NASA/TP—2010–216447

Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Analysis

J.M. Hanson and B.B. Beard
Marshall Space Flight Center, Marshall Space Flight Center, Alabama

September 2010

The NASA STI Program…in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and mission, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results…even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>

- E-mail your question via the Internet to <help@sti.nasa.gov>

- Fax your question to the NASA STI Help Desk at 443–757–5803

- Phone the NASA STI Help Desk at 443–757–5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TP—2010–216447

Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Analysis

J.M. Hanson and B.B. Beard
Marshall Space Flight Center, Marshall Space Flight Center, Alabama

National Aeronautics and
Space Administration

Marshall Space Flight Center • MSFC, Alabama 35812

September 2010

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076–1320
443–757–5802

This report is also available in electronic form at
<https://www2.sti.nasa.gov>

FOREWORD

Recent NASA Engineering and Safety Center (NESC) experiences have pointed to the need for the establishment of consistent criteria and recommended practices for performing Monte Carlo analyses across the Agency. The purpose of this Technical Publication (TP) is to recommend Monte Carlo method practices and serve as a tutorial guide for guidance, navigation, and control (GN&C) engineering practitioners at NASA, and its industry and academic partners. It is believed that the use of the guidelines suggested in this TP will greatly help to assure the completeness and correctness of Monte Carlo analyses, and will ensure against errors, omissions, and fallacious assumptions which may later impact the mission success.

This TP provides a needed overview of the application of the Monte Carlo method with a particular focus on probabilistic launch vehicle design and requirements verification. Although it employs several launch vehicle trajectory analysis applications as illustrative examples, the general Monte Carlo approach and methodology outlined in this TP can be extended to other complex NASA space and aeronautic system problems. Several important aspects of properly formulating, and determining the necessary number of Monte Carlo samples, for probabilistic requirements verification are reviewed in this TP. Also, this TP identifies several types of uncertainties and describes how to handle different types of uncertainties in the development of vehicle models. Engineering-level explanations of the fundamental steps in the Monte Carlo process are provided while the supporting detailed mathematical analysis is deferred to the appendices, where issues of consumer risk, multivariate optimization, and multiple extreme constraint optimization are discussed.

This TP was generated under the sponsorship of the NESC GN&C Technical Discipline team and it is envisioned that similar GN&C tutorial-type documents will be developed in the future under NESC sponsorship. Comments on the overall utility and value of this TP can be directed to the NASA GN&C Technical Fellow at NESC and suggestions for other needed GN&C tutorial-type guideline documents are also solicited.

Neil Dennehy
NASA GN&C Technical Fellow
July 2010

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ACRONYMS AND SYMBOLS

BDM	booster deceleration motor
CDF	cumulative distribution function
CR	consumer risk
DOF	degrees of freedom
FPR	flight performance reserve
GRAM	Global Reference Atmosphere Model
LH_2	liquid hydrogen
LO_2	liquid oxygen
OC	operating characteristic
1D	one dimensional
PDF	probability density function
PMBT	propellant mean bulk temperature
PR	producer risk
RCS	reaction control system
RNG	random number generator
RSRMV	five-segment reusable solid rocket motor
RSS	root-sum-square
RUP	remaining usable propellant
SRB	solid rocket booster
SRM	solid rocket motor

LIST OF ACRONYMS AND SYMBOLS (Continued)

TVC	thrust vector control
2D	two dimensional
USE	upper stage engine
USM	ullage settling motor

NOMENCLATURE

a	mean offset of dispersed stage separation clearance data
cov	covariance
erf	Gaussian error function
$F_{(m)}(x/N)$	cumulative distribution function of the mth order statistic for a variable x given N samples or observations
$F(x)$	cumulative distribution function of the probability distribution function $f(x)$
$f(x)$	probability density function
G, $G(x)$	cumulative distribution of a probability function $g(x)$;
$g(x)$	generic probability density function
I_{sp}	specific impulse
k	maximum allowable number of failures
N	number of random samples
P	true or actual probability
P_{BIN}	binomial probability
$P_{\text{BIN}}(k/p,N)$	binomial probability density function for k failures where p is the probability of failure and N is the number of samples
P_{coll}	probability of collision
P_{fail}	actual failure probability
P_{success}	actual success probability
$P(x,y)$	probability density function of the random variables x and y
$\hat{p}$	maximum likelihood estimate of the actual failure rate
p_A, p_C, P_{upper}	consumer-acceptable failure probability
p_B, p_P, P_{lower}	producer-allowable failure probability
p_L	failure probability
$p(\alpha_i)$	joint probability density of the α_i
$Q\alpha$	dynamic pressure times angle of attack

NOMENCLATURE (Continued)

$Q\alpha$-total	dynamic pressure times the root-sum-square of the angles of attack and sideslip
$Q\beta$	dynamic pressure times sideslip
R	recontact radius, equal to the nominal clearance
r	radial coordinate
run	a Monte Carlo simulation consisting of N samples
sample	an individual random choice within a Monte Carlo run
sgn	sign (signum) function
var	variance
$\bar{x}_i, \bar{y}_i$	mean of observed data
$x_{(m)}$	order statistic
Z	normal probability parameter for an experimental distribution
z	target value for an output variable for an extreme vehicle model
α	probability of type I (producer risk) error
α_i	m stochastic input parameter
β	probability of type II (consumer risk) error
β_L	acceptable risk for the producer
β_U	acceptable risk for the consumer
$\Phi(x)$	cumulative distribution function for the standard normal distribution
δ_{PL}	uncertainty in the failure rate
$\Theta(x)$	step function
θ	polar angle
μ	mean of a probability distribution
ρ	correlation coefficient
σ	standard deviation
σ/R	radial uncertainty
χ^2	a type of distribution; standard calculation for variations in the output variables as a function of the input variations

TECHNICAL PUBLICATION

APPLYING MONTE CARLO SIMULATION TO LAUNCH VEHICLE DESIGN AND REQUIREMENTS ANALYSIS

1. INTRODUCTION TO MONTE CARLO SIMULATION AND APPLICATION TO LAUNCH VEHICLES

Broadly speaking, the term 'Monte Carlo' in the field of numerical analysis applies to any of a number of techniques for using random numbers in computation. It has been known since at least the time of the Comte de Buffon (1707–1788)—the author of *Buffon's Needle Problem*—that random numbers can be used to achieve a variety of numerical results, most famously as a substitute for uniform quadrature (covering every value of each parameter) in the evaluation of definite integrals.[1]

Monte Carlo can be used for generating results for a process that includes some probabilistic nature and is too complicated to evaluate directly. The name refers to the famed gambling resort in the Principality of Monaco, and is credited to Nicholas Metropolis.

The following is an illustration of the use of a Monte Carlo method. In this case, there is an explicit solution to the problem, but it is a useful example. Suppose one wants to find the probability that, out of a group of 30 people, at least 2 people share a birthday. It is a classic problem in probability, with a surprisingly large answer.

The probability that there are no shared birthdays can be determined explicitly by the following reasoning (leap years are ignored):

- The first person can have any birthday, and there is a 100% chance of no shared birthdays.
- The second person has one chance of overlapping with the first person, so there is a 364/365 chance of placing him/her without an overlap. The probability of no shared birthdays is 364/365.
- The third person has two chances of overlapping with the first two people, so there is a 363/365 chance of placing him/her without overlaps (2 days are taken). The probability of no shared birthdays is now (364/365)×(363/365).
- The fourth person has three chances of overlapping with the first three people, so there is a 362/365 chance of placing him/her without overlaps. The probability of no shared birthdays is now (364/365)×(363/365)×(362/365).
- . . .

• The 30th person has 29 chances of overlapping with another, so there is a 336/365 chance of having no overlaps. The probability of having no shared birthdays is now (364/365)×(363/365)×(362/365) ×…×(336/365).

In general, for a group of m people, the probability (p) of a shared birthday is

$$p(m) = 1 - \frac{365!}{365^m (365-m)!} \ . \tag{1}$$

The overall probability of no overlapping birthdays is then 29%, giving a 71% chance that at least one pair of people have the same birthday.

However, suppose the explicit solution were not available. The Monte Carlo approach is simply to pick 30 birthdays randomly, check for duplications, and repeat:

(1) Pick 30 random numbers k_i in the range [1,365].
(2) Check to see if any of the k_i are equal. Count '1' if any two or more are equal, '0' otherwise.
(3) Go back to step (1) and repeat, say, $N = 10{,}000$ times.
(4) Report the fraction of trials that have matching birthdays.

Formally, the Monte Carlo approach is the stochastic evaluation of a multiple sum:

$$p(m) = \frac{1}{365^m} \sum_{k_1=1}^{365} \sum_{k_2=1}^{365} \sum_{k_3=1}^{365} \cdots \sum_{k_m=1}^{365} \delta(k_i) \ , \tag{2}$$

where

$$\delta(k_i) = \begin{cases} 0 & \text{if all } k_i \text{ are different} \\ 1 & \text{if any two } k_i \text{ are the same.} \end{cases}$$

The formula above gives the exact answer to the probability question by evaluating all possible combinations.

Explicit evaluation of the multiple sum for $m = 30$ would require checking $365^{30} \approx 10^{77}$ terms (again, setting aside the explicit result above). The Monte Carlo method can give a good estimate for the sum in astronomically fewer terms. However, note that the Monte Carlo answer is not exact; instead, the estimate of the mean is reported along with a standard error of the mean, which is simply the standard deviation of the individual evaluations (zero or 1) divided by the square root of N.

This last observation is the key to the utility of the Monte Carlo method. In this example, the Monte Carlo algorithm is sampling over a thirty-dimensional space. Yet, the accuracy of the method always increases as $\sqrt{N}$; i.e., the uncertainty goes as $1/\sqrt{N}$. To double the accuracy, four times as

many samples are needed. Contrast this to, say, an integral over 30 dimensions performed using uniform quadrature; e.g., the trapezoidal rule. In general, doubling the accuracy of quadrature in m dimensions requires 2^m times as many evaluations of the integrand (like the pairing function $\delta(k_i)$) – and $2^{30} \approx 10^9$.

In the birthday problem, the answer is available without Monte Carlo simulation, but in common aerospace engineering problems, it is not.

The modern Monte Carlo method gained its name and its first major use in the 1940s, in the research work to develop the first atomic bomb.[2] The scientists working on the Manhattan Project had intractably difficult equations to solve in order to calculate the probability with which a neutron from one fissioning uranium atom would cause another to fission. The equations were complicated because they had to mirror the complicated geometry of the actual bomb, and the answer had to be right because, if the first test failed, it would be months before there was enough uranium for another attempt.

The problem was solved by realizing that they could follow the trajectories of individual neutrons, one at a time, using teams of humans implementing the calculation with mechanical calculators.[3] At each step, they could compute the probabilities that a neutron was absorbed, that it escaped from the bomb, or that it started another fission reaction. They would pick random numbers, and, with the appropriate probabilities at each step, stop their simulated neutron or start new chains from the fission reaction.

The key insight was that the simulated trajectories would have identical statistical properties to the real neutron trajectories so that they could compute reliable answers for the important question, which was the probability that a neutron would cause another fission reaction. All they had to do was simulate enough trajectories.[4]

1.1 A Launch Vehicle Example

An aerospace example of this need is the stage separation of a launch vehicle. There are many components of this problem that have uncertainty, including influences like the aerodynamic forces and torques, the forces and directions from rocket motors used to separate the stages, forces from the remaining thrust of the stage being jettisoned, rotation rate of the vehicle, and possible failures of some of the components needed for separation. The requirement is that the stages separate cleanly, without any components coming into contact after the initial pyrotechnic separation. Typically, an engine nozzle must be cleared.

One possible approach to simulating this situation is to apply worst-on-worst effects (choose an extreme value for every input parameter and apply these worst values all at the same time), and if there is still clearance, nothing else is necessary. But what if there is not so much margin that a worst-on-worst approach works? Also using worst-on-worst assumes that the extreme parameter choices lead to the worst results, for all outputs of interest. Another standard approach is to evaluate the effect of each uncertain variation individually and to do a root-sum-square (RSS) evaluation of the overall effect. Performing an RSS evaluation assumes the effects are all linear and independent.

It ignores potentially important dependencies between the parameters. The Monte Carlo approach randomly varies each of the input quantities according to their input statistics and computes a large number of outcomes through a large number of individual simulations. Figure 1 shows this process pictorially.

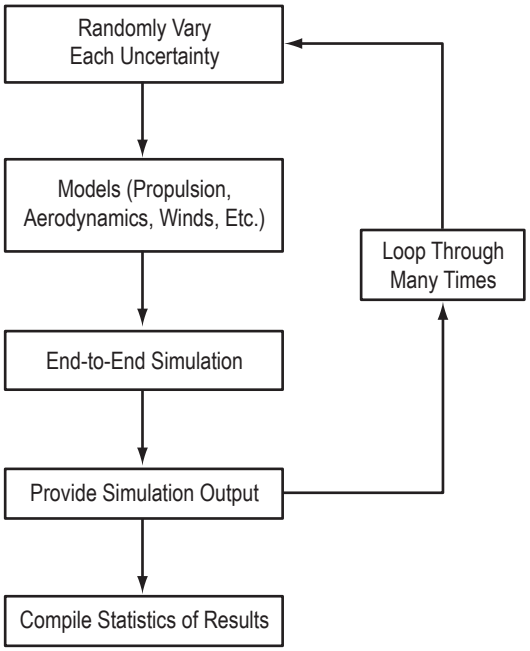

Figure 1. Monte Carlo process.

The results of the Monte Carlo calculation (for stage separation) are a recontact probability and an estimate of the uncertainty. Again, this estimate can be thought of in terms of a multidimensional integral:

$$P(\alpha_i) = \int p_1(\alpha_1)\, p_2(\alpha_2) \ldots p_m(\alpha_m)\theta(\alpha_i)\, d^m\alpha \ , \tag{3}$$

where $p_i(\alpha_i)d\alpha_i$ is the probability density for the ith input and $\theta(\alpha_i)$ is 1 if the combination of inputs results in a recontact, and zero otherwise. Because the determination of a recontact condition is a complicated result of a time-dependent dynamic simulation, the integral usually has no closed-form solution. And because the number of inputs $m \gg 1$, Monte Carlo is favored over uniform quadrature by a large factor.

The computation of recontact probability for stage separation (and probabilities for any two-dimensional problem) is discussed in section 4 and in detail in appendix A.

1.2 Trajectory Analysis

Monte Carlo simulation is used extensively for launch vehicle trajectory analysis. One of the principal purposes is to supply data for the probabilistic design of vehicle components and elements. As such, the role of the flight dynamics Monte Carlo simulation is not necessarily to provide

a probability-weighted integral of some function, as is the case for staging recontact (a single parameter), but rather to provide a set of trajectory ensembles with which other 'customer' engineering groups can assess and optimize their designs. For example, the design of the reaction control system (RCS) may depend on the probabilistic distribution of RCS propellant usage, so that the tanks are sized to cover 99.865% of the potential operating envelope, or the trajectory data can be input to a model that produces a load indicator for the vehicle, against which the structural design is measured. So in this sense, the product is more open-ended than a conventional Monte Carlo integral, where the output of interest is specified a priori. Instead, the priority is to generate an ensemble of trajectories that accurately reflects all the relevant stochastic uncertainties.

Monte Carlo simulation applied to trajectory analysis is a method of generating a suitably sized collection of trajectories via random selection of input variations. It can be used whenever inputs or processes are not completely deterministic (which they never are), when the outcome is too complicated to calculate easily, and when a simple worst-on-worst approach (choosing the extreme value of each input parameter and showing success for the combination of these worst values) is either too difficult or too expensive to achieve. It is heavily dependent upon correct uncertainties in the inputs (inputs that correctly model reality) and upon correct modeling of the system.

One of the most common pitfalls is trusting the results when the simulation has not been thoroughly tested or when the expected results are not known. It is easy to do some testing and think that the results are correct, but it is very easy to miss some input or some effect and to get results that are not correct. So it is imperative that there be an expectation for what should result from the simulation, with physical reasons to back up the effects that are seen. Prior to running the Monte Carlo simulation, nominal cases should be run and checked with independent analysis. Monte Carlo simulations can be run with one input uncertainty modified to see that the change in results is as expected. Deterministic simulations with one parameter modified can be compared to the Monte Carlo results to see that the effects are as expected. If the results are not as expected, the physical cause for the difference should be studied. Testing will be discussed in more detail in section 3.

Early on in a program, the subsystem uncertainties will not be well known. If these are underestimated (the modeled uncertainty is less than the true uncertainty), it could result in a design that is not flyable. It is critical that sufficient uncertainty be included early in a program. This is discussed more in section 2.

1.3 Success Percentages and Consumer Risk (or Confidence)

Suppose 500 Monte Carlo samples are obtained, and suppose a 99% overall success is the target. Suppose five quantities are being measured that are all needed for success. If any of the necessary quantities fails, that should be considered an overall failure. If each is 99% successful, the overall success is only 95.1%. The overall success requirement means that the success for each critical quantity must be higher.

If a 99% success rate is achieved out of the 500 samples, how confident should one be that the actual success rate is 99%? Suppose the results are ordered from best to worst, using whatever measuring criteria is appropriate for success. If cases 496 through 500 fail, 99% have succeeded. Thus,

taking sample 495 as the design case might seem to capture the desired 99% result. But if a new set of 500 cases is run, using 500 new random numbers, the 495th point will change in value. How can one be confident that the 99% value is captured when only one Monte Carlo set has been run? If the 99% value is taken, it turns out the confidence (that the 99% success rate was captured) is <50%. This confidence is complementary to 'consumer risk'; i.e., the risk that the customer thinks the product is 99% successful when in fact it is <99% successful. Consumer risk is 10% if the confidence is 90%. Suppose one wants to be 90% confident (10% consumer risk) that the 99% value is captured. That is, if 10,000 sets of 500 Monte Carlo samples each were run, the 99% value should fall below the one chosen 90% of the time. One thousand runs would have a higher value of the key parameter than the one specified.

Consumer risk is discussed in more detail in section 4 and appendix B. Numerical tables for running a single Monte Carlo set are given in section 4, more extensive tables are given in appendix C, and more detail on calculating these values appears in appendix B.[5] Some conservatism needs to be included in order to have 90% confidence, as described above. Of course, more conservatism needs to be added if the number of Monte Carlo samples is smaller, since the results are less accurate in representing the true results.

1.4 Parametric or Nonparametric? ("To fit or not to fit, that is the question…")

One of the basic decisions that must be made when manipulating statistical data—the Monte Carlo output—is whether to resort to assumptions about the type of the probability distribution; e.g., normal (Gaussian), log-normal, beta, uniform, Poisson, etc. If it is assumed that a particular variable has a Gaussian distribution, for example, that distribution can be parameterized with a mean and a standard deviation, and the values can be estimated ('fit') of those parameters with standard techniques. When specific probability distributions like this are dealt with, 'parametric statistics' are being used.

When using parametric statistics, due diligence requires that a significance test be applied to verify that they are indeed Gaussian (or log-normal, etc.). In the case of the Gaussian distribution, there are a number of tests available. Some are more sensitive to variations near the mean; others are more sensitive to the tail. Different tests have different 'statistical power,' which is the likelihood that they will reject the assumption of a Gaussian distribution when it is false. So the assumption of the type of distribution, the choice of significance test for that assumption, and the choice of significance level for that test all bring some additional uncertainty into the statistical analysis.

In some cases, it can be more appropriate to use nonparametric statistics; i.e., statistics that do not assume any particular form for the distribution of a random variable. 'Order statistics' are the most common type of nonparametric statistics used in trajectory analysis. Use of order statistics involves ordering the samples from lowest to highest (for the output parameter of interest) and examining the statistics of the tails, regarding values above some threshold as failures and values below the threshold as successes. It turns out, as discussed in detail in appendix B, that conclusions can be drawn from order statistics without assuming a particular probability distribution. It also turns out that the results are not dependent on the number of uncertain inputs, which is very beneficial when deciding how many runs are necessary.

It is difficult to formulate exact criteria for deciding whether to use parametric or nonparametric statistics. There are advantages and disadvantages to using each, and the judgment of the analyst is often required. Some of the questions that should be asked when making a choice of approach are as follows:

- Does the selected approach have to apply to a wide variety of random variables (favors order statistics)?

- Are there good reasons to believe the variables have a particular distribution; e.g., are they linear superpositions of many contributing effects, so that the central limit theorem applies?

- Will order statistics suffice to answer questions about the variable, or will more complicated manipulation of the output be required?

- Is the variable in question one-dimensional, or is it characterized by a probability density function (PDF) in a higher dimensional space? Are those extra dimensions necessary to quantify the variable, or is it satisfactory to reduce the problem to one dimension? (More dimensions usually means a fit to a distribution is desirable.)

- Is it practical to obtain enough trials in a sample so that order statistics can reasonably answer the question being asked? (Order statistics requires many runs in order to reduce the level of conservatism required, since only the tails are used.)

- Conversely, parametric statistics allow us to use all the information in a sample, and not just the top few trials, so any estimates derived using parametric statistics typically have a much higher imputed accuracy (roughly by a factor of $\sqrt{N}$). Is this increase in accuracy beneficial enough to outweigh the uncertainties associated with the parametric assumption?

- If the desired probability level is of necessity outside the range of data (for example, if a 1×10^{-6} level is desired but no more than 2,000 samples are possible), a parametric fit is a necessity. On the other hand, if there are data points at the percentage levels of interest, and these points are in the tail of the distribution, a parametric fit might not capture this region well.

Many of these questions come up, and are answered, in the discussions below. For example, trajectory Monte Carlo typically generates hundreds of output parameters, which makes testing each output for normality unwieldy (and some outputs are notably not Gaussian; e.g., the maximum dynamic pressure for a trajectory follows an extreme value distribution). Some analyses, such as the estimation of propellant reserve, or staging recontact analysis, are inherently two dimensional. In the case of staging recontact analysis, where the desired output is either a probability of recontact or a circular drift radius, the higher accuracy of parametric statistics is attractive.

1.5 Summary of Remaining Sections

Section 2 covers discussion of the input distributions for Monte Carlo runs, and gives some examples of what can be done to get these distributions correct. The Monte Carlo results represent

the truth as to how a system will behave only if the system models and the uncertainties sufficiently represent the truth. Any underestimate of the input variations (modeling smaller variations than the actual vehicle will experience) will usually lead to an underestimate of the output variations (unless the particular inputs are insignificant in their impact on the output).

Section 3 covers testing the simulation, and gives some examples of how this can be done.

Section 4 covers a number of topics, including how many runs are necessary and what to do if results are desired for events that happen only rarely, such as failure scenarios or severe wind gusts.

Section 5 covers postprocessing, including analyzing any failed runs, examples of useful output products, and statistical information for generating desired results from the output data.

1.6 Definitions

Confidence: Normally applied to the consumer, as the confidence that the design actually meets the requirement when it is believed to. A 90% confidence is the same as a 10% consumer risk.

Consumer risk: The risk of thinking the requirement is met (the design satisfies the need) when in fact it is not. To keep this risk low, the system is typically overdesigned and has a high producer risk.

Order statistics: Ordering the Monte Carlo results from lowest to highest (for the parameter of interest) and evaluating the success or failure based on the values in the tails (percentages that exceed some desired threshold).

Parametric statistics: Choosing a distribution type for the output statistical data and then fitting the data to the distribution type, along with testing, to show that the type of distribution is reasonable.

Producer risk: The risk of thinking the requirement is not met when in fact it is (and thus rejecting a good product).

Uniform quadrature: Integrating over all values of all parameters to obtain the result—an alternative to Monte Carlo simulation.

Worst on worst: Choosing an extreme value for each input parameter and simulating with all these extreme values operating together. Assumes that the extreme values lead to the worst results (that some combination of nonextreme values cannot be worse).

2. INPUT DISTRIBUTIONS: WHAT UNCERTAINTIES GO INTO THE MONTE CARLO SIMULATION?

It is imperative that the physics models in the simulation have the appropriate level of fidelity, so that all effects important to the outcome be included. It is also imperative that the input uncertainties are modeled correctly. If either of these are not right, then the output results will also be wrong.

Usually, for new systems, there is some historical data available for old systems with which to estimate the uncertainties for the new system. It is very important to be conservative at the beginning, choosing an uncertainty that is at least large enough, not just a best guess as to the uncertainty. This is because, once a system design is set in stone, it is typically too late to fix it if the uncertainties used turn out to be too small. For example, if the engine efficiency is not varied enough in the simulation, low-performing engines will be missed and ultimately the system will not be able to meet its requirements. Depending on the expense involved with missing poor performing engines, this could be a showstopper. As the engine design matures and the uncertainty becomes better known, the values of the input uncertainties can be refined. Of course, being overconservative is no virtue since it leads to increased costs and possibly infeasible designs as well. The right level of conservatism is an always-present issue.

2.1 Epistemic and Aleatory Uncertainties

In the last few decades, it has become more common in systems engineering to discriminate between different types of uncertainty.[6] Some parameters are believed to have precise, but currently unknown, values, and these parameters are said to be subject to 'epistemic' (knowledge) uncertainties. Other parameters will always show stochastic variability, and these parameters are said to have 'aleatory' (luck) uncertainties. A typical epistemic variable is drag coefficient, which is believed to have a fixed value for a given flight condition, a value that is only crudely approximated by simulations and scale model testing. Another example is a specific engine's performance, which has manufacturing uncertainty but will be a specific value for each engine that comes off the line. If the engines are tested prior to flight, estimates of these values may be available prior to launch, but are not available during vehicle design. A clear aleatory variable is wind gust, for which enough information will never be accumulated to make a deterministic calculation, and thus will always appear as a random variable in trajectory simulations (although the part of the wind variation that can be measured on launch day would be considered epistemic uncertainty for evaluation before it is measured). Besides uncertainty in the input parameters, epistemic uncertainty can also include uncertainty in the model dynamics.

It is a question of active research how best to account for epistemic uncertainties in systems design.[7] Since there is a true value (unknown), lumping these uncertainties in with the ones that randomly vary for each flight would yield different results than if certain values were chosen for the knowledge uncertainties (to represent the potential true values) and the luck uncertainties were varied separately.

The approach taken here represents a compromise between the two extremes of ignoring the differences in the uncertainties and running a separate Monte Carlo simulation for all possible combinations of epistemic uncertainty that might possibly be the real combination (which would be prohibitive computationally). The recommended practice is to pick specific combinations of epistemic parameters that represent challenging vehicle models and generate Monte Carlo ensembles around those vehicles. These could be vehicle models that stress performance (a model with low propulsion performance values, high drag, and high masses), structural and aerothermal loads (a model with high propulsion performance, low drag, and low masses), or flight control (challenging aerodynamic, slosh, and flexible body parameters). In the case of flight control, results are so nonlinear that besides examining challenging vehicle models, the epistemic and aleatory parameters should still be lumped into one Monte Carlo simulation in case certain combinations prove to be worse than expected. Mission choice (target orbit and payload, for example) and flight month represent epistemic dimensions for vehicle performance, so these would be discretely varied. Covering the primary epistemic parameters is believed to offer a satisfactory trade between uncertainties and vehicle robustness.

Investigating the sensitivity of the vehicle design to the parameter in question, and determining where the vehicle 'breaks' or fails to satisfy requirements, is also recommended as a general practice. This is done by varying specific parameters in single simulations to find out how sensitive the vehicle is to those parameters. This method provides an analytical limit against which future refinements of those parameter uncertainties can be gauged. For example, suppose the rolling moment coefficient has some nominal value derived from analysis and test, but that some experts believe that the nominal value could be as much as 20% different from the true value. Further, assume that the vehicle configuration meets requirements with the nominal value. How to handle the epistemic uncertainty remains a choice. A vehicle configuration showing a tolerance of 20% higher rolling moment coefficients could be required, or failing that, require the vehicle be modified to meet that higher requirement. Or the nominally designed vehicle could be taken and find out what increase in rolling moment coefficient could be tolerated without losing roll control of the vehicle. The margin thus demonstrated is a measure of the importance of resolving that particular epistemic uncertainty. A 2% margin would have different implications than a 15% or 25% margin, but would not force a modification to the vehicle to meet the subjective 20% margin. Factored into the importance of resolving each uncertainty more clearly would be a measure of its impact on mission safety, mission success, performance, and cost.

In addition to the distinction between epistemic and aleatory uncertainties, the current practice for trajectory Monte Carlo takes into account the distinction between current epistemic uncertainties that can be used to determine the payload capability, epistemic uncertainties that are known by the time of flight, and can be used for the go/no go decision, and flight-day uncertainties. The next section delves into this aspect of simulation.

2.2 Flight-Day Uncertainties

For a launch vehicle design, propulsion parameters are not known that well at the start of the design effort. They will be known much better by the time the vehicle is launched, but there will still be uncertainty on launch day. The design uncertainties will be gone by the time of launch and what

remains is flight day uncertainty. Also, variations in the propulsion parameters for the ensemble of engines may be known better for a particular engine if it is tested prior to flight. The trajectory will be designed for the known variations. The flight performance reserve (FPR), the extra propellant needed onboard to cover for uncertainties, should be designed to cover for flight day uncertainties, not for all the uncertainty that exists early in design. Likewise, the accuracy of the orbit insertion, the structural and aerothermal loads, and other parameters, will be variations from the trajectory that include the uncertainties remaining during flight and not the full set of early design uncertainties. The parameters known better for a specific vehicle on flight day are epistemic parameters, in that they have a true value for the vehicle to be flown but that value is unknown during design.

A way to handle this is to come up with multiple system models. For example, a sluggish ('heavy, slow') and a sporty ('light, fast') launch vehicle model is chosen, covering the range of the design uncertainties (and other uncertainties that are known better for a particular launch vehicle before it flies). Next, trajectories are designed for how these would be flown. Finally, the Monte Carlo simulation is run for each, using the estimated uncertainties remaining on flight day. If the Monte Carlo simulation had all uncertainties included as if they were all unknown, then a 99.865% value (with 10% consumer risk) from this simulation would not cover the 99.865% value for the sluggish or sporty vehicle models; their success might be much lower.

Suppose this process is used and it results in a vehicle model that assumes a known engine, weighed components, and a specific value of the axial force coefficient at each flight condition. (Early in design, the sluggish and sporty models are used to cover the range of cases.) The trajectory is designed with this model in mind, and the payload can be adjusted, if desired. On flight day, the winds are normally measured to help with the go/no go decision. The temperature is also known (particularly important for solid rocket motor (SRM) performance). Most of the wind variation and temperature variation becomes a known quantity and is no longer a flight day unknown. These variations can be used for the trajectory design if day of launch trajectory design is used. The FPR, that part of the propellant set aside to cover for the flight day uncertainties, can be a bit less since these variations are no longer unknown. While the vehicle is still in the design phases, the practice (for payload performance studies) is to postulate a sluggish vehicle, then to determine the expected performance for the vehicle with challenging winds and temperatures, and then to show through the Monte Carlo simulation that the FPR is sufficient to achieve orbit. The sluggish vehicle model choice represents the manifesting likelihood (probability of being able to launch a vehicle that has been assembled). The challenging winds and temperatures represent the ability to launch in a certain percentage of natural environments. The result of the Monte Carlo simulation represents the probability of being able to take the sluggish vehicle on the challenging day and get it to orbit.

Figure 2 shows the effect of lumping all parameters into one Monte Carlo simulation (blue) as compared to running the flight day uncertainties (green) for a vehicle model (red). In this example, the distributions of both are assumed to be the same. The figure assumes that a 3σ vehicle model must be able to fly successfully. The effect is significant, demonstrating the need to break the parameter out separately.

The sluggish and sporty vehicle models might have to hit multiple parameter targets, if an unrealistically large number of Monte Carlo runs are to be avoided. Suppose, for example, that

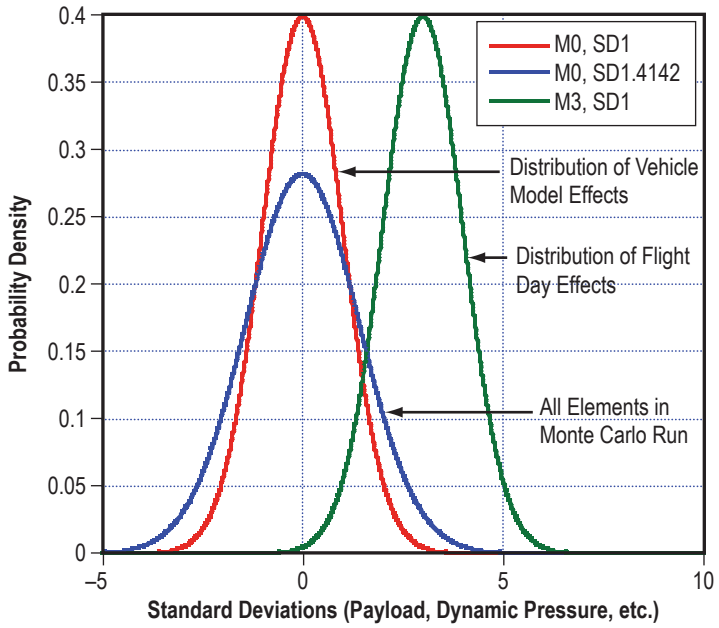

Figure 2. Normal distribution probability densities (x-axis is the sigma level). The red and green curves are assumed to have the same standard deviations.

the launch vehicle has to launch at any time of the year, has multiple missions it will fly, and for the sporty version, the desire is to cover for 99% of vehicle models in terms of time to clear the launch tower, maximum dynamic pressure, maximum acceleration, and maximum heat rate—there is a 1% chance that an even sportier version could be the end product. Launch months and mission models that yield higher values of these parameters can be chosen, rather than running all combinations. As to the vehicle model, appendix D discusses ways to design the vehicle model so that it hits the multiple desired 99% parameters. If an actual vehicle fell in the 1% of cases that exceed these values, a decision could be made to launch that vehicle at a different time of year or swap out some of the components. Of course, these options are unavailable for one-of-a-kind spacecraft.

There is a distinction here between use of epistemic parameters for performance and loads analysis (where the consequence of not capturing the case means having to swap out a vehicle component) and analysis of epistemic parameters for flight control (where the consequence of not capturing the case means the vehicle may lose control). For the flight control models (discussed in sec. 2.1), being able to fly with whatever combination of actual aerodynamic, slosh, and flexible body parameters the actual vehicle ends up with is a necessity, because changing vehicle components will not, in general, help much. On the other hand, most of the important flight control parameters are epistemic, with the primary exception of winds, so that lumping all the parameters into a Monte Carlo simulation provides a good way to sample the various possibilities. It does not cover for a possible bad combination of epistemic parameters, which is why, as discussed earlier, sensitivity analysis is so important, and why it is recommended that a flight control-challenging case be examined separately.

2.3 Types of Distributions

There are many types of possible input distributions for the uncertainties. Some common possibilities are listed below.

2.3.1 Normal (Gaussian)

Normal (Gaussian) is a bell-shaped curve that is typically used when the system is understood to provide more likely values near the mean and decreasing likelihood further away from the mean (fig. 2). The PDF follows the form

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right) , \tag{4}$$

where μ is the mean of the distribution and σ is the standard deviation. When referring to a 3σ result, a Normal distribution is typically being assumed, and specific probabilities are being referred to.

2.3.2 Triangle

The triangle contains a peak value and two end values (fig. 3). The triangle provides for more data points closer to the end values and for a mean that is not in the middle, but contains no data points outside the end values.

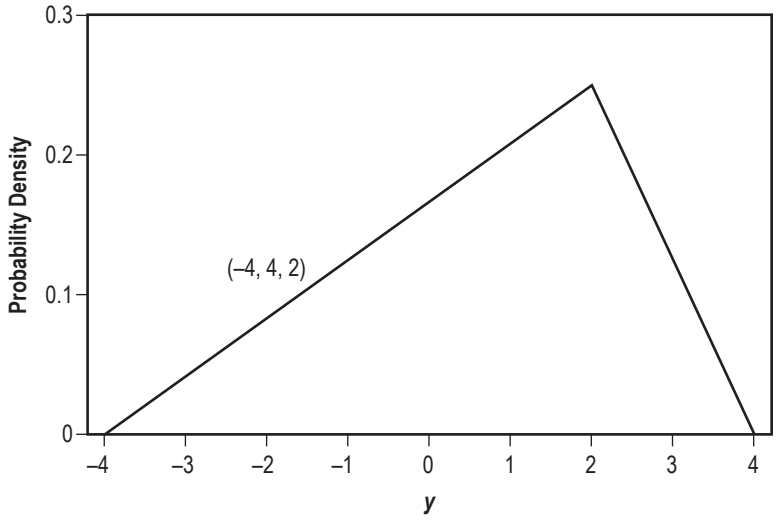

Figure 3. Probability density for a triangular distribution between –4 and 4 with the peak at 2.[8]

2.3.3 Uniform

Uniform is defined on an interval (fig. 4), with each value within the interval being equally likely, and no values allowed outside the interval. This distribution is typically conservative in the sense that values near the edge are considered just as likely as values in the center, even though most engineering systems have parameters that are more likely to be near the mean value.

