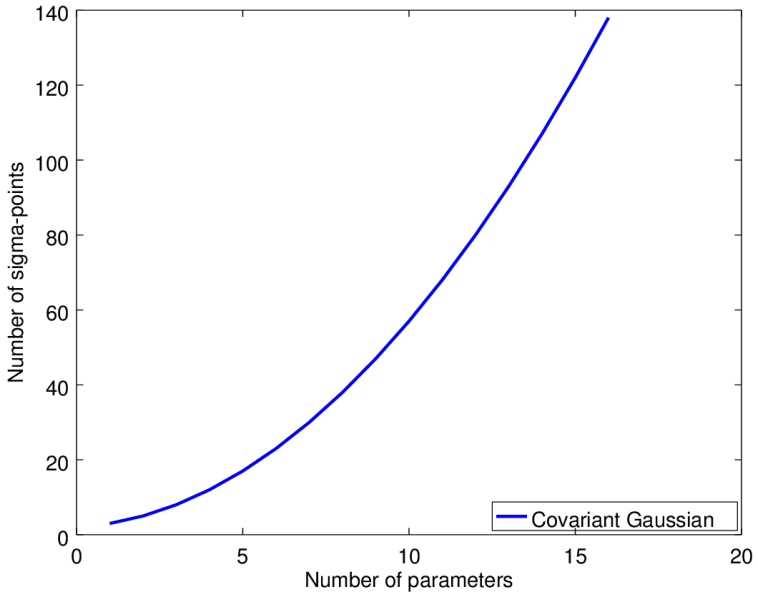


Figure 4.22: Ensemble size of the ensemble created by the Corner Method.

The ensemble size as of covariant ensembles generated by the Skewing Method is shown in Figure 4.23 for parameters with Gaussian distributions and Weibull Distributions.

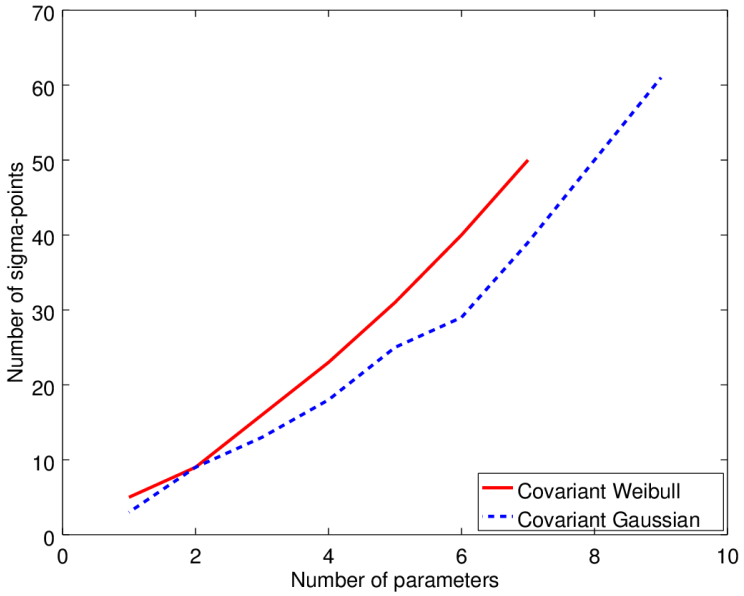


Figure 4.23: Ensemble size of ensembles created by the Skewing method.

5. Discussion

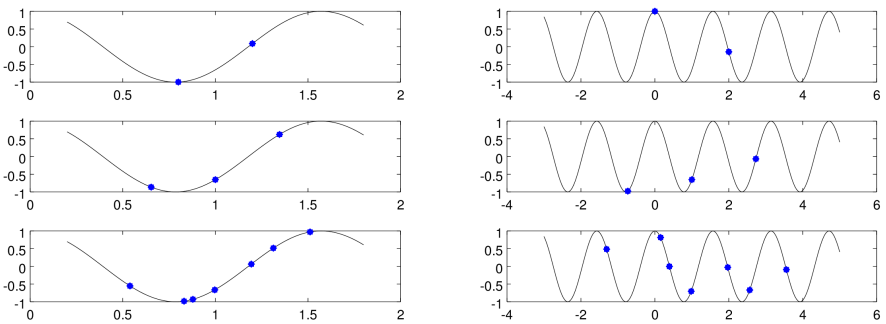
5.1. Correctness of the propagation

The tests of the propagation of mean value and variance, done in section 4.1, clearly show the general pattern that including more moments in the ensemble will give a more accurate result. The issue is only a matter of how many moments one need to encode. This question has a definitive answer which can be found when the function propagated through is explicitly known. Then it simply becomes a matter of calculating the size of the Taylor-terms in equations (2.13), or its higher-dimensional counterpart, and seeing when the terms are small enough to ignore. Often in the case of running some simulation, though, an explicit expression for the function is not known, and if one did, and if its derivatives were easily calculated, one could just calculate the uncertainty from the Taylor expansion. So in the cases when deterministic sampling is likely to be used, the number of moments to be encoded needs to be estimated in some way.

5.1.1. The tests with one parameter

The tests which were done with a reasonably "narrow" distribution with $\sigma = 0.2$ shown in Figures 4.2 to 4.5 (and Table F.1) revealed that for many functions, even functions very non-linear such as e^{2q} two moments are enough to give a good approximation for the propagated mean and variance.
The most difficult function of the once tested showed to be q^8, in the case of $\sigma = 0.2$, where six moments were needed to give a great estimation of the variance. One has to remember though that a standard deviation of 3.7, which was provided by the ensemble with four moments compared to a variance of 4.5, which is here interpreted as the actual value, are not extremely far apart. If the computations were massive, it might be worth losing a certain precision since the ensemble with four moments only needs to run 3 simulations compared to running 7 for the better value.
For the other functions tested two or four moments has shown to work well.
The tests with a wider distribution where $\sigma = 1$, shown in Figures 4.6 to 4.9, illustrates a similar pattern, but with a higher threshold for a good value of the propagated mean and variance. With such a wide distribution the error can become extremely high, even orders of magnitude, for the variance in difficult functions.
The function which was most difficult in the case if the narrow distribution, q^8, became with a wider distribution even more of a problem. A new issue arises from e^{2q} which wasn't a big problem in the case of a narrow distribution, has now become one. For q^8 and e^{2q} it was now not even enough with six moments to get a good result for the variance. The variance did, at best, reach the right order of magnitude when the ensemble carried six moments.

An interesting aspect is that periodic function $\cos(4q)$ has, in this case, caused very much trouble, which it did not do at all for the narrow distribution. This becomes a problem as the distribution has become broad enough to "feel" the periodicity of the function. Since the period now is $2\pi/4$ and the distribution has $\sigma = 1$ the deterministic ensemble now has points distributed over different periods, which makes both the propagated mean and variance get values distributed all over the map and including more moments are not necessarily better. The ensemble with four moments actually gave worse values than the one with only two. One might think that, since the six-moment-ensemble got better values again it is on its way of converging to the right value, but this turns out to just be a lucky coincidence. For example, one calculation of the propagated mean and variance with an eight-moment-ensemble is further off from the true values than the six-moment ensemble was.
The difference between these two cases is shown in Figure 5.1. One can see that in the case where $\sigma = 0.2$ there is no problem with periodicity since the function has no periodicity in the region where the ensemble has sigma-points. Here a curve can be traced through the points and will give an estimation of the functions shape. The ensembles for the wider distribution with $\sigma = 1$ does not have these properties. If one were to trace a curve through these sigma-points, one would not recognize the function at all.



(a) $\sigma = 0.2$ (b) $\sigma = 1$

Figure 5.1: The sigma-points for Deterministic Sampling applied to $\cos(4q)$ when q has (a) a thin distribution with $\sigma = 0.2$ and (b) a wide distribution with $\sigma = 1$. It is shown with three ensembles with 2, 4 and 6 moments encoded.

These cases when Deterministic Sampling fails are addressed in more detail in Section 5.3. It should be emphasized that the situations where deterministic sampling has problems are rare occasions constructed for testing purposes and in general real-world scenarios it should work perfectly well.

5.1.2. Tests with three independent parameters

The tests with three parameters showed a similar result as the one-parameter test, but the problem now has an extra layer of complexity added, since there are mixed moments to encode, which also affect the result.
In most of the cases tested encoding four moments and keeping the covariance zero gives a good result for both mean and variance.
Again some problems occur with the more difficult trigonometric function but in the other cases, it works fine.
In short, it operates in the same way as the one-dimensional case.

5.1.3. Tests with three covariant parameters

The tests with dependence encoded shown in Figures 4.13 to 4.15 (and Table F.4) also indicates that encoding more moments will give a better result. Interestingly, this one has more trouble propagating the variance correctly in some of the cases.
An issue when only encoding covariance and leaving higher order mixed moments untouched is that many of those may be relevant, in some cases maybe even more useful than the non-mixed moments.

5.1.4. Tests with discontinuous functions

Applying Deterministic Sampling to propagate uncertainty through discontinuous functions, or non-analytic functions in general, could potentially be a problem. Since the concept of Deterministic Sampling has been, in this report, motivated solely by a Taylor expansion, but a Taylor expansion is not valid for discontinuous functions.
This does not mean Deterministic Sampling should be discarded here. There is another motivation for Deterministic Sampling, used by Hessling[1], which is that there is an inherent truthfulness in encoding statistical knowledge in an ensemble, whatever that knowledge may be. Encoding the lowest order moments is still a true, although not complete, representation of the uncertainty of the parameters and by this reasoning one can see a reason as to why Deterministic Sampling may work in this case as well.
The results when the functions had discontinuities, which were presented in Figures 4.5, 4.9 and 4.18 (and Table F.1, F.2 and F.5), show that the method does indeed work somewhat okay, but the discontinuity can cause problems with the results. The mean value is somewhat off, significantly so in some cases. An interesting find is that the mean value is, in the tests done here, more off than the standard deviation. This differs from all the other cases it has been the other way around.
It is not difficult to imagine that a discontinuous step can bias the mean and variance in Deterministic Sampling since the ensemble has so few points and if there is a discontinuity at some place, this may be missed by the ensemble or affect the result more than it should.

The one-dimensional tests when the mean and standard deviation is propagated through the function $f(q) = q\mathrm{H}(q-1.1)$, who's ensemble is visualized i Figure 5.2 and results are shown in Figure 4.5, gets the mean value wrong consistently, although it seems to become better with more moments, at least with six moments. This pattern can be both due to a more accurate representation of the distribution as well as because the six-moment-ensemble has more points and is less likely to miss a discontinuity or simply more likely to get the points more evenly distributed on either side of the step. A problem occurs in the four-moment-ensemble which gives a worse result than the two-moment ensemble here. Why this happens is likely since the three points of the four-moment ensemble get a more uneven distribution around the discontinuity than the two points in the two-moment ensemble, which can be seen in Figure 5.2. The six-moment-ensemble gets the best result since it has points distributed more evenly on both sides the discontinuity.

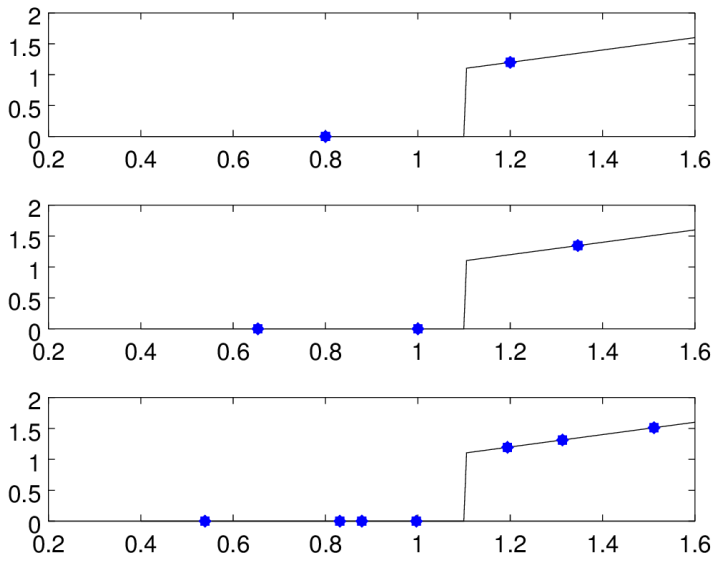


Figure 5.2: The three ensembles tested for propagating uncertainty of Gaussian parameter with $\sigma = 0.2$ through the function $f(q) = q\mathrm{H}(q-1.1)$.

Both the results for the wider Gaussian distribution from Figure 4.9 and the real-world parameters from Figure 4.18 the mean misses a little, but the standard deviation is very close to the actual value.

5.2. Ensemble size

5.2.1. Symmetrical ensembles size

Block-Diagonal Gaussian Ensemble size

The size of the Block-Diagonal Gaussian (BDG) ensemble can be seen in Figure 4.19 which shows that this ensemble grows in steps. This behavior is expected since this ensemble is built up from Extended Hadamard Matrices and they grow in steps as well. On average it seems to increase linearly. If a linear function is fitted to its growth rate, it will be roughly $2.1x + 4.7$. This line plotted along with the actual values is shown in Figure 5.3.

