Gap Conductance in Power Generator Clad Temperature Measurement

Variance Modeling in a Nonlinear, Non-Normal Setting

Tom Balmat

Abstract

Two important objectives in the design of mechanical systems are that they serve some useful function (fit-for-use) and that they operate reliably and consistently. Often, fit-for-use criteria are emphasized over consistent operation (what good is a reliable device that does nothing?) but, within the range of what would be considered good service, are all possible system settings explored to identify maximum consistent, or minimal variation, performance, thus striking better balance between fit-for-use and reliability? Real world systems are often nonlinear and modeling their subsystems may involve non-normal statistical distributions. This paper explores the use of linear approximation of variance (LAV), nonlinear programming (NLP), and non-normal statistical modeling in the context of a parameter design problem (PDP), that is, the specification of new parameter levels that maintain desired system performance, while reducing variation in performance.

Keywords: Quality; Parameter Design; Variance Modeling; Non-normal Statistical Modeling; Nonlinear Programming

1. Introduction

In their book, *Statistical Modeling Techniques* (Shapiro and Gross pg. 314), Samuel Shapiro and Alan Gross present a physical model that relates a controllable factor, gap conductance, of a power generator subsystem to the generator's clad temperature, which must be monitored. Statistical distributions and operating ranges are given for each of seven independent parameters and it is suggested that the model and parameter distributions be used to simulate, in Monte Carlo fashion, the distribution of expected response values for the purpose of stating confidence intervals on gap conductance. But this merely models the subsystem within the current independent parameter operating regions, with expected value as a function of the expected values of the parameters. Is there an alternate set of parameter levels at which gap conductance remains at the current specified level, yet with reduced variance?

2. Statement of the Problem

The gap conductance model is given as (eq 1.1)

$$h_g = \frac{k_g}{t_g + 5.448\left(\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right)} + h_r$$

where

h_g = gap conductance (the response)
k_g = gas thermal conductivity of helium
t_g = gap thickness (result of manufacturing operations)

u = viscosity of one of the components
P = internal pressure
T = internal temperature
M = molecular weight of helium
h_r = radiant heat transfer coefficient

Units of measure are not provided. We will use this equation to model the effect of parameter level changes on gap conductance, to solve for optimal (minimal variance) parameter level combinations, and to develop an empirical probability distribution from which response operating confidence intervals can be inferred.

3. Method

h_g is nonlinear in several of the parameters and its distribution is not obvious. To assess its distributional form, variance, and confidence regions, two models will be developed: one for variance optimization using LAV with NLP (ref Fathi) and one for probability distribution and confidence interval development by matching moments and/or percentiles of an appropriate theoretical distribution to a large Monte Carlo generated set of response values simulated using an optimal set of parameter levels from the NLP solutions.

NLP on an LAV Model

h_g is approximated by a first order Taylor polynomial as

$$\begin{aligned} h_g(k_g, t_g, u, P, T, M, h_r) \\ &\approx h_g(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r) + \frac{\partial h_g}{\partial k_g}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(k_g - \hat{k}_g) \\ &+ \frac{\partial h_g}{\partial t_g}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(t_g - \hat{t}_g) + \frac{\partial h_g}{\partial u}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(u - \hat{u}) \\ &+ \frac{\partial h_g}{\partial P}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(P - \hat{P}) + \frac{\partial h_g}{\partial T}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(T - \hat{T}) \\ &+ \frac{\partial h_g}{\partial M}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(M - \hat{M}) + \frac{\partial h_g}{\partial h_r}(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)(h_r - \hat{h}_r) \end{aligned}$$

where $\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r$ are target parameter operating levels. Being linear in its parameters $(k_g, t_g, u, P, T, M, h_r)$ the right hand side has variance

$$Var[h_g(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)] + \left(\frac{\partial h_g}{\partial k_g}\right)^2 Var(k_g - \hat{k}_g) + \left(\frac{\partial h_g}{\partial t_g}\right)^2 Var(t_g - \hat{t}_g) + \left(\frac{\partial h_g}{\partial u}\right)^2 Var(u - \hat{u})$$

$$+ \left(\frac{\partial h_g}{\partial P}\right)^2 Var(P - \hat{P}) + \left(\frac{\partial h_g}{\partial T}\right)^2 Var(T - \hat{T}) + \left(\frac{\partial h_g}{\partial M}\right)^2 Var(M - \hat{M}) + \left(\frac{\partial h_g}{\partial h_r}\right)^2 Var(h_r - \hat{h}_r)$$

where each partial derivative is evaluated at $(\hat{k}_g, \hat{t}_g, \hat{u}, \hat{P}, \hat{T}, \hat{M}, \hat{h}_r)$. This gives a LAV for h_g of (eq 1.2)

$$Var(h_g) \approx \left(\frac{\partial h_g}{\partial k_g}\right)^2 \sigma_{k_g}^2 + \left(\frac{\partial h_g}{\partial t_g}\right)^2 \sigma_{t_g}^2 + \left(\frac{\partial h_g}{\partial u}\right)^2 \sigma_u^2 + \left(\frac{\partial h_g}{\partial P}\right)^2 \sigma_P^2 + \left(\frac{\partial h_g}{\partial T}\right)^2 \sigma_T^2 + \left(\frac{\partial h_g}{\partial M}\right)^2 \sigma_M^2 + \left(\frac{\partial h_g}{\partial h_r}\right)^2 \sigma_{h_r}^2$$

The partial derivatives are

$$\frac{\partial h_g}{\partial k_g} = \frac{1}{t_g + 5.448\left[\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right]} \qquad \frac{\partial h_g}{\partial T} = \frac{-k_g(5.448)\left(\frac{u}{P\sqrt{M}}\right)\left(\frac{1}{\sqrt{T}}\right)}{\left(t_g + 5.448\left[\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right]\right)^2}$$

$$\frac{\partial h_g}{\partial t_g} = \frac{-k_g}{\left(t_g + 5.448\left[\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right]\right)^2} \qquad \frac{\partial h_g}{\partial M} = \frac{k_g(5.448)\left(\frac{u\sqrt{T}}{P}\right)\left(\frac{1}{M^{3/2}}\right)}{\left(t_g + 5.448\left[\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right]\right)^2}$$

$$\frac{\partial h_g}{\partial u} = \frac{-k_g(5.448)\left[\frac{1}{P}\left(\frac{T}{M}\right)^{1/2}\right]}{\left(t_g + 5.448\left[\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right]\right)^2} \qquad \frac{\partial h_g}{\partial h_r} = 1$$

$$\frac{\partial h_g}{\partial P} = \frac{k_g(5.448)\left[u\left(\frac{T}{M}\right)^{1/2}\right]\left(\frac{1}{P^2}\right)}{\left(t_g + 5.448\left[\frac{u}{P}\left(\frac{T}{M}\right)^{1/2}\right]\right)^2}$$

All but one are nonlinear, which necessitates the use of NLP to solve for variance optimal parameter solutions. The NLP objective is to minimize Var(h_g) given constraints 1.) h_g remains at its current specified operating level, and 2.) each parameter level remains within current specified operating boundaries or within a 99% confidence interval given individual respective probability distributions. Note that the σ_i^2 are provided from historical process knowledge or derived analytically from known parameter probability distributions, which are described in the next section. Derivations are given in appendix 1.