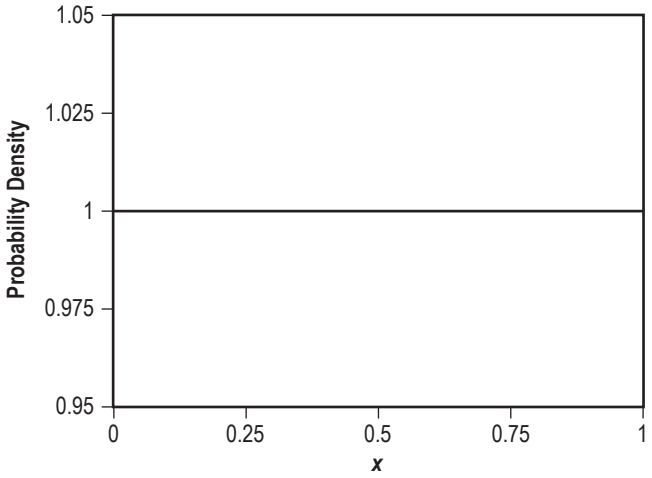

Figure 4. Uniform distribution probability density (variable is equally likely for values between zero and 1).

2.3.4 Lognormal

In lognormal, values less than zero are not possible, and the likelihood falls off further away from zero (fig. 5).

Many other distribution types are available depending on the behavior that is desired. Another possibility is that the input variations will be given by a model of the specific system of interest. For example, the Global Reference Atmosphere Model (GRAM) provides random wind profiles that are correlated with altitude and location.[10] In this case, rather than specifying the distribution, the user simply calls the model and receives the data.

2.4 Uncertainty Examples

Usually, when modeling the interaction of subsystems in a Monte Carlo simulation, the person conducting the simulation is not the same as the expert on each subsystem. So, in general, the uncertainties and distributions should come from the subsystem experts. Should a Gaussian (normal) distribution, a uniform distribution, or some other distribution be used? This may be known if the subsystems are known well enough; if not, a uniform distribution is typically a conservative choice, giving equal likelihood to values in the middle of the distribution as to the edges. The subsystem experts would also provide any information about correlation between the uncertain parameters, for any that are not independent.

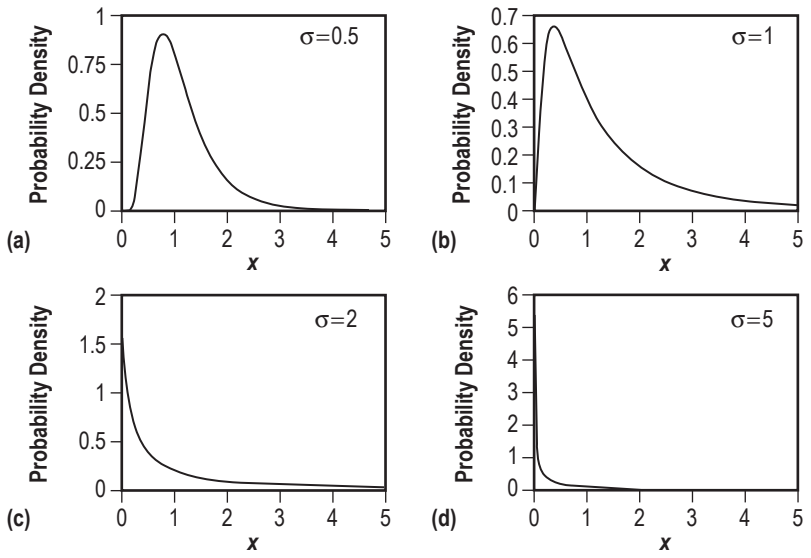

Figure 5. Probability density for a lognormal distribution: (a) $\sigma = 0.5$, (b) $\sigma = 1$, (c) $\sigma = 2$, and (d) $\sigma = 5$.[9]

If there is a bound to the input parameter's extreme values, it cannot be a true Gaussian distribution. This is immaterial if the standard deviation is small relative to the distance to the boundary. In practice, the supplier of the uncertainty for a particular input might want a Gaussian to be used but might specify that values beyond a certain boundary be dropped.

It is, of course, most important that the uncertainties that affect the outcome the most are known well, or have sufficient conservatism included. Adequate resources must be invested to develop and refine the most important uncertainties. Running single simulations that vary individual parameters will show which are the most sensitive (although this approach assumes that the sensitivity does not change if other parameters move away from their nominal values, which is not always the case). Computing the correlation between important output parameters (from a Monte Carlo simulation) and the input values is another way to indicate which ones most affect the outputs. Doing this also provides a means for making sure the results are as expected.

Following are three examples of uncertainty inputs for a launch vehicle simulation, for aerodynamics, propulsion, and winds aloft modeling:

2.4.1 Aerodynamics

There are a number of contributing causes for aerodynamic uncertainty. Some of these are wind tunnel or computational fluid dynamics accuracy, Reynolds number effects, interpolation effects (data are unavailable for every possible condition), and others. These uncertainties can combine in different ways for the different aerodynamic coefficients, and may be different at each flight condition (each Mach number, angle of attack, and sideslip, for example). The fidelity of the data may be different at different flight conditions due to the expense involved in wind tunnel tests. For an example

case of modeling the Ares I launch vehicle, some of the uncertainties are modeled as uniform distributions because it is unknown where in the uncertainty range they might be, whereas others are modeled as Gaussian distributions because it is considered more likely they will be close to the nominal (mean) values and less likely as the value gets further from the mean. It is also possible to model the aerodynamics uncertainty as an interval (min, max) with no associated probability distribution, based on the argument that these uncertainties are epistemic and other requirements must be met with worst combinations of coefficients. This worst-on-worst approach is not generally feasible for a launch vehicle design, but reinforces the importance of sensitivity analysis.

The aerodynamics community studied the various uncertainty components and developed a table of uncertainties for each Mach number, each angle of attack, and each vehicle roll angle (angle of incidence of the relative wind). The table is different for each aerodynamic coefficient and is also different for design uncertainty versus flight day uncertainty. The design uncertainty for the axial force coefficient is included for the sluggish and sporty vehicle models, with the rest of the axial force coefficient uncertainty modeled as flight day uncertainty. The overall vehicle performance is not affected by the other aerodynamic force and moment coefficients, but the flight control must be able to successfully fly anything in the full design range for these coefficients. For this reason, for all coefficients except for the axial force coefficient, the entire design uncertainty is included in each Monte Carlo simulation.

2.4.2 Propulsion

Various levels of propulsion system models are possible. For propulsion systems that are well known, the engine community may furnish a detailed model that includes uncertainties. In this case, calls to the model with a random seed will yield an appropriately varying propulsion system performance so that, overall, the correct statistics and correlations are modeled. There is a detailed model for the shuttle solid rocket boosters (SRBs), for example, that models not only overall performance uncertainties, but variations during flight and variations in the behavior of the thrust tailoff.

Earlier in a program, or if the detail described above is not needed, simpler propulsion models may be used. For example, an engine may be modeled with a nominal thrust, specific impulse (efficiency), and mixture ratio (ratio of oxidizer to fuel), along with uncertainties in each of these parameters. These uncertainties would be larger when the engine is being designed than after it has been tested and is ready to fly. In the case of the J-2X powering stages on the Ares launch vehicles, each engine would be put on a test stand and ignited prior to launch. So each engine has measured values (called 'tag' values) of thrust, specific impulse, and mixture ratio that can be incorporated into the sluggish or sporty vehicle models as described above. Measurement uncertainty of the ground tests with respect to flight and 'run-to-run' variation for each engine would be modeled as flight day uncertainty in the Monte Carlo simulation. The tag values and run-to-run variations may have distributions that are uniform or Gaussian, depending on the maturity of the design.

2.4.3 Winds Aloft

The winds in the altitude range where jet aircraft fly are also very important to launch vehicle ascent success. There is a substantial database of winds aloft for this altitude range, for many points

on the globe, using various data sources. Winds are correlated from one altitude to the next, and there is a substantial database for this correlation as well. For example, thousands of balloons have been launched at Cape Canaveral and subsequently tracked to give wind profile statistics. There are time as well as spatial correlations between the winds. The data have been analyzed and a detailed model for winds (as well as density, temperature, and other parameters) has been incorporated into the GRAM.[10] Using this model as a subroutine, with a random seed input, any desired number of random wind profiles can be obtained for use in the Monte Carlo simulation. So each time the launch is simulated, a different randomly correlated wind profile will result, such that over many runs, the statistics of the real wind data will be seen.

In all the cases above, the uncertainty models that are used come from the discipline experts for those systems. Usually the determination of the models to use also includes interaction between the flight mechanics engineers and the discipline experts so that an uncertainty model results that correctly reflects the system, and at the same time, fits the simulation and vehicle integration needs.

2.5 Sample Uncertainty

A sample of what an uncertainty table might look like is shown in table 1.

Table 1. Sample of uncertainty table.

Category	Parameter	Mean	Units	Dispersed Values (3σ)	Units	Distribution Type/Model
Five-segment SRB	PMBT* (winter/summer)	61/82	°F	11, 3	°F	Normal
	Burn rate	0.349	in/s	4.0000E-03	in/s	Normal
	I_{sp}	Derived from thrust trace	N/A	1.3000E+00	%	Normal
J-2X	Mixture ratio dispersion	5.3	N/A	5.4000E-01	%	Normal
	Thrust dispersion	296,000	lbf	8.1000E-01	%	Normal
	I_{sp}	450	s	4.8000E-01	%	Normal
Upper stage	LO_2 loaded	238,186.68	lbm	7.3000E-01	%	Normal
	LH_2 loaded	43,784.32	lbm	5.7000E-01	%	Normal
Aero	Aero coefficients (all, including base force)	Nominal coefficient data (tabular data)	N/A	See table vs Mach, aero angle	% of nominal	Uniform
Atmosphere	Winds	Mean wind profile for month of interest	N/A	N/A	N/A	Perturbed GRAM-07 for launch month

* Propellant mean bulk temperature.

3. ASSEMBLING AND TESTING THE SIMULATION

The trajectory simulation is assembled from models of how the physics of the various systems behaves during flight. In early design phases, these are simple models, and get more complicated as design proceeds. Each subsystem design discipline defines how to best model their systems. For example, the designers of an SRM will provide details on how thrust, propellant burn rate, and mass properties vary with time, as a function of environmental conditions such as ambient temperature, and how the subsystem will react in likely failure scenarios. Some of the models in the simulation will be code furnished by other discipline areas, and some will be developed by the simulation developer based on definitions from the other discipline areas. It is likely that for some of the models, the expertise resides in the same organization that is developing the simulation and would be developed 'in-house.' Testing each model for the correct behavior is an important part of verifying that the results of the overall simulation are correct.

It is critical that the simulation be thoroughly tested prior to trusting the answers; otherwise, incorrect conclusions are likely to be drawn that will lead to poor design decisions. The level of testing necessary will depend on the stage of the vehicle or spacecraft program and on its scope. For a large, expensive program, thorough verification of the simulation is necessary, and typically an independent analysis will be performed to identify discrepancies. Even early on during a program, and for smaller programs, testing is very important as results drive design decisions.

Testing takes time. Experience shows that testing and debugging typically takes significantly longer than running production analyses.

One specific test that can be conducted is to ensure that the distributions of the inputs, when randomly varied, match the expected distributions. This test can also be conducted on the outputs to see if they are normally distributed. A way to do this is to graph the experimental data. For example, if testing for normality (Gaussian distribution), the value of the parameter versus the sigma level of the parameter can be graphed, where the sigma level is taken from the percentages that would be true if the distribution were Gaussian. If the graph shows that the values fall along a straight line, then the assumption of normality is true, and if the values deviate significantly (particularly in the tails), then it is not Gaussian. There are also several quantitative tests for normality, such as the Anderson-Darling test, which provide numerical confidence estimates.

Specific testing procedures will be individual to a given project. Here are some tests that are regularly conducted on the Ares I launch vehicle simulation at the preliminary design level of the project. Implementers of individual subsystem models in the Monte Carlo simulation test these models to show that they are getting the expected results. This in itself can be a complicated process depending on the sophistication of the model. Vehicle models and nominal trajectories are implemented in the simulation in 3 degrees of freedom (DOF) (translational dynamics only, with ascent guidance flying the vehicle in simulation). The results are compared with the results of the independent trajectory design program and should match closely, including not only payload to orbit

but also the various trajectory parameters. Any discrepancies are investigated and solved. Next, the nominal trajectories are implemented in 6 DOF, including rotational dynamics and flight control, and the results are compared to the 3 DOF results. These should match closely, again mass to orbit and the various trajectory parameters. Any differences are investigated. Engineering judgment (and sometimes analysis) is used to decide what is due to the higher level of fidelity and what may be a simulation problem.

Monte Carlo simulations are then performed using the models from the last design cycle, and ideally the results are duplicated. This is not always realistic with the fidelity updates for the new iteration included, but any discrepancies must be understood. If resources allow for maintaining all the old models in the newer simulation, an exact duplication might be possible, followed by introducing each new model and understanding the differences. Experience shows that duplicating Monte Carlo runs is not that simple in that the user needs to make sure all the random variations for each input variable are duplicated. Any changes to the input variables might make this problematic. Appendix E shows a way to ensure that new runs duplicate the old variations (assuming the compiler, random number generator (RNG), or some other factor does not change).

Next, test Monte Carlo runs are performed for the latest vehicle models and the results are examined. Changes from the previous design cycle must be understood. If any of the key output results change significantly, the cause must be determined. All output parameters must be within ranges that make sense from a physics standpoint. If the expected values are not known, some individual runs can be made with a specific parameter modified to see what value is needed to generate the results obtained. Typically, for the Ares simulation, a list of discrepancies is identified and time is spent closing each one.

Using the correlation coefficients as defined in section 5, it is possible to determine the effect of each key input on each key output. Do the relative impacts make physical sense for all the parameters? For key parameters, another Monte Carlo run can be made with a change to an individual parameter. Does the change in output match the expected result? When there are failed runs or outliers in a Monte Carlo run, examining the sigma levels of the various input dispersions, does the result make sense? Any failed runs or significant outliers should always be investigated, even if a certain number of failures are allowed. It is often the case that a failure or outlier will point the way to either a simulation problem or something in the design that can be improved.

When the results of Monte Carlo simulations are being used for important design decisions, verifying the results using an independent simulation developed elsewhere is advisable since the impact of mistakes in the simulation could mean huge expenses if they are not caught until later in the program. Eventually, the simulation should be validated with flight data, for example, an early test flight that does part of the mission of the new vehicle, or a similar spacecraft that has flown in the past.

For many programs, independent simulations are used to improve confidence in the results. It will not be possible with an independent Monte Carlo simulation to exactly duplicate all the random choices. However, with an independent simulation, a number of cases can be run to show that with the same set of inputs, the resulting trajectories are nearly duplicated. The results of Monte Carlo

runs from both simulations can be compared to see if the results are as close as should be expected if the only differences are from the different randomizations. A detailed example of assessing whether two sets of Monte Carlo results are drawn from the same population is provided in appendix F, section F.10.

4. MAKING MONTE CARLO RUNS

This section discusses five topics: (1) How many samples are needed, (2) whether there are any differences in how Monte Carlo runs should be set up if the application is human space flight versus nonhuman space flight, (3) what to do when wanting to examine the statistics of events that are rare (e.g., certain failure events), (4) alternative sampling methods, and (5) RNGs.

4.1 How Many Monte Carlo Samples?

To be clear about nomenclature, a 'sample' will mean a random trajectory that is one of N in a Monte Carlo run. A 'run' is the collection of all N samples.

How does an engineer determine that a sufficient number of simulations has been performed in a particular Monte Carlo analysis? For the general case, with a general Monte Carlo algorithm, there is no way to estimate a priori how many samples are required to generate an answer to a desired precision. Instead an analyst must implement the algorithm, run sample calculations, observe the mean and variance of the output, and then use the aforementioned $1/\sqrt{N}$ scaling of the standard error to estimate the number of runs required.

For example, if one intends to use trajectory Monte Carlo to determine the impact location of a discarded first-stage motor, one might generate 1,000 Monte Carlo samples, and track the cumulative mean and variance of the impact coordinates. Then the observed variance 'var' is used to estimate the standard error of the mean, $\sqrt{\text{var} / N}$. If the uncertainty is, say, twice as large as the desired accuracy (accuracy of the standard error), then the number of runs must be quadrupled to attain the desired accuracy. In practice, the desired accuracy is itself not specified a priori, and so the diminishing returns of increased accuracy must be traded against the increasing expense of more Monte Carlo samples.

However, flight trajectory Monte Carlo is often used to estimate design limits such as maximum propellant usage, or maximum load indicators such as '$Q\alpha$-total' (dynamic pressure times the RSS of the angles of attack and sideslip). It turns out the 'order statistics' used in sampling theory; i.e., the theory underlying acceptance testing, dictate minimum numbers of runs for a given acceptable level of consumer risk. Order statistics[11] refers to ordering the Monte Carlo results from lowest to highest (for the parameter of interest) and evaluating the success or failure based on the values in the tails (percentages that exceed some desired threshold).

One item of particular interest in the order statistics approach is that the necessary number of samples does not depend on how many uncertain variables are varied. (This is not true if one is trying to determine the distribution of the output; it only applies to the use of order statistics.) This will be clear when the outcome is examined in terms of successes and failures. If the parameter of interest is the 99% value of a certain output, values above 99% may be viewed as failures and values below

99% as successes. So the question becomes one of determining the probability of success given the success criteria, or alternatively, the success criteria given the required probability of success. Working the problem this way, by counting successes and failures, also avoids the problem that would arise if a distribution type for the outcome of the simulation was assumed. Assuming a Gaussian, for example, and using the mean and standard deviation can often lead to missing important issues in the tails where our main interest lies. Deriving the desired probabilities without assuming anything about the output distribution can be done by examining the experimental distribution that results from the computer runs.

Note that reference is made to percentages and not to sigma levels. First, if one assumes that a certain sigma level corresponds to a certain percentage, then one is assuming the type of distribution, whereas Monte Carlo results are not typically that clean. Many of the launch vehicle output parameters are not Gaussian, for example, although some are. Second, even if the output is Gaussian, if it is not one-dimensional, the probabilities change. For example, the one-dimensional 3σ percentage for a Gaussian distribution (both tails) is 99.73%. The two-dimensional, 3σ percentage for a Gaussian distribution is only 98.9%, so covering for the 3σ case is probably not sufficient. An example of this is where a nozzle is being cleared from the inside of an interstage. The outside of the back end of the nozzle can be represented by a circle, and the interstage represented by a larger circle. If the inner circle contacts the outer circle, clearing has failed. The problem is two-dimensional, and 3σ is not enough. (See app. A for more details.) One could argue that if the problem were posed as the radius $r(x,y) > R$, the problem becomes one-dimensional again. However, the dimensionality here refers to the number of degrees of freedom in the two-dimensional Gaussian distribution; the distribution of the one-dimensional $r(x,y)$ is not Gaussian (it is a Rayleigh distribution).

Now suppose the 99.73% value of a particular parameter, say, acceleration, that results from running a simulation is needed; i.e., a structural engineer wants to design to the 99.73% high acceleration level. If 1,000 Monte Carlo samples are run, the answer will have quite a bit of error if the 99.73% value is asked for. There are just very few data points in the tail of the experimental distribution. If 10,000 simulations are run, less error would be expected, and so on. If the resulting accelerations are ordered from smallest to largest in the 1,000-sample case, interpolation between cases 997 and 998 could be done to get 997.3, and say that is the answer. But since the tails of our distribution are sparse in terms of the number of data points, there is a lot of inaccuracy in the estimate of point 997.3. It can be expected that if the Monte Carlo set is run a large number of times, different answers would be obtained, and eventually a distribution for the 99.73% value. So, to have confidence that the 99.73% value was captured, some conservatism has to be added to the result (confidence and risk were introduced in sec. 1). Intuitively, it makes sense that more conservatism is needed in this sense when the number of Monte Carlo samples is smaller. And this is indeed the case—the variance of the output is proportional to $1/\sqrt{N}$ where N is the number of samples.

What is the confidence level that the actual 99.73% value is captured? If the 99.73% case is taken from the experimental data, intuitively one might think that it would be an average value and there would be a 50% chance for a higher or lower value. However, it turns out that the level of confidence is <50% that the true value was captured. This is because, for a small number of samples, the tail is relatively very sparse; i.e., very few data points are in the tail when the sample size is not large. For 100 samples, for example, one would expect all of them to be below the actual 99.73% value.

Suppose one wants to be 90% confident that the 99.73% value was captured. The phrase used is the '99.73% value with 90% confidence.' This confidence is the complement of what is called 'consumer risk.' The 99.73% value would be captured with 10% consumer risk. That means that there is only a 10% chance that the 99.73% value is thought to be covered when actually it is not. The flip side of this is that, if there is a 10% consumer risk, there is a 90% producer risk; i.e., there is a 90% chance that the 99.73% value picked is greater than the actual 99.73% value and the case at hand is overdesigned.

Figure 6 illustrates the paradigm used in sampling theory. For a specified number of samples (N) and allowable number of 'failures' (k), the binomial probability of seeing k or fewer failures (y-axis) is plotted as a function of the actual failure probability (x-axis). The resulting curve (blue in the figure) is called the 'operating characteristic (OC)' of the (k, N) sampling plan. The numbers k and N are chosen so that there is a 90% confidence level that a mistake was not made in accepting a design with k or fewer failures; i.e., if the actual failure probability exceeds the design requirement, there will be less than a 10% chance of seeing k or fewer failures out of a sample of size N. In the figure, consumer risk is 10% for a failure probability of 0.25%. Also, the converse is used for producer risk: no more than a 10% chance of rejecting a good design (producer risk) may be taken when a separate and smaller acceptance failure rate is specified. If the design is made to the lower failure rate, and this (k, N) sampling plan is used, there is only a 10% chance a good product will be rejected. If the consumer is told that the product has no worse a failure probability than 0.25%, there is only a 10% chance the consumer will accept a bad product. The design conservatism is in designing for a 0.125% failure rate when the required consumer value is 0.25%.

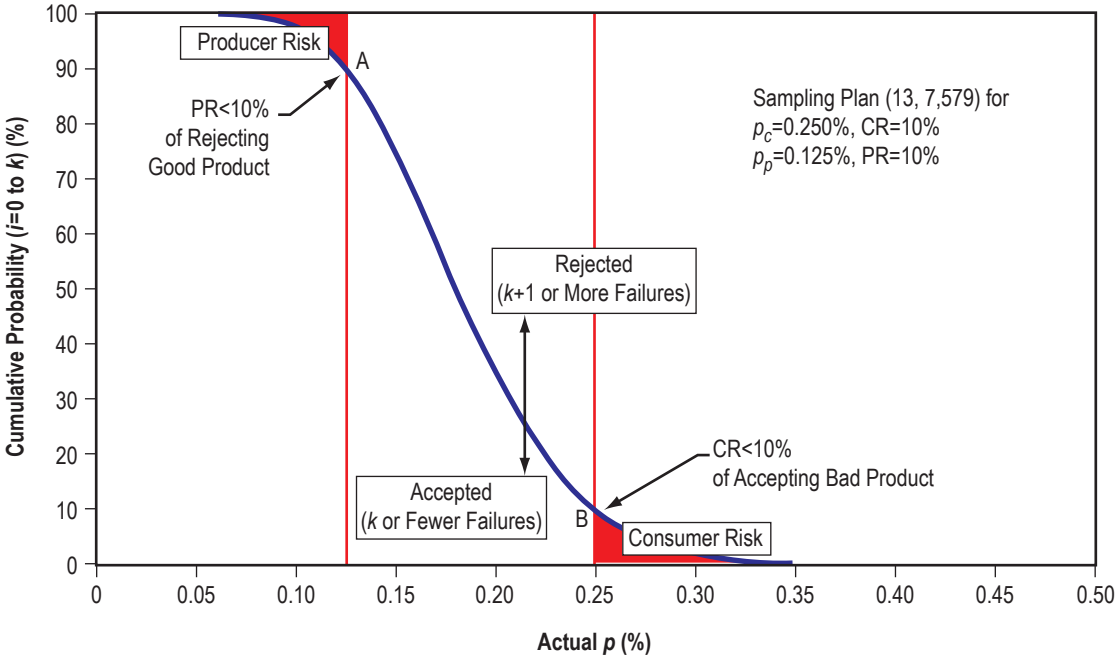

Figure 6. Example of an OC for a sampling plan devised to provide 10% consumer risk and 10% producer risk if maximum accepted $k = 13$ when $N = 7{,}579$.

Much of this development is the same as 'acceptance sampling' in statistical quality control, and the development in this TP is not new in this area.[12] This approach is less well known in aerospace simulation arenas.

In many NASA programs, no producer risk requirement is imposed, so there is actually a degree of freedom in choosing k and N. One may choose a smaller value of N and an associated smaller value of k, with the result that the design conservatism increases. It is important in this case to run some cases with a larger number of Monte Carlo samples (or run some cases with different random numbers) in order to see how much the design is driven by this numerical conservatism. Because of practical limitations on computer speed and availability, the number of Monte Carlo samples for a given launch vehicle configuration is on the order of 10^3. A typical practice is to generate $N = 2{,}000$ samples for each configuration and launch condition, which dovetails with the lower limit of N that will capture the 99.865% (what is commonly understood to be a 3σ level of success by those used to Gaussian distributions) extreme value success rate sampling plan. See appendix B for the derivation of the sampling plans.

For more detail on the estimation of consumer risk, including specific procedures for interpolating between sampling plans, consult appendix B. The order statistics used to derive these results are also shown in appendix B. Appendix C gives tables of the number of runs and the number of failures for several choices of success probability and consumer risk. With a larger number of runs, a higher percentage of failures is allowed (less conservatism) in order to get a certain percentage success with a desired level of consumer risk. Besides other topics, appendix F contains a section (F.10) that discusses the expected run-to-run Monte Carlo variations and provides additional insight concerning the number of samples.

A few illustrative cases are shown in table 2. Note how the allowable failure fraction increases as the number of runs increases and also as the consumer risk increases.

Table 2. Examples of sampling plans.

Failure Fraction Allowed	Consumer Risk (%)	Allowable Number of Failures	Minimum Number of Runs Necessary	Allowable Failure Fraction
0.02	10	0	114	0
0.02	10	1	194	0.005155
0.02	10	2	265	0.007547
0.0027	10	0	852	0
0.0027	10	1	1,440	0.000694
0.0027	10	2	1,970	0.001015
0.00135	10	0	1,705	0
0.00135	10	1	2,880	0.000347
0.00135	10	2	3,941	0.000507
0.00135	10	10	11,410	0.000876
0.00135	10	20	20,030	0.000999
0.00135	50	0	514	0
0.00135	50	1	1,243	0.000805
0.00135	50	2	1,981	0.001010
0.00135	50	10	7,903	0.001265
0.00135	50	20	15,310	0.001306

One way to try to obtain improved accuracy with fewer runs is to use Importance Sampling. This is described later in this section and in appendix G.

Note that it is not acceptable to make individual Monte Carlo sample simulations, continually increasing the number of runs, and then stop and declare success if a case is found that meets the required probability levels. For example, in table 2, for a failure fraction of 0.00135 and a consumer risk of 10%, suppose the allowable number of failures is exceeded for one failure, two, three, etc., but that with 10 failures, the test is passed and the requirement is met. The problem is that it is known that the test was failed for all previous values of the number of failures up to that point. This probably means that the case where the test passed falls in that 10% (10% consumer risk) of cases where the test was passed but the requirement was not actually met.

Suppose a Monte Carlo run is made and the number of failures indicates that the system success requirement is not met. The vehicle design must then be changed (or the requirement changed, if it is too stringent) in order to improve the system success.

4.1.1 Order Statistics Versus Fitted Distributions

In general, the good thing about order statistics is that they are nonparametric; i.e., they do not depend on an assumption or model of the underlying probability distribution. The downside is that most of the information 'bought' with the computer time is not used; consequently, there is an undesirable tradeoff between producer and consumer risk. The nice thing about fitted distributions (parametric statistics) is that they use basically all the information from the generated data, and of course the bad thing is that the uncertainty associated with the choice of distribution is difficult to control (one can test for normality, although necessarily there will be a certain number of false negatives in those tests, where the distribution is accepted as being normal when it is really not. Section 3 discusses a test for normality.) If the data are clearly not Gaussian, it may be more appropriate to fit an extreme value distribution[13] to the tail (say, the outlying 5%) of the sampled data rather than fitting the whole distribution.

In the specific case of recontact probability (app. A), the distributions have a few, or even zero, recontacts, and a way to compare two configurations even with zero recontacts being sought. The assumption of normality enables a reproducible measure of goodness. Although it would be possible to use order statistics directly on the generated ensemble, they have a significant producer's risk. The Gaussians seem to be pretty well behaved, and therefore are used to generate the circular error. It is a judgment call.

4.2 Human Versus Nonhuman Space Flight

There is no real difference in how to set up a Monte Carlo run for human versus nonhuman space flight. The only differences might be in how much attention is paid to the inputs, to the model verification, and to any failed runs. The tolerance to error is very low for human space flight, but it should also be low for expensive nonhuman missions. For application to human missions, and where improper simulation could impact flight safety or mission success, each component of the work is scrutinized heavily. All key inputs are typically examined and certified and are listed as 'critical math models' that receive extra attention. Simulations used to verify that requirements are

met must themselves be fully verified so that they can be trusted. Typically, besides testing the functions in the simulation, testing the simulation results, and documenting why it should be believed, an independent simulation would be used and compared, and eventually flight data would be used to show simulation accuracy and make any necessary improvements in the simulation. Finally, the percentage of success in simulation results is typically required to be very high when the mission is human. A typical required value is 99.865% success with 10% consumer risk. For uncrewed missions of decreasing cost, parts of this effort are relaxed depending on the risk posture of the project.

4.3 Rare Events

What if statistics about rare events are needed? For example, severe wind gusts occur rarely, but the system design must include these and be successful when they occur. This gust could be anywhere in flight and could be from any direction. If a Monte Carlo simulation was run with 2,000 samples, there might be zero or one severe gust that randomly occurs, and the random gust is not likely to be at a challenging part of the flight (e.g., high dynamic pressure) that needs to be covered. It may not be of a wavelength that challenges the vehicle—that also needs to be covered.

Similarly, how are statistics obtained for failure events that can occur at any time in flight but are rare? If a Monte Carlo simulation is simply run with the rare events seldom occurring, the statistics will be ruined. Statistics for the no-failure or no severe gust trajectories will be skewed by the one or two rare events. An example of this is where a severe gust randomly only occurred in May when running Monte Carlos for every month, making May look like the worst month, when in fact the gust could have occurred in any month. There are also no statistics for what happens in the bad cases overall because there is only one data point.

What is needed in this case is to make Monte Carlo runs where the failure or gust occurs for every sample flight. Then the low probability event will happen at random times in flight and with random directions or consequences (since the rest of the vehicle and environmental parameters are also dispersed in these runs), and statistical results will be obtained for what happens when these events occur. An example would be statistics of crew safety when the flight control (e.g., engine actuator) becomes stuck.

If flight must succeed when a bad wind gust (with a wavelength that challenges the flight control) is applied at an undesirable time of flight, then the Monte Carlo environment is not the appropriate one for this test; instead, a single simulation would be run with this gust applied.

4.4 Alternative Sampling Methods

There are other ways to do sampling when running Monte Carlo runs other than just randomly choosing every parameter. For example, ‘importance sampling,’ discussed in more detail in appendix G, involves recognizing that some input variations have more impact than others. The basic idea is to sample at locations where the impact on the output is largest in order to minimize the variance of the measurement.[14] Usually, the data of interest are in the tails where there are few data points. Most of the random output is away from the tails and does not contribute much to understanding values of interest. If values are emphasized that give outputs in the tails, the variance can

be reduced with the same number of samples. Then the outputs are weighted to correct for the use of the biased input distribution. If the outcome has a known relationship to the inputs, importance sampling can be valuable. For example, with stage separation, the trend of inputs to output clearance is known. In probabilistic risk assessment, when the interest is in what happens when any of a number of unlikely failures could occur, focus can be made on the tails of the distributions.

This becomes more complicated as the system complexity grows. Also, some input parameters may be important to some outputs (e.g., navigation initialization relative to orbit plane injection error) and not at all important to other outputs (e.g., structural loads). If a trajectory Monte Carlo simulation is being run where the desire is to observe a number of output parameters related to different parts of the design, importance sampling will not be applicable. It will not work, in general, for examining failure situations where the desire is to escape before reaching a structural load limit, since the load value is a complicated function of the particular wind profile and its interaction with the trajectory (it is not clear how to weight the inputs). Importance sampling is discussed in more detail in appendix G.

Latin hypercube sampling selects N different values from each of k input variables $X_1, \ldots, X_k$ in the following manner. The range of each variable is divided into N nonoverlapping intervals on the basis of equal probability. One value from each interval is selected at random with respect to the probability density in the interval. The N values thus obtained for X_1 are paired in a random manner (equally likely combinations) with the N values of X_2. These N pairs are combined in a random manner with the N values of X_3 to form N triplets, and so on, until N k-tuplets are formed.[15] The N k-tuplets are the input variables to the N Monte Carlo samples. In median Latin hypercube sampling, the middle value is chosen in each interval. In Hammersley and Halton sequences, the space of the random interval is sampled according to a defined procedure.[16] These methods use deterministic sampling to spread the input points over the probabilistic space rather than to randomly choose all the points as in simple Monte Carlo sampling. After the samples for each variable are determined, the values to use for the different variables are randomly chosen from this set.

Examining a two-dimensional, uniformly-distributed space yields behavior that looks like that in figure 7.[17] Inspection shows that Monte Carlo sampling appears more random than the other approaches, which sample the space more uniformly. These random numbers are converted into the uncertain variables in the same fashion as done with simple random sampling using a RNG. The resulting variable choices are now spread more uniformly across the probability space.

For a number of problems that have been studied, these alternative sampling approaches lead to significantly faster convergence (for a smaller number of samples) for the mean and standard deviation of the output as compared to simple random sampling.[16,17] This would be advantageous when fitting a distribution with the data.