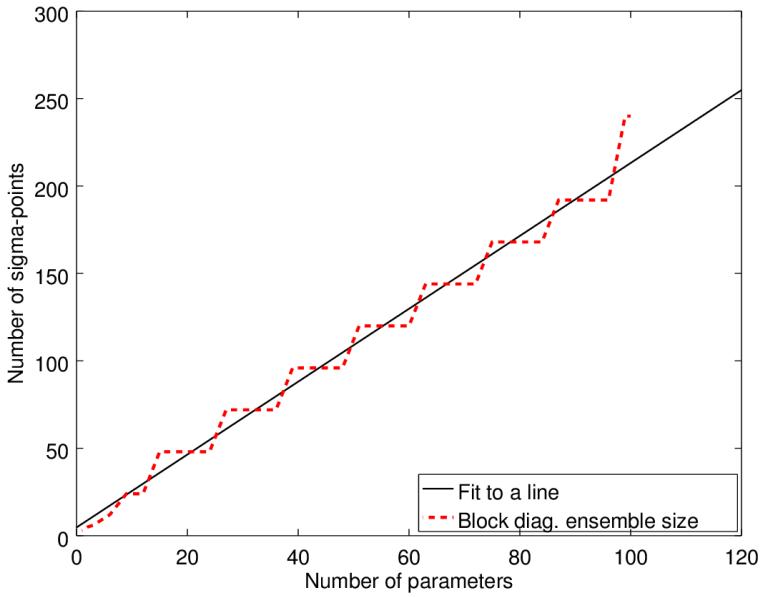


Figure 5.3: The size of the BDG ensemble along with a fitted linear function.

The approximated linearity is probably true for a higher number of parameters as well. One can calculate the ensemble sizes for ensembles with 100, 200, 400 and 800 parameters to 240, 480, 960 and 1920 samples respectively, which is exactly behaving as a linear growth.
The ensemble size can as shown in Figure 4.19, be reduced by Simplex Reduction. This plot shows a significant decrease in ensemble size, even halving its size for some ensembles while some are not reduced at all. As the figure shows, at each ”step” the reduced ensemble has the same number of parameters as the non-reduced one. This curious pattern shows up consistently at each new step. It very likely comes from some behavior in how the Simplex Algorithm works, but what specifically causes it is not currently known.
This ensemble is very quick to build and can be generated almost instan-

taneously. The reduction takes some more time though but is still not an issue until the number of parameters becomes several hundred, which is an unreasonably high number of parameters.

Heavy-Middle Ensemble size

The Heavy Middle (HM) ensemble grows in a similar way as the BDG ensemble, in the way that its size increases in steps in steps, which can be seen in Figure 4.20. Reducing it with Simplex Reduction also shrinks it significantly and just like the BDG ensemble it also has some values which are not reduced at all.
It also appears to, on average, grow linearly. If a linear function is fitted to this growth rate is will be approximately $2.2x + 1.15$.

Comparison

Both the HM ensemble and the BDG ensemble can be used to encode the moments of a Gaussian distribution, but only the HM ensemble can represent symmetric distributions in general. In the real world most parameters will be Gaussian, though, and assuming a set of Gaussian parameters the question is now which ensemble to use.
Comparing the HM ensemble to the growth of the BDG ensemble they seem to grow roughly as quickly. In Figure 5.4 and 5.5 these ensembles are plotted next to each other and in original and reduced form.

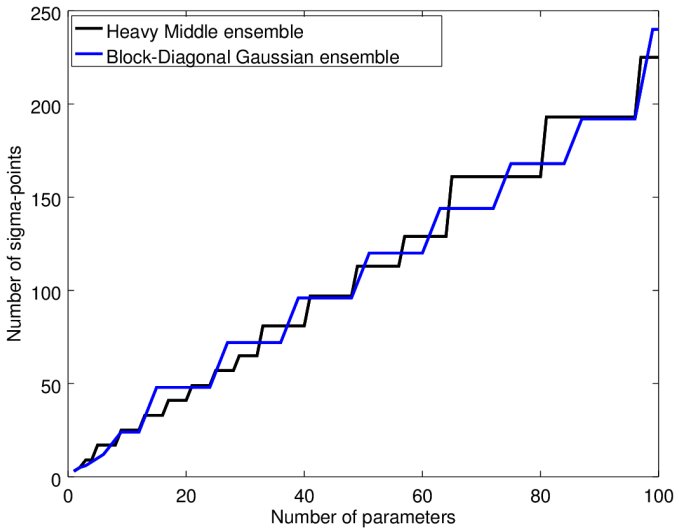


Figure 5.4: The size of the BDG ensemble and the HM ensemble compared in their original form.

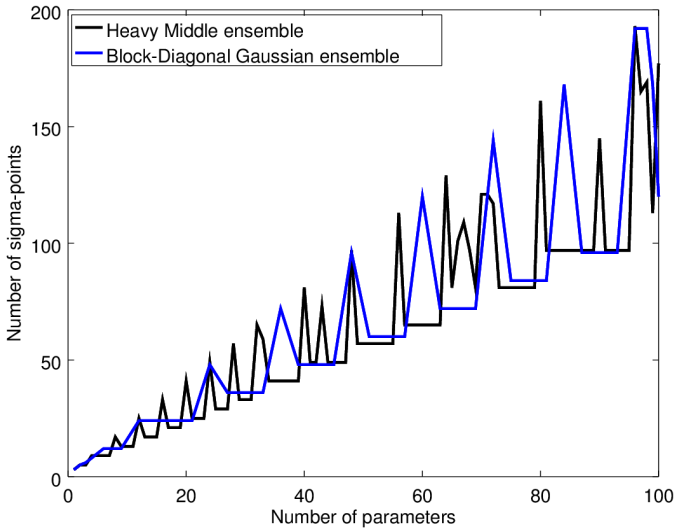


Figure 5.5: The size of the BDG ensemble and the HM ensemble compared after they have been reduced.

It does seem like they are indeed roughly growing at the same rate. So on average they are both equal in size, but at specific positions they can differ a lot. In Figure 5.5, one can see that which ensemble is the most efficient differs from point to point. Moreover, in the reduced ensembles, which probably are the ones to use, it can differ by a lot.
For 84 parameters, for example, the reduced HM ensemble has 71 sigma-points less than the reduced BDG ensemble. This is, of course, an extreme number of parameters. A more reasonable, but still high, a number is 20 parameters, and in this case, the difference is 17 points in less for the BDG ensemble.
One could define a new Gaussian ensemble which is the smallest one of the reduced BDG or HM ensembles for that number of parameters. This ensemble is plotted along with the others in Figure 5.6. As one can see this approach avoids most of the ”spikes” in the reduced ensembles.

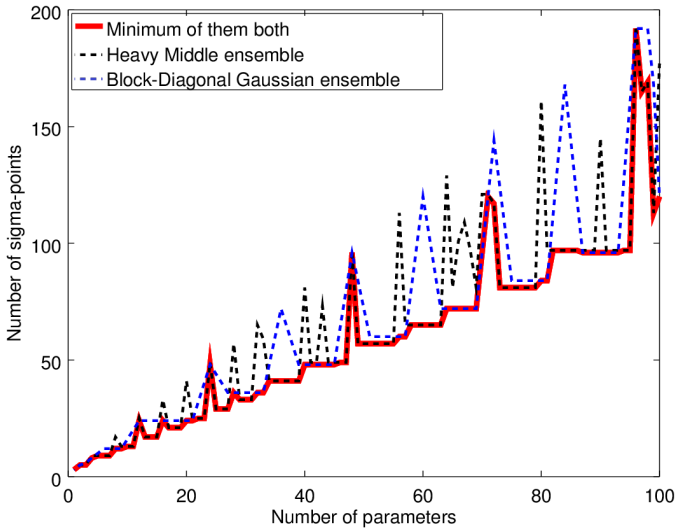


Figure 5.6: The reduced BDG and HM ensembles along with the minimum of both of them.

5.2.2. Combined ensemble size

The combined ensembles grow much quicker, as seen in Figure 4.21. This technique for building an ensemble turns out to be much less efficient in that sense. Already at 20 parameters the ensemble has over 250 sigma-points. The Gaussian ensemble which was built by combining ensembles and reducing them can be compared to the Block-Diagonal Gaussian ensembles, which shows the ineffectiveness of this way of combining ensembles when the number of parameters gets too big.
Another downside to this method is that the ensembles shown in Figure 4.21 only has the covariance set to zero while higher mixed moments are left unchecked. Often this is enough, but the much smaller Gaussian ensembles from Figure 4.19 has up to most of the fourth mixed moments encoded correctly. The Gaussian ensemble seems to be simply a better ensemble, but for now, there is no known way of making the non-symmetric ensembles better.
One could estimate the growth rate of combined ensembles to appear quadratic. A quadratic function fitted to the growth rate, about $0.50x^2 + 3.52x + 0.39$, seems to work well with the data. It is plotted in Figure 5.7 for up to 30 parameters and one would expect over 500 samples for such an ensemble. This number of sigma-points is larger than one wants to use with DS and is a problem with this technique for creating ensembles if the number of parameters is too high.

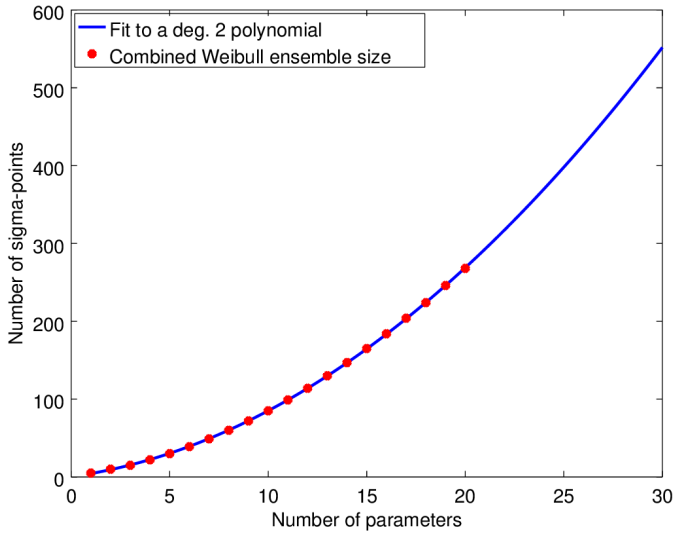


Figure 5.7: Growth rate of combined and reduced Weibull ensembles along with a second degree polynomial fitted to the data.

The combined ensemble has not been tested for ensembles with more than 24 parameters since the ensembles become so large, before reduction, that computer memory becomes an issue. Due to the combined ensemble's tendencies to grow exponentially before reduction there is a problem with creating too large ensembles.
One way around this is to, when combining several ensembles, to reduce the size before each new ensemble is added. This technique eliminates the problem of having the ensemble size grow exponentially and potentially becoming too large for the computer to handle, before finally being reduced in size. Using this technique does not mean the problem has gone away completely, just that the number of parameters which can be included in a combined ensemble has been increased.
In practice, this could mean creating an ensemble as described by the example from Appendix E.3.3. This case gives a final ensemble with 280 sigma-points, which may be a bit more than one wants. One has to remember though that this is an extreme case with 21 parameters all with unsymmetrical distributions and dependence. Often the case is rather that most parameters have Gaussian distributions and for such the number of parameters is decreased a lot. In the example from section E.3.2 the same case is presented but with Gaussian distributions instead of Weibull distributions. Here the number of sigma-points of the final ensemble is only 34, which is much more reasonable.

5.2.3. Size of covariant ensembles

Corner Method

The size of covariant Gaussian ensembles built with the Corner Method can in Figure 4.22 be seen for up to 17 parameters. The curve seems to behave in a quadratic way, and if a quadratic polynomial is fit to this data, it will indeed show to be a second-degree polynomial. The polynomial $\frac{x^2}{2} + \frac{x}{2} + 2$ fits the data exactly and this is shown in Figure 5.8.

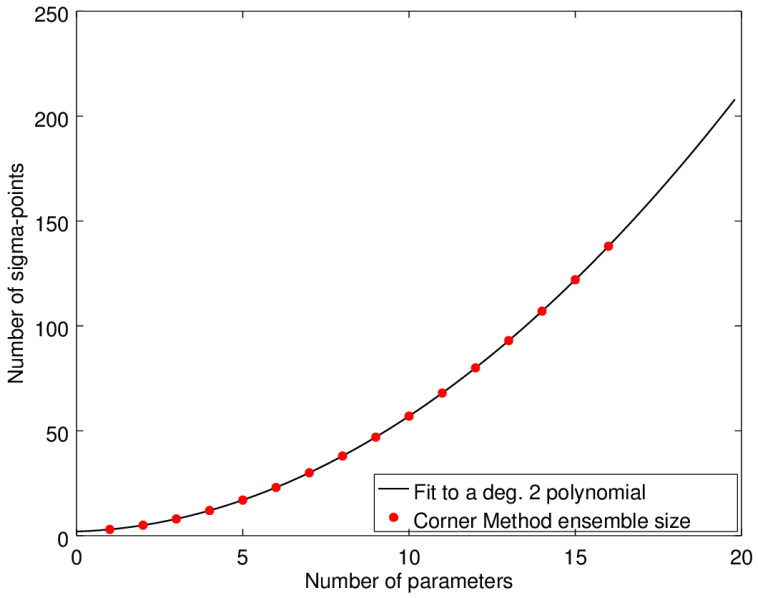


Figure 5.8: The size of the covariant Gaussian ensemble from the Corner Method fit to a second degree polynomial.