Modeling the Response with a Theoretical Probability Distribution: Moment/Percentile Matching

Given the form of eq 1.1, the distribution of h_g is not obvious. To make confidence interval statements and to project new operating levels an appropriate probability distribution must be specified. This can be done by matching moments or percentiles of a chosen theoretical distribution to those of actual response levels from the system. Lacking any empirical data, we must generate our own. NLP gives an estimate of variance (standardized second moment), but this explains dispersion only. We have no information on location or other distributional shape properties. To get these, we can run a Monte Carlo simulation by generating sets of pseudo-random parameter levels (taken from their respective probability distributions), calculating theoretical h_g values (from eq 1.1), and calculating pseudo-observed moments and percentiles to be matched to candidate theoretical distributions. Using probability plots or formal goodness of fit

tests, a reasonable theoretical distribution can be selected that, given the validity of the simulation, should agree with system results if operated at parameter levels identical to those used in the simulation.

To simulate h_g, each parameter must in turn be simulated and, for this, the probability distribution and associated parameters of each must be known. With this knowledge, pseudo-random parameter levels can be generated using a set of pseudo-random uniform numbers, one for each parameter, on the interval (0,1) and using these to derive corresponding percentiles from the inverse of respective cumulative density functions (cdf). Each set of pseudo-random parameter levels is a simulated sample operating condition from which h_g is simulated. Following are descriptions of operating constraints and accepted distributions of the parameters. Distribution properties, moment derivations, moment generating functions, and percentile methods are given in appendix 1.

k_g: Gas thermal conductivity of helium. This value is given as 0.0049. However, it is known to vary between 0.0043 and 0.0054. While the distribution is unknown, it is believed to be symmetrical, with values about 0.0049 most likely. Lacking knowledge of specific distributional form, a parabolic model on the interval [0.0043, 0.0054] will be used.

t_g: Gap thickness is a dimensional quantity and is the result of manufacturing operations. It has a tolerance specification of 0.025 ± 0.001 and is assumed to be normally distributed. Using nominal as the expected value and setting a 99.8% confidence interval to the tolerance range, we have estimates for μ_g and σ_g of 0.025 and 0.00033, respectively.

u: Viscosity is a physical material property believed to be 0.000039. However it is known to vary between 0.000030 and 0.000048, with all values in the range equally likely. Accordingly, a uniform distribution on the interval [0.000030 and 0.000048] will be used to model this parameter.

P: Pressure is observed to have a right-skewed distribution with minimum value of 450, mean levels of 500, and variance equal 100. It is believed that a lognormal model describes this parameter.

T: Temperature is expected to be symmetrical about 1600 with a standard deviation of 30. Clustering about the mean with rapid decline on either side, it is accepted that these values are normally distributed with said mean and standard deviation.

M: The molecular weight of helium is a physical constant, with value 4 and little expected variation. To account for possible slight deviations, it will be modeled by a uniform distribution on the interval [3.9999, 4.0001].

h_r: The radiant heat transfer coefficient has a minimum value of 0.6, which is its most likely value. Probabilities decrease as h_r increases. Given the distributional shape of this parameter, an exponential model, with mean of 0.8, will adequately represent it.

4. Solution

Variance Optimization Using NLP on the LAV model

The objective is to minimize the variance function

$$Var(h_g) \approx \left(\frac{\partial h_g}{\partial k_g}\right)^2 \sigma_{k_g}^2 + \left(\frac{\partial h_g}{\partial t_g}\right)^2 \sigma_{t_g}^2 + \left(\frac{\partial h_g}{\partial u}\right)^2 \sigma_u^2 + \left(\frac{\partial h_g}{\partial P}\right)^2 \sigma_P^2$$

$$+ \left(\frac{\partial h_g}{\partial T}\right)^2 \sigma_T^2 + \left(\frac{\partial h_g}{\partial M}\right)^2 \sigma_M^2 + \left(\frac{\partial h_g}{\partial h_r}\right)^2 \sigma_{h_r}^2$$

subject to constraints on the parameter levels. This effectively searches for parameter levels, different than the current operating averages but within specified operating ranges, such that the variance of h_g is minimized, while maintaining its specified expected value, or mean operating level. All partial derivatives but the last are nonlinear in some of the parameters, requiring NLP. The constraints, taken from supplied engineering data and known or historical operating regions, along with historical operating averages (nominal) and variance estimates are summarized below. All variances, except for P and h_r, are considered to be constant, but due to the functional relationship between the mean and variance of the lognormal and exponential models (details in appendix 1), the variances for P and h_r will be recalculated for each NLP iteration in terms of current corresponding NLP parameter levels. This involves aspects of Tolerance Design where, in addition to specifying new mean operating levels, parameter tolerances are also re-specified. Our object here is merely to find new process settings at which the system will operate with reduced variation in $h_g$and involving no component redesign or tolerance adjustment. However, due to the functional relationship of the mean and variance for some pdfs employed, such as the exponential and lognormal, changes in mean levels will introduce changes in variance. For more on Tolerance Design, see (Taguchi chapt. 4).