The approach used in this TP is simple random sampling, since the statistics of the output are well understood and how many cases to run and how many failures to allow in order to achieve a desired percentage outcome with a fixed consumer risk can be determined. In most of the cases of interest for trajectory Monte Carlo work, capturing requirements that specify high percentage success rates (e.g., 99.865%) is of interest. What is critical is capturing the possible random tail values

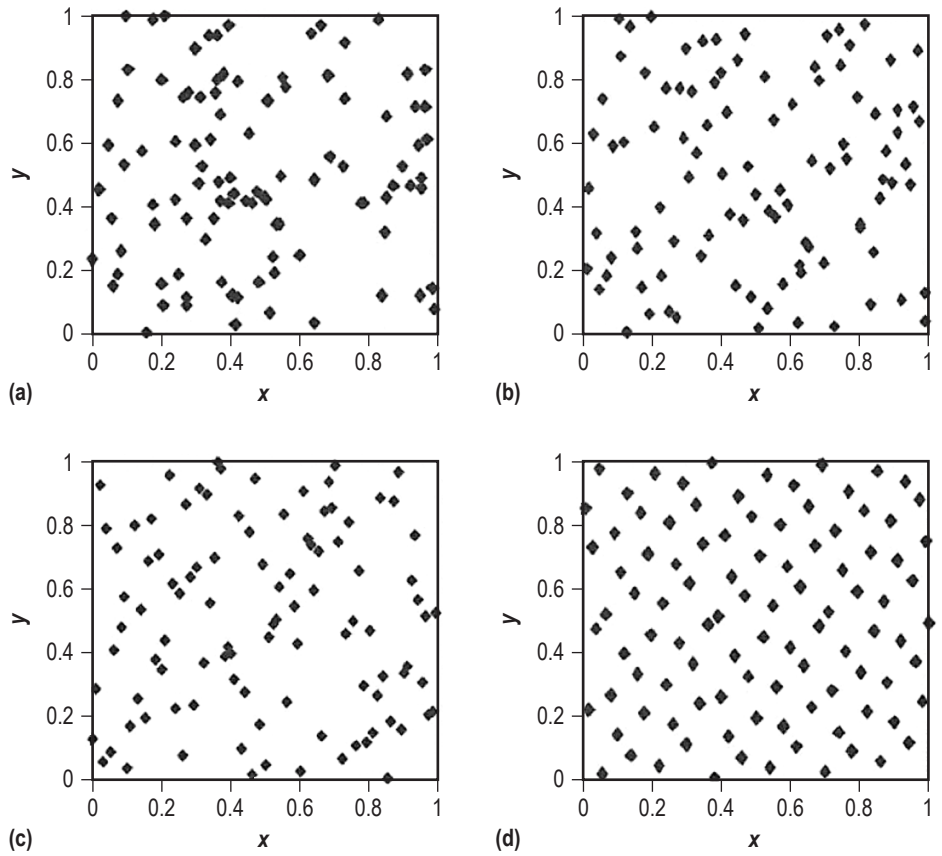

Figure 7. 100 sample points on a unit square for (a) Monte Carlo, (b) Latin hypercube, (c) median LHS, and (d) Hammersley sequence sampling.

more so than capturing the mean and variance. Investigation of the possibility of improved behavior with some of these alternate sampling approaches is a possible topic for future work. It is possible that the process of choosing 2,000 points spread across the probability space more uniformly will yield better results than randomly choosing the 2,000 points. It would be necessary to identify the corresponding order statistics for these different sampling techniques. A complication is how one would sample the more sophisticated subsystem models that are furnished by other disciplines. For example, for the random correlated winds coming from GRAM, how would one implement these techniques? It would likely be necessary to modify each of the more complicated models internally in order to use different sampling procedures.

4.5 Random Number Generators

Users running Monte Carlo simulations should be careful that a high-quality RNG is chosen. In point of fact, the only truly random numbers known in nature arise from quantum mechanical processes, such as radioactive decay. The sequences generated by computers are at best 'pseudo-random.'

By convention, a standard RNG generates uniformly-distributed numbers in the interval (0,1); i.e., including zero but excluding 1. Some RNGs are set up to generate 32-bit integers, while others provide single- or double-precision reals. Because of the finite precision of computer arithmetic, eventually all RNGs repeat the sequence of random numbers they generate. The length of this sequence is called the 'period' of the RNG.

It is up to the programmer to map the standard RNG to generate a random variable with a desired distribution. For a one-dimensional variable, in principle, this 'only' requires inverting the CDF. Sometimes this inversion is computationally costly, and notably so for a Gaussian distribution, which would require inverting the error function. Fortunately, there are several clever techniques for efficiently mapping standard uniform random numbers to a Gaussian deviate, the well-known Box-Müller method being only one example.[1] The trajectory simulations for Ares I use an inversion method called the 'ziggurat method,' which is more efficient than Box-Müller when large numbers of variables must be generated.[18] Methods for computing these inverses for a very large number of distributions are '1-liners' in MATLAB® and other high-level numerical packages.

Donald Knuth advises "a random number generator should never be chosen at random";[19] i.e., one should never assume that default RNGs which typically accompany compilers or higher level languages are satisfactory for extensive computation. One problem that occurs in some RNGs is called the 'serial correlation problem.' This means that successive *k*-tuplets of random numbers fall near each other in k-dimensional space. One particularly notorious example is a RNG called RANDU, which was supplied on IBM mainframes for many years. Triplets of RANDU output values fall on precisely 15 planes in three-dimensional space, making it undoubtedly the worst RNG ever to see widespread use.[20] The RNG supplied with early versions of Microsoft® Visual Basic® had visible correlations in four dimensions.[21]

The serial correlation problem may seem arcane and 'improbable,' but consider the implications for trajectory simulations. If a given trajectory simulation requires 24 inputs, then an RNG that has undesirable correlation in twenty-four-dimensional space may produce highly correlated (nonindependent) sets of inputs, resulting in a simulation with measurable bias. This problem can be very hard to detect if not anticipated. One way of mitigating this problem is to use separately seeded RNG sequences for each input parameter.

A standard battery of tests for RNGs was developed by George Marsaglia. The test series is known as DIEHARD.[22] It comprises roughly a dozen types of tests, including 'birthday tests' (based on the problem analyzed in sec. 1), 'monkey tests,' 'parking tests,' various kinds of serial correlation tests, and so forth. When implementing a RNG, it is a good idea to verify its performance under DIEHARD.

The RNG currently used in Ares I trajectory simulations is a variety of 'Mersenne Twister' called MT19937. This RNG has a long period and scores well on the battery of DIEHARD tests.[23]

One of the advantages of using pseudorandoms, of course, is that the entire sequence can be regenerated if the seed is reused, which is a boon to debugging, but also facilitates variance-reduction techniques such as 'common random numbers.'[24] As a practical matter, some care should

be exerted in seeding the RNG. For example, a commonly used strategy for seeding uses the time of day returned from a system call at job submittal. If the resolution of this time is only 'seconds' and not hundredths of a second, a maximum of 86,400 sequences can be generated. Since many RNGs require odd-number seeding, this number is really only 43,200.

5. OUTPUTS AND POSTPROCESSING ANALYSIS

Before using the output data, it is important to examine any failed cases. If the simulation bombs or completes with any significant outliers, these cases should be studied, even if this number of failures is allowed in terms of the overall success requirements. The first reason is part of the overall testing philosophy. The simulation may have a bug, or there might be something about the vehicle models or the way the vehicle is being flown that causes the bad outlier.

There are many ways that Monte Carlo output can be examined. Some examples of possible products are shown below.

Table 3 is sample data of output parameters with statistics included. Any variable and any desired statistic can be presented, and various different cases can be compared in a summary table. Table 3 uses order statistics to provide the percentage results. From this, design levels (of $Q\alpha$-total, maximum heat rate, etc.) can be set. The worst-case runs can be examined. Whether or not there is sufficient propellant to meet requirements, and whether attitude rates, orbit insertion accuracies, or any number of other parameters meet requirements can be examined.

Table 3. Sample statistics from a Monte Carlo simulation.

Trajectory Variables	Minimum	Maximum	Average	99.73% Low With 10% CR	99.73% High With 10% CR
Injected mass (lbm)	95,056	99,082	97,147	95,253	98,935
Usable LO_2 remaining (lbm)	291	4,385	2,350	598	4,041
Usable LH_2 remaining (lbm)	–136	1,908	1,019	37	1,893
Maximum dynamic pressure (psf)	682	887	770	695	866
Maximum $Q\alpha$-total (psf-deg)	874	5,412	2,495	1,120	5,198
Maximum axial accel. (1st stage) (g's)	3.605	3.894	3.734	3.611	3.871
Maximum heat rate (1st stage) ($Btu/ft^2/s$)	3.269	4.049	3.600	3.288	3.977
Maximum heat rate (2nd stage) ($Btu/ft^2/s$)	1.452	2.127	1.828	1.504	2.124
Total heat load (Btu/ft^2)	204.5	220.5	211.3	204.8	218.9
Mach at maximum heat rate	4.20	5.89	5.09	4.33	5.85
SRB apogee (ft)	319,188	350,651	334,401	321,174	347,915
Time of tower clearance (s)	7.20	8.05	7.63	7.24	8.01

Table 4 compares statistics for particular flight events, the example here being maximum dynamic pressure. If an engineer is examining the behavior of the system at these flight events, the variation of the trajectory parameters for the events must be known. For example, for each jettison event, the jettison must be successful for the range of trajectory parameter variations.

One can plot results versus sample number, as in figure 8. This is useful for displaying the behavior of the results and makes for easy visualization of the outlying cases. Use of Excel allows the user to determine the sample number for any point of interest by simply placing the cursor on that data point.

Table 4. Sample statistics for a flight event.

Max-*Q* Flight Condition Statistics	Minimum	Maximum	Average	99.86% Low With 10% CR	99.86% High With 10% CR
Time (s)	50	76	62.5	50.2	75.7
Dynamic pressure (psf)	682	887	770	684	882
Mach number	1.14	2.45	1.66	1.15	2.45
Altitude (ft)	26,777	58,186	41,074	27,053	58,141
Axial acceleration (g's)	1.74	2.66	2.10	1.74	2.65
Normal acceleration (g's)	–0.035	0.136	0.026	–0.035	0.135
Angle of attack (deg)	–6.27	2.07	–1.33	–6.16	2.05
Sideslip angle (deg)	–4.06	4.92	0.43	–3.81	4.87
Total angle of attack (deg)	0.02	6.33	1.97	0.02	6.32
Qα-total (psf-deg)	13	5,412	1,532	16	5,337

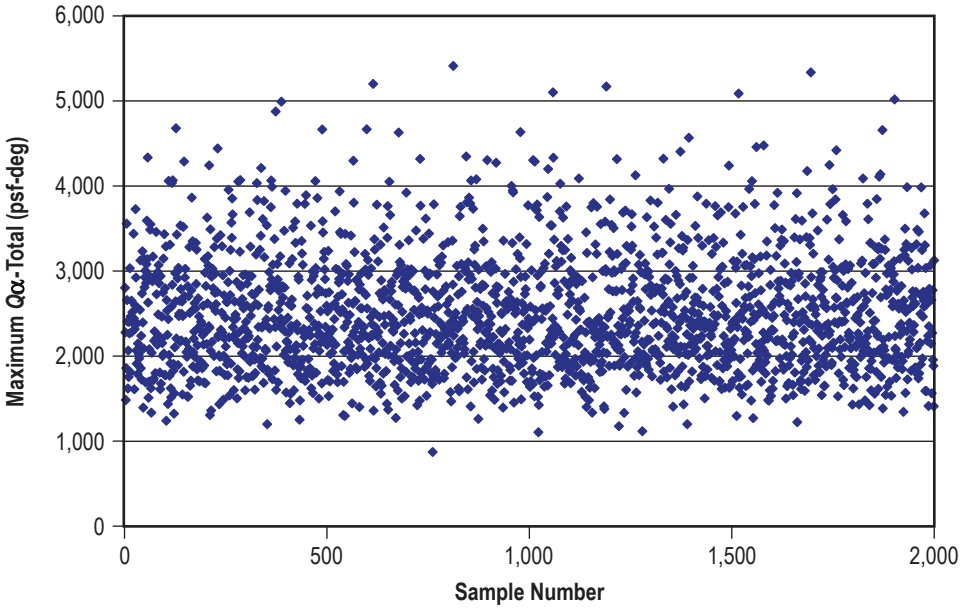

Figure 8. Scatter plot of maximum dynamic pressure times total angle of attack versus run number.

Scatter plots can be made versus another parameter of interest, such as in figure 9. Maximum dynamic pressure does not generally occur at the Mach number where it occurs in the nominal trajectory. Many questions come up in design analysis for which the behavior of a design indicator versus some other parameter needs to be known.

Plots can be used to compare two similar variables of interest, as in figure 10, which shows the in-plane orbit insertion variations.

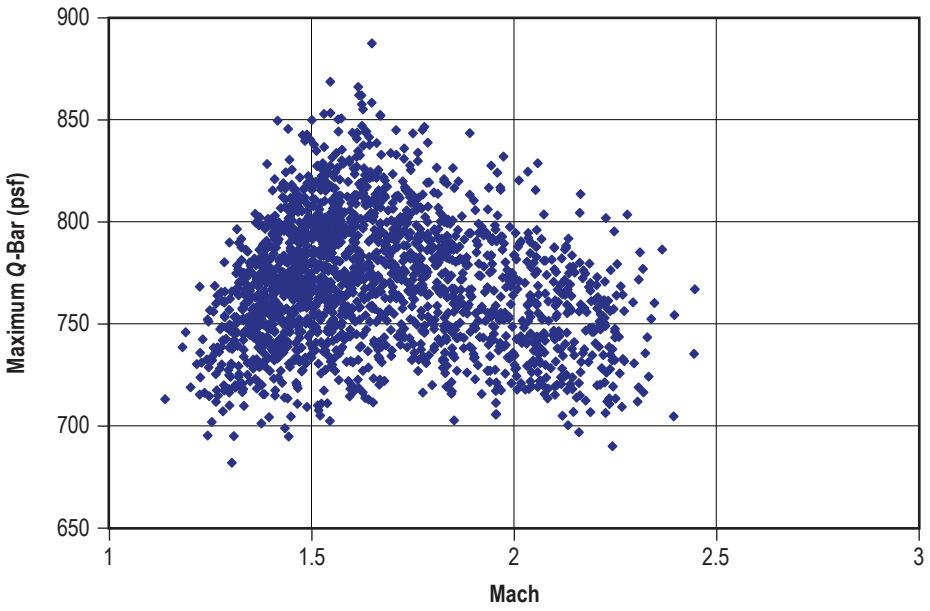

Figure 9. Scatter plot of maximum dynamic pressure versus Mach.

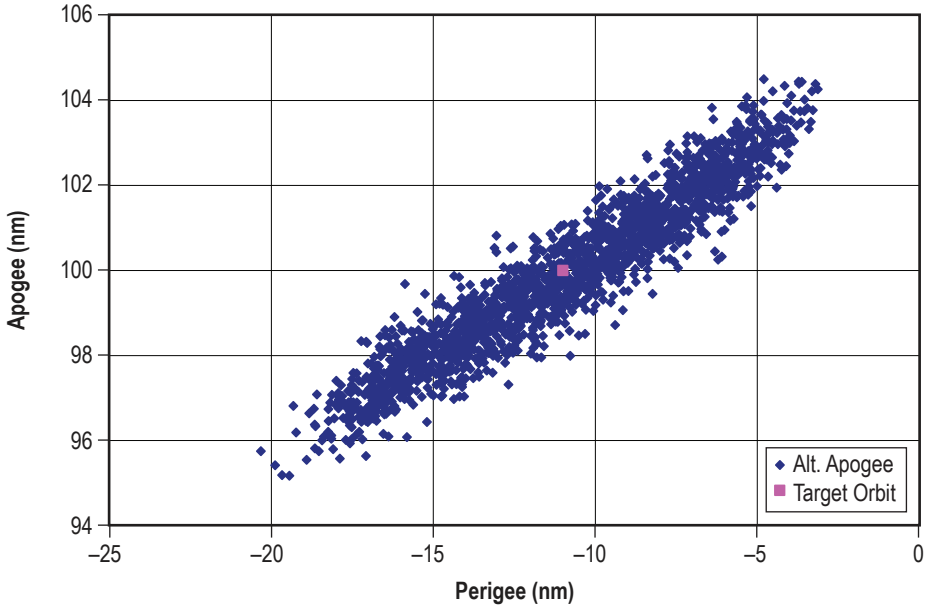

Figure 10. Scatter plot of insertion apogee versus perigee.

Envelopes of trajectory variables of interest may be plotted. Shown in figure 11 is one of the two actuator angles for part of a flight. The red curve shows the nominal case, and all the Monte Carlo results are within the blue curves. Any issues that show up where the envelopes do not behave as expected can be investigated. Note that this plot is for a particular combination of mission and vehicle model, as indicated in the caption.

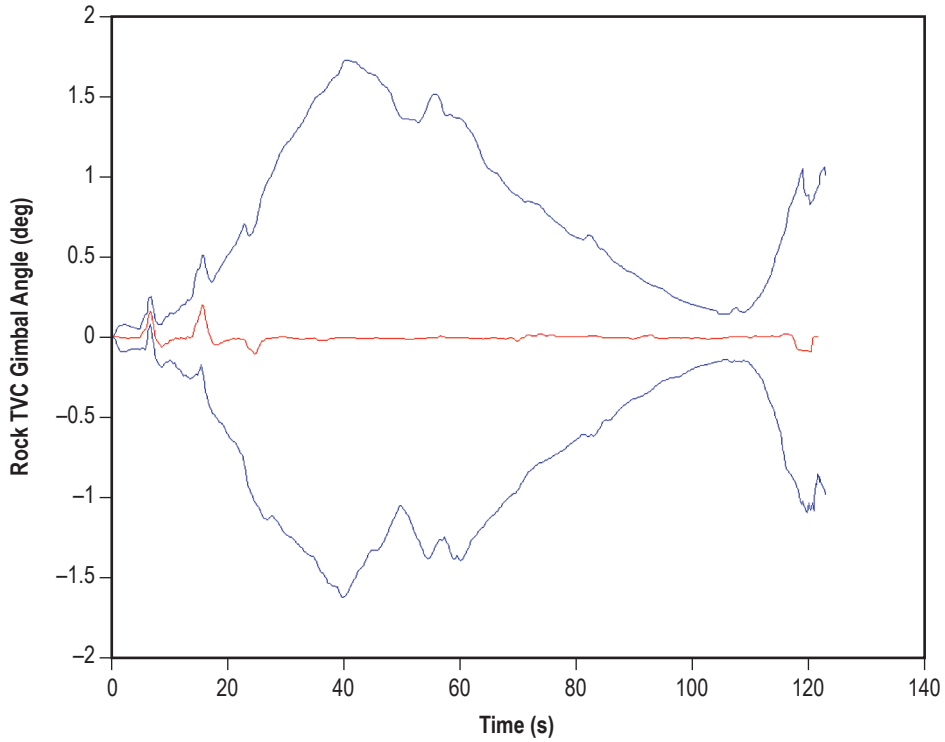

Figure 11. Envelope plot of rock TVC angle versus time—minimum/maximum envelopes (Aug. winds, Space Station mission, light/fast vehicle model, close of launch window).

One may derive correlation coefficients. The correlation between two random variates x and y is

$$\rho = \frac{\operatorname{cov}(x,y)}{\sigma_x \sigma_y} \ , \tag{5}$$

where the sample covariance is

$$\operatorname{cov}(x,y) = \frac{1}{N-1}\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i) \ . \tag{6}$$

Here, σ_x and σ_y are the standard deviations of the individual random variates, N is the number of samples, and $\bar{x}_i$ and $\bar{y}_i$ are the means.

For example, table 5 shows hydrogen remaining highly correlated with mixture ratio whereas oxygen remaining is less correlated with mixture ratio. Although these coefficients provide an understanding of which parameters most affect the important output, they do not indicate how much the output would improve (or deteriorate) if one of the parameter variations were changed. For that, a new run would be needed with the specific parameter's variation changed.

Table 5. Correlation coefficients for propellant remaining. Correlations with input vehicle variations.

LO_2 remaining	First stage I_{sp}	0.59
	First stage burn rate	0.45
	Upper stage mixture ratio	0.48
	Upper stage I_{sp}	0.31
LH_2 remaining	Upper stage mixture ratio	0.84
	Upper stage inlet LH_2 temperature	0.31

Time histories that show the spread of cases are also of interest. These may be for the whole flight or for a key part of flight, as shown in figure 12. Any unexpected behavior in these plots can be investigated.

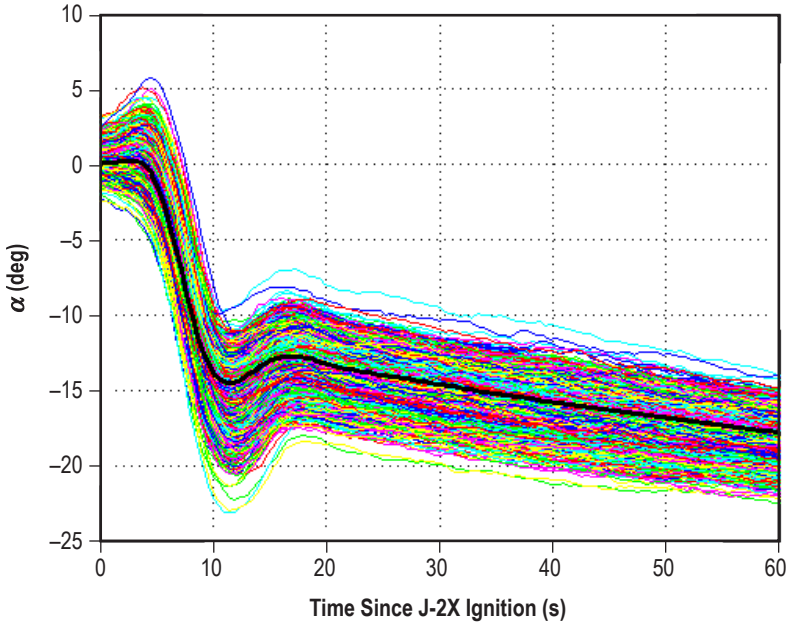

Figure 12. Time history plot of angle of attack after upper stage engine ignition (Ares I Monte Carlo results).

Figure 13 compares launch vehicle ascent structural bending load indicators at a particular altitude for different ways of modeling the winds. Here, dynamic pressure times sideslip angle is graphed versus dynamic pressure times angle of attack. Use of a measured wind database is compared to an 'enveloping vector wind model' and to the GRAM.[10] There are many more runs for the GRAM case than for the measured database since GRAM generates a new random wind profile each time it is called. The red circle is the 99.73% overall value with 10% consumer risk (for the GRAM data). A probability ellipse would be a natural curve to draw; in this case, a circle was used because it represents the vehicle capability at the desired percentage value. In this particular figure, it can be seen that there is a measured wind outlier relative to the GRAM results, even though the measured database is much smaller, which should lead to investigation.

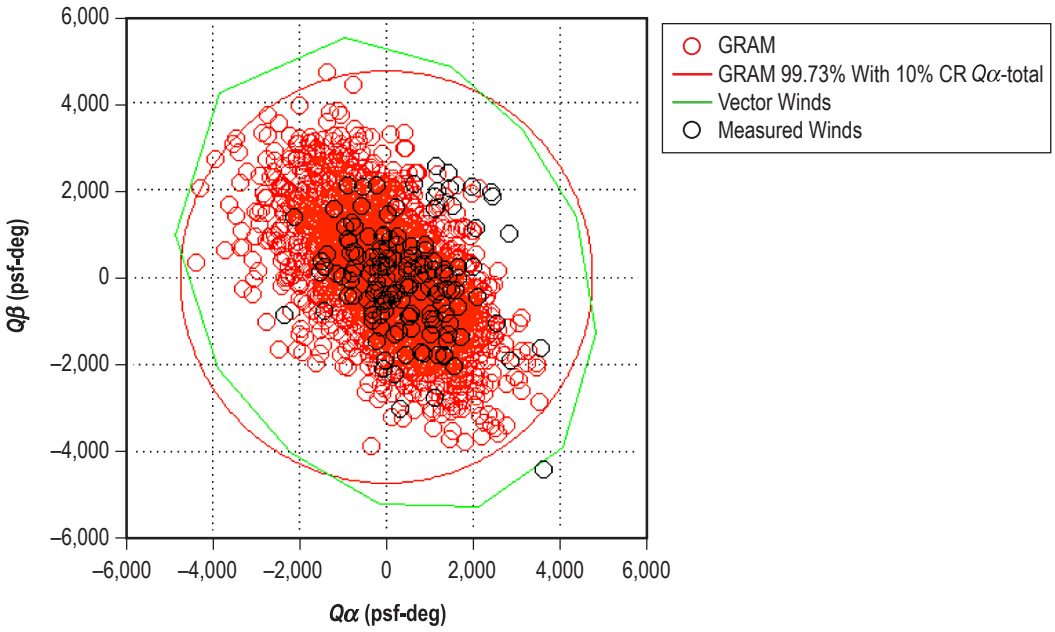

Figure 13. Scatter plot of dynamic pressure times sideslip versus dynamic pressure times angle of attack for various wind inputs (altitude = 10 km (32,808 ft)).

When examining failure cases, it has already been mentioned that it is best to always cause a failure in a particular Monte Carlo run so that reasonable statistics of the failure results may be obtained. Figure 14 is an example for a stage separation study where the outer red circle represents recontact between the engine nozzle and the separating interstage. The 10 clusters of data points are the nearest points on the nozzle to recontact for failure of each of the 10 boost deceleration motors (small rockets that pull the first stage back from the upper stage). Green probability ellipses are drawn around the scatter plots for each motor failure.

More complicated output may be calculated as opposed to simple trajectory output parameters. Figure 15 is another example where failures were caused on every Monte Carlo trajectory. In this case, a type of thrust vector control (TVC) actuator failure occurred. For each failure time in flight (x-axis), and each Monte Carlo sample, different abort detection triggers may be the first to see the problem (different colors). The time when the trigger is passed is the zero value on the y-axis. It is assumed it takes 0.55 s for the crew to depart after the trigger indicates the need for abort. The y-axis shows the time duration between when the trigger is passed and the vehicle exceeds a moment-based load indicator (MBLI) structural limit, thus providing a measure of crew survivability.

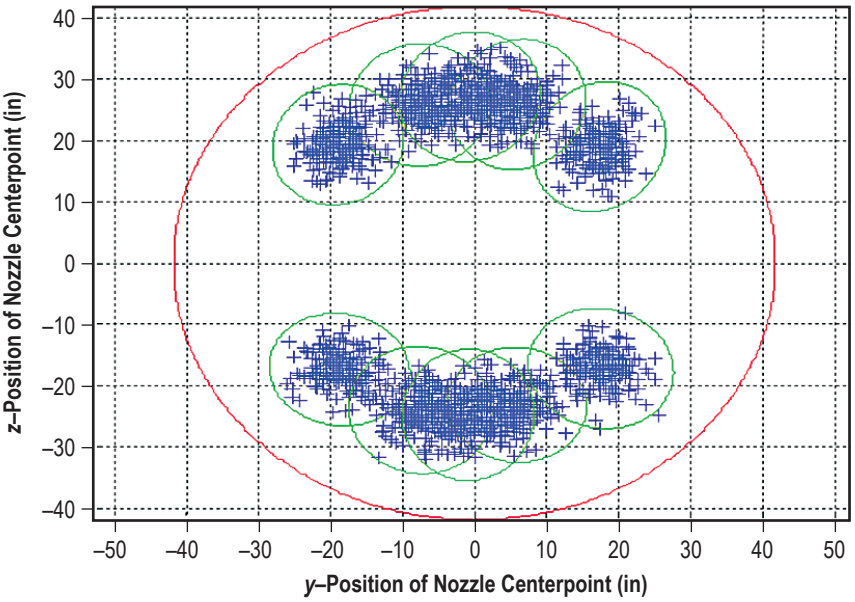

Figure 14. Scatter plot of nozzle clearance at the separation plane for stage separation—one boost deceleration motor failure.

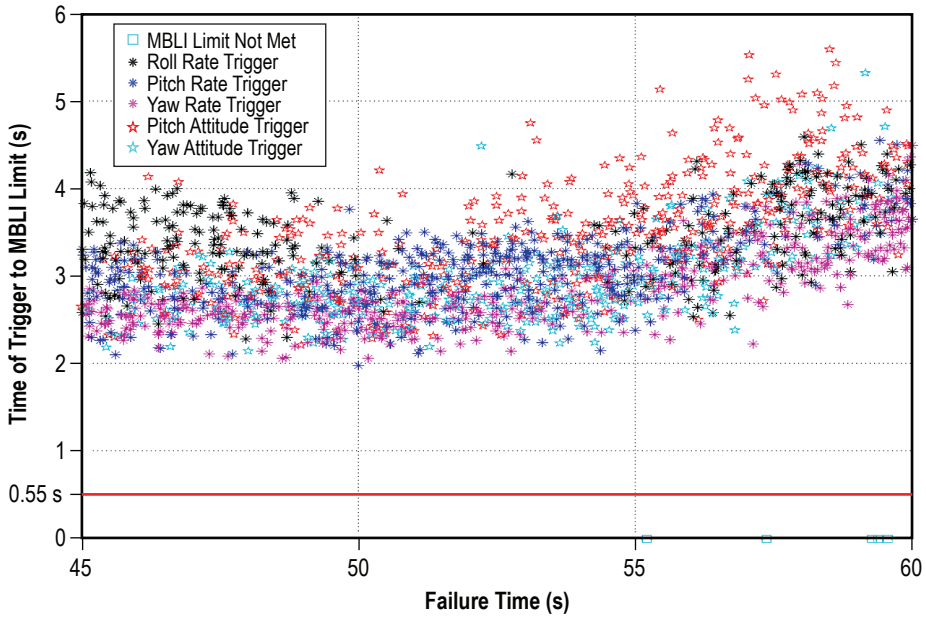

Figure 15. Time available to escape after failure detection, as a function of time, for TVC failure.

Use of appendix F would allow a user to minimize a consumable to meet a two-dimensional goal. In figure 16, running out of either oxygen or hydrogen results in a failed case. The question is how to minimize the total of oxygen plus hydrogen to achieve the required overall success. The success percentage curves are shown on the graph, along with a line of constant total propellant. If this line is tangent to the desired percentage curve, the total propellant will be minimized.

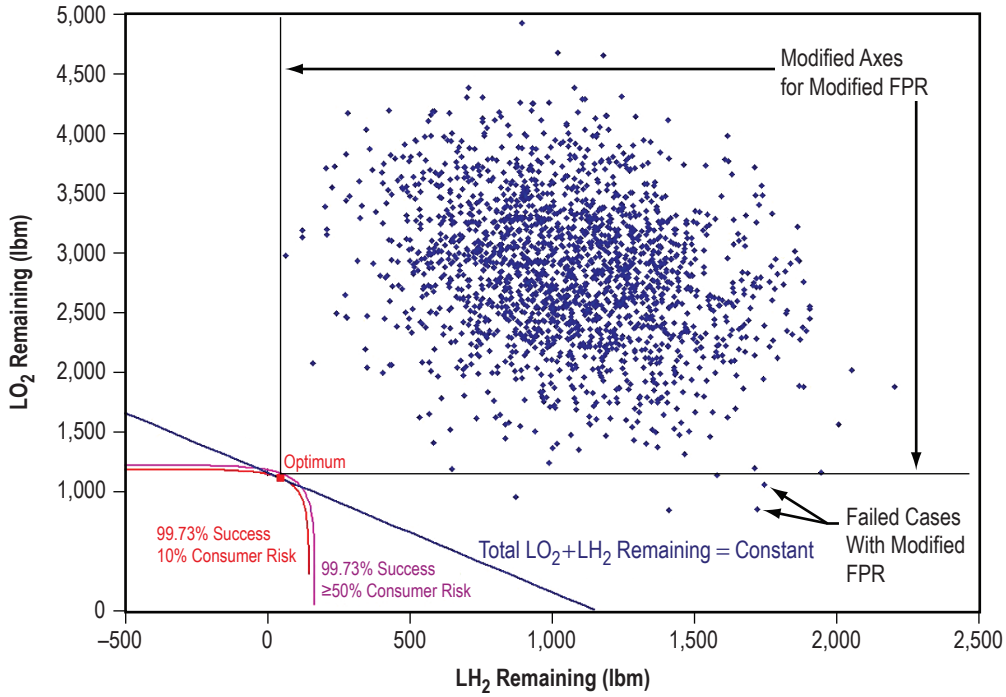

Figure 16. Usable LO_2 and LH_2 remaining scatter plot for determination of required FPR.

Use of Monte Carlo products also enables the engineer to investigate any unexpected behavior. As an example, figure 17 shows yaw error during first stage flight. Error is expected to grow around the time of maximum dynamic pressure, but as dynamic pressure becomes smaller at higher flight times, the error is expected to decrease. The relatively high error from about 70 to 110 s was unexpected. After some investigation, the cause was found and this behavior was corrected.

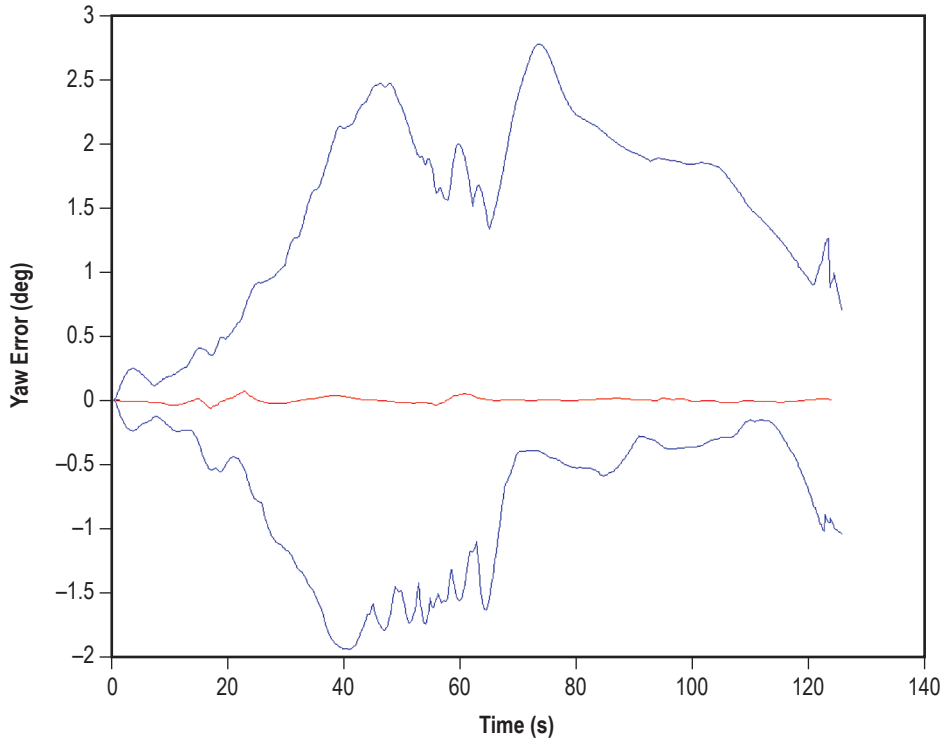

Figure 17. Use of minimum/maximum envelope plots to locate issues with flight control (lunar light/fast, close of launch window, Feb. winds).

6. SUMMARY AND OUTLOOK

An overview of the application of the Monte Carlo method to the trajectory analysis of flight vehicles has been presented. Monte Carlo enables the practical evaluation of complicated summations over multiple stochastic input variables. The trajectory ensembles thus derived can also be used to establish design envelopes for flight vehicle components and elements.

Explanation of the steps in this process have been provided. The steps include the gathering and verification of inputs to the simulation, testing of the simulation, the generation of the ensemble with Monte Carlo sampling, and postprocessing.

Detailed analysis is deferred to appendices A–G, where issues of consumer risk, multivariate optimization, and multiple extreme constraint optimization are discussed. A detailed method for analyzing stage separation recontact probability (a two-dimensional result) is also presented. Appendix E discusses how to duplicate earlier Monte Carlo random variations with a newer version of the simulation in order to compare the modeling changes in an apples-to-apples fashion. Appendix G addresses 'importance sampling.'

APPENDIX A—RECONTACT PROBABILITY DURING STAGING

A.1 Introduction

Stage separation is one of the riskiest phases in the launch of a space vehicle. In the design of a new launch vehicle, considerable attention is devoted to the configuration of the stage separation system, with the intention of minimizing potential adverse outcomes. Launch vehicle stages can fail to separate because of the failure of avionics, pyrotechnics, or retrorockets, or they can separate but bump into each other during separation (a possibility known as 'recontact'), or the stage separation process can generate forces and torques adverse to vehicle control, structural integrity, or propellant fluid dynamics.

This appendix focuses on the possibility of recontact during stage separation. The best way to quantify the risk of recontact is the calculation of a recontact probability using a combination of trajectory analysis and probability theory. Deficiencies of several commonly used metrics of recontact are discussed, and a computational model that has both theoretical and practical justification is proposed.

Most of this appendix uses the Ares I primary stage separation event as an example, but the techniques and considerations discussed herein are quite general for any two-dimensional analysis. The Ares I system presented here was the baseline system at one point in the program.

A.2 Ares I Stage Separation System

The first stage of Ares I is a five-segment reusable solid rocket motor (RSRMV) evolved from the space shuttle boosters. The upper stage is liquid fueled. The two are connected by an interstage that comprises a cylindrical section atop a conical frustum that adapts the different diameters of the two stages. The layout of the interstage and separation plane is shown in figure 18.