Since the size of the ensemble is huge before it is reduced, due to how it is constructed, there are memory issues with creating an ensemble much larger than this with the only the Corner Method.

Skewing method

The more general way of setting up a covariant ensemble which works for any distribution, the Skewing method, has even bigger problems creating large ensembles where all parameters are interdependent. The ensemble sizes for covariant Weibull and Gaussian distributions set up with the Skewing method shown in Figure 4.23 shows ensembles for up to 7 parameters for Weibull distributions and 9 parameters for Gaussian distributions. This is due to memory issues with a higher number of parameters than this. This problem appears due to the exponential increase in size as the ensemble

is combined and skewed and since it cannot be reduced until all covariant ensembles are combined it will become gigantic before it is reduced.
One can see that this ensemble grows quicker than the one created with the Corner method. There can be a problem if too many non-Gaussian variables are all interdependent on each other, but usually, this will not be a problem. In general, the ensemble size can be kept down by combining the different techniques for creating ensembles and only use the Skewing Method on small subsets of the parameters.

5.2.4. In general on ensemble size

In general, one wants the ensemble size to be as small as possible since the idea of Deterministic Sampling is to keep this number down. For the common model, most of the parameters will likely be Gaussian and if so the ensemble size will not be a problem. The both the Block-Diagonal Gaussian ensemble and the Heavy Middle ensemble can be created for any number of parameters and will keep the ensemble size reasonably small.
If some of the parameters have non-Gaussian distributions, it is still not a problem, since an ensemble for the Gaussian parameters can be created, be combined with the non-Gaussian parameters and reduced to give it it a decent size.
If some parameters have covariance, they also need to be combined separately. For Gaussian parameters with covariance, one can use the Corner-method to create a covariant ensemble for up to around 17 parameters before the Corner Matrix becomes too large and computer memory becomes an issue. There is usually no need to create an ensemble with this many dependent parameters, however. Usually, there are rather many small sets of parameters with interdependence and not one large set of parameters depending on each other, which means one can create several small ensembles with covariance encoded and then combine them to one larger ensemble.
If there are non-Gaussian parameters with covariance an ensemble for those is set up with the Skewing Method and combined with the rest of the ensemble.
In short, for an ensemble of small size independent Gaussian, or other symmetric, parameters are preferred. The more parameters which are not independent Gaussian, the more compromised the ensemble size will be.

5.3. Limitations of Deterministic Sampling

The correctness of the propagation in Deterministic Sampling is dependent on how well the moments can be encoded in the ensemble represent the actual value of the uncertainty. Equations (2.13) show the series expression of the propagated mean value and variance in the one-dimensional case.
The answer to the question of how many moments need to be encoded can not be answered completely without knowing the function through which

uncertainty is propagated, but as a general rule one can say that four moments in usually fine and the more the better. This rule is not the entire truth, however. What is needed is for the largest terms in equations (2.13) to be fulfilled, and while those usually are the first terms, which correspond to the lowest order moments, examples can be constructed where the dominating terms are those corresponding to much higher order moments.
For example, the mean value propagated by an ensemble is in one dimension

$$\langle f(\tilde{q})\rangle = \sum_{i=0}^{\infty} \frac{1}{i!} \langle \delta^i \tilde{q}\rangle \frac{\mathrm{d}^i f}{\mathrm{d}q^i}(\mu)$$

and if this is calculated for an extremely wide Gaussian distribution with, for example, $\sigma = 4$ one gets

$$\begin{aligned}\langle f(\tilde{q})\rangle = f(\mu) + 8\frac{\mathrm{d}^2 f}{\mathrm{d}q^2} + 32\frac{\mathrm{d}^4 f}{\mathrm{d}q^4} + 85.333\frac{\mathrm{d}^6 f}{\mathrm{d}q^6} + 170.667\frac{\mathrm{d}^8 f}{\mathrm{d}q^8} + \\ + 273.1\frac{\mathrm{d}^{10} f}{\mathrm{d}q^{10}} + 364.1\frac{\mathrm{d}^{12} f}{\mathrm{d}q^{12}} + 416.1\frac{\mathrm{d}^{14} f}{\mathrm{d}q^{14}} + 416.1\frac{\mathrm{d}^{16} f}{\mathrm{d}q^{16}} + \\ + 369.1\frac{\mathrm{d}^{18} f}{\mathrm{d}q^{18}} + 295.9\frac{\mathrm{d}^{20} f}{\mathrm{d}q^{20}} + 215.2\frac{\mathrm{d}^{22} f}{\mathrm{d}q^{22}} + 143.5\frac{\mathrm{d}^{24} f}{\mathrm{d}q^{24}} + \ldots.\end{aligned}$$

Depending on how the derivatives of f are, the terms corresponding to very high order moments may be the dominating terms. If, for example, all derivatives were equal (i.e. if $f(x) = e^x$) there would be no use encoding even the first ten moments in the ensemble since the largest terms in the series would be the ones corresponding to moments of order 12-18. In fact, an ensemble with just the four moments 12-18 encoded would, in this case, give a more accurate estimation of the mean value than even encoding the first twelve moments in the ensemble.
The variance, which is

$$\left\langle \delta^2 f(\tilde{q})\right\rangle = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \frac{1}{i!j!}\left[\left\langle \delta^{i+j}\tilde{q}\right\rangle - \left\langle \delta^i \tilde{q}\right\rangle \left\langle \delta^j \tilde{q}\right\rangle\right] \frac{\mathrm{d}^i f}{\mathrm{d}q^i}\frac{\mathrm{d}^j f}{\mathrm{d}q^j},$$

would in the same example need to go much higher to get anything close to a good estimation. In fact, it is completely unrealistic to use Deterministic Sampling at all here. The calculated expression for the variance will not be written out here, but it is first at the term

$$\frac{1}{33!^2}\left[\left\langle \delta^{66}\tilde{q}\right\rangle - \left\langle \delta^{33}\tilde{q}\right\rangle \left\langle \delta^{33}\tilde{q}\right\rangle\right]\left(\frac{\mathrm{d}^{33} f}{\mathrm{d}q^{33}}\right)^2 \approx 5.3\cdot 10^{11}\left(\frac{\mathrm{d}^{33} f}{\mathrm{d}q^{33}}\right)^2$$

that the sum's terms stop increasing in size, of course still assuming all derivatives are equal. So in this example, if the function has higher order derivatives not decreasing is size sufficiently fast this example shows a case where Deterministic Sampling is useless.

The described example is, of course, an extreme case constructed to break the concept of Deterministic Sampling. An important thing to learn though is that if the probability distribution is very broad, and the model used has significant high-order derivatives, DS is not the method to choose for uncertainty propagation.
This is what happens in the tests shown in Figure 4.7 (and Table F.2), when mean and standard deviation is propagated through the function $\cos(4q)$, with q having a Gaussian distribution with $\mu = 1$ and $\sigma = 1$. Here neither mean nor standard deviation gets an accurate result even with six moments encoded in the ensemble. The terms for the mean value will be

$$\begin{aligned}\langle\cos(4\tilde{q})\rangle = \sum_{i=0}^{\infty} \frac{1}{i!} \langle \delta^i \tilde{q} \rangle \left.\frac{\mathrm{d}^i \cos(4q)}{\mathrm{d}q^i}\right|_{q=1} \approx \\ \approx -0.65 + 5.23 - 20.9 + 55.8 - 111 + \\ +178 - 238 + 272 - 272 + 242 - 193 + \ldots\end{aligned}$$

where it is first at the term representing order 16 where the terms start decreasing in size. This explains why 6 moments encoded, or even 10, would not give a good estimation of the propagated mean. This is even worse for the propagated variance who's terms are too many to write down explicitly, but it is series will not start decreasing in size until the term

$$\frac{1}{31!^2} \left[\langle \delta^{62} \tilde{q} \rangle - \langle \delta^{31} \tilde{q} \rangle \langle \delta^{31} \tilde{q} \rangle\right] \left.\left(\frac{\mathrm{d}^{31} \cos(4q)}{\mathrm{d}q^{31}}\right)^2\right|_{q=1} \approx 3.2 \cdot 10^{11}.$$

The problem goes away if the distribution is thinner, however. The result shown in Figure 4.3 (and Table F.1) indicates that if the standard deviation instead is $\sigma = 0.2$ there is no problem getting the propagated mean and standard deviation correct. This is also explained by the equations (2.13). The propagated mean value now becomes, with only the terms corresponding to the first four moments

$$\langle\cos(4\tilde{q})\rangle = \sum_{i=0}^{\infty} \frac{1}{i!} \langle \delta^i \tilde{q} \rangle \left.\frac{\mathrm{d}^i \cos(4q)}{\mathrm{d}q^i}\right|_{q=1} \approx -0.65 + 0.21 - 0.033 = -0.47$$

which is very close to the real value. The variance becomes, if only the terms including moments up to the fourth

$$\begin{aligned}\left\langle \delta^2 \cos(4\tilde{q}) \right\rangle = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{1}{i!\,j!} \left[\langle \delta^{i+j} \tilde{q} \rangle - \langle \delta^i \tilde{q} \rangle \langle \delta^j \tilde{q} \rangle\right] \frac{\mathrm{d}^i \cos(4q)}{\mathrm{d}q^i} \frac{\mathrm{d}^j \cos(4q)}{\mathrm{d}q^j} \approx \\ \approx 0.367 - 2 \cdot 0.117 + 0.0875 = 0.2195,\end{aligned}$$

giving a propagated standard deviation of $\sigma \approx 0.469$, which is also close to the actual value of around 0.50. The value calculated here differs some from

the value gotten from the Deterministic Sampling, 0.498, (shown in Table F.1) since this calculation has simply cut the higher order terms but the deterministic ensemble actually has higher order moments, they are just not exactly the same as those of the parameter's distribution.
In summary, when using Deterministic Sampling, it is important to know it is limitations and if the simulation behaves in a periodic way or has large higher order derivatives, and the distributions of the parameters have large standard deviations, then one should use another method. If the distributions are not very wide, or the simulation behaves well, Deterministic Sampling will likely work well for the problem. It is also worth noting that the results for the real-world parameters of the κ-ε-model presented in Figures 4.16 to 4.18 there is no function tested which does not work, and even the discontinuous function worked fine.

6. Outlook

6.1. Exploring and improving the ensembles for DS

The problem of uncertainty propagation with Deterministic Sampling may appear to be solved, and in principle it may be, but naturally there are aspects which can be improved.
For example, the covariant Gaussian ensemble gained from the Corner method does have some problems being constructed for a high number of parameters which all have interdependence. The inability to create such an ensemble hints of a lack of understanding of what such an ensemble should be like. Perhaps there could be a more memory efficient way of producing such an ensemble, or perhaps an expression for it could be derived.
The same thing goes for the covariant ensembles with general distributions found from the Skewing method. This approach has an even bigger problem producing large ensembles than the Corner method. Although this is probably not a big issue in the most cases either, it should probably be explored further.
Finally, the most useful ensembles presented here are probably the ones for independent symmetrical distributions. Their behavior, in their reduced form, is not understood here, and should probably be explored further. Their "spikey" behaviour, as shown in Figures 4.19, 4.20 and 5.6. This behavior could probably be studied further and is actually kind of a mystery to the author.

6.2. Other potential applications of Deterministic Sampling

Although this report specifically has looked at the use of Deterministic Sampling for uncertainty propagation, it has been sold as an alternative to Monte Carlo-methods. Since Monte Carlo methods have all kinds of

applications this begs the question, can we replace Monte Carlo methods with Deterministic Sampling in other applications, for example, integration? Doing integration with Monte Carlo is often used when performing integration in higher dimensions, and it turns out that Deterministic Sampling can be used here, with some limitations and some cases where it works excellently well. Although this is not the purpose of the report, a quick, preliminary, investigation of this has been made as some final thoughts.
Integration with Monte Carlo is done by using that

$$\int_a^b f(x)\mathrm{d}x = (b-a)\left\langle f(x)\right\rangle_{x\in[a,b]} = (b-a)\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} f(x_i),$$

where x_i is random variables uniformly sampled from $[a,b]$, and approximating this by letting N be some high value. One could use Deterministic Sampling for this, that is one could calculate an ensemble $\tilde{q}$ representing the uniform distribution used in the Monte Carlo integration and use this ensemble to calculate an approximated value of the integral as

$$\int_a^b f(x)\mathrm{d}x = (b-a)\left\langle f(x)\right\rangle_{x\in[a,b]} = (b-a)\sum_{i=1}^{n} w^{(i)} f(q^{(i)}), \tag{6.1}$$

which would of course use way less points than the Monte Carlo integral. In general one will only use the Monte Carlo integral if integrating over very many variables though, but if it works in one variable there should not be any problem scaling it up. An ensemble for a hundred uniformly distributed variables can easily be created with the Heavy Middle ensemble.
The question remains, though, would equation (6.1) be a good approximation of the integral? This is, of course, an equivalent question to how well the ensemble can approximate $\left\langle f(x)\right\rangle_{x\in[a,b]}$ and it turns out that this depends greatly on what the function looks like and often on what the interval is. For some classes of functions, though, this method can work extremely well. The expression for the mean value as calculated from a set of weighted points is as previously noted

$$\left\langle f(x)\right\rangle_{x\in[a,b]} \approx \left\langle f(\tilde{q})\right\rangle = \sum_{i=0}^{\infty}\frac{1}{i!}\left\langle \delta^i\tilde{q}\right\rangle\frac{\mathrm{d}^i f}{\mathrm{d}q^i}(\mu). \tag{6.2}$$

where, in a uniform distribution the moments will be

$$\left\langle \delta^i\tilde{q}\right\rangle = \frac{1}{b-a}\int_a^b\left(x-\frac{a+b}{2}\right)^i\mathrm{d}x = \begin{cases} 0, & i \text{ odd} \\ \frac{(b-a)^i}{2^i(i+1)}, & n \text{ even.}\end{cases} \tag{6.3}$$

Those moments will grow with increasing i and with an increasing integration interval $[a,b]$. In Deterministic Sampling, the four first moments are usually encoded so either the interval or the higher order derivatives must be sufficiently small.