Response and Parameter Ranges				
Parameter	Lower Bound	Nominal	Upper Bound	Note
h_g	0.994	0.996	0.998	Based on expected gap conductance at mean parameter levels ± 0.002 (for numerical calculation space)
k_g	0.0043	0.0049	0.0054	Historical region end points
t_g	0.024	0.025	0.026	Normal mean ± 3 std dev
u	0.000030	0.000039	0.000048	Historical region end points
P	471	500	531	Lognormal mean and variance of 500 and 100 give normal mean and variance of 6.21 and .0004 (explanation in appendix 1). Lower and upper bounds, here, are exp(normal mean ± 3 normal std dev).
T	1510	1600	1690	Normal mean ± 3 std dev
M	3.9999	4.0000	4.0001	Physical constant, small deviation for natural variation
h_r	0.6	0.8	1.2	Lower bound from engineering spec, upper bound is 99% confidence limit of exponential distribution (explanation in appendix 1), λ calculated from historical mean and LB

Variance Estimates			
Parameter	Source*	Estimate	Note
$\sigma_{k_g}^2$	$\frac{(b-a)^2}{20}$	7.2E-8	Parabolic distribution
$\sigma_{t_g}^2$	± .001 tolerance / 6	1.11E-7	Normal distribution
σ_u^2	$\frac{(b-a)^2}{12}$	2.7E-11	Uniform distribution
σ_P^2	Historical	100	Lognormal distribution (recalculated each NLP iteration)
σ_T^2	Historical	900	Normal distribution
σ_M^2	$\frac{(b-a)^2}{12}$	3.33E-9	Uniform distribution
$\sigma_{h_r}^2$	$\frac{1}{\lambda^2} = (\bar{y} - .6)^2$	.04	Exponential distribution (recalculated each NLP iteration)

* theoretical derivations in appendix 1

SAS Proc NLP was used to solve this problem (ref SAS NLP). A 7-dimensional grid of initial value points was configured using the boundaries and two evenly spaced interior points for each parameter (program listing *NLP Initial Values* in appendix 2). This gave a total of 4^7=16,384 initial value sets. The objective function and parameter boundaries were programmed into Proc NLP (program listing *NLP Procedure* in appendix 2). Boundaries for the response, h_g, were set at nominal ± .002 to provide a small amount of numerical space for calculations, rounding, etc.

The NLP solutions demonstrated an optimal variance range from 0.0279198 to 0.0279237, with 764 solutions resulting in the lower value. Within these 764 solutions, all specified k_g at its upper bound, t_g at its lower bound, u at its lower bound, P at its upper bound, T at its lower bound, M distributed across its range, but concentrated at its upper bound, and h_r= 0.7671.

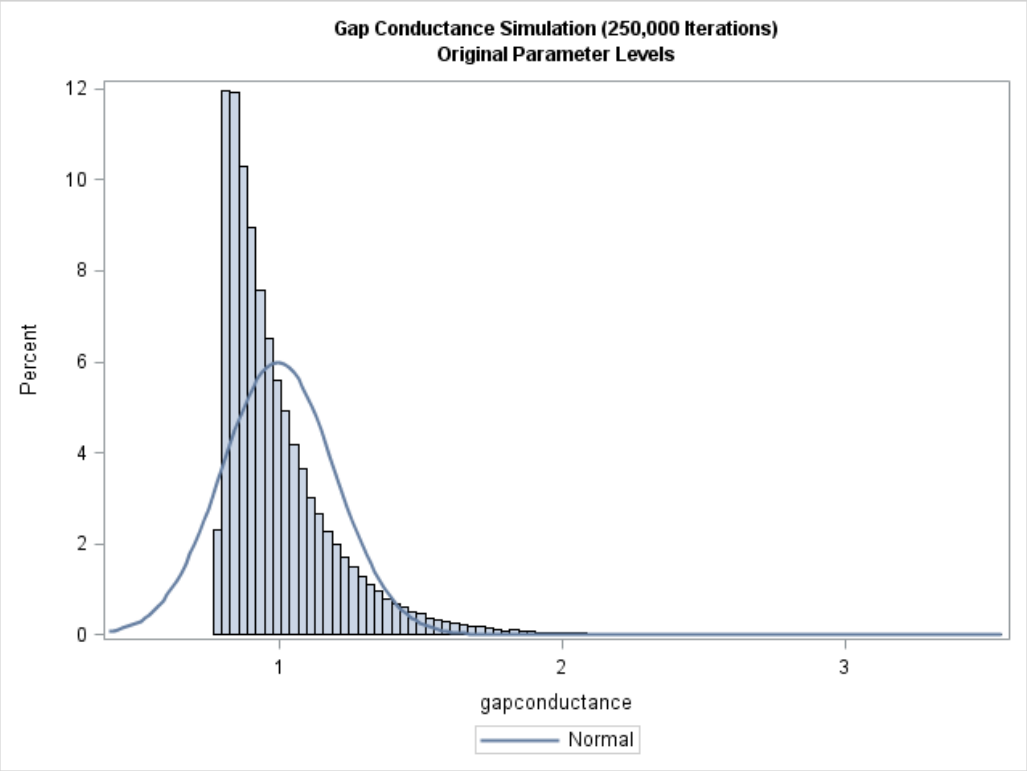
Following is a summary of the NLP results. The bottom row contains the recommended optimal solution.

NLP Variance Improvement								
Parameter Levels	k_g	t_g	u	P	T	M	h_r	Response (h_g) Variance
Original	0.0049	0.025	0.000039	500	1600	4.0000	0.8000	0.0400
NLP	0.0054	0.024	0.000030	531	1510	4.0001	0.7671	0.0279

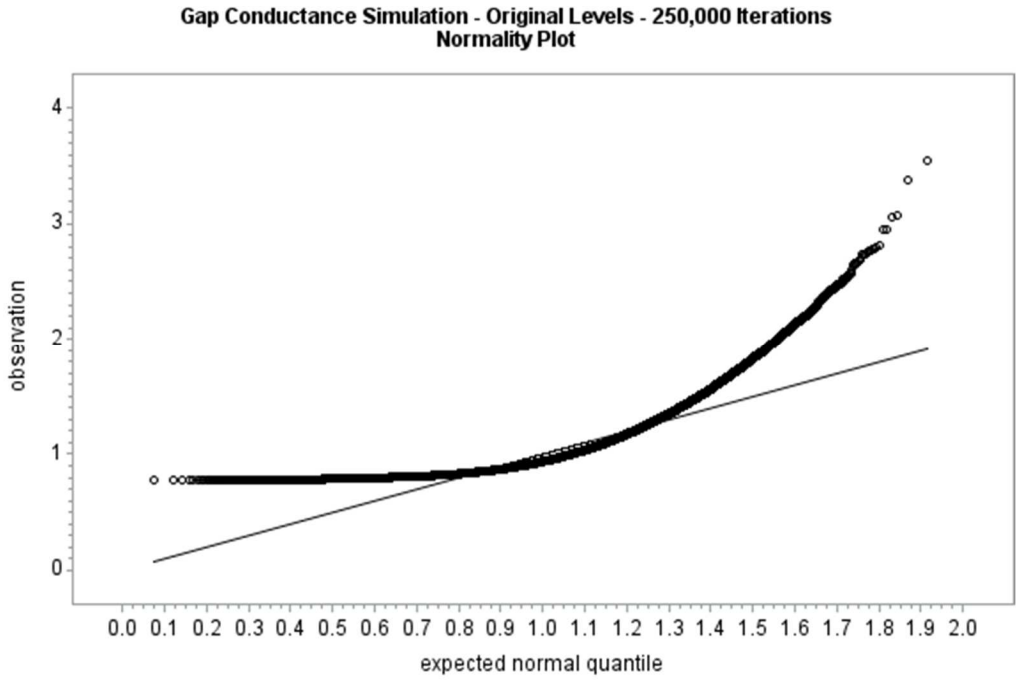
Operating the system at the NLP optimal parameter levels gives a theoretical 31.25% reduction in gap conductance variance. Understanding that some parameter distributions are based on variation in physical material properties and not subject to selection, it may not be feasible to operate at the NLP optimal settings. However, the results do indicate that there is opportunity for significant gap conductance stabilization with no change to response operating level.