When the RSRMV reaches the end of its burn, ≈2 min into flight, a sequence of events is initiated by the onboard mission manager. The focal event in the separation is the detonation of a separation mechanism that cuts the structure connecting the first stage and the upper stage. Booster deceleration motors (BDMs)—small solid rockets adapted from the space shuttle's booster separation motors—are attached to the aft skirt of the first stage and fire upward to pull the booster away from the upper stage. At the same time, ullage settling motors (USMs) attached to the upper stage fire in the opposite direction, providing thrust to maintain a positive acceleration on the upper stage propellant (so that the inlet pressure into the engine is conducive to engine start) during the brief coast period between the RSRMV separation and the firing of the upper stage engine (USE), which is a J-2X. The timings of the USM and BDM ignitions relative to the separation pyro initiation are design parameters that enable optimization of the acceleration environment of the upper stage propellant and the loading of the separation joint at the moment of separation.

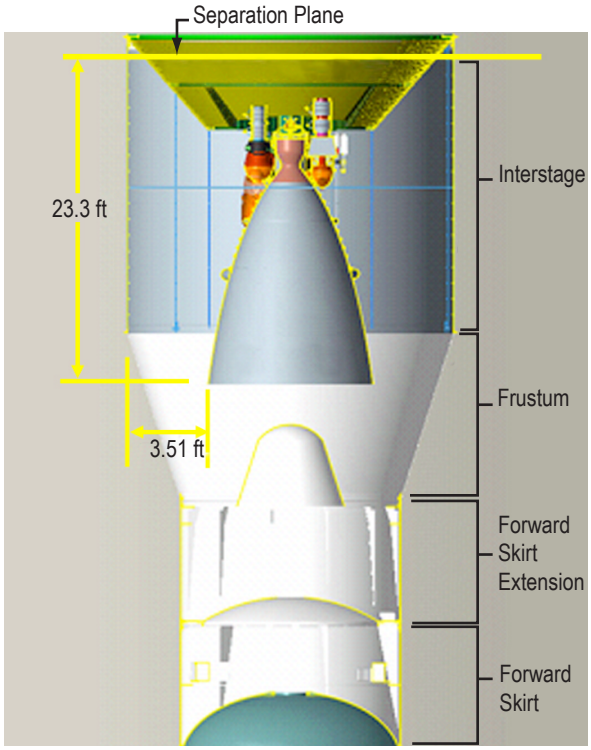

Figure 18. Ares I interstage and separation plane.

Ares I provides a challenging configuration for the stage separation design team, in part because the USE nozzle protrudes a considerable distance (≈23.3 ft) below the separation joint (the top of the interstage), with an exit diameter that is a large fraction of the upper stage diameter. The maximum clearance of the USE nozzle when it passes the separation joint is 3.51 ft. One measure of the challenge of Ares I stage separation is the separation angle $\beta = \tan^{-1}(3.51/23.3) = 8.56°$. This is the maximum average angle at which the upper stage could drift from the first stage and still allow the USE nozzle to clear the top of the interstage.

A number of random variables affect the stage separation trajectory. Variations in the timing, thrust, and thrust angle of the BDMs and USMs affect the relative motion of the first stage and upper stage. Variations in vehicle loading and flight condition affect the pitch and yaw rates of the vehicle at separation. Importantly, the RSRMV thrust may display anomalous transient side forces. In addition, the stage separation system is designed to be tolerant of single-point failures, such as the failure of a single BDM or USM, which can cause a significant imbalance of forces and torques on the launch vehicle stages.

NASA formalizes the list of assumptions in the trajectory analysis as part of accreditation of the analysis as a 'critical math model.' Altogether, there are several dozen dynamic inputs to staging that are modeled as random variables.

A.2.1 A Sampling of Separation Trajectories

Given estimates of the mean and variance of all these random variables, 6 DOF multibody numerical simulations are used to compute the relative trajectories of the first stage and upper stage during separation. Trajectory simulations conducted to date suggest that, for Ares I, the relative trajectories are smooth and show monotonically increasing radial offset, so that the risk of recontact can be formulated in terms of the displacement of the USE nozzle as it passes through the separation plane. Figure 19 shows a typical set of radial trajectories for a particular set of assumptions about the random inputs. The red lines in this figure represent several hundred trajectories, portrayed here as the radial ('lateral') displacement of the USE nozzle outer edge from its starting point. (Note the vertical and lateral scales are not the same.) The solid line starting at a 2.025-ft displacement and increasing to 3.51 ft represents the interstage inner wall. Values of the mean and one, two, and three standard deviations are shown by stars at the separation plane level (23.3 ft).

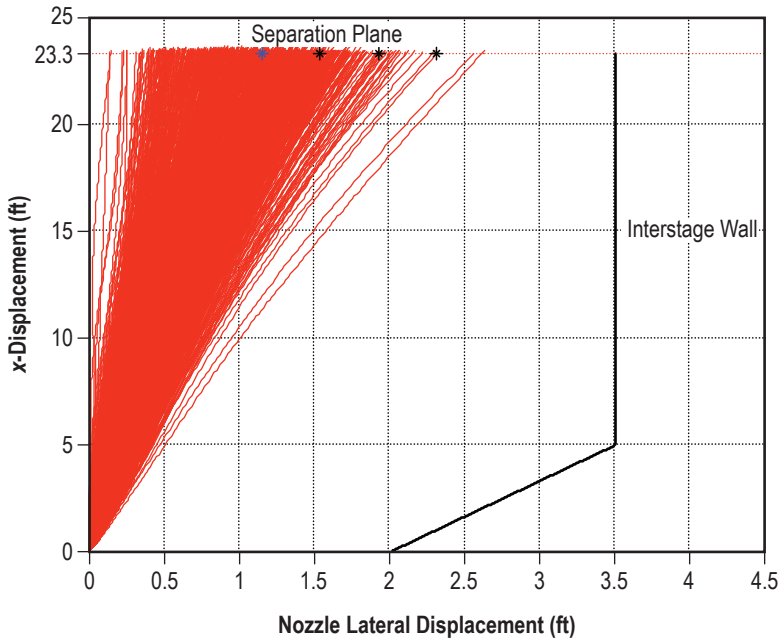

Figure 19. Typical radial trajectories during stage separation—J-2X nozzle lateral displacement.

Some comments and considerations about this behavior are in order. It is important to note that the monotonicity of the radial coordinate is not strict. Some trajectories could, in principle, curve out and then back toward the centerline, for example. There is no law of physics that dictates this monotonicity. It is instead a manifestation of the particular time-dependence of the force and moment inputs to the stages, as well as of the magnitude of these inputs relative to the inertias of the stages. Were the first stage to be subject to a large and chaotically varying force, for example, the relative trajectory of first stage and upper stage could resemble a sinusoid or a zig-zag. If that were the case, the mathematical treatment would be considerably more complicated, because it would be necessary to model the internal 'keep-out zone' inside the interstage in order to determine if a given

trajectory resulted in a recontact. Simply taking the maximum radial excursion of each trajectory would not suffice. However, all the modeling of Ares I staging has been consistent with the simple monotonic assumption.

From this point on, it will be assumed that the radial coordinate of the USE nozzle center relative to the interstage is monotonic in direction during separation. Note that this assumption of monotonicity enables a simple test for recontact: simply compute the radial coordinate at the moment the nozzle clears the separation plane, and check whether it exceeds the recontact value.

In terms of the goal of computing a recontact probability, the simplification here is significant. Instead of having to deal with an ensemble of three-dimensional paths, the ensemble to be analyzed contains points where each trajectory terminates as it penetrates the separation plane. The distribution of these points, and various techniques for extracting a recontact probability, are the objects of study.

A.2.2 The Importance of Being Two-Dimensional

One major subject of this section is that the dimensionality of this problem is crucially significant. The goal is to calculate a probability, or confidence, that recontact will be avoided during stage separation. It will be shown that inferring this probability from one-dimensional approximations to separation is inadequate and risky.

Before establishing this point, it is helpful to have some two-dimensional probabilistic models to serve as paradigms.

A.3 Simple Probability Models for Recontact

A.3.1 A Simple Two-Dimensional Calculation

Consider first the simplest recontact scenario, where all the inputs to the dynamics of stage separation are isotropic in the *y*-*z* (separation) plane. This assumption is equivalent to saying that the distribution of trajectories is symmetric under rotations in the *y*-*z* plane. Further, assume that the distribution of points is Gaussian, as it might be if it were the result of summing many random isotropic inputs.

Therefore, suppose, as shown in figure 20, that:

- A disk of radius R_1 (a model for the USE nozzle exit), which is randomly eccentric to a circle of radius $R_2 > R_1$ (the model for the interior of the interstage).
- The probability distribution for the location of the center of the disk is isotropic with respect to the circle R_2.
- The distribution is Gaussian in the *y* and *z* directions, with variance σ^2.

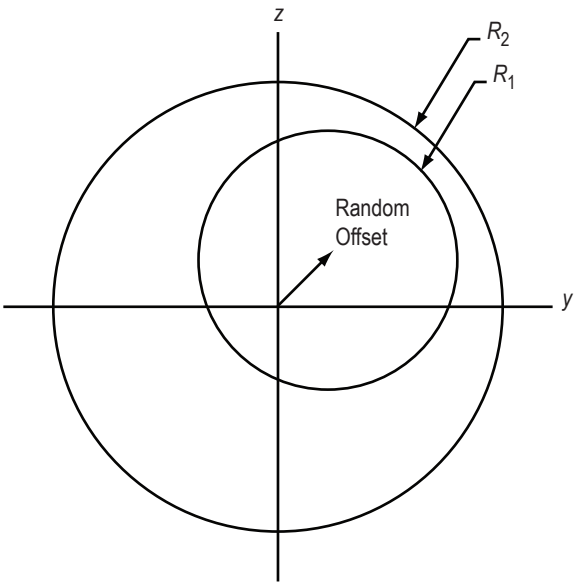

Figure 20. Eccentric nozzle and interstage.

The probability distribution for the trajectory is

$$P(y,z)\,dy\,dz = \frac{1}{2\pi\sigma^2}\exp\left(-\left(y^2+z^2\right)\Big/2\sigma^2\right)dy\,dz \ , \tag{7}$$

or, in radial coordinates,

$$P(r,\theta)\,r\,dr\,d\theta = \frac{1}{2\pi\sigma^2}\exp\left(-r^2\Big/2\sigma^2\right)r\,dr\,d\theta \ . \tag{8}$$

A typical distribution of points selected from such a distribution is shown in figure 21.

This figure shows 500 points (y/R, z/R), selected according to the above two-dimensional Gaussian using $\sigma/R = 0.3$.

Consider the question, "What is the probability that the disk will intersect or fall outside the circle R_2?" This is the same as asking for the probability P_{coll} that r will exceed the collisional displacement radius $R \equiv R_2 - R_1$. In this section, R will sometimes be called the 'recontact radius.' (Note R is the same as the nominal clearance of the nozzle and interstage; i.e., for the Ares I, this is 3.51 ft). In the next section, R will be used to nondimensionalize distance and length parameters, but for now it will be retained as R in formulae.

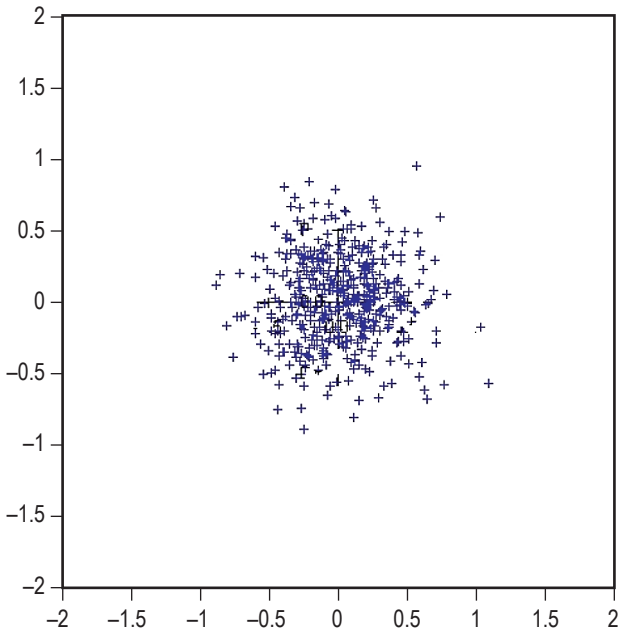

Figure 21. Two-dimensional centered isotropic Gaussian with $\sigma/R = 0.3$.

The recontact probability is easily computed:

$$P_{\text{coll}} = \int_{\theta=0}^{2\pi} \int_{r=R}^{\infty} P(r,\theta) r \, dr \, d\theta$$

$$= \int_{\theta=0}^{2\pi} \int_{r=R}^{\infty} \frac{1}{2\pi\sigma^2} \exp\left(-r^2/2\sigma^2\right) r \, dr \, d\theta$$

$$= \left[-\exp\left(-r^2/2\sigma^2\right) \right]_R^{\infty} = \exp\left(-R^2/2\sigma^2\right) . \tag{9}$$

How sensitive is the collision probability P_{coll} to the relative radial uncertainty σ/R? To answer this, tabulate and plot $\exp(-R^2/2\sigma^2)$. Figure 22 illustrates this sensitivity. Note that P_{coll} stays insignificant ($P_{\text{coll}} < 10^{-5}$) provided $\sigma/R < 0.2$. However, the collision probability rapidly rises to levels of concern ($P_{\text{coll}} > 4\%$) as $\sigma/R \to 0.4$ and beyond.

Note that the distribution of the radial coordinate is not Gaussian. Instead, it follows a chi distribution for 2 DOF, also known as the Rayleigh distribution. The result in equation (9) is simply the complement of the CDF for a Rayleigh distribution.

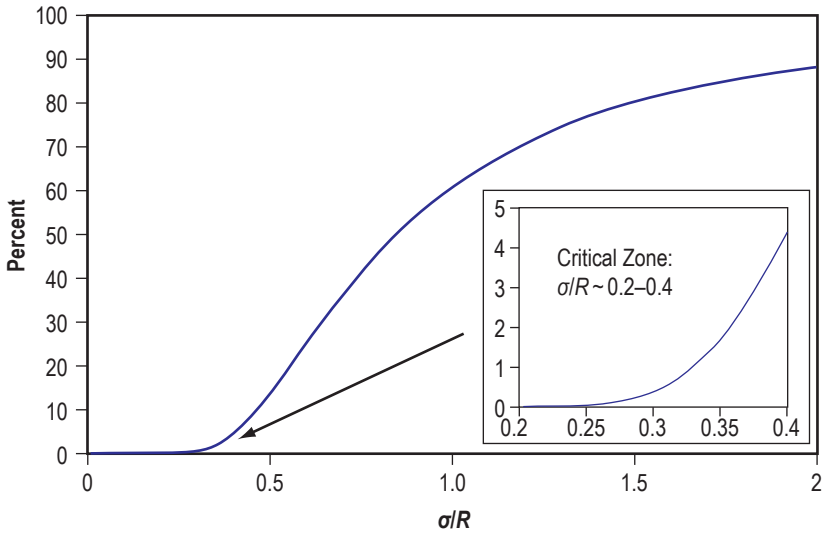

Figure 22. Recontact probability versus radial uncertainty (disk-to-circle collision probability, $\exp(-R^2/2\sigma^2)$).

An important point is that a radial coordinate that follows a Rayleigh distribution has 'fatter tails' than a Gaussian-distributed variable. The radial distance is not the sum of random variables—it is the square root of the sum of squares of random variables; i.e., $r = \sqrt{y^2 + z^2}$. Thus, r is a 'convex' function of y and z. The effect of this convexity is that the outlying values of r are more populated than they would be if the relation between r and the components y and z were linear. For the present purposes, this is an especially pertinent observation, because the recontact probability 'lives in the tail' of the recontact distribution.

A.3.2 The Fallacy of the One-Dimensional Root-Sum-Square

A reliable principle for dealing with distributions of additive one-dimensional random variables is that, when summing two independent random variables, the sample means of the distributions are additive, and the sample variances are additive. That is, if X_1 is drawn from a distribution with mean μ_1 and variance ${\sigma_1}^2$, and likewise X_2 with mean μ_2 and variance ${\sigma_2}^2$, then the sum $X_1 + X_2$ has mean $\mu_1 + \mu_2$ and variance ${\sigma_1}^2 + {\sigma_2}^2$. By induction, this gives rise to the principle that the standard deviation of the sum of N random variables is the RSS of the individual standard deviations; i.e.,

$$\sigma_{\text{total}} = \sqrt{\sum_{i=1}^{N} \sigma_i^2} \quad . \tag{10}$$

This principle can be extended to higher dimensional distributions provided that the components are uncorrelated, as is the two-dimensional Gaussian distribution in equation (7). In the uncorrelated, isotropic case, the joint or total standard deviation is also given by the RSS formula, equation (10), because the distributions can be decomposed into independent Cartesian coordinates.

If the components are uncorrelated but not isotropic, the standard deviations of the components of the joint distribution can still be computed with the RSS formula.

However, a subtle fallacy threatens when computing confidence intervals from these distributions. In particular, it is incorrect to compute the probability of recontact for the two-dimensional scenario using the one-dimensional normal distribution ('Z-test') and the RSS-calculated standard deviation. There are several ways to illustrate this fallacy.

Explicitly, the probability of recontact using a one-dimensional normal distribution and the parameters defined above is

$$P_{1D}(\sigma, R) = 1 - \mathrm{erf}\left(R/\sigma\sqrt{2}\right) , \tag{11}$$

which is not the same function as equation (9).

Another way to illustrate the fallacy of the one-dimensional RSS is to compute the recontact probability using the two-dimensional Gaussian (eqn. (9)) and the 3σ standard:

$$P_{2D}(R = 3\sigma) = \exp\left(-3^2/2\right) = 1.11\% \ . \tag{12}$$

Thus, ignoring the difference between one- and two-dimensional distributions can result in a significant underestimation of the likelihood of recontact. For 3σ, this 'overconfidence' amounts to about a factor of 4.1.

Figure 23 shows the comparison of the one- and two-dimensional methods, equations (8) and (9). The 3σ criterion corresponds to the abscissa $\sigma/R = 1/3$. Note that the one-dimensional method significantly underpredicts the recontact probability.

It is commonplace to mandate that the design of a component or system meet a two-sided 3σ standard of reliability; i.e., the failure rate must be less than

$$P_{1D,3\sigma} = 1 - \mathrm{erf}\left(3/\sqrt{2}\right) = 100\% - 99.73\% = 0.27\% \ . \tag{13}$$

For the reasons already discussed and reiterated below, use of the terminology '3σ equivalent' for this level of reliability is recommended.

The one-dimensional approach can be partially salvaged—for any given known distribution—by defining equivalent one-dimensional deviations that provide the same recontact probability. Thus, the stage separation success rate of 99.73% could be called a 3σ-equivalent capability. This language must be used with some care, however, as the potential for misunderstanding is obvious.

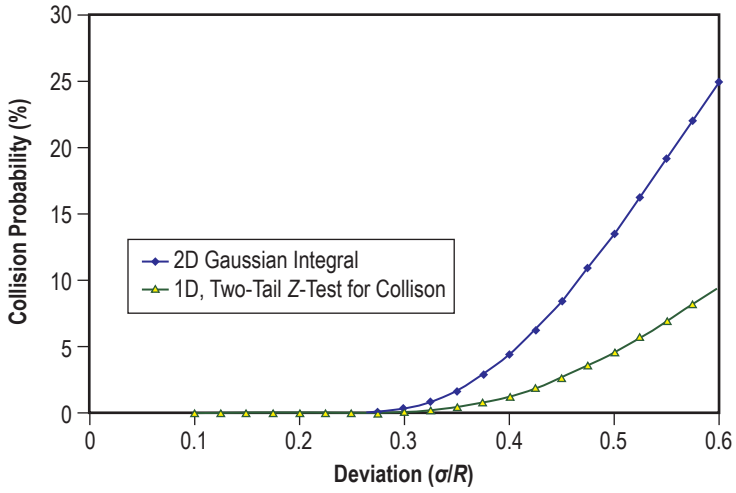

Figure 23. Comparison of collision probabilities computed with one- and two-dimensional Gaussian distributions.

A.3.3 A More Useful Model: The Offset Two-Dimensional Gaussian

This section presents a model that has a baseline ('nominal') offset and a dispersion centered about that offset. This provides a better approximation to the results of Monte Carlo trajectory calculations, since these often have deterministic inputs that generate an average offset, together with randomized inputs that are almost isotropic.

To that end, consider a two-dimensional Gaussian centered on the coordinate $(y,z)=(a,0)$, with a dispersion that is isotropic about that point. Using the equivalent polar coordinates, the PDF is

$$P(y,z)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{(y-a)^2+z^2}{2\sigma^2}\right)$$

$$=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{(r\cos\theta-a)^2+r^2\sin^2\theta}{2\sigma^2}\right)$$

$$=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{r^2-2ar\cos\theta+a^2}{2\sigma^2}\right). \qquad (14)$$

As observed in the last section, it is useful to nondimensionalize the radial coordinate r, the uncertainty σ, and the offset a by using the recontact radius $R=R_2-R_1$. That is, these symbols are shorthand for r/R, σ/R, and a/R. Figure 24 shows a typical 500-point sample from such a PDF with $\sigma=a=0.3$.

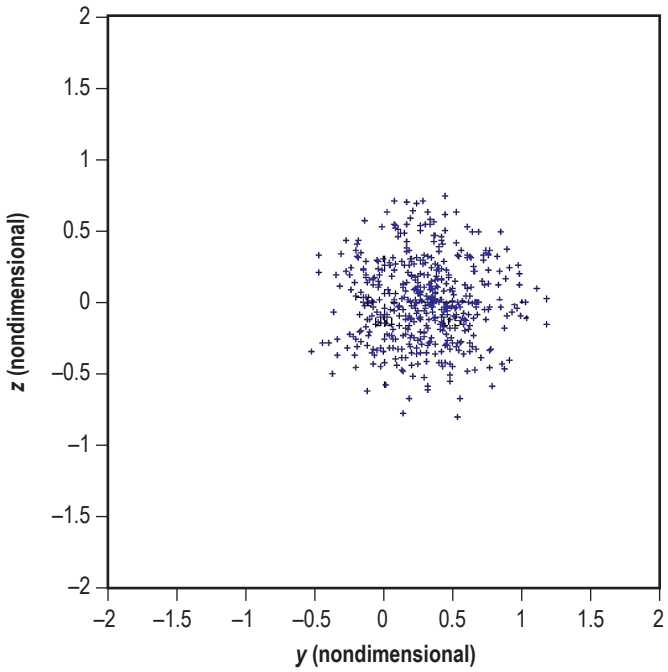

Figure 24. Two-dimensional offset isotropic Gaussian with $\sigma/R = a/R = 0.3$.

In order to find the recontact probability, compute the total weight of this PDF outside the unit disk $r = 1$ (i.e., $r = R$ in dimensional units). This will be a model of the probability of nozzle/booster recontact due to random effects. The collision probability will be

$$P_{\text{coll}} = \int_{\theta=0}^{2\pi}\int_{r=1}^{\infty} r\,dr\,d\theta\,P(r,\theta)$$

$$= \int_{\theta=0}^{2\pi}\int_{r=1}^{\infty} r\,dr\,d\theta\,\frac{1}{2\pi\sigma^2}\exp\left(-\frac{r^2 - 2ar\cos\theta + a^2}{2\sigma^2}\right)$$

$$= \frac{\exp\left(-a^2/2\sigma^2\right)}{2\pi\sigma^2}\int_{r=1}^{\infty} r\exp\left(-\frac{r^2}{2\sigma^2}\right)\left(\int_{\theta=0}^{2\pi}\exp\left(\frac{ar\cos\theta}{\sigma^2}\right)d\theta\right)dr \quad . \tag{15}$$

The details of this calculation are provided in reference 5. The recontact probability can be expressed in terms of the incomplete gamma function, or equivalently in a series expansion:

$$P_{\text{coll}} = \exp\left(-\frac{1+a^2}{2\sigma^2}\right)\sum_{m=0}^{\infty}\frac{1}{m!}\left(\frac{a^2}{2\sigma^2}\right)^m\sum_{k=0}^{m}\frac{1}{k!}\left(\frac{1}{2\sigma^2}\right)^k \quad . \tag{16}$$

Note when $a = 0$, this reduces to

$$P_{\text{coll}} = \exp\left(-\frac{1}{2\sigma^2}\right), \tag{17}$$

which is the same result as equation (9), with $\sigma \rightarrow \sigma/R$.

Figure 25 shows P_{coll} for a range of σ and a.

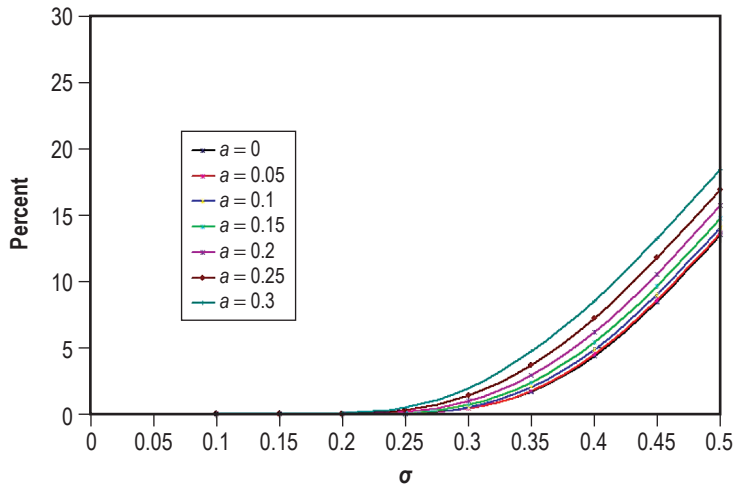

Figure 25. Recontact collision probability for offset two-dimensional Gaussian.

Figure 26 shows an equivalent way to represent the same relation, as contours in the a, σ plane for constant recontact probability. This presentation emphasizes the sensitivity of the recontact probability to a and σ.

A.3.4 The One-Dimensional Fallacy for the Offset Two-Dimensional Gaussian

There is also a 'one-dimensional fallacy' for the offset two-dimensional Gaussian.

Consider an alternative analysis of the statistical ensemble for staging recontact analysis. It is tempting to conduct experiments or simulations for this two-dimensional distribution, compute the mean and standard deviation of the radial coordinate alone, and then to make inferences of collision probabilities based on this one-dimensional analysis and a normal distribution Z-test. This procedure is incorrect. Unlike the zero-offset case, the Rayleigh distribution also does not apply, so there is no conveniently invertible function as in equation (17).

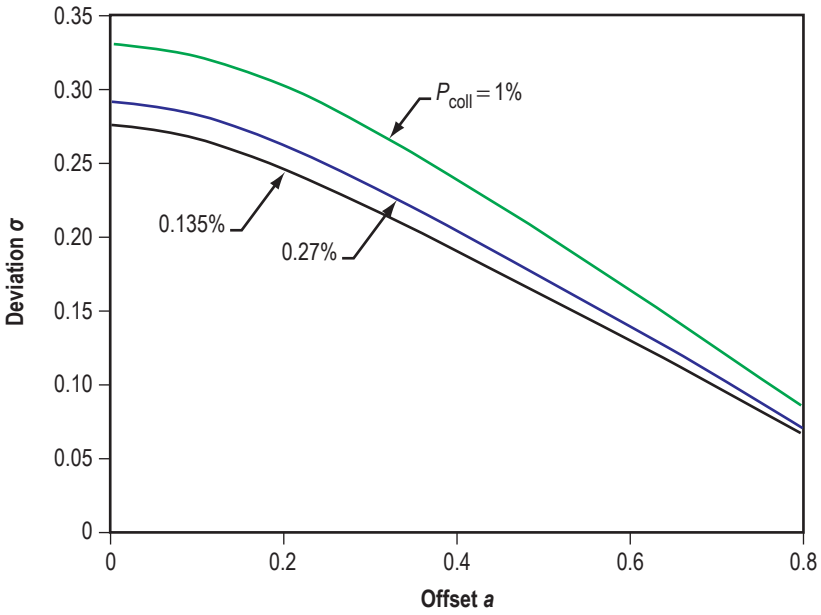

Figure 26. Contours of constant recontact probability for offset two-dimensional Gaussian.

To illustrate this point, compute the mean and standard deviation of the radial coordinate. Start with the first and second moments:

$$\mu_1 = \langle r \rangle = \iint r\, P(r,\theta) r\, dr\, d\theta \tag{18}$$

and

$$\mu_2 = \left\langle r^2 \right\rangle = \iint r^2\, P(r,\theta) r\, dr\, d\theta \ , \tag{19}$$

where, as usual, the brackets indicate an expectation value; i.e., a probability-weighted average, then, as usual, $\sigma_r^2 = \mu_2 - \mu_1^2$, which is different from the σ^2 of the parent Gaussian. The computation is similar in flavor to the calculation in the first part of this discussion, with two key refinements. First, the radial coordinate is taken to a higher power. Second, the limits of integration extend from the origin to infinity rather than from unity to infinity.

The details of the calculation are provided in reference 5. The principal results are

$$\mu_1 = \sigma\sqrt{\frac{\pi}{2}} \exp\left(-\frac{a^2}{2\sigma^2}\right) \sum_{m=0}^{\infty} \frac{2m+1}{m!} \left(\frac{a^2}{2\sigma^2}\right)^m \frac{1}{2^{2m}} \binom{2m}{m} \ , \tag{20}$$

$$\mu_2 = 2\sigma^2 + a^2 \ , \tag{21}$$

and

$$\sigma_r = \sqrt{\mu_2 - \mu_1^2} \quad , \tag{22}$$

Using these results, for any a and σ a Z-value for the radial coordinate can be computed:

$$Z = \left(r - \mu_1\right)/\sigma_r \quad . \tag{23}$$

As discussed above, it is tempting to map this Z-value to a confidence interval based on a standard normal distribution. Figure 27 shows that this procedure can severely underpredict the collision probability (given correctly in eqn. (16)). The data in the figure are for an offset $a = 0.25$. The subgraphs show the mean μ_r and the radial standard deviation σ_r. The dispersion of the parent two-dimensional offset Gaussian is used as a parameter. Note σ_r underpredicts the dispersion σ.

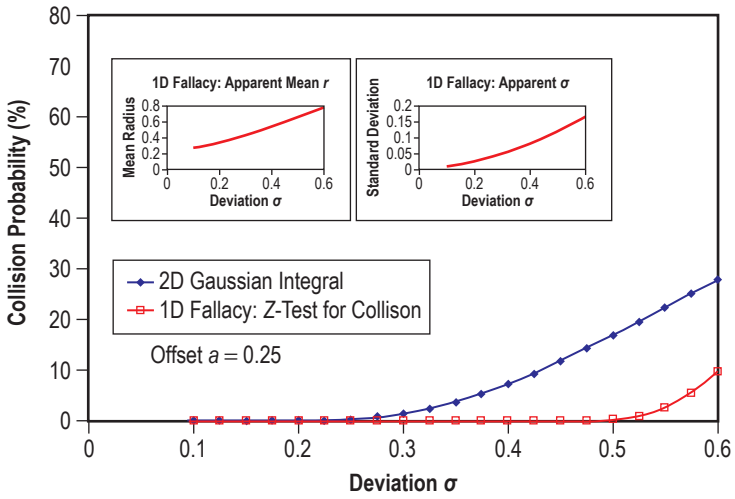

Figure 27. Comparison of explicit calculation of recontact probability and one-dimensional fallacy.

A.4 Inference and Fitting Procedures

In this section, some practical issues associated with determining the recontact probability from a finite Monte Carlo sample are discussed.

A.4.1 Estimating Recontact Probability by Counting

One of the simplest ways to estimate recontact probability from a Monte Carlo sample is to count the number of trials that result in a recontact. The probability of recontacting k times out of N trials with a given recontact probability p is given by the binomial probability distribution:

$$P\left(k|N,p\right) = \binom{N}{k} p^k \left(1-p\right)^{N-k} \quad . \tag{24}$$

Conversely, if k out of N trials result in recontact, the maximum likelihood estimate of the recontact probability is $P = k/N$.

The shortcoming of this approach is that the relative uncertainty in this estimator gets very large as the number of 'hits' gets small. The variance of the binomial distribution is

$$\left\langle \left(k - \langle k \rangle\right)^2 \right\rangle = Np(1-p) \doteq Np = \langle k \rangle \ , \tag{25}$$

i.e., the standard deviation is approximately $\sqrt{\langle k \rangle}$ and thus the relative error goes as $1/\sqrt{\langle k \rangle}$. If the number of trials (N) is insufficient to generate a large number of recontacts, the recontact probability is not determined with much precision. Once $k = 0$, the counting method gives

$$P(0|N,p) = (1-p)^N \ . \tag{26}$$

The smallest recontact probability which gives no more than $P = 10\%$ chance of zero recontacts out of N trials (i.e., the 10% consumer risk (CR) probability) is approximately $p_{10\%} \doteq \ln(10)/N$; e.g., $p_{10\%} \doteq 0.115\%$ for $N = 2{,}000$. This probability is the upper bound on the recontact probability, given the willingness to accept 10% CR; i.e., the recontact probability could be as high as $p_{10\%}$ and still generate zero hits out of N trials. Now, it may appear from inspection of the scatter plot of recontacts (the 'recontact patch') that the patch is well inside the recontact circle, and that the recontact probability is therefore considerably less than $p_{10\%}$. However, the counting method cannot provide a better estimate (for a very low probability) and therefore a parametric method (one where the type of distribution is assumed), such as the method described in the next section, is needed to refine the estimate in such cases.

These counting method (or order statistics) issues, together with the related story of producer and consumer risk, are developed in more detail in appendix B. In the next two sections, a distribution is fit to the data. Doing so is preferable when it must be shown that the probability of recontact is very small (e.g., 1×10^{-6}). However, the counting method (or order statistics) is a good approach (maybe better if the experimental distribution is not truly Gaussian) for estimating a percentage clearance level such as 99.865%.

A.4.2 Best Two-Dimensional Gaussian Parameters

In some cases, there is a better approach than binomial failure analysis. If the ensemble of trajectories corresponds to a general two-dimensional Gaussian distribution, the parameters of that distribution can be estimated and analytical methods can be used to compute the recontact probability. In principle, this enables more of the information computed in the trajectory simulation to be used, and thereby obviates the problem of small numbers.

Consider a correlated Gaussian in two dimensions, with offsets (y_0, z_0), variances σ_y^2, σ_z^2, and correlation ρ. The PDF is

$$P(y,z)\,dy\,dz = \frac{\exp\left(-\frac{1}{2}\frac{1}{1-\rho^2}\left(\frac{(y-y_0)^2}{\sigma_y^2} - 2\rho\frac{(y-y_0)}{\sigma_y}\frac{(z-z_0)}{\sigma_z} + \frac{(z-z_0)^2}{\sigma_z^2}\right)\right)}{2\pi\sigma_y\sigma_z\sqrt{1-\rho^2}}\,dy\,dz \quad . \tag{27}$$

As before, the recontact probability will be the total weight outside the unit circle $y^2+z^2=1$ (as before, all lengths are normalized by R).

For a set of points (y_i, z_i) (suitably normalized by $R = R_2 - R_1$), the parameters for the general two-dimensional Gaussian can be estimated from the usual relations:

$$y_0 = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{28}$$

and

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - y_0)^2 \quad , \tag{29}$$

and analogously for z. The covariance is

$$\mathrm{cov}(y,z) = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_0)(z_i - z_0) \tag{30}$$

and the correlation is

$$\rho = \mathrm{cov}(y,z)/\sigma_y\sigma_z \quad . \tag{31}$$

It should be mentioned that one advantage of this method is that the Gaussian distribution thus derived is useful not only for estimating the recontact probability, but also for providing an accurate estimate of the 'circular drift.' The circular drift is the radius of the circle, centered on the origin, that contains a specified percentage of the total probability distribution; e.g., 99.73%. Because all N data points are used to determine the parameters of the two-dimensional Gaussian, a much higher accuracy estimate can be obtained in this case than with nonparametric methods; i.e., order statistics.