A few tests have been done to check how integration with DS works, and the result is shown in Table 6.1, along with the analytically calculated value. The interval used is $[0,4]$ in each case. The function with two variables have been integrated over $x,y \in [0,4]$. The results have some interesting properties.

Table 6.1: Tests for integration of some functions over the interval $[0,4]$ with DS. In the case of the function of two variables both variables have been integrated from 0 to 4. The analytically calculated result is shown for comparison.

Function	DS	Analytical
$x^4 - 4x^3 + 3x^2 - 2x + 1$	0.800	0.800
$x^4 + y^3 - 3(x+y)^2$	179.20	179.20
e^x	53.530	53.598

One can note that this technique of numerical integration does work fine for all of the functions tested at this interval. Actually, the two first functions, the polynomials of order 4, are more than fine. They do, in fact, get the exact result.
This is not an approximation, and the reason is the way the mean value depends on the encoded moments, as in equation (6.2). Since the first four moments are exactly correct and the function, being a polynomial of order four, has no higher derivatives than the first four, the result is not an approximation but an exact result.
This means that the integral of any polynomial of order four or less can be calculated exactly from this ensemble, and other functions integrals may be approximated by this.
One subject for further study would be to examine whether Deterministic Sampling works as a substitute for Monte Carlo integration in the real world application when it is used and whether this can be a more efficient way of performing integrations in higher dimensions.

References

[1] P. Hessling, Deterministic Sampling for Propagating Model Covariance, SIAM/ASA Journal of Uncertainty Quantification, 2013

[2] P. Hedberg and P. Hessling, Use of Deterministic Sampling for Uncertainty Quantification in CFD, NURETH-16, Chicago, IL, August 30-September 4, 2015

[3] J. Uhlmann Dynamic Map Building and Localization: New Theoretical Foundations, Ph.D Thesis, University of Oxford, UK, 1995

[4] J. Uhlmann and S, Julier, New Extension of the Kalman Filter to Nonlinear Systems, The Robotics Research Group, University of Oxford, UK, 1997

[5] Spanos, Aris, Probability Theory and Statistical Inference, New York: Cambridge University Press, 1999

[6] Casella, Berger Statistical Inference (2 ed.), Pacific Grove: Duxbury,2002

[7] G. B. Dantzig Maximization of a linear function sibject to linear inequalities, Activity Analysis of Production and Allocation, John Wiley & Sons, New York, 1951

[8] D. Gale Linear Programming and the Simplex Method, Notices of the AMS, Volume 54, Nr 3, 2007

[9] SRSA, Strålsäkerhetsmyndighetes föreskrifter och allmänna råd om säkerhet i kärntekniska anläggningar, SSMFS 2008:1 , 2008

[10] W.P Jones, B.E Launder, The prediction of laminarization with a two-equation model of turbulence, International Journal of Heat and Mass Transfer, Volume 15, Issue 2, 1972, Pages 301-314, ISSN 0017-9310

[11] Dunn MC, Shotorban B, Frendi A. Uncertainty Quantification of Turbulence Model Coefficients via Latin Hypercube Sampling Method. ASME. J. Fluids Eng. 2011;133(4):041402-041402-7. doi:10.1115/1.4003762.

[12] D.P. Kroese, T. Taimre, Z.I. Botev Handbook of Monte Carlo Methods. Wiley Series in Probability and Statistics, John Wiley and Sons, New York, 2011

Appendices

A. Theorems

A.1. About combining ensembles

Assume two ensembles $\tilde{q}_1$ and $\tilde{q}_2$. Those can be combined, as described in section 3.5, to create the new ensemble

$$\tilde{q} = \begin{pmatrix} q_1^{(1)} & q_2^{(1)} \\ q_1^{(1)} & q_2^{(2)} \\ \vdots & \vdots \\ q_1^{(1)} & q_2^{(N_2)} \\ q_1^{(2)} & q_2^{(1)} \\ \vdots & \vdots \\ q_1^{(N_1)} & q_2^{(N_2-1)} \\ q_1^{(N_1)} & q_2^{(N_2)} \end{pmatrix} \quad w = \begin{pmatrix} w_1^{(1)} w_2^{(1)} \\ w_1^{(1)} w_2^{(2)} \\ \vdots \\ w_1^{(1)} w_2^{(N_2)} \\ w_2^{(1)} w_2^{(1)} \\ \vdots \\ w_{N_1}^{(1)} w_2^{(N_2-1)} \\ w_{N_1}^{(1)} w_2^{(N_2)} \end{pmatrix}. \tag{A.1}$$

This is what is here referred to when it is said that ensembles are combined. Here the combined ensemble is written as just two parameters, but the reasoning could be generalized for an arbitrary number of parameters.
Two theorems about combined ensembles are here presented.

Theorem A.1. Two ensembles combined will give an ensemble where all the moments of the original ensembles are conserved.

Proof. The n-th moment of $\tilde{q}$ is calculated by summing the points distance to the mean value, to the power of n, multiplied by corresponding weights. It can in general terms for a combined ensemble $\tilde{q}$ be expressed as

$$\langle \delta^n \tilde{q} \rangle = \sum_{i=1}^{N_{\text{cmb}}} w_{\text{cmb}}^{(i)} \delta^n q_{\text{cmb}}^{(i)}, \tag{A.2}$$

but in this case we can rewrite the sum in terms of the original ensembles which becomes

$$\langle \delta^n \tilde{q} \rangle = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_1^{(i)} w_2^{(j)} \begin{pmatrix} \delta^n q_1^{(i)} & \delta^n q_2^{(j)} \end{pmatrix}. \tag{A.3a}$$

Moving the sums inside the vector, the expression can be rewritten as

$$\langle \delta^n \tilde{q} \rangle = \begin{pmatrix} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_1^{(i)} w_2^{(j)} \delta^n q_1^{(i)} & \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_1^{(i)} w_2^{(j)} \delta^n q_2^{(j)} \end{pmatrix} \tag{A.3b}$$

and noting that $w_1^{(i)}$ can be extracted from one of the sums and $w_2^{(j)}$ from the other we get

$$\langle \delta^n \tilde{q} \rangle = \left(\sum_{j=1}^{N_2} w_2^{(j)} \sum_{i=1}^{N_1} w_1^{(i)} \delta^n q_1^{(i)} \quad \sum_{i=1}^{N_1} w_1^{(i)} \sum_{j=1}^{N_2} w_2^{(j)} \delta^n q_2^{(j)} \right). \tag{A.3c}$$

Since the sum of the weights always is 1, the expression finally becomes

$$\langle \delta^n \tilde{q} \rangle = \left(\sum_{i=1}^{N_1} w_1^{(i)} \delta^n q_1^{(i)} \quad \sum_{j=1}^{N_2} w_2^{(j)} \delta^n q_2^{(j)} \right) = \left(\langle \delta^n \tilde{q}_1 \rangle \quad \langle \delta^n \tilde{q}_2 \rangle \right) \tag{A.3d}$$

meaning that both parameters in the new ensemble have the same moments as before.

□

The proof can be done in the same way for raw moments, meaning the mean value is also conserved. Now, let's look at the combined ensemble's mixed moments.

Theorem A.2. Two ensembles combined will give an ensemble where all the mixed moments, up to the order encoded in the original ensembles, represent independence.

Proof. The mixed moments of order $n+m$ of an ensemble $\tilde{q}$ is in general calculated as

$$\langle \delta^n \tilde{q}_a \delta^m \tilde{q}_b \rangle = \sum_{i=1}^{N} w^{(i)} \delta^n q_a^{(i)} \delta^m q_b^{(i)}. \tag{A.4a}$$

The sum can be rewrite rewritten in terms of the original ensembles leading to the expression as

$$\langle \delta^n \tilde{q}_1 \delta^m \tilde{q}_2 \rangle = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_1^{(i)} w_2^{(j)} \delta^n q_1^{(i)} \delta^m q_2^{(j)} \tag{A.4b}$$

which can be rewritten as

$$\langle \delta^n \tilde{q}_1 \delta^m \tilde{q}_2 \rangle = \sum_{i=1}^{N_1} w_1^{(i)} \delta^n q_1^{(i)} \sum_{j=1}^{N_2} w_2^{(j)} \delta^m q_2^{(j)} \tag{A.4c}$$

giving the final result for the mixed moments of combined ensembels.

$$\langle \delta^n \tilde{q}_1 \delta^m \tilde{q}_2 \rangle = \langle \delta^n \tilde{q}_1 \rangle \langle \delta^m \tilde{q}_2 \rangle. \tag{A.4d}$$

Note that this has been shown to be true in the case when the parameters have been combined according to equation (A.1) only. For example one can not let $q = q_1 = q_2$ and get $\langle \delta^{n+m} \tilde{q} \rangle \neq \langle \delta^n \tilde{q} \rangle \langle \delta^m \tilde{q} \rangle$ for distributions in general.
From (A.4d) we can see that the covariance of the combined ensemble, which we get when $n = m = 1$, will be zero. All mixed moments won't be zero though, but expression (A.4d) is in fact the expression for mixed moments for independent variables.

To show that these mixed moments are the correct ones for independent ensembles this case will now be examined. Assume two parameters q_1 and q_2 are independent random variables and are sampled independently N times. Also, let N be a sufficiently high number so that the samples represents the distribution of q_1 and q_2 well. Just calculating the mixed moments of order $n+m$ between them would give

$$\langle \delta^n q_1 \delta^m q_2 \rangle = \frac{1}{N} \sum_{i=1}^{N} \delta^n q_1^{(i)} \delta^m q_2^{(i)}$$

which does not seem to be equal to $\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle$ in general. Let's instead start from the other side and see that

$$\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle = \frac{1}{N} \sum_{i=1}^{N} \delta^n q_1^{(i)} \frac{1}{N} \sum_{i=1}^{N} \delta^m q_2^{(i)} \tag{A.5a}$$

which can be rewritten as

$$\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta^n q_1^{(i)} \delta^m q_2^{(j)}. \tag{A.5b}$$

This sum can be written in many ways, but one way is to first write all the terms where $i = j$ and then all terms where j is shifted one cyclic step from i and next all terms where j is shifted two steps and so on which will include all the terms in the sum and in this case be useful. The expression would be

$$\begin{aligned}
\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta^n q_1^{(i)} \delta^m q_2^{(j)} = \\
= \frac{1}{N^2} \big(&\delta^n q_1^{(1)} \delta^m q_2^{(1)} + \delta^n q_1^{(2)} \delta^m q_2^{(2)} + \ldots + \delta^n q_1^{(N)} \delta^m q_2^{(N)} + \\
&\delta^n q_1^{(1)} \delta^m q_2^{(2)} + \delta^n q_1^{(2)} \delta^m q_2^{(3)} + \ldots + \delta^n q_1^{(N)} \delta^m q_2^{(1)} + \\
&+ \delta^n q_1^{(1)} \delta^m q_2^{(3)} + \delta^n q_1^{(2)} \delta^m q_2^{(4)} + \ldots + \delta^n q_1^{(N)} \delta^m q_2^{(2)} + \\
&\vdots \\
&+ \delta^n q_1^{(1)} \delta^m q_2^{(N)} + \delta^n q_1^{(2)} \delta^m q_2^{(1)} + \ldots + \delta^n q_1^{(N)} \delta^m q_2^{(N-1)} \big).
\end{aligned} \tag{A.5c}$$

Looking at this expression one can realize another way of writing the sums which is

$$\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle = \frac{1}{N} \sum_{k=0}^{N-1} \left(\frac{1}{N} \sum_{i=1}^{N} \delta^n q_1^{(i)} \delta^m q_2^{((i+k) \mathrm{mod} N)} \right) \tag{A.5d}$$

where $(i+k)\mathrm{mod}N$ is the modulos operator making $(i+k)$ cyclic with period N. We now note that since q_1 and q_2 are independently sampled it does not matter which point get's paired with which when taking the sum. Each of

the inner sums inside the parenthesis in (A.5d) results in the mixed moment we were looking for, i.e.

$$\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle = \frac{1}{N} \sum_{k=0}^{N-1} \left\langle \delta^n q_q \delta^m q_1 \right\rangle \tag{A.5e}$$

and since the sum is over a constant now we get the final result for the sampling

$$\langle \delta^n q_1 \rangle \langle \delta^m q_2 \rangle = \langle \delta^n q_1 \delta^m q_2 \rangle \tag{A.5f}$$

which is the same expression as in equation (A.4d). This means that the mixed moments of the combined ensemble be those representing independence between the parameters.