Modeling the Response: Moments, Percentiles, and Monte Carlo

We now have parameter levels that yield optimally reduced response variance. But what is the effect on the expected range of gap conductance during system operation? What confidence interval statements can we make regarding the response? Following is a histogram of 250,000 Monte Carlo simulated gap conductance values using the original nominal parameter levels. Pseudo-random parameter levels were generated about their nominals using computer generated pseudo-random probability values converted to percentiles from respective probability distributions (as explained appendix 1). SAS program instructions are listed under *Monte Carlo Simulation*, in appendix 2.



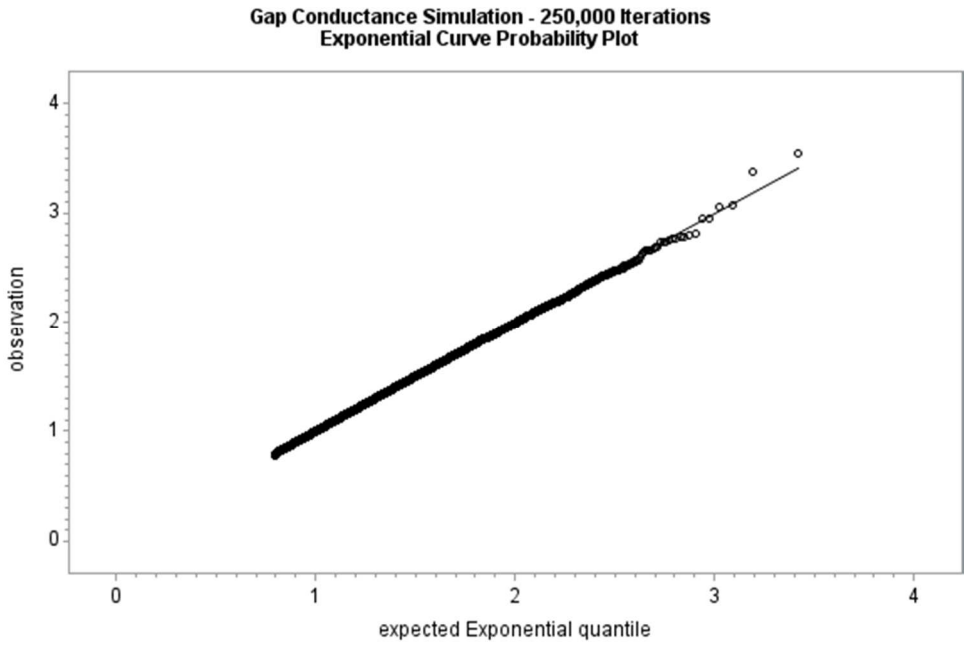
We often make an assumption of normality, but in this case the superimposed normal curve is clearly not a very good model, grossly over-predicting for small values and under-predicting for large ones. A normal probability plot shows how far these data are from a normal distribution:



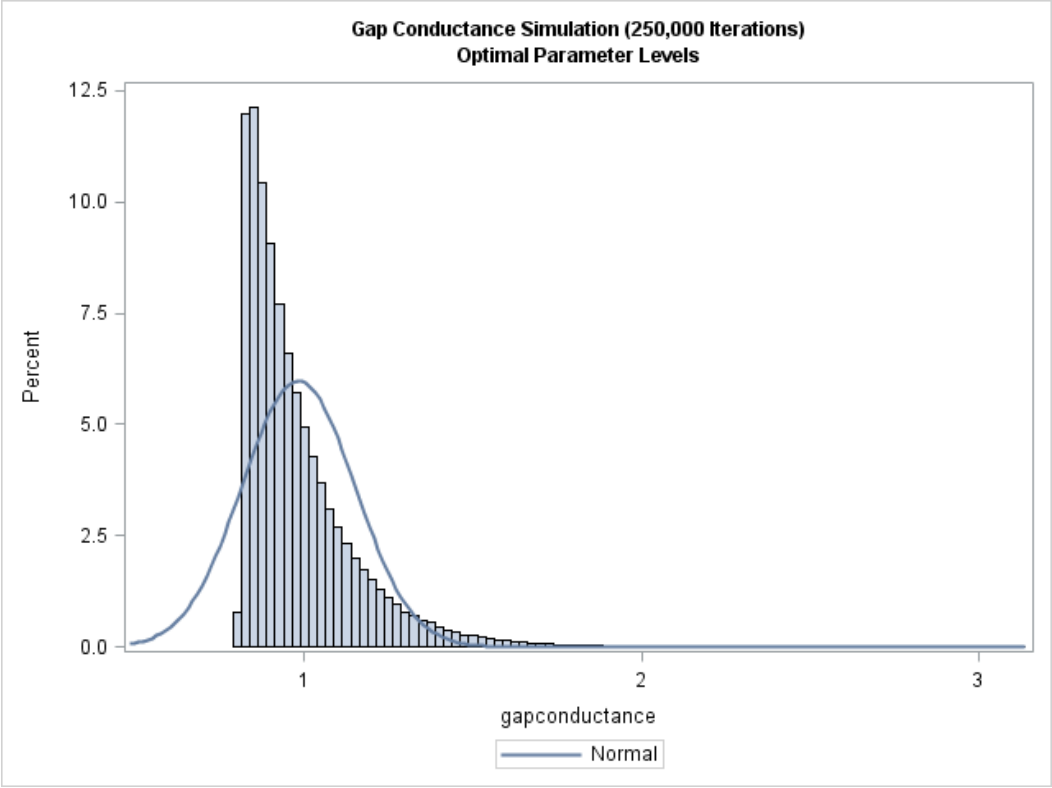
Formal normality tests confirm lack of fit. The following table is output from SAS Proc Means with the NormalTest option. Shapiro-Wilk is a preferred test (Shapiro and Gross chapt. 6), but is valid only for sample sizes<2000.