A.4.3 Semianalytical Method for Computing Recontact Probability

Once the parameters of the two-dimensional Gaussian are computed, the recontact probability can be calculated. Details are in reference 5. The basic idea is to rotate and dilate the coordinate system until the probability distribution is a standard two-dimensional Gaussian, and the recontact circle is an ellipse. The resulting integral for the recontact probability can be analytically performed over the radial coordinate, leaving a simple numerical integral over the (coordinate-transformed) angle. The technique requires the solution of a quadratic equation to find the intersection of the recontact ellipse with each ray from the origin, which yields zero, one, or two roots, depending on the direction and the parameters of the distribution.

The final step is a numerical integration over the polar angle θ:

$$P_{\text{coll}} = \frac{1}{2\pi} \int_{\theta=0}^{2\pi} g(\theta)\, d\theta \ , \tag{32}$$

where

$$g(\theta) = \begin{cases} \exp\left(-r_S^2/2\right) & \text{if one root } r_S \text{ is positive} \\ \left(1-\exp\left(-r_{\min}^2/2\right)\right)+\exp\left(-r_{\max}^2/2\right) & \text{if both roots } r_{\min}, r_{\max} \text{ are positive} \\ 1 & \text{otherwise} \end{cases} \tag{33}$$

In equation (33), a quadratic equation with zero, one, or two roots must be solved. This technique provides a more robust method for evaluating the recontact probability than the conventional counting method, especially when the recontact probability is low and the number of failures is small. Figure 28 shows a comparison of the relative errors of the two methods for a test case. The thick black lines are fitted trendlines. The counting method shows the expected $1/\sqrt{k}$ dependence.

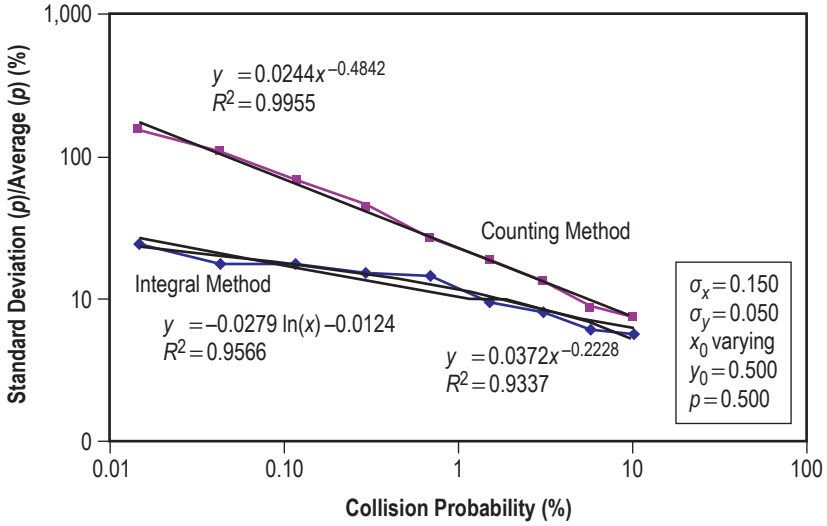

Figure 28. Comparison of relative errors for counting and integral methods as functions of collision probability.

The integral method has significantly tighter confidence bounds, but the functional dependence on the number of failures is not so obvious (logarithmic and power-law fits are shown). From an analytical standpoint, it is feasible to compute the expectation values of terms in the error matrix

for the Gaussian parameters, and propagate the uncertainties through the integral method. However, it is even more straightforward to conduct numerical simulation using the estimated Gaussian parameters to establish 10%–90% confidence bounds on the estimated recontact probability, as in figure 28.

A.5 Summary

In this appendix, several key recommendations have been presented for the statistical analysis of stage separation trajectory simulations and the prediction of recontact probability. The importance of the dimensionality of the problem was stressed. One-dimensional methods, including one-dimensional RSS and radial projection methods, severely underpredict the recontact probability when incorrectly tied to a Gaussian Z-test. Two-dimensional models are potentially more robust, although different ways to use simulation results give different tradeoffs of producer and consumer risk. For estimating '3σ equivalent,' i.e., 99.865%, clearances, the counting method (or order statistics), described in appendix B, is valuable. For estimating very low probabilities of recontact (e.g., 1×10^{-6}), fitting a distribution to the data is appropriate.

Besides using a Gaussian distribution, an extreme value distribution, for example, using a peak-over-threshold approach and generating a generalized pareto distribution,[13] may be fit to the tail of the data. This approach might fit the outlying data better than a Gaussian, although there are potential pitfalls in this approach as well.

APPENDIX B — CONSUMER RISK

B.1 Introduction

'Success probability' is a predicted engineering parameter, like tank pressure or rocket thrust. The method used to determine such a parameter, whether a projection from an empirical success rate or a fraction of a Monte Carlo ensemble, will also lead to an uncertainty estimate (or equivalently, to a confidence interval around the prediction). Intuitively, more data or more extensive simulations are expected to lead to a better estimate of the parameter. However, there is always some noninfinitesimal confidence interval around the prediction. Conversely, an engineering estimate and confidence interval will rarely have 100% likelihood of containing the actual value of the parameter in question. Usually something like a 90% confidence value is used; i.e., a 10% risk that the parameter will not be in the quoted interval is deemed acceptable.

These intuitive ideas are formalized in the concept of 'consumer risk,' which derives from sampling theory, and this leads in turn to a discussion of order statistics. NASA program requirements are often written in terms of consumer risk, which makes it vital for engineers to understand this concept. The material developed in this appendix using order statistics is closely related to 'acceptance sampling' in quality engineering.[12]

B.2 A Paradigm for Consumer and Producer Risk

The paradigmatic case involves the use of a finite sample to estimate the failure rate of some component or system. To be concrete, suppose $N = 2{,}000$ identical widgets are subjected to a stressful duty cycle, and five of them fail. What can be said of the failure rate of the widgets? The immediate temptation is to say that the empirical failure rate is 5/2,000 or 0.25%, and to assume that this sample is typical. In the spirit of the introduction, though, what confidence interval can be put on this estimate? In order to be '90% sure' that a quoted failure rate is an upper bound on the actual failure rate, what rate should be quoted?

The rigorous approach, called binomial failure analysis, assumes that there is some actual failure probability (p), and that the 2,000 widgets form one sample from an infinite ensemble of samples. (To be careful about language, the individual test of each of the widgets will be called a 'sample' and the collection of N trials will be called a 'run.') For any given p, the probability of seeing exactly k failures in a run of size N is given by the binomial probability formula:[25]

$$P_{\mathrm{BIN}}(k|p,N) = \binom{N}{k} p^k (1-p)^{N-k} \quad . \tag{34}$$

Given the parameters, it is easy to plot this probability. Figure 29 shows the results for three failure rates near the intuitive result. Notice that any of these actual failure rates gives nearly the same

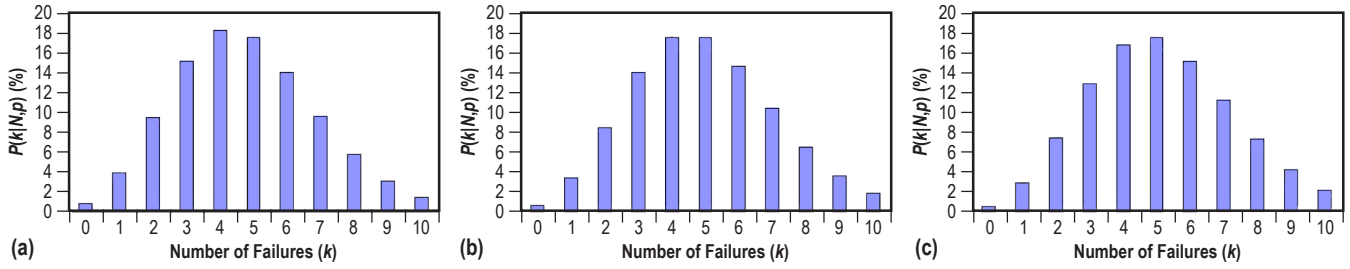

Figure 29. Probabilities $P_{\mathrm{BIN}}(k|p,N)$ for different k for three different actual failure rates with binomial distributions of $N=2{,}000$: (a) $p=0.24\%$, (b) $p=0.25\%$, and (c) $p=0.26\%$.

result: around $P_{\mathrm{BIN}}=18\%$, which means that the expectation is that about 18 out of every 100 runs of 2,000 will exhibit exactly five failures. Also notice that k's in the interval from about 2 to 8 have nonnegligible probabilities P_{BIN}. Conversely, this means that it would be mildly coincidental if the actual failure rate were really 0.25% ; and, if the actual failure rate were really 0.25%, one should bet against getting exactly five failures out of 2,000 samples.

However, the significance of the intuitive result can be easily demonstrated. Suppose the probability P_{BIN} of getting five failures out of 2,000 samples is calculated as the actual probability is varied. The result is given in figure 30; the probability P_{BIN} peaks at the intuitive value of failure probability, $\hat{p}=k/N$.

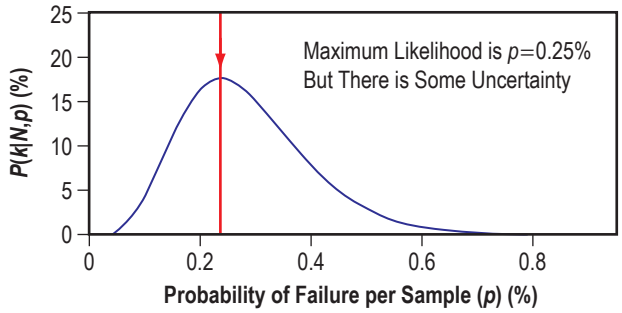

Figure 30. Probability $P_{\mathrm{BIN}}(k|p,N)$ as a function of actual failure rate—binomial possibility given $k=5$, $N=2{,}000$.

Thus, $\hat{p}=k/N$ is what is called a 'maximum likelihood estimator' for the actual failure rate p. The wording here is important: $\hat{p}$ is not the 'most likely' value of p. In fact, p has some exact unknown value, and there is not a PDF associated with different values of p. However, $\hat{p}$ is the value of p that maximizes the probability $P_{\mathrm{BIN}}(k|p,N)$—and 'maximum likelihood estimator' is the agreed terminology.

What does this mean in terms of a confidence interval on p? There appears to be a range of actual failure rates consistent with the observed failure fraction k/N. But p is *not* a random variable (although k is). If it were, its PDF could be integrated to give its cumulative distribution function (CDF), then the CDF could be inverted at 0.10 and 0.90 to get the 10%–90% confidence interval. Instead, the standard approach to binomial failure analysis is to compute, for a range of possible failure rates p, the probability of getting k or fewer failures, and define the confidence in terms of the resulting function of p. For example, for $k = 5$, then for each of the three subplots in figure 29, imagine adding up the heights of the columns for $0, 1, \ldots, 5$ failures. The result is the cumulative binomial probability and is a decreasing function of p.

The conventional way to put confidence limits on empirical binomial failures is to compute upper and lower bounds on the probability p, not using a cumulative distribution of the 'probability of the probability', but by answering the following questions:

- What is the largest probability p_{lower} that will result in no more than a chance β_L of generating k or more failures out of N samples?

- What is the smallest probability p_{upper} that will result in no more than a chance β_U of generating k or fewer failures out of N samples?

The probabilities β_L and β_U are acceptable probabilities of error, typically taken as 5% or 10% for practical calculations. Answering these questions requires the solution of the equations,[26]

$$\sum_{j=k}^{N} \binom{N}{j} p_{\text{lower}}^{j} \left(1 - p_{\text{lower}}\right)^{N-j} = \beta_L \tag{35}$$

and

$$\sum_{j=0}^{k} \binom{N}{j} p_{\text{upper}}^{j} \left(1 - p_{\text{upper}}\right)^{N-j} = \beta_U \quad . \tag{36}$$

Figure 31 shows an example of this upper- and lower-bound confidence interval calculation. The monotonically decreasing line is the CDF (zero to k) for the binomial distribution function, the left side of equation (36), while the increasing line is the overlap complement (k to N), the left side of equation (35), for the case $k = 5$, $N = 2{,}000$. The dotted line is the 10%–90% confidence interval.

Note these expressions are related to the cumulative probability of seeing k or fewer failures, which is

$$F_{\text{BIN}}\left(k \mid p, N\right) = \sum_{j=0}^{k} \binom{N}{j} p^{j} \left(1 - p\right)^{N-j} \quad , \tag{37}$$

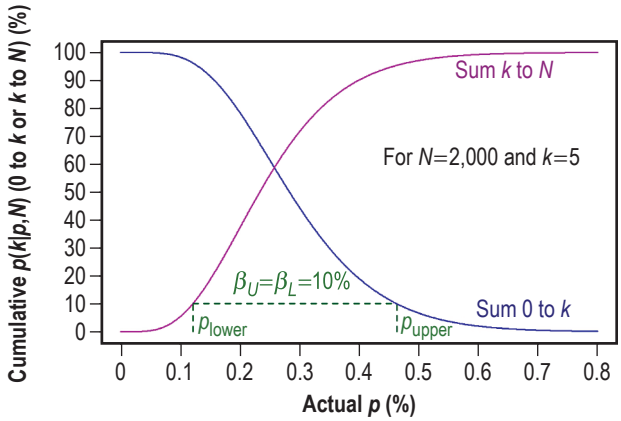

Figure 31. Confidence interval for binomial trials.

i.e., $F_{\mathrm{BIN}} = \beta_U$ can be solved to find p_{upper}. If this function of p is plotted for a given k and N, the result is the curve in figure 32. The curve is called the operating characteristic (OC) for the sampling plan (k,N). The terminology comes from the discipline of quality control.

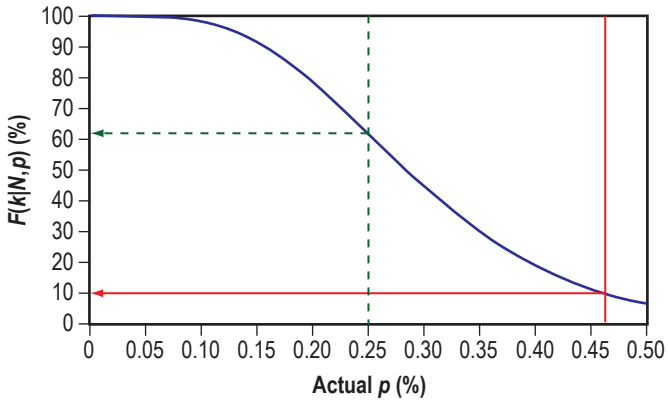

Figure 32. Cumulative probability $F(k\,|\,p, N)$ as a function of actual failure rate if maximum accepted $k = 5$ when $N = 2{,}000$.

Note that if the actual failure rate were 0.25%, the plot shows that ≈62% of 2,000 trial samples exhibit five or fewer failures. If there were a strict requirement for the widget failure rate not to exceed 0.25%, then a single sample of 2,000 trials with five failures would not give much confidence that the widgets meet the requirement. There is still a 58% chance of acceptance if $p = 0.26\%$, ≈50% chance if $p = 0.28\%$, and so on. In fact, the probability of getting five or fewer failures in 2,000 trials drops below 10% only when p increases to 0.463%. (Note this is equivalent to saying that $p = 0.463\%$ is the solution to the equation $F_{\mathrm{BIN}}(k\,|\,p,N) = 10\%$ for the given k and N.) This suggests that a (5, 2,000) sampling plan is likely to be unsatisfactory to verify a requirement like $p < 0.25\%$. This risk of accepting an invalid design is called consumer risk (CR), and can be cast in the form of a 'type II' error in statistics. From the consumer's standpoint, p_{upper} is the important limit.

An aside concerning type I and II errors—It is a long-standing statistical practice to express the objects of statistical analysis in terms of 'hypotheses,' which are falsifiable statements about the state of the world, and to compute the probabilities of accepting or rejecting hypotheses in terms of the outcome of statistical tests. Because a hypothesis or its negation may be true, and because a system can pass or fail a statistical test, there are four possible outcomes:

(1) The hypothesis could be true, and the system could pass the statistical test.
(2) The hypothesis could be true, and the system could fail the statistical test.
(3) The hypothesis could be false, and the system could pass the statistical test.
(4) The hypothesis could be false, and the system could fail the statistical test.

Outcomes (2) and (3) are 'errors.' Rejecting a valid hypothesis is called a 'type I error,' while accepting a false hypothesis is called a 'type II error.' Because a major application of this nomenclature is to the diagnosis of disease, type I errors are sometimes called 'false positives' and type II errors are sometimes called 'false negatives.' In terms of systems engineering, it is convenient to take 'the design configuration meets the requirement' as the generic hypothesis. The four outcomes can then be depicted as shown in table 6:

Table 6. Hypothesis (H_0): The design configuration meets the requirement.

	Inference From Analysis/Test	
Actual Situation	**H_0 is Accepted**	**H_0 is Rejected**
H_0 is true (design meets requirements)	Correct inference	Type I error (producer risk)*
H_0 is false (design does not meet requirements)	Type II error (consumer risk)**	Correct inference

* Acceptable Probability = α
** Acceptable Probability = β

As indicated, in a given situation, the probability of these errors for a given statistical test can often be quantified. The probability of type I (producer risk) error is usually denoted α, while the probability of type II (consumer risk) error is usually denoted β.

The traditional approach to choosing a sampling plan exploits the fact that two independent parameters (k and N) can be specified. This approach also reflects the fact that the producer and the consumer of the widgets have distinct concerns about the transaction. In this paradigm, the consumer will posit a requirement that the widgets have a specified not-to-exceed failure rate. The producer, however, would like not to waste good products because of inadequate sampling. It is in the interest of the producer, therefore, to specify an allowable failure rate. In order to satisfy both parties though, the producer-allowable failure rate should be less than or equal to the consumer-specified requirement.

In this context, the CR for a given sampling plan (k,N) is the upper bound on the probability that widgets will be accepted that have greater than the consumer-required failure rate (a type II

error). The producer risk (PR) for a given sampling plan (k,N) is the lower bound on the probability that widgets will be rejected that have less than the producer-allowable failure rate (a type I error). As tersely as possible,

- CR is the risk of accepting a bad product.
- PR is the risk of rejecting a good product.

Figure 6 shows the traditional setup of an OC that provides 10% CR and 10% PR for consumer and producer required/allowable failure rates of $p_C = 0.250\%$ and $p_P = 0.125\%$, respectively. The parameters k and N, in this case (13, 7,579), are selected to be the minimum values that meet the goals of the sampling plan. Note that if the actual failure rate is less than p_P, there is a finite probability that the widget will be rejected, and if the actual failure rate is greater than p_C, there is a finite probability that the widget will be accepted.

However, typical NASA programs do not specify a producer risk and producer-allowable failure rate. Without a separate PR requirement, the producer risk is the complement of the consumer risk, PR = 1 – CR. Failure to specify or to account for producer risk leads to a verification framework with a very large producer risk; i.e., there is a significant probability that the resulting design will be conservative because acceptable designs will be rejected. (This could be viewed as design margin, although care must be taken that the result is not an over-conservative design).

Sampling plans accounting only for CR can be designed. If the conservatism penalty is not large, accounting only for CR is reasonable.

What is the best way to choose a sampling plan for a given consumer-required failure rate? It may be worthwhile to backtrack a bit and approach the problem from a slightly different direction.

The essential process involves the following steps:

(1) Specify an acceptable failure rate (a not-to-exceed requirement) (p_A).

(2) Choose a sampling plan (k,N), in which a design is accepted if, in a sample of N trials, no more than k failures are encountered.

(3) Compute the upper bound of the probability of accepting the design if the actual failure rate exceeds the requirement; i.e., if $p > p_A$.

The upper bound probability in step (3) is the consumer risk. If the CR is unacceptable for a given (k,N), it is necessary to revise the sampling plan (go to a smaller k or a larger N).

Typically, requirements are written with CR limited to 10% or less; i.e., the verification of various aspects of system reliability and performance is to allow no more than 10% chance that an unsatisfactory design will be accepted because of statistical errors.

For any given acceptable failure rate p_A and number of trials N, specifying an upper bound on consumer risk of 10% is equivalent to putting an upper bound on the allowable number of failures k. Conversely, for any given acceptable failure rate p_A and allowable number of failures k, specifying CR is equivalent to putting a lower bound on the number of trials N. Using the formulas previously developed, here are some typical results for 10% CR, $p_A = 0.27\%$, which is the complement of the two-tailed 3σ value 99.73% (table 7).

Table 7. Extract of study design table for 10% consumer risk.

N	k	p_A (%)	CR (%)
852	0	0.27	10
1,440	1	0.27	10
1,970	2	0.27	10
2,473	3	0.27	10
2,959	4	0.27	10
3,433	5	0.27	10
3,899	6	0.27	10
4,358	7	0.27	10
4,811	8	0.27	10
5,259	9	0.27	10
5,704	10	0.27	10

More extensive tables are given in appendix C. Note that the allowable number of failures is noticeably less than what one gets using an intuitive rule like $k \cong Np_A$.

Consider the OC for a sampling plan based on this table. Figure 33 is a typical result. (For this figure, $N = 1{,}970$ has been rounded up to $N = 2{,}000$). This figure clarifies the meaning of type I and type II errors. Designs actually meet or fail to meet the requirement if they reside to the left or to the right of the vertical line at $p = p_A = 0.27\%$. However, they are accepted or rejected according to whether the results of an 'experiment' (or simulation, in the case of trajectory dispersions) lie below or above the OC. The red area to the right and below represents consumer risk; i.e., the acceptance of a design that actually fails to meet requirements. (Note the upper bound on CR is 10%, as promised.) The red area to the left and above represents the producer risk for this situation where there is no separate specification for the producer-allowable failure rate.

Note that PR is not specified, so any of these sampling plans has 90% PR at the specified failure rate p_A. However, the advantage of increasing N is that the PR at failure rates less than specified failure rate becomes significantly less. Suppose, hypothetically, that the producer set a maximum allowable failure rate of $p_B = 0.20\%$ (less than the consumer specified rate $p_A = 0.27\%$). For the (2, 2,000) sampling plan, the PR is ≈75%. But the OC shifts as N is increased. Figure 34 shows the (44, 20,000) and (509, 200,000) sampling plans, for which k was selected so that CR is 10%. Thus,

- For the (2, 2,000) sampling plan, the PR at 0.20% is ≈75%,
- For the (44, 20,000) sampling plan, the PR at 0.20% is down to ≈25%,
- For the (509, 200,000) sampling plan, the PR at 0.20% is essentially zero.

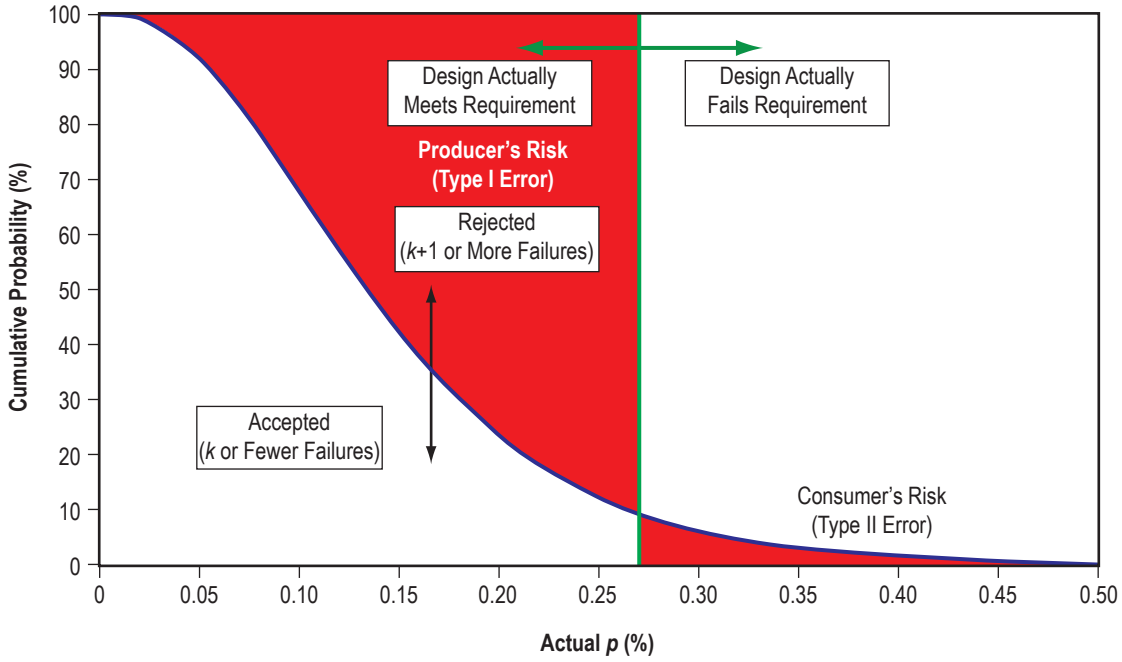

Figure 33. Operating characteristic for the (2, 2,000) sampling plan if maximum accepted $k = 2$ when $N = 2{,}000$.

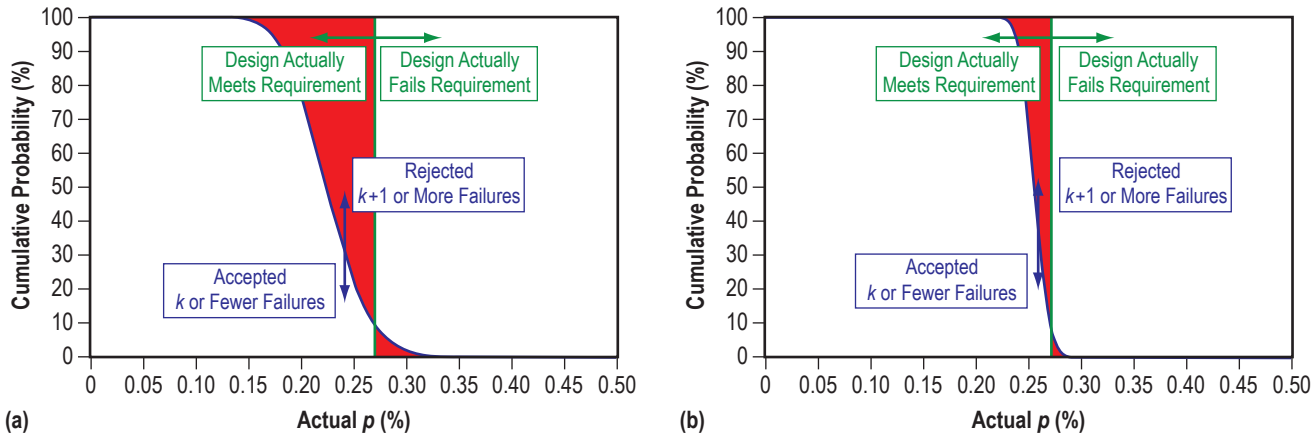

Figure 34. Operating characteristics for sampling plans if maximum accepted is: (a) $k = 44$ when $N = 20{,}000$ and (b) $k = 509$ when $N = 200{,}000$.

Note that as the sample size N is increased, the test aligns more and more closely with the hypothesis. The ideal test would look like a step function, with value 1 for $p < p_A$, stepping down to zero for $p > p_A$.

Specifying 'producer allowables' distinct from 'consumer allowables' enables the establishment of a minimum sample size, but more importantly, ensures that the design is not overly conservative; i.e., that acceptable designs are not tossed aside unnecessarily.

B.3 From Binomial Failures to Order Statistics[11]

Trajectory dispersions are frequently used to establish extreme values for launch vehicle operating parameters; e.g., maximum dynamic pressure, maximum pitch attitude error, maximum longitudinal g's, etc. Simply using the PERCENTILE function from Excel to compute the 99.865 percentile value from a run of 2,000 trajectories will usually underestimate the population 3σ-equivalent value, however. More importantly, it will also involve considerable consumer risk. The probability distribution of the mth ordered value of N trials is computed using 'order statistics.'

There is a close relationship between binomial failure analysis and order statistics. This enables a nonparametric estimate of population extremes; i.e., an estimate that does not depend on the specifics of the underlying distribution, be it normal, log-normal, beta, etc. It also enables accounting for consumer risk.

To see the connection, suppose the PDF of some continuous variable x is $g(x)$, with CDF:

$$G(x) = \int_{-\infty}^{x} g(x')\,dx' \quad . \tag{38}$$

Consider a sample of N independent trials taken from $g(x)$. Order the sample from least to greatest x value, numbering them $1, 2, \ldots, N$. Then the mth order statistic is defined to be the mth smallest value in the list, and is a random variable with a computable PDF and CDF. (Note that $m = 1$ corresponds to the minimum, $m = N$ corresponds to the maximum, and, if N is even, $m = N/2$ is the median.) The mth order statistic is denoted $x_{(m)}$. The CDF for $x_{(m)}$ is given by[11]

$$F_{(m)}(x \mid N) = \sum_{j=m}^{N} \binom{N}{j} [G(x)]^{j} [1 - G(x)]^{N-j} \quad . \tag{39}$$

This is a basic result from order statistics. From the binomial theorem,

$$\sum_{j=0}^{N} \binom{N}{j} [G(x)]^{j} [1 - G(x)]^{N-j} = 1 \tag{40}$$

and

$$\Rightarrow \sum_{j=m}^{N} \binom{N}{j} [G(x)]^{j} [1 - G(x)]^{N-j} = 1 - \sum_{j=0}^{m-1} \binom{N}{j} [G(x)]^{j} [1 - G(x)]^{N-j} \quad . \tag{41}$$

The last summation is the cumulative binomial probability for $p = G(x)$, $k = m-1$, hence

$$F_{(m)}(x \mid N) = 1 - F_{\mathrm{BIN}}\left((m-1) \middle| G(x), N\right) \ . \tag{42}$$

Because of the symmetries of the binomial formula, this is equivalent to

$$F_{(m)}(x \mid N) = F_{\mathrm{BIN}}\left((N-m) \middle| (1-G(x)), N\right) \ , \tag{43}$$

which is more computationally friendly than the other form when N is large and $N–m$ is small. Excel has a convenient worksheet function, BINOMDIST. This expression is equivalent to '=BINOMDIST(N-m,N,1-G,TRUE)' where N, m, and G refer to the appropriate cells. However, large N or large N-m can produce #NUM! errors. For reference, the PDF for the mth order statistic is the derivative of the CDF (the sum telescopes; i.e., successive terms cancel, leaving only the first) and is given by

$$f_{(m)}(x \mid N) = N \binom{N-1}{m-1} [G(x)]^{m-1} [1-G(x)]^{N-m} g(x) \ . \tag{44}$$

As an example, figure 35 shows the comparison of a numerical simulation with the predictions of order statistics theory. In this example, 1,000 samples of 10 trials were generated for a standard Gaussian random variable. The eighth smallest value out of each sample of 10 was sorted into bins of size 0.10, resulting in empirical density and CDFs for $N = 10$, $m = 8$. The results are compared to the $f_{(m)}$ and $F_{(m)}$ using $G(x) = \Phi(x)$ (the standard Gaussian CDF), plotting versus the variable x.

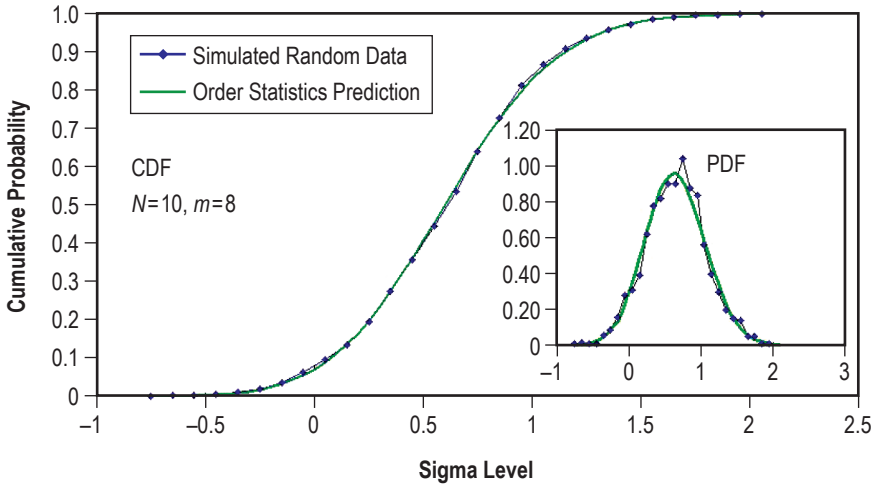

Figure 35. Example of use of order statistics for $N = 10$, $m = 8$, plotted for a standard Gaussian x.

A digression about the PERCENTILE function in Excel is in order. Generally, a quantile function takes as arguments a sample of N trials, and an input probability value p, and then estimates an x value by interpolating between order statistics using some model for the relation between p and the order statistics. The quantile algorithm can be defined in a number of nonequivalent ways.

For example, in the open-source statistics package *R*, the function 'quantile' in the 'stats' package supports nine different algorithms for calculating the quantile.[27] The Excel PERCENTILE function is equivalent to *R* type 7. Type 7 is based on linear interpolation of order statistics using the definition

$$p(m,N) = \frac{m-1}{N-1} \ . \tag{45}$$

Thus, in a sample {71, 69, 58, 81, 72} where $N = 5$, the order statistics are $x_{(1)} = 58$, $x_{(2)} = 69$, etc. The zeroth percentile of the sample is 58, and the 100th percentile is 81. The 60th percentile is determined by linear interpolation between $x_{(3)} = 71$ and $x_{(4)} = 72$, which are respectively the 50th and 75th percentiles of the sample. Excel thus returns a value of 71.4 for the 60th percentile.

It is easy to confuse the PERCENTILE function with an inverse CDF, but it is important to understand the distinction. Just as the sample CDF is only an estimator for the real underlying population CDF, quantile functions are only estimators for the population inverse CDF. Depending on the algorithm, the quantile estimator can be biased in one direction or the other. As discussed in detail below, it turns out that for high quantiles (p near 1), PERCENTILE (i.e., the *R* type 7 quantile) is biased low.

Other algorithms for defining the quantile function are less biased, but without manipulation they also are not usable for establishing design parameters that satisfy a benchmark success rate. The reason is that a nearly unbiased estimator will correspond to a consumer risk of ≈50%, whereas typically specifications are for 10% consumer risk.

The approach discussed here uses order statistics to quantify the consumer risk. The PERCENTILE function is relegated to being a convenient lookup function for interpolating between order statistics. PERCENTILE should not be used directly for estimating population quantiles because the consumer risk is thereby uncontrolled.

Thus, keep in mind the distinctions among several different things being discussed:

(1) The 'population' random variable, which has some (ostensibly unknown) probability distribution characterized by a PDF and CDF.

(2) The sample $x_{(1)}, x_{(2)}, \ldots, x_{(N)}$ of N trials drawn from the population.

(3) The order statistics $x_{(1)}, x_{(2)}, \ldots, x_{(N)}$ that result from sorting the x_i.

(4) The PERCENTILE function used to interpolate between order statistics.

Note that for a given N, particular arguments to the PERCENTILE function will return particular order statistics. For example, when $N = 2{,}000$, the call =PERCENTILE($A$1:$A$2000,99.849925%) will return the 1,997th smallest element in the array (located in the first 2,000 cells of column A); i.e., it returns $x_{(1,997)}$. This works because $(1{,}997-1)/(2{,}000-1) \doteq 0.99849925$. Note this is within 1 part in 10^6 of $1{,}997/2{,}000 = 99.85\%$, so from here on, only the first four significant figures will be quoted.

For a given sample of $N = 2{,}000$, $x_{(1,997)}$ is a biased estimator for the 99.85% quantile of the population. For example, suppose again that the underlying population has a standardized normal probability density. The x value giving $\Phi(x) = 0.9985$ is 2.9676; i.e., just under 3σ. How does the probability distribution for $x_{(1,997)}$ compare to this value? Figure 36 shows that the mean, median, and mode of the $x_{(1,997)}$ order statistic all underpredict the 99.85% population quantile. The mean, median, and mode of $x_{(1,997)}$ in this case are 2.9152, 2.9051, and 2.8855, respectively. All these are less than the population 99.85% quantile 2.9676. Consumer risk, defined here as the probability that the order statistic will underestimate the corresponding population quantile, is ≈64.7%.