□

B. Theory

B.1. A more detailed description of linear optimization with the Simplex Method

Assume we have a quantity z we want to maximize or minimize, and that z can be written as a linear function of some positive parameters $\{x_1, x_2, \ldots x_n\}$: $x_i > 0 \ \forall \ i$.

$$z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \tag{B.1a}$$

Also assume there are some linear constraints to these parameters described as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned} \tag{B.1b}$$

This can be written as a linear system of equations as

$$\begin{aligned} z - c_1 x_1 - c_2 x_2 - \ldots - c_n x_n &= 0 \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned} \tag{B.1c}$$

and in matrix form as

$$\begin{pmatrix} 1 & -c_1 & -c_2 & \ldots & -c_n \\ 0 & a_{11} & a_{12} & \ldots & a_{1n} \\ 0 & a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} \begin{pmatrix} z \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \tag{B.1d}$$

Unless $n = m$ this will not have a unique solution, but can be subject to optimization, in this case, assume we want to maximize z.
By using the technique from linear algebra known as Gaussian elimination, one can rewrite the system into the following equivalent form

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & a'_{0m+1} & \dots & a'_{0n} \\ 0 & 1 & 0 & \dots & 0 & a'_{1m+1} & \dots & a'_{1n} \\ 0 & 0 & 1 & \dots & 0 & a'_{2m+1} & \dots & a'_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & a'_{mm+1} & \dots & a'_{mn} \end{pmatrix} \begin{pmatrix} z \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ \vdots \\ b'_m \end{pmatrix}. \tag{B.2}$$

Upon visual inspection one can note that one solution to (B.2) is to set all x_i which appear in more than one row to zero which leads to the solution

$$\begin{aligned} x_i = & \begin{cases} b'_i & i \leq m \\ 0 & i > m \end{cases} \\ z = & \quad b'_0. \end{aligned} \tag{B.3}$$

This is at least one working solution, and it gives $z = b'_0$, which is some number gained from the Gaussian elimination. This solution is valid as long as all b'_i are positive since all x_i are supposed to be positive, so if we are really lucky, this will be a valid solution.
There is little chance that just pivoting the first m columns this way would give a solution with only positive x_i, but there is a way to find one, or see that one does not exist. If we choose very carefully which columns and rows to pivot we can get the system into some form,

$$\begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ a_{11} & \dots & 0 & a_{1,i} & \dots & 0 & a'_{1j} & 1 & \dots & a'_{1N} \\ a_{21} & \dots & \vdots & a_{2,i} & \dots & 0 & a'_{2j} & \vdots & \dots & a'_{2N} \\ \vdots & \dots & 1 & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & a_{2,i} & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & a_{2,i} & \dots & 1 & a'_{m,j} & 0 & \dots & \vdots \\ a_{m+1,1} & \dots & 0 & a_{2,i} & \dots & 0 & a'_{m+1,j} & 0 & \dots & a'_{m+1,N} \end{pmatrix} \begin{pmatrix} z \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} 1 \\ b'_1 \\ b'_2 \\ \vdots \\ b'_{m+1} \end{pmatrix}, \tag{B.4}$$

where all b'_i corresponding to the pivoted columns are greater than or equal to zero and hence find a solution where all $x_i \geq 0$.
Assuming this is done, the question is now, is this the optimal solution which maximizes z? To know this one need only look at the top row of (B.2) and see the values of the coefficients a'_{0i}. If they are all positive we know any change in any of the parameters would give a smaller z, and hence z is maximized by this solution.
The Simplex Method offers an algorithm for performing the Gaussian elimination according to certain rules which give only positive coefficients in the top row, and keeps all $b'_i > 0$, or shows that such a solution does not exist[8], hence finding a valid solution which maximizes z. This method will not be described any further here, however.

A note is that same approach can be used to minimize z, simply by choosing to maximize $z' = -z$.
Another more important note, in this case, is that many of the x_i will be set to zero, if possible. It is for this feature the Simplex algorithm will be used in this project, rather than for optimization.

B.2. A more detailed description of Simplex Reduction

Assume a set of N sigma points $\tilde{q}$ intended to represent the distribution of a parameter q and encode the first m moments. Hence, the ensemble should, with weights fulfill equation (2.21). This can be set up in matrix form as

$$\underbrace{\begin{pmatrix} 1 & 1 & \dots & 1 \\ q^{(1)} & q^{(2)} & \dots & q^{(N)} \\ \delta^2 q^{(1)} & \delta^2 q^{(2)} & \dots & \delta^2 q^{(N)} \\ \vdots & \vdots & \vdots & \vdots \\ \delta^m q^{(1)} & \delta^m q^{(2)} & \dots & \delta^m q^{(N)} \end{pmatrix}}_{(m+1)\times N} \underbrace{\begin{pmatrix} w^{(1)} \\ w^{(2)} \\ w^{(3)} \\ \vdots \\ w^{(N)} \end{pmatrix}}_{N\times 1} = \underbrace{\begin{pmatrix} 1 \\ \langle q \rangle \\ \langle \delta^2 q \rangle \\ \vdots \\ \langle \delta^m q \rangle \end{pmatrix}}_{(m+1)\times 1} \tag{B.5}$$

Note that, if $N > m+1$, this equation has not a unique solution in general. This can now be treated as an optimization problem. Since the weights should all fulfil $w_i \geq 0$ the Simplex Method described briefly in Section 2.5 and more in depth in Appendix B.1, turns out to be useful here. This may seem counter-intuitive since the Simplex Method is designed specifically for minimization or maximisation of a quantity, which is not the aim here. The aim here is a solution where all weights are positive or zero and this is something the Simplex Algorithm can find.
What is done here is to define the quantity $z = w_1 + w_2 + \ldots + w_n$ and apply the Simplex Algorithm to it for minimization or maximization. This is fed into the Simplex Algorithm, along with equation (B.5) as constraints. The Simplex Algorithm will set up the system as

$$\begin{pmatrix} 1 & -1 & -1 & \dots & -1 \\ 0 & 1 & 1 & \dots & 1 \\ 0 & q^{(1)} & q^{(2)} & \dots & q^{(N)} \\ 0 & \delta^2 q^{(1)} & \delta^2 q^{(2)} & \dots & \delta^2 q^{(N)} \\ 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & \delta^m q^{(1)} & \delta^m q^{(2)} & \dots & \delta^m q^{(N)} \end{pmatrix} \begin{pmatrix} z \\ w^{(1)} \\ w^{(2)} \\ w^{(3)} \\ \vdots \\ w^{(N)} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \langle q \rangle \\ \langle \delta^2 q \rangle \\ \vdots \\ \langle \delta^m q \rangle \end{pmatrix} \tag{B.6}$$

It is now apparent that whether the algorithm intended to minimize or maximize z is irrelevant, it will always be 1 and the whole first row is unnecessary, but most pre-written Simplex Methods require it as input. The Simplex method will still start working and perform the Gaussian elimination according to certain rules which, if possible, gets the system in the following form, with all of the constants on the right-hand side are positive. If this procedure is successful, the system will be in the form

$$\begin{pmatrix} 1 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ a_{11} & \dots & 0 & a_{1,i} & \dots & 0 & a'_{1j} & 1 & \dots & a'_{1N} \\ a_{21} & \dots & \vdots & a_{2,i} & \dots & 0 & a'_{2j} & \vdots & \dots & a'_{2N} \\ \vdots & \dots & 1 & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & a_{2,i} & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & a_{2,i} & \dots & 1 & a'_{m,j} & 0 & \dots & \vdots \\ a_{m+1,1} & \dots & 0 & a_{2,i} & \dots & 0 & a'_{m+1,j} & 0 & \dots & a'_{m+1,N} \end{pmatrix} \begin{pmatrix} z \\ w^{(1)} \\ w^{(2)} \\ \vdots \\ w^{(N)} \end{pmatrix} = \begin{pmatrix} 1 \\ b'_1 \\ b'_2 \\ \vdots \\ b'_{m+1} \end{pmatrix}, \quad \text{(B.7)}$$

where for every row there is a column with the number one in that row and zeroes everywhere else. From this form one can trivially find a solution by setting $w^{(i)} = b'_i$ for weights corresponding to these columns and all other weights zero.

A pre-written Simplex Method will perform all of this automatically and return a vector of weights where some of them may be zero. If some weights are zero, this means those points are not needed in the ensemble and can be removed.

This method of using the Simplex Algorithm to not only calculate a valid set of positive weights but also to remove unnecessary sigma-points has been named Simplex Reduction.

Simplex Reduction can be used just as well for ensembles with several parameters. If this is done, the equations for each of the parameters moments should be included in (B.5), as well as the equations for mixed moments one wants to encode. This is a way of encoding dependence or independence into the ensemble.

If one wanted, one could write an algorithm not requiring the top row of (B.6) as input since for this purpose it is not needed. The author has used the open source GNU Linear Programming Kit (GLPK) for the Simplex Method. In MATLAB's optimization toolkit there is a linear programming function which also works for this purpose, but only if it is set to specifically use the Simplex Method. Other linear optimization methods will give a result, but not set the excessive weights to zero. This is why Simplex is used.

C. Expressions for the ensembles

C.1. The Block-Diagonal Gaussian Ensemble

Extending the reasoning about Gaussian ensembles done in Section ??, a general expression for independent Gaussian parameters can be written. Now the standard deviation and mean value will be taken into account which means the ensemble will be scaled to fulfill the standard deviation and translated to fulfill the mean value. The general expression for the Block-Diagonal Gaussian Ensemble, encoding four moments of n independent parameters, is

$$\tilde{q}_n = \begin{cases} \sqrt{3}\sigma \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \begin{pmatrix} \langle q \rangle \\ \langle q \rangle \\ \langle q \rangle \end{pmatrix} & n = 1 \\ \sqrt{3} \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \end{pmatrix} \Sigma + M, & n = 2 \\ \sqrt{3} \begin{pmatrix} C_a & & \\ & C_b & \\ & & C_c \end{pmatrix} \Sigma + M, & n \geq 3. \end{cases} \tag{C.1a}$$

The scaling is done by the matrix Σ, which is a diagonal matrix including all the standard deviations. Multiplying with this from the right scales the ensemble to give each parameter the correct standard deviation. It is written as

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{pmatrix}. \tag{C.1b}$$

The matrix M provides the translation of the ensemble giving each parameter the correct mean value. It is written as

$$M = \begin{pmatrix} \langle q_1 \rangle & \langle q_2 \rangle & \dots & \langle q_n \rangle \\ \langle q_1 \rangle & \langle q_2 \rangle & \dots & \langle q_n \rangle \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}. \tag{C.1c}$$

The weight-vector w_n corresponding to the ensemble $\tilde{q}_n$ is

$$\tilde{w}_n = \begin{cases} \begin{pmatrix} 1/6 & 1/6 & 2/3 \end{pmatrix}^{\mathsf{T}}, & n = 1 \\ \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/3 \end{pmatrix}^{\mathsf{T}}, & n = 2 \\ \begin{pmatrix} W_a \\ W_b \\ W_c \end{pmatrix}, & n \geq 3. \end{cases} \tag{C.1d}$$

Here, for $n \geq 3$, the weights come in chunks W_k corresponding to the rows with the matrix C_k. If the number of rows in C_k is r, W_k will simply be

$$W_k = \frac{1}{3r} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{r \times 1} \tag{C.1e}$$

In equation (C.1), the order of the matrices C_a, C_b and C_c are increased incrementally, i.e. one of them increases per added dimension according to

$$\begin{aligned} a = &\quad \lfloor \tfrac{n}{3} \rfloor + \delta_{\mathrm{rem}(\frac{n}{3}),2} \\ b = &\quad \lfloor \tfrac{n}{3} \rfloor + \delta_{\mathrm{rem}(\frac{n}{3}),1} \\ c = &\qquad \lfloor \tfrac{n}{3} \rfloor, \end{aligned} \tag{C.1f}$$

where $\lfloor \frac{n}{3} \rfloor$ is the floor function, $\mathrm{rem}(\frac{n}{3})$ is the remainder from the division and δ_{ij} is Krönecker's delta function.
For the actual matrices C_k the Extended Hadamard Matrices described in Appendix D.2 are recommended. The Corner Matrix described in Appendix D.1 also works but gives a much larger ensemble if the number of parameters is big.
This way an ensemble can be constructed for any number of independent parameters with a Gaussian distribution, with the first four moments encoded.

C.1.1. The Heavy Middle Ensemble

Assume n independent parameters q_i all with symmetric distributions of the same shape. Their ensemble can be written as

$$\tilde{q}_n = \begin{pmatrix} C_n \\ 0_{1\times n} \end{pmatrix} \Omega + M, \quad \tilde{w} = \begin{pmatrix} W \\ 1 - \frac{\langle \delta^2 q \rangle^2}{\langle \delta^4 q \rangle} \end{pmatrix}. \tag{C.2a}$$

The scaling is here done by the matrix Ω, similarly to how the matrix Σ scales the Block-Diagonal Gaussian ensemble. Ω is defined as

$$\Omega = \begin{pmatrix} \sqrt{\frac{\langle \delta^4 q_1 \rangle}{\langle \delta^2 q_1 \rangle}} & 0 & \dots & 0 \\ 0 & \sqrt{\frac{\langle \delta^4 q_2 \rangle}{\langle \delta^2 q_2 \rangle}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\frac{\langle \delta^4 q_n \rangle}{\langle \delta^2 q_n \rangle}} \end{pmatrix} \tag{C.2b}$$

The matrix M translates the ensemble to the correct mean value and is defined the same way as in equation (C.1c). The weights to the non-zero samples, which here are collected in the column vector W, will all be equal and

$$W = \frac{1}{r} \frac{\langle \delta^2 q \rangle^2}{\langle \delta^4 q \rangle} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{r\times 1}$$

where r is the number of rows in the matrix C_n.