SAS Tests for Normality				
Test	**Statistic**		**p Value**	
Kolmogorov-Smirnov	D	0.057208	Pr > D	<0.0100
Cramer-von Mises	W-Sq	381.2447	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	2779.811	Pr > A-Sq	<0.0050

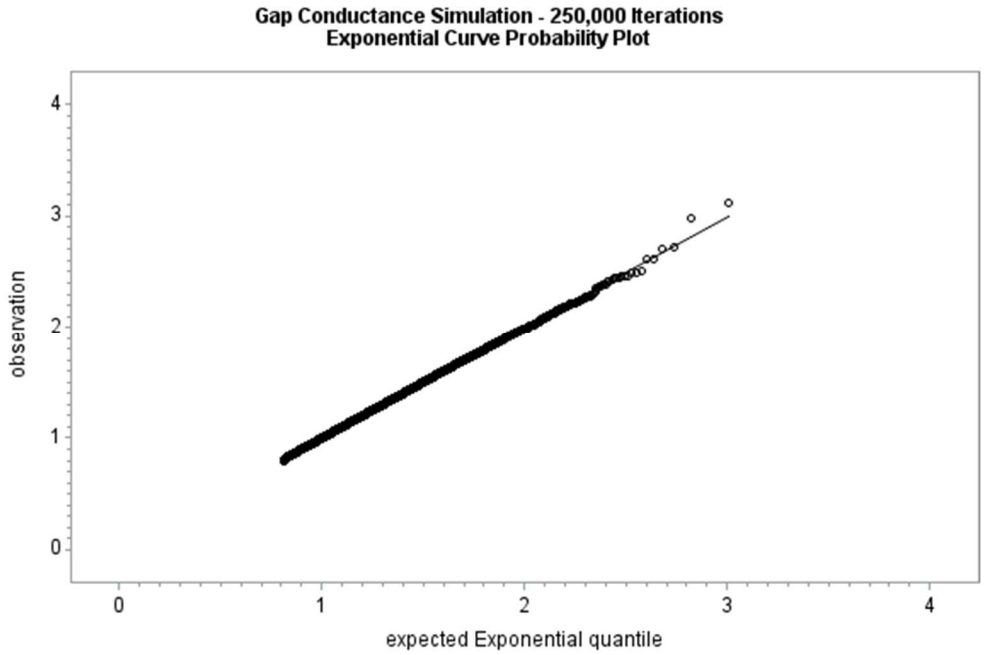
Several non-normal models, such as the Johnson S_L (Shapiro and Gross chapt. 7) and a two parameter lognormal were fit with poor results. A review of the simulation histogram, however, revealed a clear exponential shape with a lower bound in the vicinity of 0.7. Fitting a two parameter exponential distribution ($f(x) = \lambda e^{-\lambda(x-\varepsilon)}$, where λ determines shape and ε position) by matching its first two moments to those of the simulated data (method explained under *Exponential Model, Moment Matching*, in appendix 1) gave estimates for λ and ε of 5.01 and 0.79, and produced the following probability plot:



A vast improvement over the normal model. Next, a 250,000 iteration simulation was executed using the NLP optimal parameter levels, with slight modification. Since levels may not be possible outside of the given ranges, those parameters with NLP optimal levels at a boundary were simulated within the half of their range that the specified boundary resided in. So, for instance, instead of simulating u about 0.000030 (its lower bound), it was varied within [0.000030, 0.000039], the upper half of its range. The following distribution resulted:



An exponential model was fit, by matching moments, giving λ = 5.99, ε = 0.81, and the following probability plot:



Again, very good fit. Deriving confidence limits from an exponential distribution is straightforward (percentile explanations in appendix 1). Comparing 99% confidence intervals for simulated gap conductance (as a measure of expected total variation) using the original nominal settings and those of the NLP optimal levels, we have:

Original vs. NLP confidence Interval Comparison				
Source	λ	ε	99% Lower Bound	99% Upper Bound
Original	5.01	0.79	0.79	1.85
NLP Optimal	5.99	0.81	0.82	1.70

The reduction in range is confirmed, with a confidence interval ratio of $\frac{1.70-0.82}{1.85-0.79} = 0.83$

5. Conclusion

Although in a textbook problem setting, this exercise has demonstrated the power of combining the numerical and analytical modeling techniques of Linear Approximation of Variance, Nonlinear Programming, and Non-normal Statistical Modeling. A complex nonlinear model in several variables was approximated by a simpler one (using LAV) whose variance was modeled for optimization (using NLP). Normal and non-normal probability distributions of underlying system parameters were used to generate percentiles for specifying operating boundaries. These distributions, shifted by the NLP optimal solution, were then used to simulate the expected response distribution if operated under NLP specified parameter levels. Finally, analysis of the simulated data indicated that the response is non-normally distributed and after attempting the fit of several non-normal models, a two parameter exponential distribution was fit with good results. From this distribution, confidence limits on gap conductance, at NLP optimal parameter levels, were specified.

6. Opportunities for further research

Additional areas of research related to this problem are:

- Identify significant contributors of variance in the LAV model. These are candidates for further response variance reduction. A quick review of the variance table reveals σ_P^2 and σ_T^2 as possible candidates, having relatively large variances. However, it must be remembered that contribution to the LAV model is a product of a parameter's variance and the square of its partial derivative, which may be negligible.
- Perform Tolerance Design, where LAV-NLP is executed as above, with the additional feature of permitting parameter variance to change. However, this implies the ability to change (presumably decrease) parameter variances in the actual system.
- Study the effect of parameter level changes on the distribution of the parameter. In this exercise, we assumed no change in distribution. Additional parameter knowledge or ability to design experiments to measure parameter changes is required to accomplish this.

References

Shapiro, S., and Gross, A., *Statistical Modeling Techniques,* Dekker, 1981.

Fathi, Y., “A linear approximation model for the parameter design problem”, *European Journal of Operational Research* 97 (1997) 561-570

Taguchi, G., *Introduction to Quality Engineering,* Asian Productivity Organization, 1982.