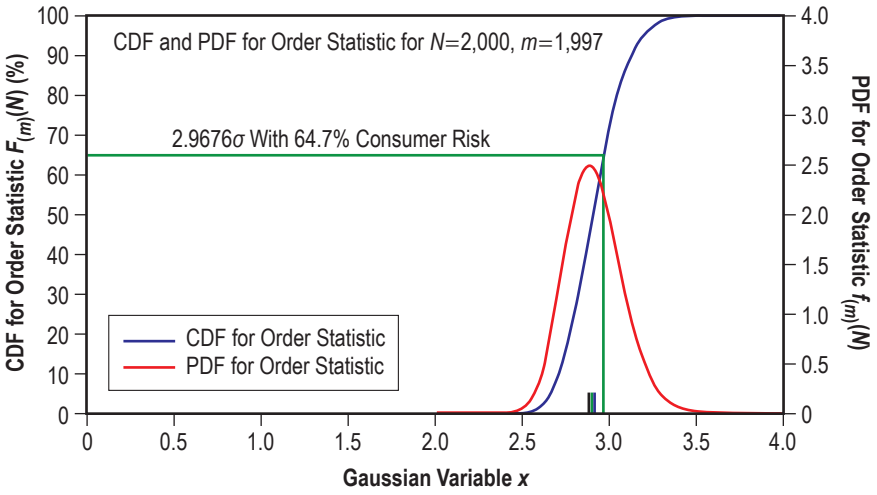

Figure 36. The 99.85 percentile (1,997 out of 2,000) estimator usually underpredicts 3σ.

Note this observation of bias is a statement about order statistics. PERCENTILE only plays a role as a convenient lookup function. Other algorithms for computing quantiles of a sample may in fact give less biased results. However, given the need to account for consumer risk, it is easier to work directly with the order statistics and relegate PERCENTILE to a subsidiary role.

Estimators can be constructed for any desired level of consumer risk and success rate. The CDF for the mth order statistic can be inverted to establish confidence bounds. But first, a discussion of the nonparametric nature of order statistics is appropriate.

So far, the examples shown here have been tied to a standard Gaussian random variable x. However, it should be noted that the CDF $F_{(m)}$ for the mth order statistic is a function of the CDF $G(x)$ of the underlying distribution, and is not tied to x directly. This means that order statistics are 'nonparametric'; i.e., they do not depend on the parameters of type of the underlying distribution. To illustrate this point, figure 37 replots the CDF from figure 36 with $G(x)$ replacing x as the horizontal axis. The resulting curve is valid for any underlying probability density.

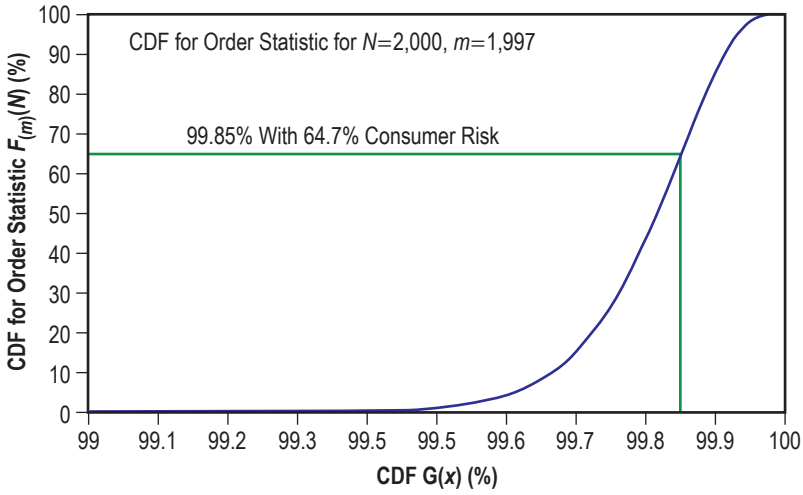

Figure 37. The CDF for an order statistic does not depend directly on the underlying distribution, only on the cumulative $G(x)$.

From this point, order statistics will be discussed in terms of the CDF $G(x)$, which makes the discussion independent of the type of the underlying probability density. It is desired to choose order statistics that satisfy a given benchmark success rate, such as $G(x) = 99.865\%$, while maintaining a 10% upper bound on consumer risk; i.e., it is necessary to choose N and m that satisfy the inequality

$$F_{(m)} = F_{\mathrm{BIN}}\left((N-m)\middle|(1-99.865\%), N\right) < 10\% \ . \tag{46}$$

These combinations are already provided in the binomial sampling plan tables in appendix C, since those tables were computed from inequalities; e.g.

$$F_{\mathrm{BIN}}\left(k\middle|0.135\%, N\right) < 10\% \ . \tag{47}$$

The equivalent order statistic is thus $m = N - k$. For example, the entries in table 10 for $p_A = 0.135\%$ show that the sampling plans $(k, N) = (0, 1{,}705)$, $(1, 2{,}880)$, $(2, 3{,}941)$, and so on, all satisfy the CR = 10% constraint. Figure 38 shows the CDFs for the first two of these cases, illustrating how they provide 90% confidence that the 99.865% level of performance will be estimated.

It is possible to define producer risk for order statistics in several ways. The approach that most closely resembles the approach of sampling theory for quality control is to specify a producer-acceptable success rate (greater than the consumer-acceptable rate of 99.865%) and to compute the probability that the order statistic will overpredict that level.

A similar approach is to compute the success rate for which there is a 10% chance that the order statistic will overpredict; i.e., the value of $G(x)$ at which the CDF $F_{(m)}$ is equal to 90%. In figure 38, these levels are illustrated along with the 10% consumer risk levels. These levels are 99.994% and 99.981% for the $N = 1{,}705$ and $N = 2{,}880$ sampling plans, respectively. Another way of saying this is that the PR for the (1,705, 1,705) plan is 10% if the acceptable success rate is 99.994% ($\approx 3.85\sigma$ equivalent).

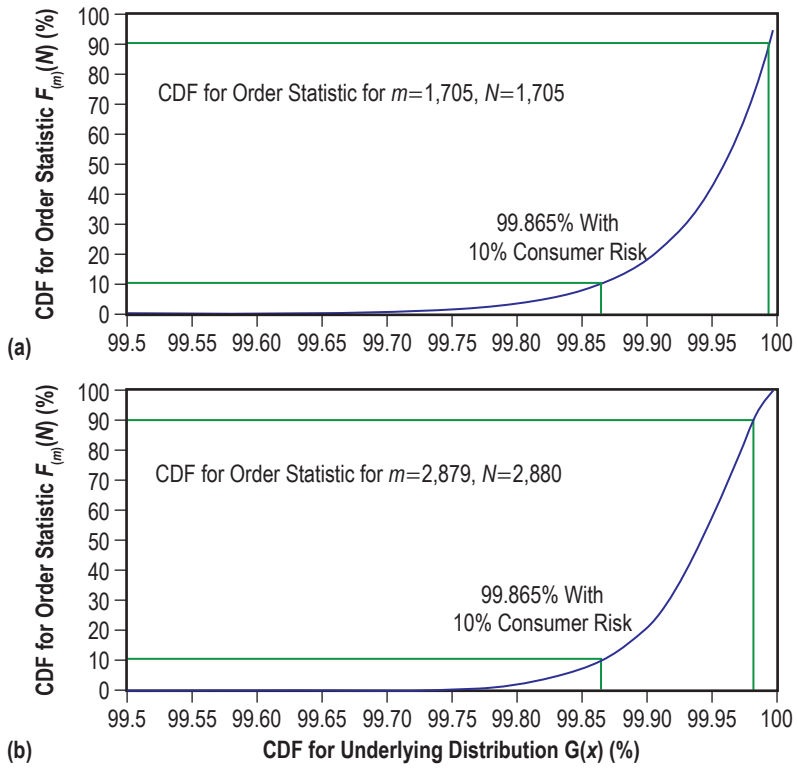

Figure 38. The CDFs for (a) $(m,N) = (1{,}705,\ 1{,}705)$ and (b) $(m,N) = (2{,}879,\ 2{,}880)$ order statistics, showing that they both provide CR = 10% for 99.865% success.

Finally, the sampling plans in appendix B are spaced closely enough that interpolation between entries is feasible. A Monte Carlo simulation might produce a sample of $N = 2{,}000$ Monte Carlo trials, and a 10% consumer risk estimator is needed for some required success probability. Figure 39 illustrates the philosophy behind the interpolation.

Figure 39 shows 10% CR contours versus N for constant $k = N - m$ values. For every N and k, the 10% consumer risk point is computed, using the relations developed above. Thus, the vertical axis is the 10% CR value from the CDF $G(x)$ for the underlying distribution (the quantity on the horizontal axis of figs. 37 and 38). The horizontal lines are the benchmark levels 99.73% and 99.865% success, corresponding to two-tailed and one-tailed 3σ levels for a standard Gaussian. Values from the binomial failure tables in appendix C are marked on the plot; e.g., $(k,N) = (1,\ 1{,}440)$.

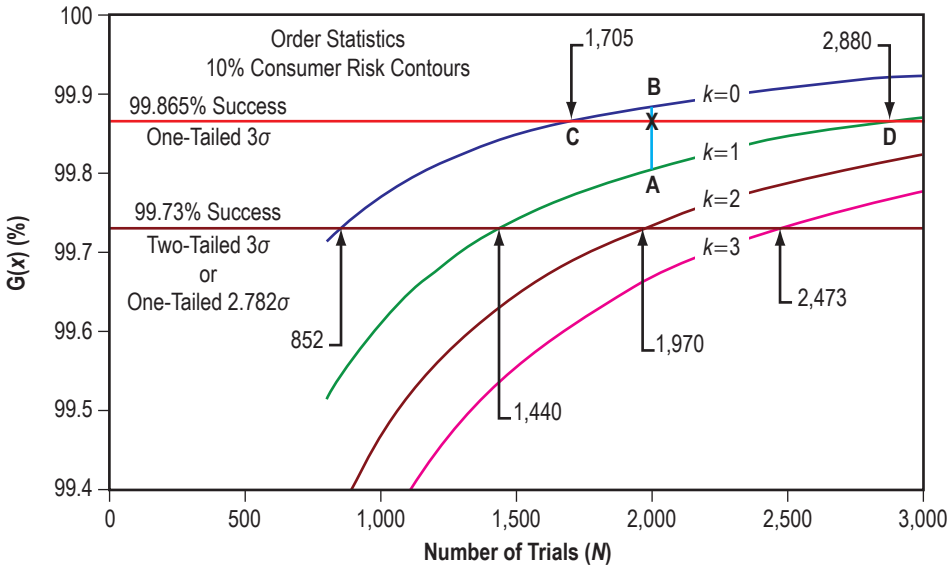

Figure 39. Consumer risk contours for various sampling plans.

The interpolation problem can be posed as follows: Suppose a sample of $N = 2{,}000$ trials is generated. It is desired to use high-end order statistics to estimate the 99.865% success value of a parameter, with no more than 10% consumer risk. This is ostensibly the point 'X' in figure 39, which lies on the line 'AB' at $N = 2{,}000$ connecting the $k = 0$ and $k = 1$ contours. How should the $m = 1{,}999$ and $m = 2{,}000$ data points (corresponding to $N = 2{,}000$ and $k = 1, 0$), denoted $x_{1,999}$ and $x_{2,000}$, be used to formulate this estimate? The answer is that BXC and AXD are almost triangles, and are nearly similar, which means that

$$\frac{\mathrm{AX}}{\mathrm{AB}} \cong \frac{\mathrm{DX}}{\mathrm{DC}} = \frac{2{,}880 - 2{,}000}{2{,}880 - 1{,}705} = 0.749 \ . \tag{48}$$

Thus, one can use

$$\hat{x}_{99.865\%} \cong x_{1,999} + 0.749\left(x_{2,000} - x_{1,999}\right) \tag{49}$$

as an estimator for the 99.865% success value of the parameter x. In this case, explicit computation of the ordinates at points A and B shows that there is <0.05% error in the coefficient 0.749 associated with the similar triangle approximation. One can use the PERCENTILE function to look up this interpolated value by using an argument $(1{,}999.749 - 1)/(2{,}000 - 1) = 99.9874\%$.

Similar interpolations can be done for other success values; e.g., one can interpolate between $(k, N) = (2, 1{,}970)$ and $(3, 2{,}473)$; i.e., between $x_{1,997}$ and $x_{1,998}$, for the 99.73% success value for $N = 2{,}000$. The values 1,970, 2,473, etc. are found in tables 6–10 in appendix C.

APPENDIX C—BINOMIAL FAILURE SAMPLING PLANS

This appendix presents tables of sampling plans for binomial failures. They can be applied to Monte Carlo results where the outcome is pass/fail.

C.1 Overview

These tables enable the design of Monte Carlo binomial-failure studies. See appendix B for a discussion of their use. Tables 8–10 are for 10% consumer risk. Table 11 is for 50% consumer risk. Sections 1 and 4 describe consumer risk and producer risk. As an example, in table 8, if one generates 265 Monte Carlo samples and sees two failed cases (e.g., value of some success parameter above a specified limit), that is 2/265 or 0.75% failed samples, which corresponds to 98% success with 10% consumer risk (10% chance that the real success rate is <98% in reality) and to 99.583% success with 10% producer risk (chance that the success rate is higher than 99.483% in reality).

Table 8. Study structure table, $P_{\text{fail}} = 2\%$ ($P_{\text{success}} = 98\%$), 10% consumer risk.

Minimum Number of Trials (*N*)	Allowable Number of Failures (*k*)	*p* at 10% Consumer Risk (%)	*p* at 10% Producer Risk (%)
114	0	2.0	0.092
194	1	2.0	0.274
265	2	2.0	0.417
333	3	2.0	0.525
398	4	2.0	0.612
462	5	2.0	0.684
525	6	2.0	0.743
587	7	2.0	0.795
648	8	2.0	0.840
708	9	2.0	0.880
768	10	2.0	0.916
828	11	2.0	0.947
887	12	2.0	0.977
1,062	15	2.0	1.050
1,349	20	2.0	1.142
1,632	25	2.0	1.210
1,912	30	2.0	1.263
2,190	35	2.0	1.306
2,465	40	2.0	1.342
3,012	50	2.0	1.399
3,555	60	2.0	1.443
4,094	70	2.0	1.478
4,630	80	2.0	1.507
5,697	100	2.0	1.552

Table 9. Study structure table, $P_{fail} = 0.27\%$ ($P_{success} = 99.73\%$), 10% consumer risk.

Minimum Number of Trials (*N*)	Allowable Number of Failures (*k*)	*p* at 10% Consumer Risk (%)	*p* at 10% Producer Risk (%)
852	0	0.270	0.012
1,440	1	0.270	0.037
1,970	2	0.270	0.056
2,473	3	0.270	0.071
2,959	4	0.270	0.082
3,433	5	0.270	0.092
3,899	6	0.270	0.100
4,358	7	0.270	0.107
4,811	8	0.270	0.113
5,259	9	0.270	0.118
5,704	10	0.270	0.123
6,145	11	0.270	0.127
6,583	12	0.270	0.131
7,883	15	0.270	0.141
10,014	20	0.270	0.154
12,112	25	0.270	0.163
14,187	30	0.270	0.170
16,245	35	0.270	0.176
18,288	40	0.270	0.181
22,343	50	0.270	0.188
26,365	60	0.270	0.194
30,361	70	0.270	0.199
34,337	80	0.270	0.203
42,243	100	0.270	0.209

Table 10. Study structure table, $P_{fail} = 0.135\%$ ($P_{success} = 99.865\%$), 10% consumer risk.

Minimum Number of Trials (*N*)	Allowable Number of Failures (*k*)	*p* at 10% Consumer Risk (%)	*p* at 10% Producer Risk (%)
1,705	0	0.135	0.006
2,880	1	0.135	0.018
3,941	2	0.135	0.028
4,947	3	0.135	0.035
5,920	4	0.135	0.041
6,868	5	0.135	0.046
7,800	6	0.135	0.050
8,717	7	0.135	0.053
9,624	8	0.135	0.056
10,521	9	0.135	0.059
11,410	10	0.135	0.062
12,293	11	0.135	0.064
13,169	12	0.135	0.066
15,769	15	0.135	0.071
20,030	20	0.135	0.077
24,227	25	0.135	0.081
28,378	30	0.135	0.085
32,493	35	0.135	0.088
36,581	40	0.135	0.090
44,691	50	0.135	0.094
52,735	60	0.135	0.097
60,728	70	0.135	0.100
68,681	80	0.135	0.101

Table 11. Study structure table, $P_{fail} = 0.135\%$ ($P_{success} = 99.865\%$), 50% consumer risk.

Minimum Number of Trials (*N*)	Allowable Number of Failures (*k*)	*p* at 50% Consumer Risk (%)	*p* at 10% Producer Risk (%)
514	0	0.135	0.020
1,243	1	0.135	0.043
1,981	2	0.135	0.056
2,720	3	0.135	0.064
3,460	4	0.135	0.070
4,200	5	0.135	0.075
4,941	6	0.135	0.079
5,681	7	0.135	0.082
6,422	8	0.135	0.085
7,162	9	0.135	0.087
7,903	10	0.135	0.089
8,643	11	0.135	0.091
9,384	12	0.135	0.092
11,606	15	0.135	0.096
15,310	20	0.135	0.100
19,013	25	0.135	0.104
22,717	30	0.135	0.106
26,420	35	0.135	0.108
30,124	40	0.135	0.110
37,531	50	0.135	0.112
44,939	60	0.135	0.114
52,346	70	0.135	0.115
59,753	80	0.135	0.117

Note: This table is 50% consumer risk.

Table 12. Study structure table, $P_{fail} = 0.02\%$ ($P_{success} = 99.98\%$), 50% consumer risk.

Minimum Number of Trials (*N*)	Allowable Number of Failures (*k*)	*p* at 50% Consumer Risk (%)	*p* at 10% Producer Risk (%)
11,512	0	0.020	0.0009
19,448	1	0.020	0.0028
26,610	2	0.020	0.0042
33,403	3	0.020	0.0052
39,966	4	0.020	0.0061
46,471	5	0.020	0.0068
52,659	6	0.020	0.0074
58,853	7	0.020	0.0079
64,972	8	0.020	0.0084
71,028	9	0.020	0.0088
77,031	10	0.020	0.0091
82,989	11	0.020	0.0094
88,906	12	0.020	0.0097
106,459	15	0.020	0.0105
135,223	20	0.020	0.0114
163,553	25	0.020	0.0121
191,572	30	0.020	0.0126
219,354	35	0.020	0.0130
246,947	40	0.020	0.0134
301,693	50	0.020	0.0139

Note: This table is 50% consumer risk.

APPENDIX D—MAXIMUM LIKELIHOOD EXTREME VEHICLES

D.1 Introduction

The standard practice for trajectory dispersions is to select particular combinations of input parameter values that are used to generate specific paradigmatic vehicles representing 'light, slow,' 'heavy, fast,' etc., classes. The purpose of this process is to capture the possible variations in vehicle 'sportiness' or 'sluggishness' (see the beginning of sec. 2) among parameters that will be better known prior to launch. Then the Monte Carlo runs capture those variations that are still unknown on flight day. A study was conducted to find a consistent way to choose the input parameter values (for these vehicle models) from assumed probability distributions for those parameters.

Notation and major results are provided here. Detailed calculations can be found in the appendices of reference 5.

Since there are many ways to combine postulated vehicle variations to achieve a desired result, the idea here is to look for the most likely combination. Two methods of modeling an extreme vehicle design were analyzed, including an 'equal input likelihood' method and the recommended 'maximum joint likelihood' method. It is recommended that the latter method be adopted as the standard practice, since it models a more likely combination of input parameters (such as burn rate and I_{sp}) for a fixed value of the targeted output parameter (such as payload).

The principal results are as follows—assume z is a desired output variable (payload, maximum acceleration, etc.):

(1) The standard deviation of a linear function of N uncorrelated input variables $z = f(x_i)$ is given by the familiar RSS formula,

$$\sigma_z = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial z}{\partial x_i} \sigma_i \right)^2} \quad . \tag{50}$$

(2) The most likely point on the $z = m\sigma_z$ level contour (e.g., $m = 3$) is given by

$$x_i^* = m \frac{\partial z}{\partial x_i} \frac{\sigma_i^2}{\sigma_z} \quad , \tag{51}$$

so that the inputs with larger impact on z have a larger variation from nominal.

The effect of correlation of the input variables was also analyzed. This effect can be significant.

For Ares, the input parameters for this trajectory model are mostly Gaussian random variables, which facilitates this analysis. However, some parameters are assumed to have a uniform distribution, which presents some minor complications. The essential result for uniformly-distributed input variables is that the maximum likelihood point is at an endpoint of the distribution interval; however, it is important to compensate for the special treatment of uniform inputs by adjusting the sums used in this algorithm for the remaining variables. An exception is if the endpoint of the uniform distribution by itself gives a result that exceeds the target z, or if the endpoint is not a feasible point. If the case at hand has an intermediate value of the uniformly-distributed variable for which the other sigmas go to zero (it covers all the variation), then that is the most likely point. In some cases where there are multiple target values; e.g., slowest liftoff and payload, the endpoint may not be a feasible point.

D.1.1 Problem Framework

To treat the problem in its simplest form, it is assumed that some vehicle parameter z, such as payload to orbit, is a function of some number of input parameters that are modeled as independent Gaussian random variables, such as engine I_{sp} and solid-fuel propellant burn rate. The goal is to define an appropriate method for picking an 'extreme' vehicle configuration, given a criterion for 'extreme' input parameters (such as a 3σ-equivalent confidence region).

Consider a sample calculation with one output as a function of two inputs. Thus,

$$z = f(x, y) \ , \tag{52}$$

where x and y are taken as independent Gaussian variables with known variances σ_x^2 and σ_y^2. To simplify the notation, it is assumed that x and y are 'offsets' from mean values of input parameters such as engine I_{sp} and propellant burn rate, and that z is also the offset from an output mean value of a parameter such as payload. Then, the joint distribution of the two random variable input parameters is

$$P(x, y)\,dx\,dy = \frac{\exp\left(-\frac{1}{2}\left(x^2/\sigma_x^2 + y^2/\sigma_y^2\right)\right)}{2\pi\sigma_x\sigma_y}\,dx\,dy \ . \tag{53}$$

A key assumption is that $f(x, y)$ is a smooth function of the input parameters, and also that its variation is (a) calculable and (b) slow enough that a first-order Taylor expansion is sufficiently accurate for vehicle design purposes. The Taylor expansion is

$$z = \frac{\partial z}{\partial x}x + \frac{\partial z}{\partial y}y + O\left(x^2, y^2\right) \ , \tag{54}$$

since the means for x, y, and z are assumed to be zero. To simplify the algebra, write

$$a \equiv \frac{\partial z}{\partial x} \tag{55}$$

and

$$b \equiv \frac{\partial z}{\partial y} \tag{56}$$

for the sensitivities. Note that a and b can be dimensional parameters (with units such as 'payload per burn rate'), and dimensional analysis may be useful as a check on the calculations. Thus,

$$z = ax + by \ . \tag{57}$$

The first question is, "What is the probability distribution for z?" It should not be surprising that if z is a linear function of Gaussian variables, it is also Gaussian. (But if higher order terms are added to the Taylor expansion, z is no longer Gaussian, and it becomes considerably more involved to carry out the following calculation.) The important result is the familiar RSS rule:

$$\sigma_z = \sqrt{a^2\sigma_x^2 + b^2\sigma_y^2} \ . \tag{58}$$

Therefore, the probability distribution for z is

$$P\left(z\right)dz = \frac{\exp\left(-\frac{1}{2}\left(z^2/\sigma_z^2\right)\right)}{\sqrt{2\pi}\sigma_z}dz \ . \tag{59}$$

Of central importance are the confidence regions on the output parameter, analogous to 3σ (99.73%) two-tailed confidence region. It is helpful to picture $P(z)$ contours superimposed on the joint probability distribution. Figure 40 shows an example constructed to illustrate the calculation below. The figure is based on the arbitrary choice of parameters (in no particular set of units):

σ_x=1.5
σ_y=0.4
a=0.1

and

b=1 .

Figure 40 shows a 1,000-point sample from the probability distribution for x and y (black +'s), with the $\chi^2 = 9$ (3σ) ellipse in green, and the 3σ rectangle in blue. Note that the point set, the ellipse, and the rectangle are determined by σ_x and σ_y and are independent of the z-sensitivities a and b.

χ^2 is calculated from

$$\chi^2 = \frac{a^2\sigma_x^2 + b^2\sigma_y^2}{\sigma_z^2} \ . \tag{60}$$

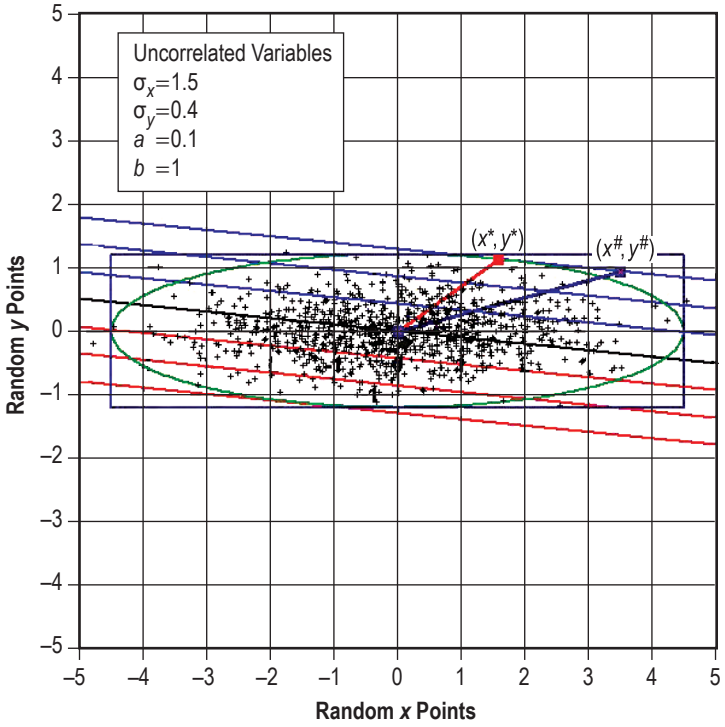

Figure 40. Example of two ways to pick a representative 3σ vehicle.

Superimposed on this diagram are the lines of constant z:

$$ax + by = m\sigma_z \ , \tag{61}$$

with $m = -3, -2, \ldots, 3$, and σ_z given by the RSS formula for the example values of a and b, which incidentally were picked to exaggerate the slope of the level contours.

Note that any point on the highest shown z-contour is a 3σ z value; i.e., there is an infinite number of (x,y) choices that could yield that payload.

One simple approach picks the point $(x^{\#}, y^{\#})$ at the intersection of the diagonal of the 3σ rectangle and the $ax + by = 3\sigma_z$ contour; i.e., it uses the solution to the equation

$$a\left(k\sigma_x\right) + b\left(k\sigma_y\right) = 3\sigma_z \Rightarrow k = 3\frac{\sqrt{a^2\sigma_x^2 + b^2\sigma_y^2}}{a\sigma_x + b\sigma_y} \tag{62}$$

to compute the payload based on the inputs $(x^{\#}, y^{\#}) = (\mathrm{k}\sigma_x, k\sigma_y)$. This point is marked by a blue square with a connected blue line to the origin. Note that the ratio $\sqrt{a^2\sigma_x^2 + b^2\sigma_y^2} \Big/ \left(a\sigma_x + b\sigma_y\right)$ reaches a global minimum of $\sqrt{2}/2$ when $a\sigma_x = b\sigma_y$. As more variables are added, the minimum decreases as $1/\sqrt{N}$.

This choice corresponds to an a priori assumption that the input parameters should have equal likelihood; i.e., should have the same σ values.

D.2 A Better Criterion

However, another criterion suggests itself. The fundamental observation is that the equal a priori probability point is not the most likely point on the $z = 3\sigma_z$ contour. The most likely point is the one with the minimum value of χ^2; i.e., it is the point on the $z = 3\sigma_z$ contour which is 'tangent' to the $\chi^2 = 9$ ellipse. This point has coordinates

$$(x^*, y^*) = \left(3\frac{a\sigma_x^2}{\sigma_z}, 3\frac{b\sigma_y^2}{\sigma_z} \right) . \tag{63}$$

The derivation is given in reference 5.

The point (x^*, y^*) is shown in figure 40 as the solid red square, with a red line connecting to the origin. For the example shown, the computed points $(x^\#, y^\#)$ and (x^*, y^*) are given in table 13, together with χ^2 values and the corresponding p values for χ^2. (To reiterate a fundamental point, note that the p value for $\chi^2 = 9$ is >1%, and not, as one might naively expect, $p = 0.27\%$ for 3σ. This is another example of the well-known result that a fixed value of χ^2 encloses a smaller and smaller fraction of the joint probability distribution as the number of dimensions is increased. The reconciliation with the one-dimensional Gaussian is the observation that 99.73% of the joint probability lies in the strip between the $z = \pm 3\sigma_z$ contours, while only 98.9% lies within the $\chi^2 = 9$ ellipse. See appendix A for more on this point.)

Table 13. Comparison of equal- and maximum-likelihood methods for example case.

	(x,y)	χ^2	*p* Value (%)
Equal component likelihood	$(x^\#, y^\#) = (3.495, 0.932)$	10.86	0.44
Maximum joint likelihood	$(x^*, y^*) = (1{,}580, 1.124)$	9	1.11

In this particular example, the conventional method of picking the point $(x^\#, y^\#)$ gives a point that is ≈2.5 times less likely than the point (x^*, y^*). That is, the tail of the χ^2 distribution with 2 DOF contains 1.11% of the total weight when χ^2 is 9, and only 0.44% of the weight when χ^2 is 10.86, making the latter point much less likely (see table 13). The amount by which the maximum likelihood method surpasses the equal a priori probability method will depend on the values of the uncertainties and sensitivities input to the respective algorithms.

D.3 Generalization to More Variables

The generalization to N variables is straightforward. It is still convenient to assume mean values have been subtracted out. The input variables are relabeled $x_1, x_2, \ldots, x_N$, with variances σ_i^2 and have $z = f(x_i)$. The Taylor expansion is

$$z = \sum_{i=1}^{N} \frac{\partial z}{\partial x_i} x_i + O\left(x_i^2\right) \ . \tag{64}$$

The standard deviation of z is

$$\sigma_z = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial z}{\partial x_i} \sigma_i \right)^2} \ . \tag{65}$$

The most likely point on the $z = m\sigma_z$ contour (e.g., $m = 3$) is then given by

$$x_i^* = m \frac{\partial z}{\partial x_i} \frac{\sigma_i^2}{\sigma_z} \ . \tag{66}$$

This is the recommended approach for modeling an extreme vehicle with uncorrelated inputs, where the inputs follow Gaussian distributions.

D.4 Generalization to Two Correlated Variables

The next question is, "Does it make a difference if the input variables are correlated?" Correlation is quantified by the covariance

$$\mathrm{cov}(x, y) \equiv \left\langle \left(x - \langle x \rangle \right) \left(y - \langle y \rangle \right) \right\rangle \ . \tag{67}$$

where $\langle\ \rangle$ means expected value or probability-weighted average.

It is convenient to define the correlation coefficient

$$\rho \equiv \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} \ . \tag{68}$$

Note that cov(x,y) is a quantity with dimensions of xy, but the correlation coefficient is nondimensional with $-1 \le \rho \le +1$.

The Taylor expansion of z is still

$$z = ax + by. \tag{69}$$

The variance of z is given by

$$\sigma_z^2 = a^2 \sigma_x^2 + 2ab\rho\sigma_x\sigma_y + b^2\sigma_y^2 \ . \tag{70}$$

Note that the variance of z can be increased or decreased by correlation, depending on the sign of the product $ab\rho$.

The most probable point on the $z = m\sigma_z$ contour is

$$\left(x^*, y^*\right) = \left(m\frac{a\sigma_x^2 + \rho b\sigma_x\sigma_y}{\sigma_z},\ m\frac{b\sigma_y^2 + \rho a\sigma_x\sigma_y}{\sigma_z} \right) , \tag{71}$$

which obviously reduces to the previous result when $\rho = 0$. The derivation is in reference 5.

For illustrative purposes, the locus of these tangent points is shown in figure 41, as ρ is varied from –0.9 to 0.9 for the preceding example. In figure 41, the standard deviation of z changes as the correlation changes, from a minimum of $|a\sigma_x - b\sigma_y|$ to a maximum of $|a\sigma_x + b\sigma_y|$. It can be seen that correlation of the input variables can significantly affect the estimate of the dispersion of z. Note that χ^2 is held constant at $3^2 = 9$, so all the points in this locus have equal p values.

D.5 Generalized Correlated Inputs

The generalization to N correlated input variables is a bit messy to write down explicitly, although the linear algebra is straightforward. With N inputs, χ^2 is given by a quadratic form involving the inverse of the error matrix (repeated indices are summed):

$$\chi^2 = x_i \left(M^{-1}\right)_{ij} x_j \ , \tag{72}$$

where the error matrix is defined by

$$M_{ij} = \left\langle \left(x_i - \left\langle x_i \right\rangle\right)\left(x_j - \left\langle x_j \right\rangle\right) \right\rangle \ ; \tag{73}$$

i.e., M is a symmetric matrix with diagonal elements equal to the variances and off-diagonal elements equal to the covariances of the input variables.

Manipulation of the error matrix and tangent equations yield the maximum likelihood point x_i^*. Details are provided in reference 5.

D.6 Uniform Distribution

Accounting for an input variable with a uniform probability distribution presents some challenges. As before, detailed calculations are available in reference 5.

Start with a two-input case, $z = z(x,y)$, assuming x is uniformly distributed over the interval $(-c/2, c/2)$ and y is Gaussian with mean zero and known variance σ_y^2. Note that the probability distribution for x is

$$P(x)dx = \frac{1}{c}\,\mathrm{rect}\left(\frac{x}{c}\right)dx \tag{74}$$

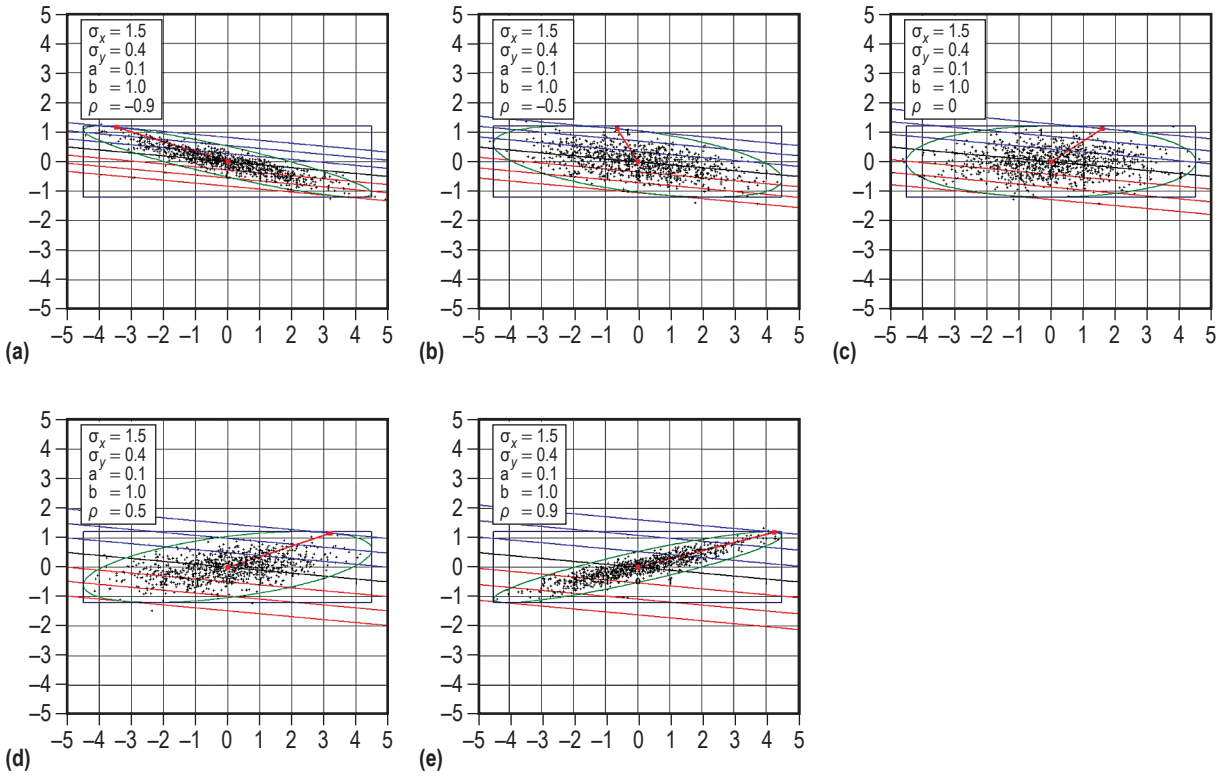

Figure 41. Effect of correlation on optimum: (a) ρ=–0.9, (b) ρ=–0.5, (c) ρ=–0, (d) ρ=0.5, and (e) ρ=0.9.