Just as in the Section C.1 the matrix C_n can be any matrix with where all columns have mean value 0, each point is -1 or 1, and all columns are orthogonal. The Extended Hadamard Matrix defined in Appendix D.2 is recommended.
This ensemble has been named the Heavy Middle Ensemble since the weight of the middle point stays the same while the weights of the surrounding points decrease with the number of parameters.
Both the Block-Diagonal Gaussian Ensemble and the Heavy Middle Ensemble can often be reduced in size by Simplex Reduction.

D. Definitions

D.1. The Corner Matrix

The Corner Matrix of order n is here defined as a matrix which has, if each row is considered a point in space, one point in each corner of the n dimensional room $[-1,1]^n$. It can be recursively constructed as

$$C_n = \begin{cases} \begin{pmatrix} 1 \\ -1 \end{pmatrix}, & n = 1 \\ \begin{pmatrix} 1_{2^{n-1}\times 1} & C_{n-1} \\ -1_{2^{n-1}\times 1} & C_{n-1} \end{pmatrix}, & n \geq 1 \end{cases} \tag{D.1}$$

The first order Corner Matrices are

$$C_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{D.2a}$$

$$C_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix} \tag{D.2b}$$

$$C_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{pmatrix} \tag{D.2c}$$

$$C_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix} \tag{D.2d}$$

D.2. The Extended Hadamard Matrix

When constructing the Block-Diagonal Gaussian Ensemble described in section C.1, a special class of matrices is used. The kind of matrix used is a matrix where every column has mean value zero, standard deviation one, every element either 1 or -1 and every column orthogonal to every other. One matrix fulfilling these requirements is an extension of the Hadamard matrices which will be described here.

A Hadamard Matrix

A Hadamard Matrix, H_n, of order n is a square $n \times n$ matrix who's elements $H_n^{(i,j)} \in \{-1, 1\}$, and who's rows are all orthogonal to each other.
One way of constructing Hadamard matrices, invented by Sylvester in 1867, is the recursive rule

$$H_{2^n} = \begin{cases} \begin{pmatrix} 1 \end{pmatrix}, & 2^n = 1 \\ \begin{pmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{pmatrix}, & 2^n > 1 \end{cases} \tag{D.3}$$

which limits their construction to order's $1, 2, 4, 8 \ldots$. The Hadamard Conjecture, though, proposes that a Hadamard Matrix or order $4k$ exists for every positive integer k. Although this has not been proven, many more ways of constructing Hadamard Matrices has been found since then.
This means the first order Hadamard Matrices are as follows.

$$H_1 = \begin{pmatrix} 1 \end{pmatrix} \tag{D.4a}$$

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{D.4b}$$

$$H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{D.4c}$$

The Extended Hadamard Matrix

It can be verified that if one constructs a matrix in the following way

$$D_n = \begin{pmatrix} H_n \\ -H_n \end{pmatrix}, \tag{D.5a}$$

the resulting matrix will be one where each column has variance 1, mean value 0, and no covariance with the other columns. Verifying this is quite trivial since every element in H_n is ± 1 and if each collumn adds a mirror of itself it will have the sum 0 and of course the standard deviation will be 1 since every element is deviating from the mean by 1.

Since n can only be a power of 2, but we want, for the purpose of building an ensemble, such a matrix but allowing n to be any integer. This is simply done by using

$$E_n = D_m(:, 1:n), \tag{D.5b}$$

where n is any integer, and m is the smallest integer $\geq n$ for which a Hadamard Matrix can be constructed. $D_m(:, 1:n)$ refers to the n first columns of D_m.
The matrix E_n is here called the Extended Hadamard Matrix of order n, although it bears a resemblance to the matrix for the Binary Ensemble described by Hessling[1].
The first orders matrices are as follows.

$$E_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{D.6a}$$

$$E_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & -1 \\ -1 & 1 \end{pmatrix} \tag{D.6b}$$

$$E_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \tag{D.6c}$$

$$E_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \end{pmatrix} \tag{D.6d}$$

$$E_5 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix} \tag{D.6e}$$

$$E_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & -1 \end{pmatrix} \tag{D.6f}$$

E. Examples

E.1. Determine ensemble without weights

This section is intended to be a clarifying example of what is described in Section 2.4.3
Assume we have one uncertain parameter q with the standard exponential distribution, i.e. it has the PDF

$$f(q) = e^{-q}.$$

Its first four moments would then be

$$\begin{aligned}\langle q \rangle &= 1\\ \langle \delta^2 q \rangle &= 1\\ \langle \delta^3 q \rangle &= 2\\ \langle \delta^4 q \rangle &= 9.\end{aligned}$$

Also assume we want use Deterministic Sampling to find the mean value and standard deviation of this the function $\cos(q)$ when q has this distribution. If we want to encode these four moments into an ensemble for Deterministic Sampling, we need to force them to fulfill four equations and hence four points are required. So we look for an ensemble $\tilde{q} = \{q^{(1)}, q^{(2)}, q^{(3)}, q^{(4)}\}$ fulfilling these moments and the equation becomes

$$\begin{cases} 1 &= \frac{1}{N}\left[q^{(1)} + q^{(2)} + q^{(3)} + q^{(4)}\right] \\ 1 &= \frac{1}{N}\left[(q^{(1)}-1)^2 + (q^{(2)}-1)^2 + (q^{(3)}-1)^2 + (q^{(4)}-1)^2\right] \\ 2 &= \frac{1}{N}\left[(q^{(1)}-1)^3 + (q^{(2)}-1)^3 + (q^{(3)}-1)^3 + (q^{(4)}-1)^3\right] \\ 9 &= \frac{1}{N}\left[(q^{(1)}-1)^4 + (q^{(2)}-1)^4 + (q^{(3)}-1)^4 + (q^{(4)}-1)^4\right] \end{cases}$$

This is a very messy system of equations, but its solutions can be found numerically. One solution is

$$\begin{aligned} q^{(1)} &\approx 0.76 - 1.36i \\ q^{(2)} &\approx 3.17 \\ q^{(3)} &\approx -0.69 \\ q^{(4)} &\approx 0.76 + 1.36i. \end{aligned}$$

The system did, in this case, turn out to only have complex solutions. This is what is expected, in the general case, since there is no guarantee that real solutions will exist.
So we have an ensemble. Now let's calculate the propagated mean value and variance. The propagated mean value becomes

$$\begin{aligned}
&\langle \cos(q) \rangle \approx \\
&\approx \frac{1}{4}\left[\cos(q^{(1)}) + \cos(q^{(2)}) + \cos(q^{(3)}) + \cos(q^{(4)})\right] \approx \\
&\approx \frac{1}{4}\left[\cos(0.76 - 1.36i) + \cos(3.17) + \cos(-0.69) + \cos(0.76 + 1.36i)\right] \approx \\
&\approx 0.69.
\end{aligned}$$

The propagated variance becomes

$$\begin{aligned}
&\left\langle \delta^2 \cos(q) \right\rangle \approx \\
&\approx \frac{1}{N}\left[(\cos(q^{(1)}) - 0.69)^2 + \ldots + (\cos(q^{(4)}) - 0.69)^2\right] \approx \\
&\approx 0.26.
\end{aligned}$$

Are complex sigma-points a problem? Not necessarily, but it should be avoided. This solution does indeed encode the first four moments, and using it to perform uncertainty propagation would probably work in most cases. A problem is though that the parameters usually represent some physical quantity, which should be a real value. Therefore, the software used for simulations will in many cases not support complex values for the parameters. The complex parameters are also in principal a bad representation of the probability distribution which should be strictly real-valued.
The problem of getting a complex ensemble can be solved by using a weighted ensemble instead.

E.2. Encoding moments into a weighted ensemble

This section is intended as a clarifying example of what is described in Section 2.4.4.
Assume, just as in Section E.1, that we have one uncertain parameter q with the standard exponential distribution, i.e. it has the PDF

$$f(q) = e^{-q}, \quad q \in [0, \infty).$$

Its first four moments would then be

$$\begin{aligned}
\langle q \rangle &= 1 \\
\left\langle \delta^2 q \right\rangle &= 1 \\
\left\langle \delta^3 q \right\rangle &= 2 \\
\left\langle \delta^4 q \right\rangle &= 9.
\end{aligned}$$

Also assume we want use Deterministic Sampling to find the mean value and standard deviation of this the function $\cos(q)$ when q has this distribution.

To avoid an ensemble with complex sigma-points, we want to use a weighted ensemble. This means each sigma-point $q^{(i)}$ gets an associated weight $w^{(i)}$. Now we want the ensemble to encode the first four moments, but also make sure the sum of the weights is 1. This means there are five constraints, and we need five sigma-points to fulfill this. The equations are now

$$\begin{cases} 1 &= w^{(1)} + w^{(2)} + w^{(3)} + w^{(4)} + w^{(5)} \\ 1 &= w^{(1)}q^{(1)} + w^{(2)}q^{(2)} + w^{(3)}q^{(3)} + w^{(4)}q^{(4)} + w^{(5)}q^{(5)} \\ 1 &= w^{(1)}(q^{(1)}-1)^2 + w^{(2)}(q^{(2)}-1)^2 + w^{(3)}(q^{(3)}-1)^2 + w^{(4)}(q^{(4)}-1)^2 + w^{(5)}(q^{(5)}-1)^2 \\ 2 &= w^{(1)}(q^{(1)}-1)^3 + w^{(2)}(q^{(2)}-1)^3 + w^{(3)}(q^{(3)}-1)^3 + w^{(4)}(q^{(4)}-1)^3 + w^{(5)}(q^{(5)}-1)^3 \\ 9 &= w^{(1)}(q^{(1)}-1)^4 + w^{(2)}(q^{(2)}-1)^4 + w^{(3)}(q^{(3)}-1)^4 + w^{(4)}(q^{(4)}-1)^4 + w^{(5)}(q^{(5)}-1)^4 \end{cases}$$

This system should be solved for the weights, meaning we need to set the sigma-points $q^{(i)}$ to some reasonable values. Any distinct five points will guarantee a solution, but knowing what the distribution looks like one should try to pick some points which are reasonable.
Let's pick points within two standard deviations from the mean value. Since the mean value is 1 and the standard deviation also is 1, let's pick

$$\begin{aligned} q^{(1)} &= 0 \\ q^{(2)} &= 1 \\ q^{(3)} &= 1.5 \\ q^{(4)} &= 2 \\ q^{(5)} &= 3. \end{aligned}$$

We now get a linear system guaranteed to have real solutions. The system becomes

$$\begin{cases} 1 &= w^{(1)} + w^{(2)} + w^{(3)} + w^{(4)} + w^{(5)} \\ 1 &= 0w^{(1)} + 1w^{(2)} + 1.5w^{(3)} + 2w^{(4)} + 3w^{(5)} \\ 1 &= 1w^{(1)} + 0w^{(2)} + 0.5^2w^{(3)} + 1w^{(4)} + 2^2w^{(5)} \\ 2 &= -1w^{(1)} + 0w^{(2)} + 0.5^3w^{(3)} + 1w^{(4)} + 2^3w^{(5)} \\ 9 &= 1w^{(1)} + 0w^{(2)} + 0.5^4w^{(3)} + 1w^{(4)} + 2^4w^{(5)}. \end{cases}$$

Solving this system gives the weights

$$\begin{aligned} w^{(1)} &\approx 0.61 \\ w^{(2)} &\approx -3.0 \\ w^{(3)} &\approx 7.1 \\ w^{(4)} &\approx -4.5 \\ w^{(5)} &\approx 0.78. \end{aligned}$$

Let's use this ensemble to propagate uncertainty through the function $\cos(q)$. The propagated mean value is calculated as

$$\begin{gathered}\langle \cos(q) \rangle \approx \\ \approx w^{(1)} \cos(q^{(1)}) + w^{(2)} \cos(q^{(2)}) + w^{(3)} \cos(q^{(3)}) + w^{(4)} \cos(q^{(4)}) + w^{(5)} \cos(q^{(5)}) \approx \\ \approx 0.61 \cos(0) - 3.0 \cos(1) + 7.1 \cos(1.5) - 4.5 \cos(2) + 0.78 \cos(3) \approx \\ \approx 0.60.\end{gathered}$$

The propagated variance is then

$$\begin{gathered}\left\langle \delta^2 \cos(q) \right\rangle \approx \\ \approx w^{(1)} \left(\cos(q^{(1)}) - 0.60 \right)^2 + \ldots + w^{(5)} \left(\cos(q^{(5)}) - 0.60 \right)^2 \approx \\ \approx -0.60.\end{gathered}$$

Here we have run into a serious problem. The variance is not only wrong but completely unreasonable. The variance, defined as a sum of square values, can not possibly be negative. This is a problem which can arise if negative weights are used and is the reason for this project.