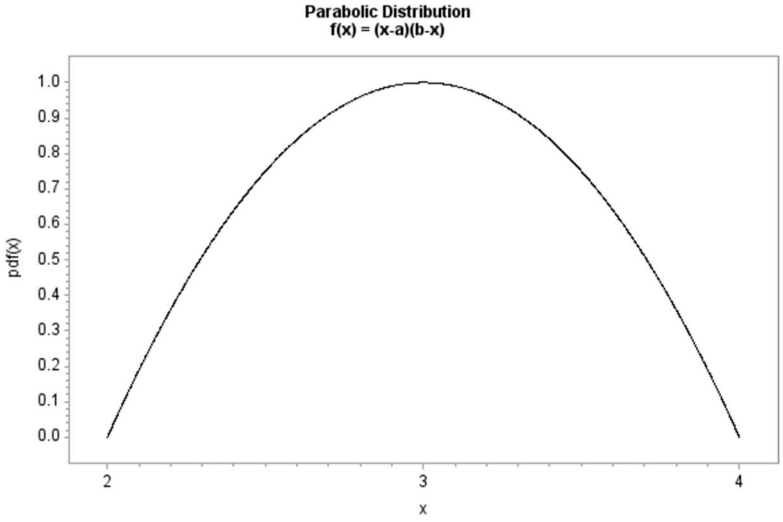
SAS NLP. http://support.sas.com/documentation/cdl/en/ormpug/59679/HTML/default/viewer.htm#nlp.htm

Appendix 1

Probability Distributions and Properties

Parabolic Distribution

The parabolic distribution has application where minimum and maximum possible values are known, but specific distributional shape is unknown, and weight is given to central observations diminishing toward both endpoints. The pdf is defined as $f(x) = K(x-a)(b-x)$, where a and b are the minimum and maximum values and K is the reciprocal of the area under the curve formed by the pdf. Note that an equivalent expression is $f(x) = K[-x^2 + (a+b)x - ab]$, an inverted parabola. Following is an example with a=2 and b=4.



Area bounded by a pdf and the x-axis must be 1. The parabolic pdf intercepts the x axis at

$$x = \frac{-(a+b) \pm \sqrt{(a+b)^2 - 4ab}}{-2} = \frac{a+b \pm \sqrt{a^2 - 4ab + b^2}}{2} = \frac{a+b \pm \sqrt{(a-b)^2}}{2} = a, b$$

and, being symmetrical has center point, $x = (a+b)/2$. Area under the curve between a and b is

$$\int_a^b -x^2 + (a+b)x - ab\, dx \;=\; -\frac{x^3}{3} + (a+b)\frac{x^2}{2} - abx \Big|_a^b$$

$$= \frac{a^2 - b^2}{3} + \frac{a+b}{2}(b^2 - a^2) - ab^2 + a^2 b \;=\; \frac{b^3 - a^3 + 3b^2 a - 3a^2 b}{6} \;=\; \frac{(b-a)^3}{6}$$

So, the complete definition of the pdf is $f(x) = \frac{6}{(b-a)^3}(x-a)(b-x)$.

Moments (intermediate algebra omitted for brevity)

$$E(x) = \int_a^b x[-x^2 + (a+b)x - ab]\,dx \;=\; \frac{6}{(b-a)^3}\left[-\frac{x^4}{4} + (a+b)\frac{x^3}{3} - \frac{abx^2}{2}\Big|_a^b\right]$$

$$= \frac{6}{(b-a)^3}\left[b^2\left(\frac{-3b^2+4b^2+4ab-6ab}{12}\right) - a^2\left(\frac{-3a^2+4a^2+4ab-6ab}{12}\right)\right]$$

$$= \frac{6}{(b-a)^3}\left[\frac{(b-a)^2(b^2-a^2)}{12}\right] = \frac{b^2-a^2}{2(b-a)} = \frac{a+b}{2}$$

In like fashion,

$$E(x^2) = \int_a^b x^2[-x^2 + (a+b)x - ab]\,dx \;=\; \frac{3b^5 - 5ab^4 + 5a^4b - 3a^5}{10(b-a)^3}$$

giving

$$Var(x) = E(x^2) - [E(x)]^2 = \frac{2(3b^5 - 5ab^4 + 5a^4b - 3a^5)}{10(b-a)^3} - \frac{5[(a+b)(a-b)]^2(b-a)}{20(b-a)^3}$$

$$= \frac{b^5 - 5ab^4 + 10a^2b^3 - 10a^3b^2 + 5a^4b - a^5}{20(b-a)^3} = \frac{(b-a)^5}{20(b-a)^3} = \frac{(b-a)^2}{20}$$

Note the binomial form in the numerator on the left side of the previous equation:

$$\sum_{k=0}^{n} \binom{n}{k} b^k(-a)^{n-k} = (b-a)^n$$

This property reduces a complicated, distributed algebraic structure into an amazingly compact and useful result. Simply knowing the endpoints of the distribution gives us both mean and variance.

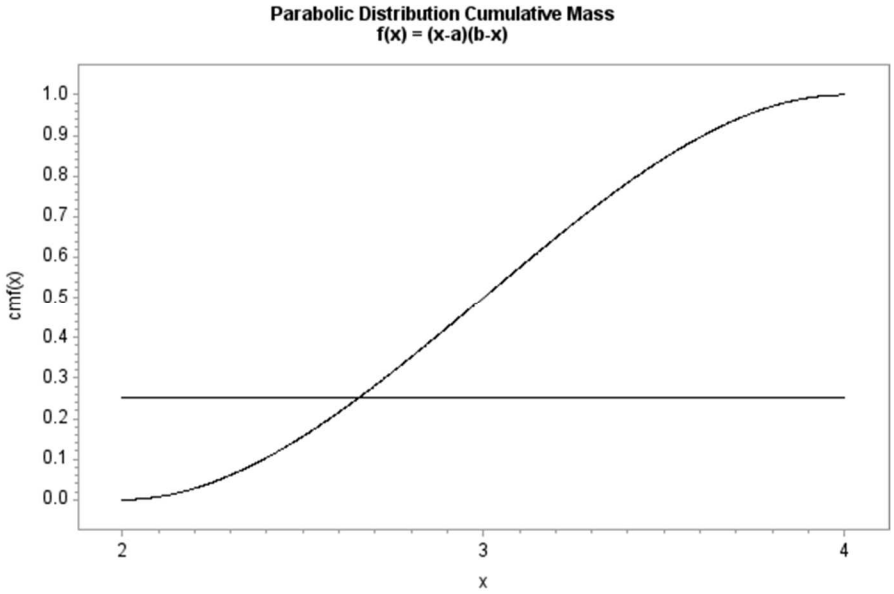
Percentiles

To facilitate simulation of a parabolic distributed variable, percentiles must be drawn from its cumulative density function (cdf) for probabilities on the interval (0,1). Given probability *p*, the percentile *x* is the value on the interval (a,b) such that cdf(*x*)=*p*. We are effectively interested in the inverse to the cdf, that is the function cdf^{-1} such that, given *p*, $cdf^{-1}(p)$=x. For this distribution, the inverse cdf is