(see ref. 28 for definition of rect(x)) and the variance of x is

$$\sigma_x^2 = \frac{1}{c}\int_{-c/2}^{c/2} x^2 dx = \frac{c^2}{12} \quad . \tag{75}$$

The joint probability distribution for x and y is

$$P(x,y)\,dx\,dy = \frac{1}{c}\,\mathrm{rect}\left(\frac{x}{c}\right)\frac{\exp\left(-y^2/2\sigma_y^2\right)}{\sqrt{2\pi}\sigma_y}\,dx\,dy \quad . \tag{76}$$

Note that, assuming a first-order Taylor expansion, the usual RSS formula still applies:

$$\sigma_z = \sqrt{a^2\sigma_x^2 + b^2\sigma_y^2} \quad . \tag{77}$$

The first wrinkle is that z is not Gaussian, which means that there is no longer the familiar relation between confidence intervals and the variance. Reference 5 gives the derivation for the probability distribution for z. The result for $P(z)$ is

$$P(z)dz = \frac{1}{ac}\left(\Phi\left(\frac{z + ac/2}{b\sigma_y} \right) - \Phi\left(\frac{z - ac/2}{b\sigma_y} \right) \right) dz \ , \tag{78}$$

where

$$\Phi\left(x\right) = \frac{1}{2}\left(1 + \text{erf}\,\frac{x}{\sqrt{2}} \right) \tag{79}$$

is the CDF for the standard normal distribution.

The good news is that this distribution will approach a Gaussian distribution, provided *ac*, or equivalently, $a\sigma_x$, is small enough. The bad news is that for large $a\sigma_x$, the exact expression becomes somewhat unwieldy for analytical solution, and one must resort to numerical evaluation of the CDF to find confidence limits.

In more nondimensional terms, let

$$\zeta \equiv z/\sigma_z \tag{80}$$

and

$$\beta \equiv b\sigma_y / a\sigma_x \ . \tag{81}$$

The parameter β is a measure of the 'Gaussian dominance' of the uni-Gauss distribution. Then,

$$P(\zeta)d\zeta = \frac{\sqrt{1+\beta^2}}{2\sqrt{3}}\left(\Phi\left(\zeta\frac{\sqrt{1+\beta^2}}{\beta} + \frac{\sqrt{3}}{\beta} \right) - \Phi\left(\zeta\frac{\sqrt{1+\beta^2}}{\beta} - \frac{\sqrt{3}}{\beta} \right) \right) d\zeta \ . \tag{82}$$

In these terms, figure 42 illustrates how the distribution of z is similar to, and diverges from, a Gaussian as the ratio $\beta = b\sigma_y / a\sigma_x$ is varied.

The significance of this observation is that the 99.73% confidence interval for z is not simply $\pm 3\sigma$ if the ratio β is too small. In that case, one must evaluate the CDF numerically to find the 99.73% value of z. In the case depicted in figure 42(d), for example, numerical evaluation gives $z = \pm 1.852\ \sigma_z$ as the bounds for 99.73% confidence.

However, if β is large, the familiar $\pm 3\sigma$ confidence bounds can be used. In the more general case of multiple Gaussian variables, the ratio $\sigma_z / a\sigma_x$ can be evaluated to check that it is large.

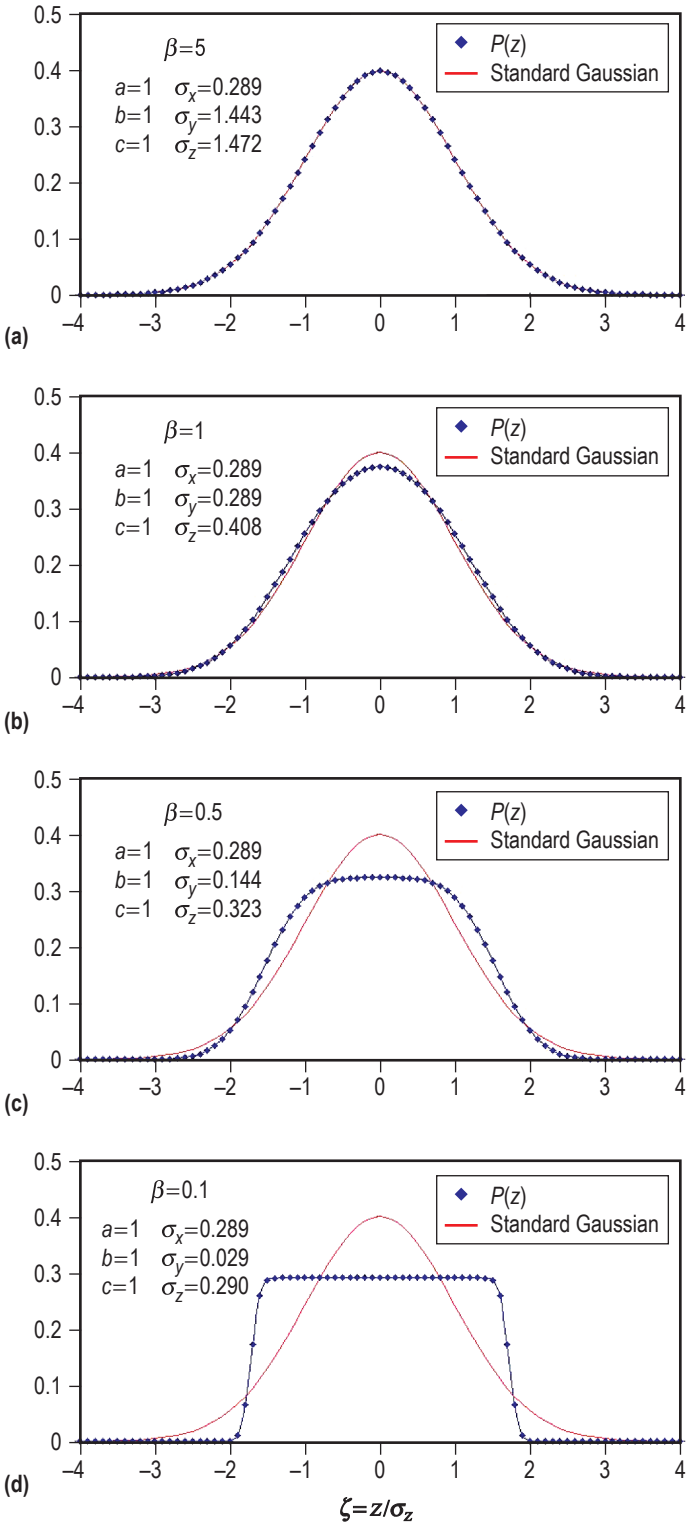

Figure 42. Distribution function for z as parameters are varied: (a) β=5, (b) β=1, (c) β=0.5, and (d) β=0.1.

Once the appropriate selection of confidence bounds is determined, the problem of selecting the maximum likelihood combination of x and y remains. Again, an illustrative figure is useful. Figure 43 is set up like figures 40 and 41, except that in this case, the probability distribution is uniform in x and Gaussian in y. The rectangle shows the boundaries $\pm c/2$ in x and $\pm 3\sigma_y$ in y. The sensitivities have been selected to exaggerate the slope of the constant z contours. The example was constructed such that

$$b\sigma_y / a\sigma_x = 5 \,, \tag{83}$$

which is large enough to use the Gaussian approximation for z and plot lines of $m\sigma_z$ as before. Note $\sigma_x = 1$ in this example and the interval is $\pm c/2 = \pm\sqrt{3}\sigma_x$.

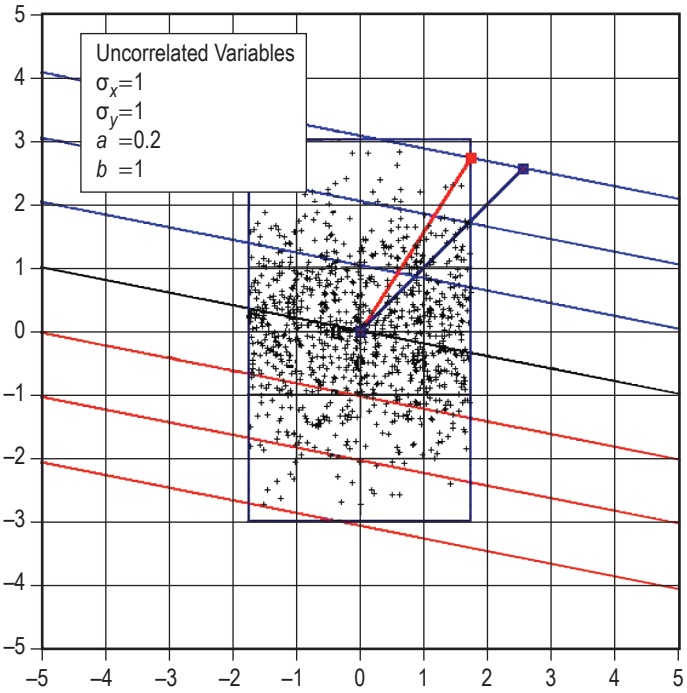

Figure 43. The maximum likelihood point is at one end of the interval when x is uniformly distributed.

The key observation is that lines of constant probability density are not ellipses, but horizontal segments in the interval $(-c/2, c/2)$. This means that the maximum likelihood point will be at the endpoint of the interval.

This statement as written in not precisely true in all cases. There is a tiny corner of parameter space—where the joint probability is essentially uniform—where the 99.73% constraint falls up to 0.27% inside the uniform distribution interval. (In this case, it is still recommended to use the endpoint $x = \pm c/2$.) This calculation, and the proof that the maximum likelihood point is nearly always at endpoint of the uniform distribution interval, are discussed in detail in reference 5. This makes intuitive sense since each of the uniform probability points is equally likely, so that by choosing the

endpoint, the sigma level required of the other (Gaussian) variable is reduced. Using the endpoint would not be the correct choice if that choice drives z beyond its target value. In that case, choosing the uniform variable at the point where the sigma level of the other variable can be zero maximizes the likelihood.

If $a > 0$, the maximum likelihood point will be at the maximum x, and conversely, if $a < 0$, the maximum likelihood point will be at the minimum x (again, unless this choice drives z beyond its target value). Hence, for the uni-Gauss distribution,

$$(x^*, y^*) = \left(\mathrm{sgn}(a) \frac{c}{2}, \frac{m\sigma_z - |\, ac/2 \,|}{b} \right) . \tag{84}$$

Note that 'sgn' is the sign 'signum.'[29] In this case, the 'equal scaling of sigmas' approach does not work at all, since it gives

$$a\left(k\sigma_x\right) + b\left(k\sigma_y\right) = 3\sigma_z \Rightarrow k = 3\frac{\sqrt{a^2\sigma_x^2 + b^2\sigma_y^2}}{a\sigma_x + b\sigma_y} = 2.55 \ , \tag{85}$$

whence

$$k\sigma_x = 2.55 \ , \tag{86}$$

which falls outside the support for x, which is confined to the interval $\pm c/2 = \pm\sqrt{3}\,\sigma_x = \pm 1.73\,\sigma_x$. That is, the 'equal k' approach yields a point with a disallowed value of x, with zero probability.

D.7 A Basket of Inputs

The precise handling of a mix of uniform and Gaussian variables requires careful consideration.

In this section, $\{x_i\}$ is a set of uniformly-distributed variables with intervals $c_i = 2\sqrt{3}\sigma_{x_i}$, and $\{y_i\}$ is a set of Gaussian distributed variables with dispersions σ_{y_i}. First, note that the maximum likelihood set of inputs requires each uniformly-distributed variable to be at an endpoint value (again unless the importance of the uniformly-distributed variables is sufficient to completely satisfy the target z):

$$x_i = \mathrm{sgn}\left(a_i\right)\frac{c_i}{2} = \mathrm{sgn}\left(a_i\right)\sqrt{3}\sigma_{x_i} \ . \tag{87}$$

Reducing the set $\{y_i\}$ of Gaussian variables to the previous all-Gaussian case requires a new 'z' variable:

$$z'\left(y_j\right) \equiv \sum_j \frac{\partial z}{\partial y_j} y_j = z - \sum_i \frac{\partial z}{\partial x_i} x_i + O\left(y_j^2\right) \tag{88}$$

with a smaller standard deviation

$$\sigma_{z'} = \sqrt{\sum_j \left(\frac{\partial z}{\partial y_j} \sigma_{y_j} \right)^2} = \sqrt{\sigma_z^2 - \sum_i \left(\frac{\partial z}{\partial x_i} \sigma_{x_i} \right)^2} \quad . \tag{89}$$

Note that by construction,

$$\frac{\partial z'}{\partial y_j} = \frac{\partial z}{\partial y_j} \quad . \tag{90}$$

Since the goal is to find the most likely set of inputs that satisfies the constraint

$$z^* = m\sigma_z \quad ; \tag{91}$$

e.g., $m = 3$), this means that the multiple contributions $|a_i c_i / 2|$ of the uniform inputs must be subtracted from the original function $z(x_i, y_j)$. This leaves a set of Gaussian variables with the same sensitivities as previously calculated. However, the constraint $z = m\sigma_z$ must be replaced by a constraint:

$$z'^* = m\sigma_z - \sum_i \left| a_i c_i / 2 \right| = m'\sigma_{z'} \tag{92}$$

and

$$\Rightarrow m' = \frac{m\sigma_z - \sum_i \left| a_i c_i / 2 \right|}{\sigma_{z'}} \quad . \tag{93}$$

This result is in the form to which the results of the previous section can be applied. In the notation of this section,

$$y_j^* = m' \frac{\partial z'}{\partial y_j} \frac{\sigma_{y_j}^2}{\sigma_{z'}} \quad . \tag{94}$$

The following summarize the handling of uniformly-distributed random variable inputs:

- The standard deviation of z can be computed with the RSS as before.

- Check whether the component $a_i\sigma_{x_i} \ll \sigma_z$:

  – If it is, assume z is normally distributed.
  – Otherwise, compute the cumulative distribution of z numerically and invert it to get confidence intervals.

- Regardless of the choice of confidence interval on z, the maximum likelihood point will fall at the end of the interval for x_i, because the extreme value for x_i minimizes χ^2 for the remaining variables (unless choosing all the uniform variables at the endpoints causes z to exceed its target value, in

which case, the value of the uniform variables that cause other variables to need no variation are chosen). And, in the case of multiple targets (sec D.9), the endpoint may not even be a feasible point.

- Once the uniform variables are accounted for, the variance and confidence level on z must be adjusted to reflect the reduced set of parameters.

- Then the previous formulas for finding the maximum likelihood point apply.

Reference 5 gives a detailed example of implementing this calculation with a spreadsheet.

D.8 Correlations for Uniform Distributions

Curiously, the question of correlated uniform and uni-Gauss distributions is relatively subtle. The reason is that, unlike correlated Gaussian variables, there is no natural Taylor expansion and χ^2 methodology to guide the selection of the functional form for the joint distribution. Another way to say this is that the covariance alone does not determine the functional form for $P(x, y)$. There are numerous approaches discussed in the literature, but selecting among these is problematic.

D.9 Extension to Multiple Target z Values

It is also possible to extend this methodology to multiple target z parameters. In such cases, it is convenient to fall back on numerical methods. The widespread use of spreadsheet programs like Microsoft Excel, which has a Solver add-on available in the Analysis Toolpak, makes it very easy to set up multiple input variables and multiple target values. The basic requirements are the same as outlined above: the mean and standard deviation of the input variables must be known, and the linear (first-order Taylor expansion) sensitivities of all the output z values to each input must be quantified.

Once these data are acquired, the following setup is necessary:

- Set up a range of cells in the spreadsheet containing guesses for the x_i values.
- Set up parallel ranges for the means, standard deviations, and sensitivities.
- Set up probability density calculations for the x_i, using the assumed PDF for each one.
- Multiply the probability densities to get a joint probability density.
- Compute the output z value as a function of the input x_i.

Instead of multiplying the PDFs, it is more numerically stable to take the natural logs of the PDF values and add the logs. Obviously, maximizing the log of the joint probability should give the same result as maximizing the probability.

Then, the Solver add-on must be invoked. The idea is to ask the Solver to maximize the log of the joint probability density by varying the x_i guess range, subject to the constraint that the target z value is greater than or equal to the $m\sigma_z$. Figure 44 is a depiction of this process for a two target z's, but the extension to multiple target z's is straightforward.

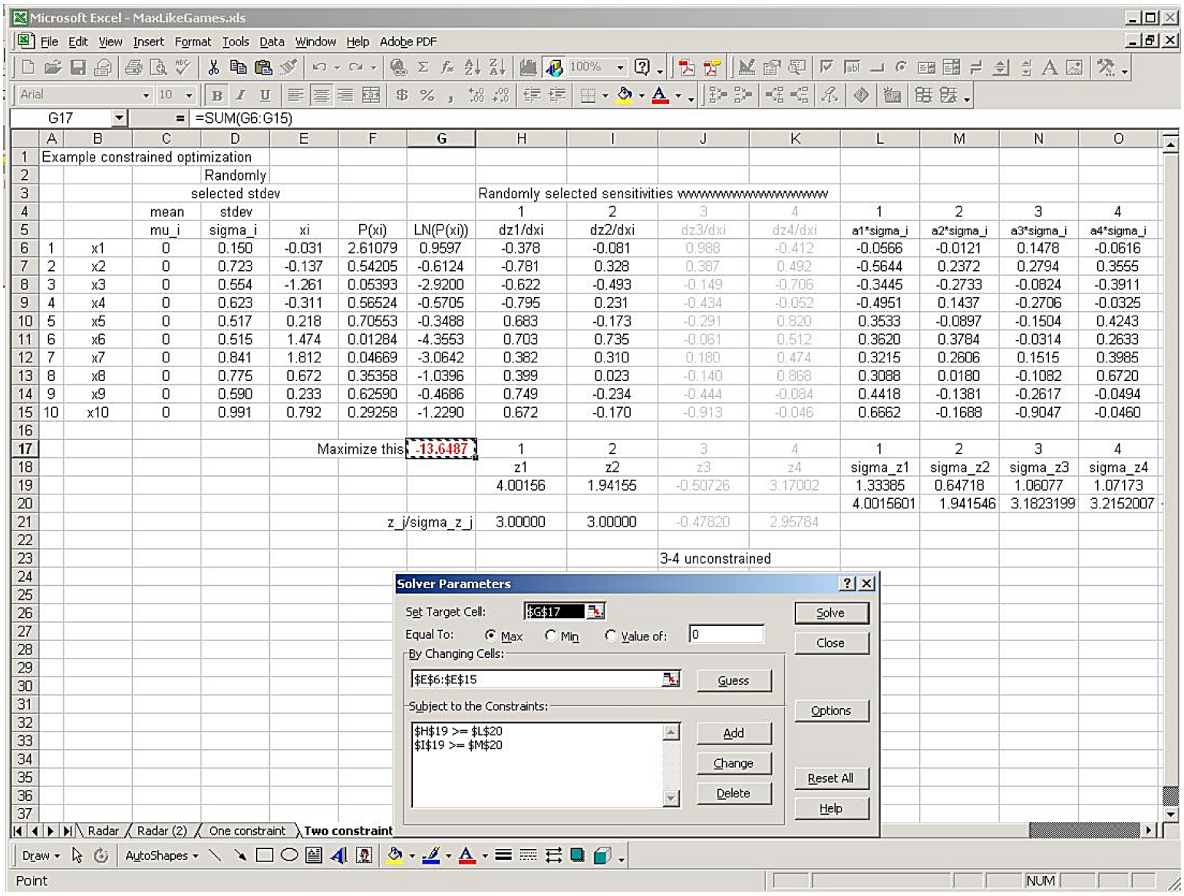

Figure 44. Depiction of Solver add-on process.

When there are multiple target z values, it will often be the case that choosing the endpoint of a uniform distribution does not work. For example, suppose the targets values are high liftoff thrust/weight, maximum dynamic pressure, maximum acceleration, and maximum heat rate, and suppose the aerodynamic axial force coefficient is uniformly distributed. Taking a value of axial force coefficient at its lower limit will result in a high dynamic pressure but will have little effect on the other target parameters. Then other major parameters, such as thrust dispersion, may be able to satisfy the other three targets but not the maximum dynamic pressure target. So an intermediate value of axial force coefficient is needed. This is easy to implement in the numerical approach, where the solver iterates on all the inputs within their allowable ranges and chooses the values that maximize the joint probability while simultaneously satisfying the target constraints. This approach should work as long as the number of input x_i parameters is greater than or equal to the number of target z parameters.

APPENDIX E—ENSURING DUPLICATED MONTE CARLO RANDOM VARIATIONS

E.1 Introduction

Suppose a simulation generates a random number sequence for each Monte Carlo sample, using the sample number (or some similar approach) as the seed. Then each variable's variation is pulled from the random number sequence. When testing a new simulation version, duplicating the old random variations to be able to look for causes for changes between the old and new versions is vital. This can be difficult if there is uncertainty as to whether the change was due to the changing random parameters or whether it was due to a change in modeling or a logic bug.

Here is a way to avoid that problem. Give each variable a unique identifier name, with the run number appended to this name. Hashing the identifier generates a seed that is used to initialize the RNG for that variable and for that Monte Carlo sample. So the RNG will get a different seed for every variable and for every sample. This fixes the previous problem because if a new version of the simulation is run, the variable will have the same seed as before. But instead of generating one sequence of random numbers with a single seed for each sample (a different sequence for each Monte Carlo sample), the seeds now change for every variable and for every run.

E.2 Description of the Method

Here is one approach to implementing this. The random number sequence is reinitialized for each variable with its own customized seed. This seed is automatically generated internal to the simulation and is partly a function of sample number so that a specific sample in a Monte Carlo set can be reproduced by specifying the sample number.

The random number initialization seed for each variable is generated as follows:

First, a string is constructed consisting of three parts: (1) The input name of the variable; i.e., the 'search' string that is specified in the data file just preceding the variable's input value, (2) a discriminator string (which is normally automatically set) that distinguishes variables that have the same input name string (e.g., 'SRB' and 'J2X' to distinguish between engine 'location' for the two different engines), and (3) a sample number string which identifies the sample number.

Following are examples of typical strings that would be generated for each variable:

location(SRB)_sample48
location(J2X)_sample84
gimbal_frame_orientation(pm_eng)_sample4
thrust_radial_offset(SRB)_sample68.

Second, this string is hashed into a unique 32-bit integer using the following hashing algorithm (in C):

```
unsigned long hash ( unsigned char *str )
{
  unsigned long hash=5381;
  int c;
  while(c=*str++) hash=((hash<<5)+hash)+c;
  return hash
}
```

where c = the ASCII code of a character in the string,

The above algorithm produces the following unique integers that are used as the seed to reinitialize the random number sequence from which the random numbers for the variable will be drawn (examples):

location(SRB)_sample48	3404552662
location(SRB)_sample84	3404552790
gimbal_frame_orientation(pm_eng)_sample4	1445673045
thrust_radial_offset(SRB)_sample68	2603420667

If the context of an input does not cause a discriminator string to be automatically assigned by the simulation (which should be rare), then the string must be supplied in the dispersion specification such as:

[N, sd=.2, “J2X”] ,

where N means the variable has a normal distribution, 0.2 is the standard deviation, and J2X is the discriminator string.

Also, a name that is used to hash into a seed may be generated as follows:

[N, sd=.2,seed=location(SRB)] .

The simulation would then append “_sampleXX” (where XX represents the sample number) to the user-specified seed string so that any sample number may be duplicated. A user might want to specify the seed this way if the desire is to force two or more different variables to be dispersed the same way (i.e., using the same random numbers).

The 32-bit integers generated by the hashing algorithm for each variable are used to reinitialize the random number sequence at the time the dispersion is about to be computed for the input variable.

When dispersing a vector where all three components are dispersed, random numbers are drawn from the same sequence which was initialized by the single seed for the vector variable. Quaternions and table columns are also handled in the same way. So each variable, whether scalar, vector, quaternion, or table column, is dispersed with the first few elements of its own unique sequence of random numbers.

APPENDIX F—FLIGHT PERFORMANCE RESERVE/REMAINING USABLE PROPELLANT

F.1 Introduction

During every design analysis cycle (DAC), Ares I has a specified amount of FPR (extra propellant of both propellant types—liquid hydrogen (LH_2) and liquid oxygen (LO_2)) set aside to ensure mission success in the presence of dispersions. When Ares guidance, navigation, and control engineers generate a trajectory dispersion, remaining usable propellant (RUP) is calculated for each trajectory. The development below shows whether or not this excess propellant meets success probability requirements. This development can be applied to other problems where one wishes to minimize the addition of two consumables and the outcome is dependent on both (running out of either one is considered a failure).

If the results using these procedures do not sufficiently cover the necessary poor performing cases, order statistics may be used in order to meet the requirements (as discussed in app. B). This may be true if the tails of the distribution are not Gaussian. Even with this approach, the resulting fuel numbers may not be very accurate since there are few points in the tail. In addition, if a low-fuel engine cutoff sensor sometimes provides the fuel cutoff, the uncertainties in the low fuel measurements might throw the statistics off, since the sensor cutoffs are probably few in number (resulting in poor statistical results for their uncertainty). Addressing this issue is beyond the scope of this TP, but distributions for uncertainties in the low fuel measurements may be used to refine the answer. Analysis of this situation is planned for future publication.

As in the previous section, detailed technical calculations are available in reference 5. The main text here gives the framework, notation, and major results.

F.2 General Framework for Optimizing Propellant Reserves

The purpose of this analysis is to optimize the distribution of RUP in order to show that success requirements are met with allowable consumer risk. RUP is the mass of LO_2 and LH_2 at the end of the second-stage burn. It is assumed herein that the quantity to be optimized is the total RUP mass. Typically, the RUP values are reported as dispersed values for a particular set of trajectory assumptions. In what follows, assume that RUP values are tabulated for N runs in a Monte Carlo sample; typically $N = 2{,}000$. This TP uses values from an example dataset. The overall goal is to minimize the FPR that enables meeting of the mission success requirements. For a fixed amount of LO_2 and LH_2 on board, finding the maximum RUP mass that meets the requirement maximizes excess propellant that is not needed. Then this excess propellant can be subtracted (or added, if any of the values are negative) from the FPR; i.e., any excess LO_2 or LH_2 can be removed from the FPR and still allow the vehicle to meet its requirements.

The recommended approach assumes that the remaining LO_2 and LH_2 values have a probability distribution that is well modeled as a correlated two-dimensional Gaussian. This seems to be

a satisfactory representation of the data in hand. An analysis by Allan Benjamin has confirmed that the example dataset is consistent with the Gaussian assumption. Other sample datasets examined have outlying data points that are not consistent with a Gaussian distribution so that the resulting FPR would not in fact cover the required outliers. In this case, order statistics (app. C) may be used to obtain better estimates for the required FPR. It is also assumed that the optimum approach is the one that maximizes the total mass of propellant included in RUP, subject to success constraints.

Thus,

$$P(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\frac{1}{\sqrt{1-\rho^2}}\exp\left(-\frac{1}{2}\frac{1}{1-\rho^2}\left(\frac{(x-\bar{x})^2}{\sigma_x^2}-2\rho\frac{(x-\bar{x})}{\sigma_x}\frac{(y-\bar{y})}{\sigma_y}+\frac{(y-\bar{y})^2}{\sigma_y^2}\right)\right), \quad (95)$$

where $x=\mathrm{LH}_2$ remaining and $y=\mathrm{LO}_2$ remaining. Note $x+y$ is the total mass of RUP to be optimized.

Given a set of $N=2{,}000$ Monte Carlo data pairs (x_i,y_i), estimates of the mean, standard deviation, covariance and correlation coefficients are computed. This gives sample parameters

$$m_x=\frac{1}{N}\sum_{i=1}^{N}x_i, \quad s_x=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-m_x)^2} \quad (96)$$

and the same for y, while the covariance is

$$\mathrm{cov}(x,y)=\frac{1}{N-1}\sum_{i=1}^{N}(x_i-m_x)(y_i-m_y) \text{ and } \rho=\frac{\mathrm{cov}(x,y)}{s_x s_y}. \quad (97)$$

For the example dataset (shown in fig. 45):

	x	y
m_x, m_y	1,058.1	2 857.9
s_x, s_y	320.4	585.4
cov(x,y)	−46,260	
ρ	−0.247	

Note in this case the x and y parameters have a mild negative correlation.

F.3 A Preliminary Calculation

Using these parameters, contours of constant probability density can be computed. Figure 45 shows contours for two different success probabilities and also a trial candidate optimum, which turns out not to satisfy system requirements. The green curve shows a proportion of propellant at the nominal mixture ratio (5.5, meaning 5.5 lb LO_2 for every pound of LH_2). This was the assumption for the candidate optimum.

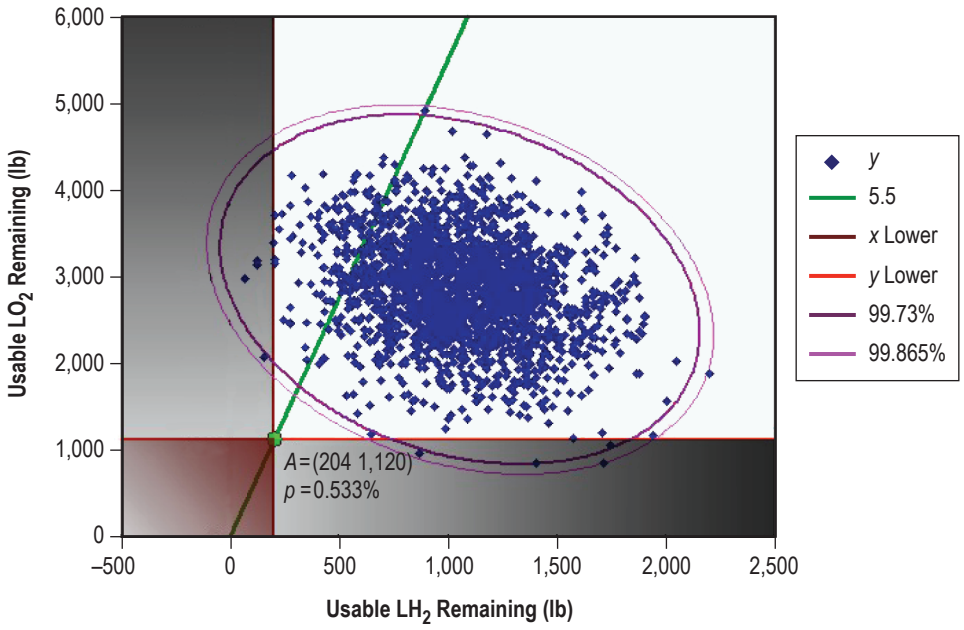

Figure 45. Example dataset with a candidate optimum RUP. Integrating the probability distribution over the L-shaped region gives the failure probability.

The aim is to pick (x_0, y_0) for RUP, satisfying a predetermined limit on failure rate. 'Failure' is defined as '(x,y) such that $(x < x_0)$ or $(y < y_0)$'. An estimate for the failure probability can be computed from the parameters of the two-dimensional Gaussian distribution by integrating over the L-shaped shaded area in figure 46. In practice, this integral can be calculated in three parts, first over the vertical and horizontal semi-infinite planes:

$$p_x = \int_{x=-\infty}^{x_0} \int_{y=-\infty}^{\infty} P(x,y)\,dx\,dy \tag{98}$$

and

$$p_y = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{y_0} P(x,y)\,dx\,dy \tag{99}$$

and then the overlap (the lower left-hand corner)

$$p_c = \int_{x=-\infty}^{x_0} \int_{y=-\infty}^{y_0} P(x,y)\,dx\,dy \ . \tag{100}$$

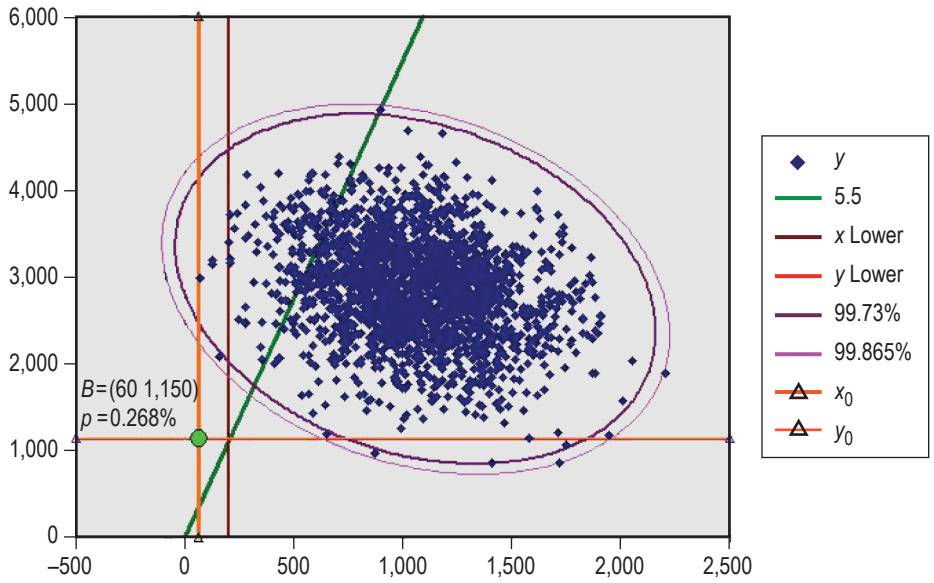

Figure 46. Revised setpoint satisfies upper bound on failure probability.

Then, $p_{\text{fail}} = p_L \equiv p_x + p_y - p_c$ (because adding the two half-planes overcounts the contribution of the corner). It is straightforward to compute p_x and p_y because the x and y distributions are Gaussian. The overlap p_c is more troublesome. Note that $p_c = p_x p_y$ only if x and y are uncorrelated, but note $p_x p_y \ll p_x + p_y$, since the probabilities are small. The double integral in general does not have a closed form. However, ignoring this contribution to p_L is slightly conservative; i.e., it allows an overestimate of the failure probability, and thus evaluating this integral is usually not necessary. On the other hand, if a case with correlation between x and y strongly positive (e.g., $\rho > 0.5$) is encountered, it may be worthwhile to account for this correction. Otherwise $p_L \doteq p_x + p_y$ is appropriate.

If p_L (204, 1,120) is computed in this way, for example, the result is $p_{\text{fail}} = 0.533\%$, which is consistent with the eight exceedances in this dataset (fig. 45).[21]

Figure 46 shows that a point can be selected that meets the target failure rate $p_{\text{fail}} = 0.27\%$. In this case, a manual search of (x,y) space produced this solution, which has the maximum value of $x + y$ which is consistent with this failure rate.

If the exceedances are counted, there are now five runs that fall outside the box, all with low LO_2. This is consistent with an overall reliability of $p_{\text{fail}} \leq 0.27\%$, but raises the interesting question of consumer risk.

F.4 Optimizing Remaining Usable Propellant With Attention to Consumer Risk

There is, of course, a formulation of this optimization problem that will converge rapidly to the maximum RUP that satisfies the bound on failure probability. Before delineating this approach, the correction for consumer risk must be considered.

This procedure does not depend on binomial statistics to generate the point (x_0, y_0). Because all $N = 2{,}000$ data runs are used to fit a Gaussian, the result is expected to be somewhat more robust than the binomial method for small numbers of failures. However, there are five parameters—two means, two standard deviations, and a correlation—that are samples of population means, deviations, etc. Each of these is dispersed around an expected value. Thus, in order to account for consumer risk, the uncertainties in the two-dimensional Gaussian parameters must be computed.