E.3. Examples of how to create ensembles

This section contains some examples of how the different methods for creating ensembles can be used together. The examples chosen here are picked to represent how ensembles can be created in general, and the process is shown as pseudo-Octave-Code

E.3.1. 5 Gaussian parameters without covariance

An ensemble for five independent parameters without dependence can be generated with the process shown here.

```
%Five Gaussian parameters without covariance
%with four moments encoded

%define moments, for example
moments = [0,2,4,3,1            % mean value for each parameter
           1,2,1,0.2,0.3        % variances
           0,0,0,0,0            % mom 3 = 0
           3,12,3,0.12,0.27];    % mom 4 = 3*variance^2

% Create ensemble with Heavy Middle ensemble
[q,w] = HeavyMiddleEnsemble(moments)

%Could also have used the BDG-ensemble
%[q,w] = BlockDiagonalGaussian(moments)

%Reduce ensemble size to get final ensemble
[q,w] = reduceEnsemble(q,moments)  %Final ensemble has 9 ...
    sigma-points
```

E.3.2. 21 parameters with Gaussian distributions and some dependence

Assume 21 Gaussian parameters, three of which have some interdependence with each other. An ensemble for this can be generated by the steps shown bellow. The final ensemble will have 34 sigma-points.

```
% Example of 21 Gaussian parameters, 3 of which have ...
    covariance described by matrix
C = [1     0.5    -0.2
     0.5    1      0.3
    -0.2    0.3     1   ] ;

% Ensemble for the 18 independent parameters. Let's pick the ...
    simple moments
m1 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3];

[q1, w1] = BlockDiagonalGaussian(m1);

% Ensemble for the 3 covariant parameters
m2 = [0,0,0
      1,1,1
      0,0,0
      3,3,3];
% Use the Corner method to get a covariant ensemble for the ...
    covariant parameters.
[q2,w2] = CornersEnsemble(m2,C);

%Combine them
[q,w] = combineEnsembles(q1,w1,q2,w2);

%Reduce ensemble to smaller size must enforce the total ...
    covariance. That is why a large covariance matrix is ...
    created, and the lower right corner becomes the one for ...
    the three dependent parameters
Ctot = diag(ones(1,21));
Ctot(19:21, 19:21) = C;

%and total moments variable
mtot = [m1, m2];

% now reduce and get final ensemble, which will have 34 ...
    sigma-points.
[q,w] = reduceEnsembleCovar(q, mtot, Ctot);
```

E.3.3. 21 parameters with unsymmetrical distributions and some dependence

Assume a model through which uncertainty should be propagated. Also assume that the model has 21 uncertain parameters, off which three have some known interdependence. Assume also all of them are Weibull-distributed and the ensemble should have four moments should be encoded, which means the individual parameters will have an ensemble of five sigma-points since the distribution is unsymmetrical.
The main point of this example is to show how the problem of combined ensembles growing exponentially in size as many parameters are combined, if in each step of combining two ensembles; they are reduced before combining with another parameters ensemble.
This ensemble is built by first creating an ensemble for the independent parameters by combining and reducing. Then an ensemble for the dependent part is created, and covariance is encoded. The two parts are then combined and reduced. The process is described step by step bellow and shown as pseudo-Octave-code.

1. Individual ensembles for each parameter are created with the Shotgun Algorithm. They get five sigma-points each.

2. The ensembles for the 18 independent parameters are combined and reduced. Their ensemble will have 221 points.

3. Now ensembles for the three dependent parameters are combined and skewed. Their resulting ensemble now has 125 samples.

4. The ensemble for the dependent parameters is skewed to encode their covariance, and Simplex Reduction is used to reduce it is size and make sure four moments are encoded. Now this ensemble has 14 samples.

5. Combine those ensembles resulting in a single ensemble for everything. It has 3094 samples.

6. The final ensemble is once again reduced by in size by Simplex Reduction, and covariance is forced in the process. The resulting ensemble has 280 sigma points.

```
% Example of how to generate 18 independent ensembles with ...
    the shotgun-algorithm

%Create and combine the independent ensembles
[qind, wind]  = shotgunAlgorithm(moments);
for i=2:18
  [qtemp, wtemp] = shotgunAlgorithm(moments);
  [qind, wind] = combineEnsembles(qind,wind, qtemp, wtemp);
  %It is important to reduce ensemble size between each ...
      step, otherwise it will grow unreasonably large
  [qind,wind] = reduceEnsemble(qcmb, moments);
end

%Now make the covariant part
for i=1:3
  [qdep{i}, wdep{i}] = shotgunAlgorithm(moments);
end

%Combine those
[qdep, wdep] = shotgunAlgorithm(moments);
for i=2:3
  [qtemp, wtemp] = shotgunAlgorithm(moments);
  [qdep, wdep] = combineEnsembles(dep,dep, qtemp, qtemp);
  %This time we do NOT reduce at each combination
end

% Skew Ensemble to encode covariance
qdep = skewEnsemble(qdep, covariance);
% Reduce to make sure higher moments are correct and reduce ...
    size
[qdep,wdep] = reduceEnsembleCovar(qdep, moments, covariance);


%Now combine both dependent and independent part and reduce
[q,w] = combineEnsembles(qind,wind,qdep,wdep);
[q,w] = reduceEnsembleCovar(q,moments, covariance);
```

F. Detailed results

F.1. 1D ensembles

Propagation of uncertainty with Deterministic Sampling has here been tested through functions of one parameter with a Gaussian distribution. Several functions has been tested.
The calculation has been done with Deterministic Sampling (DS) with 2, 4 and 6 encoded moments. A Random Sampling (RS) simulation has been done as reference.
The ensembles used are for 2 and 4 moments

$$\tilde{q}_{2\text{mom}} = \begin{pmatrix} \mu - \sigma \\ \mu + \sigma \end{pmatrix} \quad \tilde{w}_{2\text{mom}} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \tag{F.1a}$$

$$\tilde{q}_{4\text{mom}} = \begin{pmatrix} \mu - \sqrt{3}\sigma \\ \mu \\ \mu + \sqrt{3}\sigma \end{pmatrix} \quad \tilde{w}_{4\text{mom}} = \begin{pmatrix} 1/6 \\ 2/3 \\ 1/6 \end{pmatrix}, \tag{F.1b}$$

which are the previously known expressions. The ensemble with 6 moments is

$$\tilde{q}_{6\text{mom}} = \begin{pmatrix} -2.304616\sigma + \mu \\ -0.016909\sigma + \mu \\ 0.970453\sigma + \mu \\ 2.557482\sigma + \mu \\ 1.565832\sigma + \mu \\ -0.606727\sigma + \mu \\ -0.845351\sigma + \mu \end{pmatrix} \quad \tilde{w}_{6\text{mom}} = \begin{pmatrix} 0.047800 \\ 0.250499 \\ 0.316545 \\ 0.025754 \\ 0.016841 \\ 0.019078 \\ 0.323482 \end{pmatrix}, \tag{F.1c}$$

which has been found with the Shotgun Algorithm.
For a Gaussian distribution with mean value $\mu = 1$ and standard deviation $\sigma = 0.2$ the propagated mean and standard deviation are shown in Table F.1.

Table F.1: The propagated mean value and standard deviation of a Gaussian parameter with $\mu = 1$ and $\sigma = 0.2$ through various functions.

$f(q)$		DS 2 mom 2 samples	DS 4 mom 3 samples	DS 6 mom 7 samples	RS 10^7 samples
q^4	Mean	1.2416	1.2448	1.2448	1.2451
	Std	0.83200	0.96056	0.96627	0.96628
q^8	Mean	2.2338	2.4722	2.4833	2.4830
	Std	2.0660	3.7410	4.5519	4.5240
$\cos(q)$	Mean	0.52953	0.52960	0.52960	0.52965
	Std	0.16717	0.16564	0.16566	0.16564
$\cos(4q)$	Mean	-0.45540	-0.47587	-0.47462	-0.47465
	Std	0.54290	0.49764	0.50729	0.50440
e^q	Mean	2.7728	2.7732	2.7732	2.7733
	Std	0.54728	0.56001	0.56022	0.56042
e^{2q}	Mean	7.9881	8.0042	8.0045	8.0044
	Std	3.0351	3.3141	3.3318	3.3347
$q\mathrm{H}(q-1.1)$	Mean	0.60000	0.22440	0.43903	0.37785
	Std	0.60000	0.50178	0.58863	0.56974

For a wider Gaussian distribution, with $\sigma = 1$ and $\mu = 1$ the propagated results show in Table F.2.

Table F.2: The propagated mean value and standard deviation of a Gaussian parameter with $\mu = 1$ and $\sigma = 1$ through various functions.

$f(q)$		DS 2 mom 2 samples	DS 4 mom 3 samples	DS 6 mom 7 samples	RS 10^7 samples
q^4	Mean	8.0	10.0	10.0	10.003
	Std	8.0	20.445	25.785	25.721
q^8	Mean	128	518.00	764.88	766.09
	Std	128	1156.5	4054.5	6812.3
$\cos(q)$	Mean	0.29193	0.33129	0.32785	0.32762
	Std	0.70807	0.56331	0.62148	0.60377
$\cos(4q)$	Mean	0.42725	-0.60987	0.088038	-1.1963e-4
	Std	0.57275	0.26989	0.59168	0.70723
e^q	Mean	4.1945	4.4531	4.4817	4.4825
	Std	3.1945	4.9476	5.7460	5.8807
e^{2q}	Mean	27.799	44.309	53.102	54.696
	Std	26.799	85.795	193.63	373.34
$qH(q-1.1)$	Mean	1.000	0.45534	0.75857	0.86629
	Std	1.000	1.0182	1.0442	1.0155

F.2. 3D independent ensembles

Here propagation of uncertainty has been tested with functions of three parameters. The parameters chosen here are of one Gaussian parameter q_1 with $\mu = 1$ and $\sigma = 0.1$ and two Weibull-distributed parameters, q_2 and q_3, with $\alpha = 6$ and $\beta = 1$, giving them a mean value of ≈ 0.93 and a standard deviation of ≈ 0.18.
The testing has been done by Deterministic Sampling with 2 and 4 moments encoded. In the case of 4 moments the test has been performed in one case with only the covariance set to zero while the higher order mixed moments left unchecked and in one case with up to the first four mixed moments set to represent independence. RS with 10^7 samples is used as reference.
The ensembles have been found by finding the parameters individual ensembles with the Shotgun Algorithm, combined them as described in section 3.5 and Simplex Reduction has been used to reduce the number of samples. The ensembles used can be viewed in Appendix G.1.
The results as these ensembles are used to calculate the propagated mean value and variance through some test functions is shown in Table F.3.

Table F.3: The mean value and standard deviation propagated through various functions with deterministic sampling using various ensembles. The ensemble with 2 moments encoded has covariance set to zero. One of the ensembles with 4 moments encoded has covariance set to zero while higher mixed moments are left unchecked. The other ensemble with 4 moments encoded has the first 4 mixed moments set to represent independence. Random Sampling with 10^7 samples is used as reference.

$f(q_1,q_2,q_3)$		DS 2 mom cov zero 9 smp	DS 4 mom cov zero 14 smp	DS 4 mom 4 mixed 31 smp	RS 10^7 smp
$(q_1+q_2+q_3)^4$	Mean	70.569	70.073	70.098	70.087
	Std	34.453	26.058	25.897	25.852
$(q_1+q_2+q_3)^6$	Mean	638.34	616.14	616.49	616.67
	Std	560.49	353.61	343.31	340.10
$e^{(q_1+q_2+q_3)}$	Mean	18.167	18.028	18.030	18.035
	Std	6.5899	4.9255	4.8854	4.8637
$\cos(q_1+q_2+q_3)$	Mean	-0.92676	-0.92363	-0.92440	-0.92432
	Std	0.13768	0.070309	0.095134	0.094377
$\cos((q_1+q_2+q_3)^4)$	Mean	0.50285	0.76876	-0.035705	-3.4515e-4
	Std	0.35402	0.42882	0.72885	0.70692

F.3. 3D dependent ensembles

The results from the tests with three dependent parameters, described in Section 4.1.3, are shown in Table F.4. The three parameters all have Gaussian distributions with $\sigma = 0.1$. The ensembles have encoded 2, 4 and 6

moments respectively. They all have the correct covariance encoded but no higher order mixed moments are set. The ensembles used are shown in Appendix G.2. Table F.4 shows the mean value and variance gained by propagating uncertainty with these ensembles through different functions.

Table F.4: The mean value and standard deviation propagated uncertainty through various functions with Deterministic Sampling. All of the ensembles have encoded covariance but no higher mixed moments. Random Sampling is used as reference, also with the same covariance encoded.