$$cmf^{-1}(p) = \frac{6}{(b-a)^3}\int_a^X -x^2 + (a+b)x - ab\,dx \Longrightarrow X^3 - \frac{3(a+b)}{2}X^2 + 3abX + \frac{a^3 - 6a^2b - 6}{2} = \frac{p(b-a)^3}{6}$$

which leads to solving the roots of a cubic equation, with somewhat complicated selection rules (there are three possible roots) and the possibility of complex numbers. An alternative is to use Newton's method (subtract $\frac{p(b-a)^3}{6}$ on both sides of the right hand equation and solve for X). Considering that, as with all cdfs, the parabolic is an increasing function through its domain with derivative near 1 for most probable

values, it is an efficient technique. The following diagram demonstrates locating $cdf^{-1}(0.25)$. The known cumulative density is subtracted from the cdf resulting in a cubic root problem.



Following is a program outline implementing Newton's method to solve the right hand equation above.

$given\ endpoints\ a\ and\ b$

$and\ f(x) = x^3 - \frac{3(a+b)}{2}x^2 + 3abx + \frac{a^3 - 6a^2b - 6}{2} - \frac{p(b-a)^3}{6},$

$locate\ x_0\ such\ that\ f(x_0) = 0$

$let\ x = \frac{a+b}{2} \quad (begin\ at\ centerpoint)$

$while\ f(x) \neq 0$

$\quad if\ f(x) > 0$

$\quad\quad x = x - \frac{f(x)}{f'(x)}$

$\quad else$

$\quad\quad x = x + \frac{f(x)}{f'(x)}$

$note\ that\ f'(x) = 3x^2 + 3(a+b)x + 3ab$

Uniform Distribution

As its name implies, the uniform distribution applies where probability is constant, or uniform, throughout the domain of a random variable. Its pdf is defined simply as $f(x) = p$ and, given that its total cumulative density must be unity, we have

$$\int_a^b p\,dx = 1 \Rightarrow p = \frac{1}{b-a}$$

Moments

$$E(x) = \int_a^b xp\,dx = \frac{1}{2}\left(\frac{b^2-a^2}{b-a}\right) = \frac{b+a}{2}$$

$$E(x^2) = \int_a^b x^2 p\,dx = \frac{1}{3}\left(\frac{b^3-a^3}{b-a}\right) = \frac{(b-a)^3 + 3b^2a - 3a^2b}{3(b-a)} = \frac{b^2+ab+a^2}{3}$$

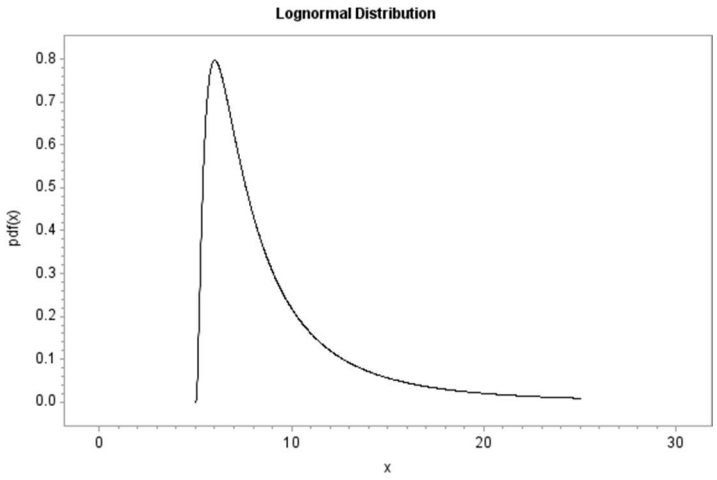
$$Var(x) = E(x^2) - [E(x)]^2 = \frac{(4-3)b^2 + (4-3)a^2 + (4-6)ab}{12} = \frac{(b-a)^2}{12}$$

Percentiles

$$p_0 = \int_a^X \frac{1}{b-a}dx = \frac{X-a}{b-a} \Rightarrow X = (b-a)p_0 + a$$

Lognormal Distribution

A log-normally distributed variable, y, is one such that $\log(y) \sim N(\mu, \sigma^2)$. In general, μ and σ are unknown and must be derived from the untransformed (original, non-log) data. A log-normally distributed variable has domain (0,∞) corresponding to (-∞,∞) for its transformed, normal variable. This distribution is generally right skewed (logarithms are compress for lower values, extended for larger ones) as seen below. It is appropriate when density accumulates below the mean and when extreme values are observed beyond the mean.



The lognormal pdf derives from the fact that the transformed logarithmic values are normally distributed. The addition of a location parameter shifts the distribution up or down (left or right) in value. +5 is used in the previous graph. Letting μ and σ^2 be the mean and variance of the normal distribution (after log) and ε be the additional location parameter, the pdf of the untransformed distribution is

$$\frac{1}{\sigma\sqrt{2\pi}}\left(\frac{1}{x-\varepsilon}e^{-\frac{(\ln(x-\varepsilon)-\mu)^2}{2\sigma^2}}\right)$$

since the pdf of a transformed variable (here x to $\ln(x-\varepsilon)$) is the untransformed pdf * the derivative of the transform. Here $\frac{d\ln(x-\varepsilon)}{dx} = \frac{1}{x-\varepsilon}$. But for cumulative density and percentile operations it is easier to use the normal distribution of the log values and convert to the original scale using the exponential function than it is to used the pdf above.