F.5 Uncertainties in the Two-Dimensional Parameters

In the recommended approach, the conservative assumption has been adopted that the double counting implicit in writing $p_L \doteq p_x + p_y$ can be ignored, which is a very safe assumption when p_x and p_y are small and the correlation ρ is not strongly positive. In this approach, the correlation is not used in the calculation. So the uncertainty in the probability p_L arises from the use of the sample means m_x, m_y and sample standard deviations s_x, s_y to estimate the population mean and variance.

These parameters have uncertainties—results are precisely true only for a Gaussian parent distribution. The Central Limit Theorem tells us this should be a good approximation for the uncertainties in the means and standard deviations for $N = 2{,}000$:

$$\delta_{m_x} = \frac{\sigma_x}{\sqrt{N}} \tag{101}$$

and

$$\delta_{s_x} = \frac{\sigma_x}{\sqrt{2N}} \tag{102}$$

and analogously for y. Despite the proportionality of these uncertainties themselves, the dispersions associated with the uncertainties are uncorrelated. This TP will use δ in place of σ to denote these uncertainties, although the uncertainties are themselves standard deviations of their respective random variables (the sample mean and sample standard deviation). Note δ_m is frequently called the standard error of the mean. In analogy, δ_s will be called the standard error of the standard deviation.

Given values of x and y, what is the uncertainty in the probability $p_L = p_x + p_y$? A detailed derivation is given in reference 5. The result is

$$\delta_{p_L} = \frac{1}{\sqrt{N}} \sqrt{f_x^2 \left(1 + \frac{z_x^2}{2} \right) + f_y^2 \left(1 + \frac{z_y^2}{2} \right)} \ , \tag{103}$$

for uncertainty in the probability of failure where

$$z_x \equiv \frac{x - m_x}{s_x} \tag{104}$$

and

$$f_x = f(z_x) = \frac{\exp\left(-z_x^2/2\right)}{\sqrt{2\pi}} \tag{105}$$

and analogously for y. This expression has been checked with numerical simulations. For example, the standard deviation of 1,000 runs of one particular 2,000-sample simulation was 0.000294, while this formula gives 0.000293.

F.6 Compensating for Consumer Risk

Up to this point, the optimization target probability has not been distinguished from the required probability. From this point on, this section will use $\hat{p}$ to denote the required not-to-exceed failure rate, and p^* to denote the probability that will be used in the constrained optimization.

To find the optimum coordinates (x_0, y_0), the constraint probability p^* must be adjusted relative to the required failure rate, $\hat{p} = 0.27\%$, in order to account for consumer risk. For 90% confidence (10% consumer risk), the appropriate adjustment is

$$p^* + z_{90}\delta_{p_L} = \hat{p} \Rightarrow p^* = \hat{p} - z_{90}\delta_{p_L} \ , \tag{106}$$

where $z_{90} = \Phi^{-1}(90\%) = 1.2815$, where $\Phi(z)$ is the standard Gaussian CDF. An implicit assumption is that the distribution of the errors associated with the uncertainty δ_{P_L} is Gaussian, which the Central Limit Theorem indicates is well justified for the relatively large $N = 2{,}000$. Since the uncertainty δ_{pL} is itself a function of (x,y), this dependence will be implemented in the computation.

F.7 Constrained Optimization—The Lazy-But-Cost-Effective Version

The constrained optimization can now be formulated. The procedure described in this section implements the simplifying assumption that the 'gradient' of δ_{P_L} can be safely ignored in the Lagrange equation, although the dependence on x and y is included. Reference 5 addresses how this assumption can be lifted, at the cost of making the RUP calculation more iterative, and with a very small increase in accuracy.

The general procedure for Lagrange multipliers is used, where the 'objective function' $g(x,y) = x + y$ is to be maximized subject to the constraint function $h(x,y) = p_L - p^* = 0$, where $p^* = \hat{p} - z_{90}\delta_{P_L}$, as described above, is slightly smaller than the target failure rate of $\hat{p} = 0.27\%$.

Note the Lagrangian has the form

$$L(x, y, \lambda) = g(x, y) + \lambda h(x, y) \ . \tag{107}$$

In this case of a single scalar constraint function, the procedure for handling Lagrange multipliers is simple: the gradients of g and h must be proportional; i.e., ignoring $\nabla\delta_{P_L}$,

$$\nabla h = \frac{\partial \Phi(z_x)}{\partial x}\mathbf{i} + \frac{\partial \Phi(z_y)}{\partial y}\mathbf{j} = \frac{f_x}{s_x}\mathbf{i} + \frac{f_y}{s_y}\mathbf{j} \tag{108}$$

must be proportional to

$$\nabla g = \frac{\partial g}{\partial x}\mathbf{i} + \frac{\partial g}{\partial y}\mathbf{j} = \mathbf{i} + \mathbf{j} \ . \tag{109}$$

This leads to the conclusion that the optimum choice of x and y will satisfy

$$\frac{f_x}{s_x} = \frac{f_y}{s_y} \ . \tag{110}$$

Note the expressions here are the normal distribution PDFs, normalized for x and y, respectively, as explained above. To avoid misunderstanding, note that the Lagrange equation does not imply $f_x = f_y$, or even $z_x = z_y$, but rather that the normalized densities are equal.

To find the optimum (x_0, y_0), the pair of equations comprising this Lagrange equation and the constraint equation

$$p_L - p^* = \Phi\left(\frac{x - m_x}{s_x}\right) + \Phi\left(\frac{y - m_y}{s_y}\right) - p^* = 0 \tag{111}$$

must be solved simultaneously.

Figure 47 portrays the meaning of the optimum point. The curves in the lower left are loci of constant failure rate, with and without accounting for consumer risk. The optimum point is the tangent point for a line of constant total RUP. Thus, the optimum is seen to maximize RUP while satisfying the constraint on failure rate.

Next, the computational flow for the iterative calculation is outlined.

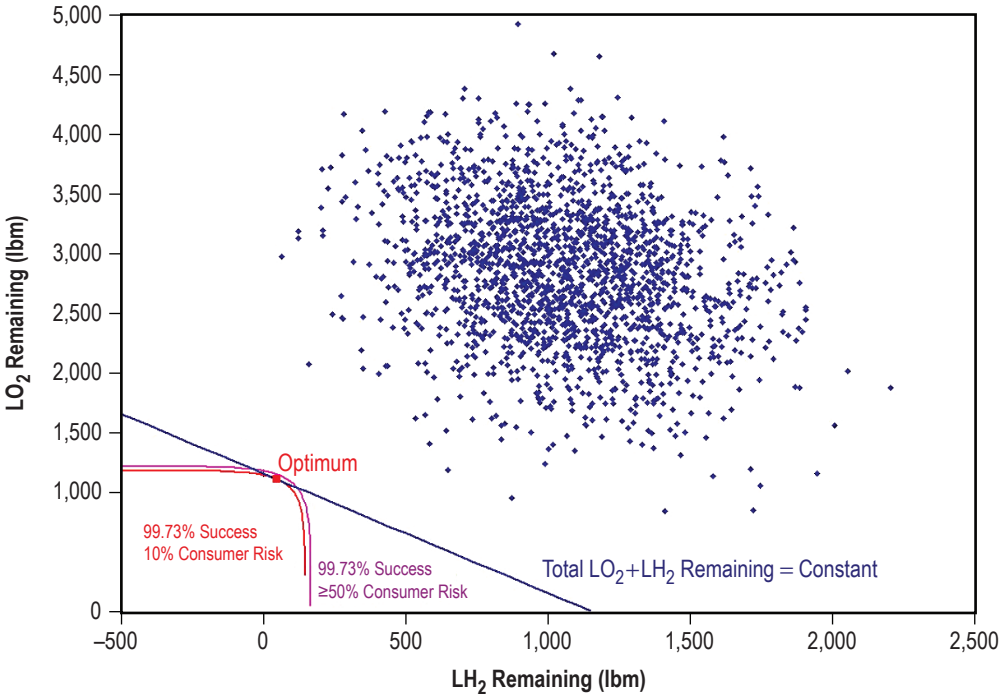

Figure 47. Depiction of optimizing procedure, usable LO_2 remaining versus LH_2.

F.8 A Computational Strategy

The computational strategy includes the following:

- Select a required value for failure probability ($\hat{p} = 0.27\%$, for example).
- Gather the $N = 2{,}000$ pairs (x_i, y_i) that define the dataset.
- Compute the sample mean and standard deviation for x and y:

$$m_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad s_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - m_x)^2} \tag{112}$$

and

$$m_y = \frac{1}{N}\sum_{i=1}^{N} y_i \quad s_y = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - m_x)^2} \tag{113}$$

- Choose a starting guess for x_0; e.g., $x = m_x - 3s_x$.

- Compute, in order,

$$z_x \equiv \frac{x - m_x}{s_x} \ , \tag{114}$$

$$f_x = \frac{\exp\left(-z_x^2/2\right)}{\sqrt{2\pi}} \ , \tag{115}$$

$$f_y = \frac{s_y}{s_x} f_x \text{ (Lagrange equation)}, \tag{116}$$

$$z_y = -\sqrt{-2\ln\left(\sqrt{2\pi} f_y\right)} \ ; \tag{117}$$

i.e., invert the expression for f_y and take the 'negative' root for z_y,

$$y = m_y + z_y s_y \ , \tag{118}$$

$$p_x = \Phi\left(z_x\right) \ ; \tag{119}$$

e.g., using Excel's NORMSDIST() function, or

$$\Phi(z) = \left(\text{erf}\left(z/\sqrt{2}\right) + 1\right)/2 \ , \tag{120}$$

$$p_y = \Phi\left(z_y\right) \ , \tag{121}$$

$$\delta_{p_L} = \frac{1}{\sqrt{N}} \sqrt{f_x^2\left(1 + \frac{z_x^2}{2}\right) + f_y^2\left(1 + \frac{z_y^2}{2}\right)} \text{ (uncertainty relation)}, \tag{122}$$

$$p^* = \hat{p} - z_{90}\delta_{p_L} \ , \tag{123}$$

$$h(x,y) = \left(p_x + p_y\right) - p^* \text{ (constraint function)}. \tag{124}$$

- Now adjust the value of x until $h(x,y) = 0$. The dependence of h on the input x should be generally well behaved.

- The optimum (x_0, y_0) gives $h(x_0, y_0) = 0$.

An example of setting up this calculation on a spreadsheet is given in reference 5.

F.9 Another Approach

It is interesting to compare this method to an alternative approach. Suppose the margins on LO_2 and LH_2 are set independently, using binomial order statistics, in such a way that the total failure rate is 0.27%. That is, choose the order statistics for each at the 99.865% level (with 10% consumer risk), which means that interpolation between the smallest and next to smallest points for each component is required. (For $N = 2{,}000$, the high index for this failure probability is 1,999.749 (see app. B), and the low is 0.251, corresponding to percentiles 99.987% and 0.013%.)

For the example dataset, this gives $(x_0, y_0) = (80.7, 855.0)$, for a total of $x_0, y_0 = 935.7$. Compare this to the output of the Gaussian method $x_0 + y_0 = 1{,}169.8$. The difference is due to two factors: (1) the nonoptimum distribution of risk between LO_2 and LH_2, and (2) the considerably higher producer risk associated with binomial statistics.

Note for $(x_0, y_0) = (80.7, 855.0)$, the expected failure rate is 0.145%, which is considerably less than the 10% consumer risk failure rate of 0.27%, reflecting the increased producer risk of this method. For this total reserve (935.7), the minimum expected failure rate is 0.099%, for the setpoint $(x_0, y_0) = (0, 935.7)$, assuming that x_0 is not allowed to drop below zero.

F.10 Run-to-Run Variability

A concern was raised by analysts that consecutive Monte Carlo runs give apparently inconsistent results for RUP, using either the previous procedure or the recommended procedure. A careful analysis showed that the run-to-run variability was within the expected range for the RUP and RUP components, considered as random variables.

To what extent should one expect run-to-run agreement on the extreme values of RUP? That is, consecutive $N = 2{,}000$ trial simulations of a specified vehicle trajectory dispersion yield minimum RUPs that vary considerably from run to run.

The short answer is that under current simulation practice, runs will exhibit considerable scatter in extreme values.

Here, the case of three 2,000 sample runs for a particular Monte Carlo ensemble is discussed. Data provided include average and standard deviation for each component (LH_2 and LO_2) as well as minima and 99.73% low with 10% consumer risk (denoted 99.73%@10%) values. The run-to-run variation was apparently the source of some concern. See table 14.

Table 14. Remaining usable propellant.

	Propellant	Official Results	Alternative Set 1	Alternative Set 2
Minimum	LO_2 LH_2	879 70	658 23	831 122
99.73% low with 10% CR	LO_2 LH_2	991 134	745 40	866 198
Average	LO_2 LH_2	2,883 1,063	2,865 1,080	2,851 1,062
Standard deviation	LO_2 LH_2	585 320	596 311	597 307

The first check is to see whether the averages and standard deviations are consistent. To that end, first assume that the $3 \times 2{,}000 = 6{,}000$ samples are drawn from the same parent population, and then test to see whether the individual runs vary significantly from the joint averages. The overall averages for LH_2 and LO_2 are simply arithmetic averages of the three runs, while the overall standard deviations are root-mean-squares of the individual standard deviations, since the variances of the runs are additive. The average and standard deviation for each run can be compared with the overall mean and standard deviation by using a Z-test (normal probability), since the standard errors of the mean (subscript m) and standard deviation (subscript s)are known to be

$$\delta_m = \frac{\sigma}{\sqrt{N}} \tag{125}$$

and

$$\delta_s = \frac{\sigma}{\sqrt{2N}} \ . \tag{126}$$

The overall averages, standard deviations, and standard errors are given in table 15.

Table 15. Overall averages and standard errors.

	Propellant	Overall	Standard Error
Average	LO_2 LH_2	2,866.3 1,068.3	13.3 7.0
Standard deviation	LO_2 LH_2	592.7 312.7	9.4 4.9

For each parameter and each run, Z-values were computed:

$$Z_i = \frac{x_i - m_i}{\delta_i} \ . \tag{127}$$

The computed Z-values are given in table 16.

Table 16. Z-values for runs, relative to overall means.

	Propellant	Official Results	Alternative Set 1	Alternative Set 2
Average	LO_2	1.258	–0.101	–1.157
	LH_2	–0.763	1.668	–0.906
Standard deviation	LO_2	–0.821	0.353	0.460
	LH_2	1.474	–0.347	–1.156

None of these Z-values are extreme, and, in fact, the average of these 12 Z's is –0.0031, while the standard deviation is 1.0336, consistent with the expected standard normal distribution.

The conclusion is that, using only the averages and standard deviations, there is no reason to suspect that these three runs are from different populations.

Next, consider the minima and the 99.73%@10% values.

F.10.1 Run-to-Run Variation of the Minimum

First consider the minimum values. The Z-values computed with the overall average and standard deviation are given in table 17.

Table 17. Minimum and Z-values.

	Propellant	Official Results	Alternative Set 1	Alternative Set 2
Minimum	LO_2	879	658	831
	LH_2	70	23	122
Z-value	LO_2	–3.3531	–3.7259	–3.4341
	LH_2	–3.1925	–3.3428	–3.0262

Note the probability distribution for minimum is the mirror image of the distribution for maximum. To avoid confusion arising from the negative signs, this discussion will be expressed in terms of the probability distribution for maximum. The absolute Z-values in table 17 will be compared with the CDF for maximum.

Either order statistics or extreme value theory can be used to generate the distribution function for maximum. These two techniques are not congruent, but give very similar results. In this calculation, the order statistics for $N = 2{,}000$, $r = 2{,}000$ will be used, where the probability density is

$$f_{Y_r} = \frac{N!}{(r-1)!(N-r)!}\big(F(x)\big)^{r-1}\big(1-F(x)\big)^{N-r} f(x) \ . \tag{128}$$

$F(x)$ is the CDF of the parent distribution function $f(x)$. This gives the smooth red curve in figure 48. Note the blue curve is the probability density, with the scale on the left, while the red curve is the CDF, with the scale on the right. The empirical cumulative distribution for the six sorted Z-values is the stair-step line on the plot.

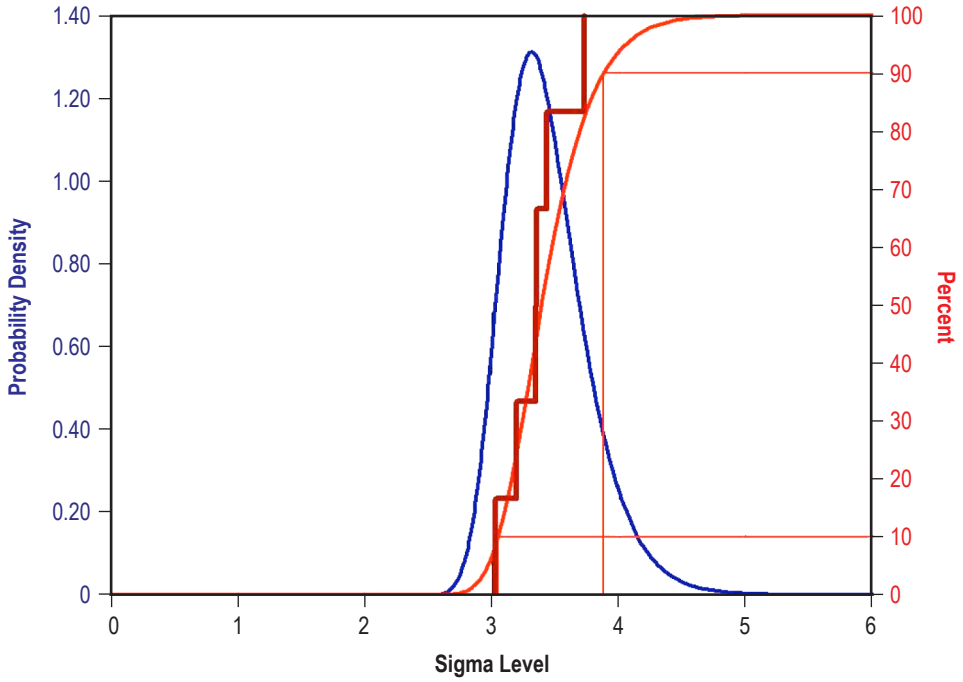

Figure 48. Cumulative distribution and PDF for $N = 2{,}000$, $r = 2{,}000$ order statistic for a standard Gaussian distribution.

The order statistics also yield the 10% and 90% confidence intervals $Z_{10} = 3.0483$ and $Z_{90} = 3.8780$ as shown on the plot.

Note that none of the six minima fall outside the 10%–90% confidence interval. This suggests that the run-to-run variability that is exhibited is within the expected range. Applying the Kolmogorov-Smirnov test gives a Q value of 65%, well within margins for the null hypothesis. Thus, the six-run CDF appears to be in good agreement with the order statistics.

The conclusion is that the scatter in the minimum values agrees with the expected scatter.

F.10.2 Comparing the 99.73% at 10% Values

Next, consider the 99.73%@10% values. According to the $N = 2{,}000$ interpolation procedure, this is equivalent to selecting the 99.897 percentile value directly (or rather, for low values, using the complementary 100%–99.897% = 0.103 percentile). The data and Z-values computed with the overall average and standard deviation are given in table 18.

Table 18. 99.73%@10% and Z-values.

	Propellant	Official Results	Alternative Set 1	Alternative Set 2
99.73% low @ 10% CR	LO_2 LH_2	991 134	745 40	866 198
Z-value	LO_2 LH_2	–3.1641 –2.9878	–3.5792 –3.2884	–3.3750 –2.7832

In practice, it is cumbersome to apply order statistics to a nonround percentile like 99.897%, since the PDF is a convolution of the nontrivial PDFs for two r values. For the purposes here, it is close enough to examine the order statistic for $r = 1{,}998$, $N = 2{,}000$, which is essentially the 99.90 percentile. In this case, the distribution function is shown in figure 49. The 10%–90% confidence interval and the empirical CDF for the six data points. The confidence bounds for this order statistic are $Z_{10} = 2.7871$ and $Z_{90} = 3.2631$.

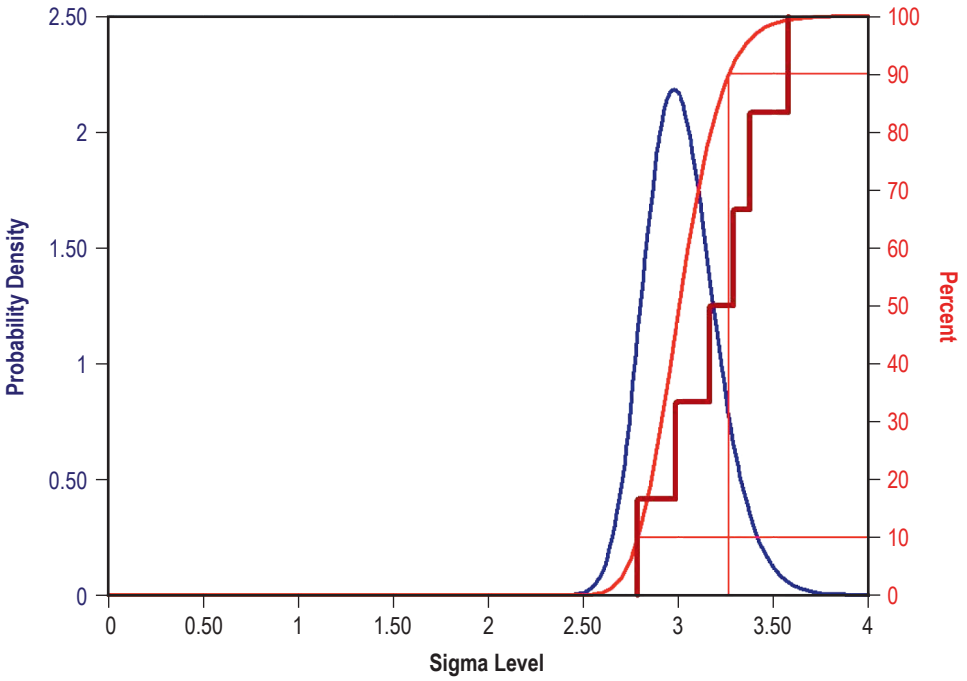

Figure 49. Cumulative distribution and PDF for $N = 2{,}000$, $r = 1{,}998$ order statistic for a standard Gaussian distribution.

In this case, the agreement is slightly less convincing. It appears that half the runs fall on the far side of the 90% confidence bound, though only very slightly. Due to the small number of runs, it is difficult to say definitively whether this is evidence of a processing error or bug. A computation of the Kolmogorov-Smirnov goodness of fit test gives a Q value of 11%, which is (barely) consistent with the null hypothesis. It may be worthwhile to recheck the inputs to the 99.73%@10% procedure to make sure that the appropriate percentile is being selected.

F.11 Summary

Using a two-dimensional Gaussian parametric approach to optimize RUP gives a straightforward optimization procedure which is superior to allocation of risk with binomial statistics. The procedure incorporates the standard error of the mean and the standard error of the standard deviation for sample parameters in order to account for consumer risk. The results should be checked to see that they do in fact cover the required percentage of cases, since multiple outliers may exist for some datasets. If the Gaussian assumption is not good, then using order statistics (app. B) would be a reasonable approach. Also note that even if the Gaussian assumption is reasonable, if the Monte Carlo simulation does not show the required number of successes are achieved, then additional propellant will be necessary, since success must be demonstrated in the Monte Carlo results in order to verify that the system satisfies the requirement. Additional efforts to characterize the tail of the distribution and to account for the effect of low fuel cutoff sensors is the subject of a future publication.

APPENDIX G—IMPORTANCE SAMPLING

In situations where trajectory Monte Carlo is used in a 'closed-end' calculation (e.g., where the average of a single parameter over an ensemble is computed), the efficiency of the calculation can be improved by using a variety of techniques that fall in the general category of 'variance reduction.' The point of variance reduction is that the standard error of a Monte Carlo calculation is proportional to the standard deviation (the square root of the variance) of the integrand. A number of techniques have been developed over the years to reduce this variance by adjusting the integrand or manipulating the traversal of the region of integration. One of the basic techniques of variance reduction is 'importance sampling.'

The idea behind importance sampling is that the variance of the integrand can be reduced if the sampling is biased toward regions where the integrand is largest. That is, using generic notation, the integrand can be rewritten

$$\int f(x)dx = \int \frac{f(x)}{P(x)} P(x)dx \ , \tag{129}$$

where $P(x)$ is a function chosen to satisfy the requirements of a probability density (finiteness, positivity, and normalization). If $P(x)$ can be devised to approximate the target function $f(x)$, then the new integrand $f(x)/P(x)$ can be sampled according to $P(x)$ and the result will be a reduction of variance. In some cases, an improvement of many orders of magnitude can be achieved. Figure 50 is a conceptual sketch of importance sampling.

So if the goal is to characterize the far tail of a distribution, the original integral (Monte Carlo run) will not provide much in the way of statistics. But if $P(x)$ is used to cause more activity in the tail, this improves the statistical data there, and the function $f(x)$ is then adjusted by dividing by $P(x)$ to compensate for the modification.

An example is the calculation of staging recontact probability. (An alternate approach to this calculation is discussed in appendix A.) Note the recontact probability can be written as

$$\int \Theta(r-R)\, p\left(\alpha_1, \alpha_2, ..., \alpha_m\right) d^m\alpha_i \ , \tag{130}$$

where α_i are the m stochastic input parameters, $p(\alpha_i)$ is the joint probability density of the α_i, $r = r(\alpha_i)$ is the radial drift during staging, R is the recontact radius, and $\Theta(x)$ is the step function.

For some circumstances, a dominant contribution to the recontact probability is the side force exerted by the BDMs. The BDM side force input to separation dynamics is modeled by a one-dimensional Gaussian with a specified mean and standard deviation. Even when the staging drift is dominated by the side force, the computed recontact probability may be small enough that very few

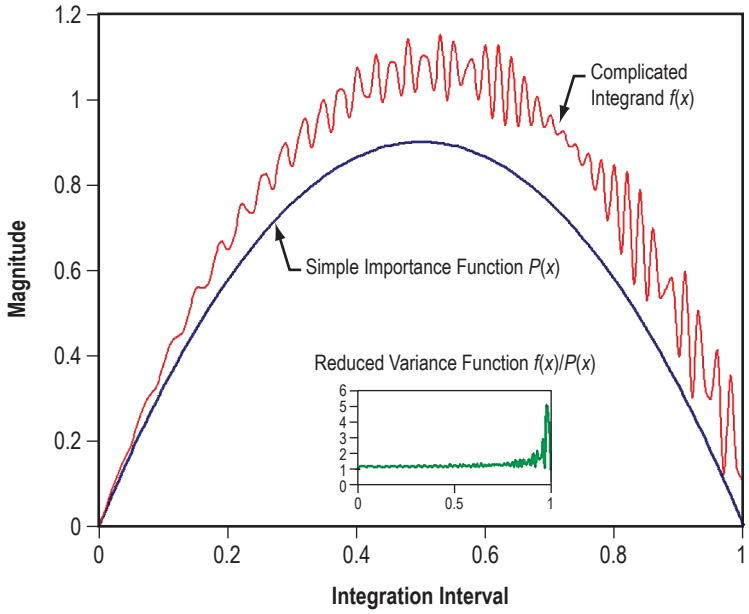

Figure 50. Conceptual sketch of importance sampling.

Monte Carlo samples generate recontacts. This means that the integrand has a relatively large variance, or worse, it may be completely misleading in that it never gets evaluated in regions where the step function is nonzero. That is, unless there is a recontact, the integral is zero.

This shortcoming can be corrected by introducing an importance sampling function. Suppose α is the BDM side force, and suppose the probability density factors into

$$p\left(\alpha_1,\ \alpha_2,\ ...,\ \alpha_m\right) = q\left(\alpha_1\right) p'\left(\alpha_2, \alpha_3, ..., \alpha_m\right) \ . \tag{131}$$

Then, a revised $q'(\alpha_1)$ with a higher mean than the original can be introduced so as to generate more recontacts in the importance-sampled ensemble. The integrand is then

$$\Theta(r-R)\, p\left(\alpha_1, \alpha_2, ..., \alpha_m\right) = \Theta(r-R) \frac{q\left(\alpha_1\right)}{q'\left(\alpha_1\right)} q'\left(\alpha_1\right) p'\left(\alpha_2, \alpha_3, ..., \alpha_m\right) \ , \tag{132}$$

so that instead of evaluating $\Theta(r–R)$ against the original probability $p(\alpha_i)$, $\Theta(r–R)\, q(\alpha_1)/q'(\alpha_1)$ is evaluated against the revised probability $q'(\alpha_1)p'(\alpha_2, \alpha_3, ..., \alpha_m)$. The revised probability causes more recontacts, and then the step function is adjusted to compensate for the modification.

Although this technique has not yet been applied to Ares I trajectory simulations, it shows promise for closed-end trajectory Monte Carlo simulations, where a single output parameter is being measured and the overall effect of the most important inputs on this parameter is understood.

REFERENCES

1. Negele, J.W.; and Orland, H.: *Quantum Many-Particle Systems*, Addison-Wesley, p. 400 and following, 1988.

2. Metropolis, N.: "The Beginning of the Monte Carlo Method," Los Alamos Science Special Issue, Vol. 15, p. 125, 1987, <http://lib-www.lanl.gov/la-pubs/00326866.pdf>, accessed September 2, 2010.

3. Feynman, R.: *Surely You're Joking Mr. Feynman*!, Bantam, 1985.

4. Kochanski, G.: "Monte Carlo Simulation," Creative Commons, Stanford, CA, <http://kochanski.org/gpk/teaching/0401Oxford/MonteCarlo.pdf>, 2005, accessed September 2, 2010.

5. Beard, B.B.: "Some Notes on the Use of Statistical Methods in the Design of Launch Vehicles," ARES Corporation Report No. 080251001-001, January 2009.

6. Ferson, S.: "Introduction to Imprecise Probabilities," Proceedings of the Summer School on Imprecise Probabilities, Lugano, Switzerland, July 2004.

7. Ferson, S.; Kreinovich, V.: Ginzburh, L.; et al.: "Constructing Probability Boxes and Dempster-Shafer Structures," SAND Report SAND2002-4015, January 2003.

8. A Compendium of Common Probability Distributions, Version 2.3, <http://www.causascientia.org/math_stat/Dists/Compendium.pdf>, accessed September 2, 2010.

9. *Engineering Statistics Handbook*, Gallery of Distributions, <http://www.itl.nist.gov/div898/handbook/eda/section3/eda366.htm>, accessed September 2, 2010.

10. Leslie, F.W.; and Justus, C.G.: "Earth Global Reference Atmospheric Model 2007 (Earth-GRAM07) Applications for the NASA Constellation Program," 13th Conference on Aviation, Range, and Aerospace Meteorology, New Orleans, LA, January 2008.

11. David, H.A.; and Nagaraja, H.N.: *Order Statistics*, 3rd Ed., Wiley-Interscience, 2003.

12. White, K.P.; Johnson, K.L.; and Creasey, R.R.: "Attribute Acceptance Sampling as a Tool for Verifying Requirements Using Monte Carlo Simulation," *Quality Engineering*, Vol. 21, No. 2, pp. 203–214, April 2009.

13. Coles, S.: *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London, 2001

14. Landau, D.P.; and Binder, K.: *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge University Press, p. 51, 2000.

15. Wyss, G.D.; and Jorgensen, K.H.: "A User's Guide to LHS: Sandia's Latin Hypercube Sampling Software," SAND98-0210, <http://prod.sandia.gov/techlib/access-control.cgi/1998/980210.pdf>, February 1998, accessed September 2, 2010.

16. Robinson, D.; and Atcitty, C.: "Comparison of Quasi- and Pseudo-Monte Carlo Sampling for Reliability and Uncertainty Analysis," Paper AIAA–99–1589, Proceedings of the AIAA Probabilistic Methods Conference, St. Louis, MO, April 12–15, 1999.

17. Wang, R.; Diwekar, U.; and Gregoire Padro, C.: "Efficient Sampling Techniques for Uncertainties in Risk Analysis," American Institute of Chemical Engineers, <http://www.vri-custom.org/pdfs/paper35.pdf>, 2004, accessed September 2, 2010.

18. Marsaglia, G.; and Tsang, W.W.: "The Ziggurat Method for Generating Random Variables," *J. Stat. Soft*., Vol. 5, No. 8, pp.? , 2000, <http://www.jstatsoft.org/v05/i08/paper>, accessed September 2, 2010.

19. Knuth, D.E.: *The Art of Computer Programming*, Vol. 2, p. 1, 2nd Ed., Addison-Wesley, 1998.

20. Negele, op. cit., 407.

21. Beard, B.B.: "Lectures on Quantum Monte Carlo Methods – Lecture 2," from lectures given at Universitá Degli Studi di Firenze (UniFi), June–July 2001, <http://bbbeard.com/QMC/Lecture2.pdf>, accessed September 2, 2010.

22. Marsaglia, G.: "Diehard Battery of Tests of Randomness," Florida State University, <http://www.stat.fsu.edu/pub/diehard/>, 1995, accessed September 2, 2010.

23. Matsumoto, M.; and Nishimura, T.: "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator," *ACM Trans. on Modeling and Computer Simulation*, Vol. 8, No. 1, pp. 3–30, January 1998; Other relevant documents available online at <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/ARTICLES/earticles.html>, accessed September 2, 2010.

24. Law, A.M.: *Simulation Modeling and Analysis*, 4th Ed, McGraw-Hill, 2007.

25. Derived by Bernoulli, Jacob: 1654–1705. See Feller, W., "An Introduction to Probability Theory and Its Applications," Volume I, 3rd Ed., J. Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, p. 148, 1968.

26. Birolini, A.: *Reliability Engineering: Theory and Practice*, 4th Ed., Berlin: Spring-Verlag, 2004.

27. Hyndman, R.J.; and Fan, Y.: "Sample Quantiles in Statistical Packages," *American Statistician*, Vol. 50, No. 4, pp. 361–365, doi: 10.2307/2684934, November 1996.

28. Rectangular Function, In Wikipedia, the free encyclopedia, <http://en.wikipedia.org/wiki/Rectangular_function>, accessed September 2, 2010.

29. Sign Function, In Wikipedia, the free encyclopedia, <http://en.wikipedia.org/wiki/Sign_function>, accessed September 2, 2010.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01–09–2010	Technical Publication	

4. TITLE AND SUBTITLE

Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

J.M. Hanson and B.B. Beard

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

8. PERFORMING ORGANIZATION REPORT NUMBER

M–1296

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITOR'S ACRONYM(S)

NASA

11. SPONSORING/MONITORING REPORT NUMBER

NASA/TP—2010–216447

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified-Unlimited
Subject Category 65
Availability: NASA CASI (443–757–5802)

13. SUPPLEMENTARY NOTES

Prepared by the Spacecraft & Vehicle Systems Department, Engineering Directorate

14. ABSTRACT

This Technical Publication (TP) is meant to address a number of topics related to the application of Monte Carlo simulation to launch vehicle design and requirements analysis. Although the focus is on a launch vehicle application, the methods may be applied to other complex systems as well. The TP is organized so that all the important topics are covered in the main text, and detailed derivations are in the appendices. The TP first introduces Monte Carlo simulation and the major topics to be discussed, including discussion of the input distributions for Monte Carlo runs, testing the simulation, how many runs are necessary for verification of requirements, what to do if results are desired for events that happen only rarely, and postprocessing, including analyzing any failed runs, examples of useful output products, and statistical information for generating desired results from the output data. Topics in the appendices include some tables for requirements verification, derivation of the number of runs required and generation of output probabilistic data with consumer risk included, derivation of launch vehicle models to include possible variations of assembled vehicles, minimization of a consumable to achieve a two-dimensional statistical result, recontact probability during staging, ensuring duplicated Monte Carlo random variations, and importance sampling.

15. SUBJECT TERMS

Monte Carlo, launch vehicle requirements verification, order statistics, acceptance sampling, launch vehicle ascent simulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk at email: help@sti.nasa.gov
U	U	U	UU	130	19b. TELEPHONE NUMBER *(Include area code)* STI Help Desk at: 443–757–5802

National Aeronautics and
Space Administration
IS20
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama
35812