$f(q_1,q_2,q_3)$		DS 2 mom 7 smp	DS 4 mom 8 smp	DS 6 mom 22 smp	RS 10^7 smp
$(q_1+q_2+q_3)^4$	Mean	1.2736	1.2938	1.3140	1.3074
	Std	0.92550	1.3430	1.1115	1.1744
$(q_1+q_2+q_3)^6$	Mean	1.7107	1.9178	1.8727	1.8755
	Std	1.7701	3.7320	2.4291	2.9516
$e^{(q_1+q_2+q_3)}$	Mean	2.7901	2.7915	2.7979	2.7961
	Std	0.56182	0.61858	0.61515	0.61309
$\cos(q_1+q_2+q_3)$	Mean	-0.9692	-0.96951	-0.96911	-0.96923
	Std	0.03882	0.056749	0.031321	0.040674
$\cos((q_1+q_2+q_3)^4)$	Mean	-0.27913	0.67729	0.29426	-4.50e-05
	Std	0.52662	0.37705	0.69734	0.70714

F.4. Semi-real world example

This is a more detailed presentation of the data from the tests described in section 4.1.4.

Two ensembles representing the five parameters of the $\kappa-\varepsilon$-model has been tested. One of the ensembles encode four marginal moments and enforces covariance to zero, but does not set any restrictions on the higher order mixed moments. The other ensemble encodes both four marginal moments as well as sets all of the mixed moments up to order four to represent independence.

These ensembles have not been tested on actual CFD-simulations, but evaluated by test-functions. The mean value and standard deviation of those tests are presented in Table F.5.

Table F.5: The mean and standard deviation as ensembles representing the κ-ε-model's uncertain parameters are propagated through various test functions.

Function		DS 4 moms cov zero 20 smp	DS 4 moms 4 mixed 53 smp	RS 10^7 smp
$(\Sigma q_i)^4$	Mean	5598.2	5598.3	5598.4
	Std	350.50	360.21	360.28
$(\Sigma q_i)^8$	Mean	3.1470e7	3.1471e7	3.1470e7
	Std	4.0231e6	4.0700e6	4.0710e6
$\cos((4\Sigma q_i))$	Mean	-0.86004	-0.85640	-0.85677
	Std	0.22535	0.18376	0.18317
$e^{(\Sigma q_i)}$	Mean	1.2107e15	1.2266e15	1.2264e15
	Std	6.4531e14	7.4206e14	7.5010e14
$\mathrm{H}(-8.6+\Sigma q_i)\Sigma q_i$	Mean	5.4516	5.8101	5.6261
	Std	4.1703	4.1120	4.1102

G. Ensembles used in the tests

G.1. Independent 3D ensembles from section 4.1.2

q_1 is a Gaussian parameter with $\mu = 1$ and $\sigma = 0.1$. q_2 and q_3 are Weibull-distributed parameters, with $\alpha = 6$ and $\beta = 1$. Ensemble $\tilde{q}_{2\mathrm{mom}}$ has the first 2 moments encoded, $\tilde{q}_{4\mathrm{mom}}$ has the first 4 and $\tilde{q}_{4\mathrm{mix}}$ has the first 4 moments encoded as well as all mixed moments up to the 4th forced. All ensembles has the covariance set to zero.
The ensemble for 2 moments is

$$\tilde{q}_{2\mathrm{mom}} = \begin{pmatrix} 0.9 & 0.92693 & 0.92801 \\ 0.9 & 0.92693 & 0.35668 \\ 0.9 & 1.3857 & 0.92801 \\ 1.1 & 0.44957 & 0.92801 \\ 1.1 & 0.44957 & 1.3104 \\ 1.1 & 0.92693 & 0.92801 \\ 1.1 & 0.92693 & 0.35668 \\ 1.1 & 1.3857 & 0.92801 \\ 1.1 & 1.3857 & 1.3104 \end{pmatrix} \quad \tilde{w}_{2\mathrm{mom}} = \begin{pmatrix} 0.49889 \\ 0.00025367 \\ 0.00085789 \\ 0.028309 \\ 0.043198 \\ 0.29398 \\ 0.059255 \\ 0.030315 \\ 0.044945 \end{pmatrix}. \quad \text{(G.1a)}$$

For 4 moments where the covariance has been set to zero, but no higher mixed moments have been forced, the ensemble used is

$$\tilde{q}_{4\mathrm{mom}} = \begin{pmatrix} 0.82679 & 0.62283 & 0.58692 \\ 0.82679 & 0.62283 & 1.2123 \\ 0.82679 & 0.95649 & 0.93679 \\ 0.82679 & 0.95649 & 1.1932 \\ 0.82679 & 1.2476 & 0.58692 \\ 1 & 0.99012 & 0.93679 \\ 1 & 0.62283 & 1.1932 \\ 1 & 0.95649 & 0.9262 \\ 1 & 0.95649 & 0.93679 \\ 1 & 1.2476 & 1.2123 \\ 1.1732 & 0.99012 & 1.1932 \\ 1.1732 & 0.95649 & 0.9262 \\ 1.1732 & 1.2476 & 0.9262 \\ 1.1732 & 0.59595 & 0.58692 \end{pmatrix} \quad \tilde{w}_{4\mathrm{mom}} = \begin{pmatrix} 0.027618 \\ 0.060407 \\ 0.012344 \\ 0.0099932 \\ 0.056304 \\ 0.0457 \\ 0.035929 \\ 0.4912 \\ 0.067585 \\ 0.02625 \\ 0.058251 \\ 0.0017574 \\ 0.036087 \\ 0.070572 \end{pmatrix} \tag{G.1b}$$

and for 4 moments where all the first 4 mixed moments have been fixed

$$\tilde{q}_{4\mathrm{mix}} = \begin{pmatrix} 0.82679 & 1.3012 & 0.96052 \\ 0.82679 & 0.61074 & 0.35621 \\ 0.82679 & 0.61074 & 0.96052 \\ 0.82679 & 0.94864 & 0.99418 \\ 0.82679 & 0.94864 & 0.61386 \\ 0.82679 & 0.94864 & 1.2499 \\ 0.82679 & 0.94864 & 0.35621 \\ 0.82679 & 0.94864 & 0.96052 \\ 0.82679 & 0.3125 & 0.61386 \\ 0.82679 & 1.2177 & 0.61386 \\ 0.82679 & 1.2177 & 1.2499 \\ 0.82679 & 1.2177 & 0.96052 \\ 1 & 0.61074 & 0.61386 \\ 1 & 0.61074 & 1.2499 \\ 1 & 0.61074 & 0.96052 \\ 1 & 0.94864 & 0.61386 \\ 1 & 0.94864 & 1.2499 \\ 1 & 0.94864 & 0.96052 \\ 1 & 0.3125 & 0.96052 \\ 1 & 1.2177 & 0.61386 \\ 1 & 1.2177 & 0.96052 \\ 1.1732 & 1.3012 & 0.96052 \\ 1.1732 & 0.61074 & 0.61386 \\ 1.1732 & 0.61074 & 0.96052 \\ 1.1732 & 0.94864 & 0.99418 \\ 1.1732 & 0.94864 & 0.61386 \\ 1.1732 & 0.94864 & 1.2499 \\ 1.1732 & 0.94864 & 0.96052 \\ 1.1732 & 0.3125 & 0.96052 \\ 1.1732 & 1.2177 & 0.61386 \\ 1.1732 & 1.2177 & 1.2499 \end{pmatrix} \quad \tilde{w}_{4\mathrm{mix}} = \begin{pmatrix} 0.00068992 \\ 8.0458\text{e-}5 \\ 0.030937 \\ 0.021847 \\ 0.026005 \\ 0.0088534 \\ 5.5219\text{e-}5 \\ 0.053019 \\ 2.7124\text{e-}5 \\ 0.0069034 \\ 0.0094503 \\ 0.0087998 \\ 0.03591 \\ 0.022321 \\ 0.065428 \\ 0.076958 \\ 0.057431 \\ 0.30383 \\ 0.00039582 \\ 0.016791 \\ 0.087604 \\ 0.0061707 \\ 0.00021641 \\ 0.028191 \\ 0.027613 \\ 0.026413 \\ 0.0086525 \\ 0.052506 \\ 0.00067619 \\ 0.0068636 \\ 0.0093649 \end{pmatrix} \tag{G.1c}$$

G.2. Covariant 3D Gaussian ensembles from section 4.1.3

Three gaussian parameters, all with $\sigma = 0.1$ and a covariance matrix

$$C = \sigma^2 \begin{pmatrix} 1 & 0.02 & 0 \\ 0.02 & 1 & 0.6 \\ 0 & 0.6 & 1 \end{pmatrix}$$

are represented in three ensembles with 2, 4 and 6 moments encoded along with the covariance. Mixed moments of higher order than 2 is not forced in any way.

The ensembles for 2 and 4 moments have been calculated with the Corner-method described in section 3.6.2, while the one with 6 moments has been

calculated with the Skewing-method from section 3.6.1.
The ensemble with 2 moments is

$$\tilde{q}_{\text{2moms}} = \begin{pmatrix} 1.1 & 1.1 & 1.1 \\ 1.1 & 1.1 & 0.9 \\ 1.1 & 0.9 & 1.1 \\ 1.1 & 0.9 & 0.9 \\ 0.9 & 1.1 & 1.1 \\ 0.9 & 0.9 & 1.1 \\ 0.9 & 0.9 & 0.9 \end{pmatrix} \quad \tilde{w}_{\text{2moms}} = \begin{pmatrix} 0.155 \\ 0.100 \\ 0.095 \\ 0.150 \\ 0.245 \\ 0.005 \\ 0.250 \end{pmatrix}, \tag{G.2a}$$

the one with 4 moments is

$$\tilde{q}_{\text{4moms}} = \begin{pmatrix} 1.1732 & 1.1732 & 1.1732 \\ 1.1732 & 1.1732 & 0.82679 \\ 1.1732 & 0.82679 & 1.1732 \\ 1.1732 & 0.82679 & 0.82679 \\ 0.82679 & 1.1732 & 1.1732 \\ 0.82679 & 0.82679 & 1.1732 \\ 0.82679 & 0.82679 & 0.82679 \\ 1 & 1 & 1 \end{pmatrix} \quad \tilde{w}_{\text{4moms}} = \begin{pmatrix} 0.051667 \\ 0.033333 \\ 0.031667 \\ 0.05 \\ 0.081667 \\ 0.0016667 \\ 0.083333 \\ 0.66667 \end{pmatrix} \tag{G.2b}$$

and the one with 6 moments is

$$\tilde{q}_{6\text{moms}} = \begin{pmatrix} 1.2491 & 1.1209 & 0.91764 \\ 0.85459 & 0.87524 & 0.97737 \\ 0.85626 & 1.0258 & 1.0276 \\ 0.8631 & 1.0232 & 1.0194 \\ 1.0112 & 0.91821 & 1.1183 \\ 1.0094 & 0.47851 & 0.47832 \\ 1.0151 & 1.1184 & 0.91804 \\ 1.0146 & 1.2186 & 1.2184 \\ 1.0133 & 1.0248 & 1.0192 \\ 0.58902 & 1.2376 & 1.2897 \\ 1.1719 & 0.73486 & 0.56295 \\ 1.1708 & 0.94989 & 1.1907 \\ 1.1729 & 1.0264 & 1.0189 \\ 1.0238 & 1.3127 & 1.3125 \\ 1.0239 & 1.2892 & 1.2419 \\ 1.0204 & 0.82464 & 0.82005 \\ 1.0201 & 0.87699 & 0.97709 \\ 1.0168 & 0.68715 & 1.1039 \\ 1.0175 & 0.5635 & 0.73299 \\ 1.0234 & 1.1709 & 1.0751 \\ 1.0221 & 0.97735 & 0.87672 \\ 1.0215 & 1.0753 & 1.1706 \end{pmatrix} \quad \tilde{w}_{6\text{moms}} = \begin{pmatrix} 0.013112 \\ 0.036678 \\ 0.15967 \\ 0.055697 \\ 0.0048227 \\ 1.2252e-05 \\ 0.0085342 \\ 0.0079886 \\ 0.075825 \\ 0.0013674 \\ 0.00068884 \\ 0.034299 \\ 0.085541 \\ 0.0010115 \\ 0.003553 \\ 0.0777 \\ 0.11153 \\ 0.0017618 \\ 0.00046851 \\ 0.10505 \\ 0.15007 \\ 0.064616 \end{pmatrix} \tag{G.2c}$$

2016:26

The Swedish Radiation Safety Authority has a comprehensive responsibility to ensure that society is safe from the effects of radiation. The Authority works to achieve radiation safety in a number of areas: nuclear power, medical care as well as commercial products and services. The Authority also works to achieve protection from natural radiation and to increase the level of radiation safety internationally.

The Swedish Radiation Safety Authority works proactively and preventively to protect people and the environment from the harmful effects of radiation, now and in the future. The Authority issues regulations and supervises compliance, while also supporting research, providing training and information, and issuing advice. Often, activities involving radiation require licences issued by the Authority. The Swedish Radiation Safety Authority maintains emergency preparedness around the clock with the aim of limiting the aftermath of radiation accidents and the unintentional spreading of radioactive substances. The Authority participates in international co-operation in order to promote radiation safety and finances projects aiming to raise the level of radiation safety in certain Eastern European countries.

The Authority reports to the Ministry of the Environment and has around 300 employees with competencies in the fields of engineering, natural and behavioural sciences, law, economics and communications. We have received quality, environmental and working environment certification.

Strålsäkerhetsmyndigheten
Swedish Radiation Safety Authority

SE-171 16 Stockholm
Solna strandväg 96

Tel: +46 8 799 40 00
Fax: +46 8 799 40 10

E-mail: registrator@ssm.se
Web: stralsakerhetsmyndigheten.se