Moments

Let y be log-normally distributed with mean μ_L and variance σ_L^2 and $x \sim \mathrm{N}(\mu, \sigma^2)$ such that $y = e^x$. Then

$$E(e^{kx}) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{kx}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx$$

$$= \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{x^2-2\mu x+\mu^2-2k\sigma^2 x}{2\sigma^2}}\,dx$$

$$= \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{\left[x-\mu-k\sigma^2\right]^2-2k\mu\sigma^2-k^2\sigma^4}{2\sigma^2}}\,dx$$

$$= e^{\frac{2k\mu\sigma^2+k^2\sigma^4}{2\sigma^2}}\left(\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-\frac{\left[x-\mu-k\sigma^2\right]^2}{2\sigma^2}}\,dx\right) = e^{\frac{2k\mu\sigma^2+k^2\sigma^4}{2\sigma^2}} = e^{k\mu+\frac{k^2\sigma^2}{2}}$$

From this we have $E(y) = e^{\mu+\frac{\sigma^2}{2}}$ and $E(y^2) = e^{2\mu+2\sigma^2}$, giving

$$Var(y) = E(y^2) - [E(y)]^2 = e^{2\mu+2\sigma^2} - e^{2\mu+\sigma^2} = e^{2\mu+\sigma^2}\left(e^{\sigma^2}-1\right) = [E(y)]^2\left(e^{\sigma^2}-1\right)$$

Note that, assuming $E(y)$and $Var(y)$ are known (they derive from the untransformed distribution), we have a pair of equations from which to express μ and σ^2, the mean and variance of the transformed (log) distribution. Letting $\mu_L = E(y)$ and $\sigma_L^2 = Var(y)$, we have

$$\sigma_L^2 = \mu_L^2\left(e^{\sigma^2}-1\right) \Rightarrow \sigma^2 = \ln\left(\frac{\sigma_L^2}{\mu_L^2}+1\right)$$

and

$$\mu = \ln(\mu_L) - \frac{\sigma^2}{2} = \frac{\ln(\mu_L^2) - \ln\left(\frac{\sigma_L^2}{\mu_L^2}+1\right)}{2} = \frac{1}{2}\ln\left(\frac{\mu_L^2}{\frac{\mu_L^2}{\sigma_L^2+\mu_L^2}}\right) = \frac{1}{2}\ln(\sigma_L^2+\mu_L^2)$$

Percentiles

Given $x \sim \mathrm{N}(\mu, \sigma^2)$, μ and σ^2 just derived, and a given cdf level, p, we have

$$p = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{X} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \Phi\left(\frac{X-\mu}{\sigma}\right) \Longrightarrow X = \mu + \sigma\Phi^{-1}(p)$$

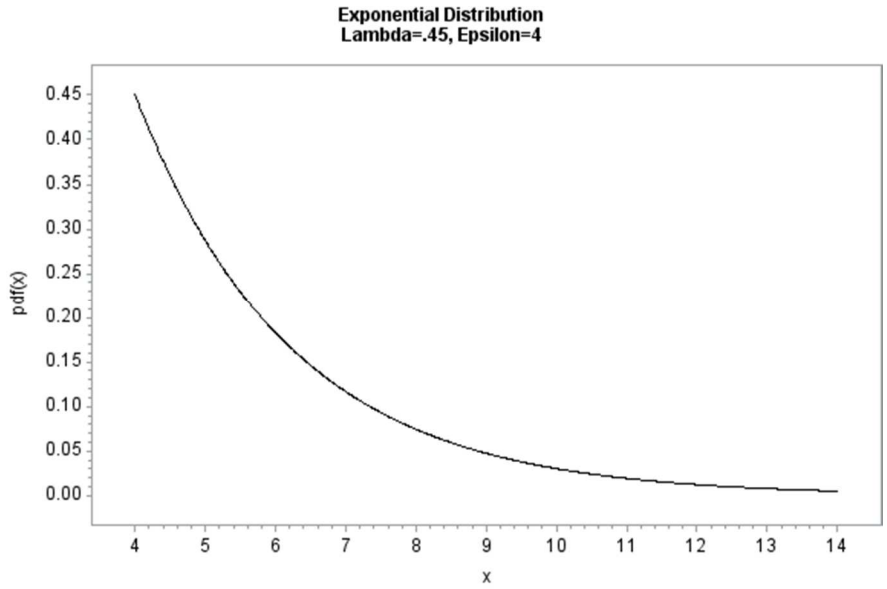
and the corresponding percentile of the untransformed distribution is simply $e^X = e^{\mu+\sigma\Phi^{-1}(p)}$

Exponential Distribution

The exponential distribution is an extension of the geometric, from the domain of discrete random variables to continuous ones. In its simplest form, it has one parameter, λ, and has pdf

$$f(x) = \lambda e^{-\lambda x}$$

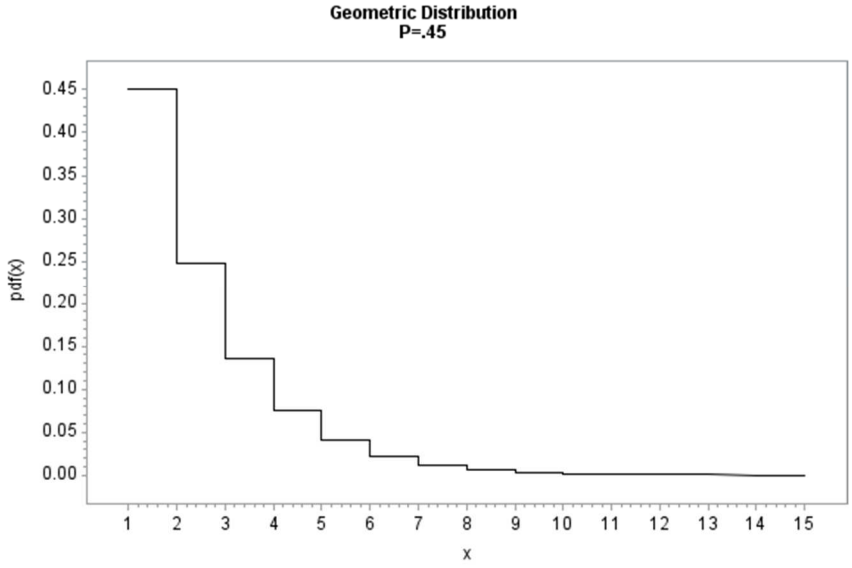
It is an appropriate distribution when smallest values have greatest probability density and with density decreasing rapidly as x increases. A location parameter, ε, can be included to give the distribution an origin at some known lower boundary. In this case the pdf becomes $f(x) = \lambda e^{-\lambda(x-\varepsilon)}$ (note that $\frac{dx-\varepsilon}{dx} = 1$ so that the new pdf becomes $\lambda e^{-\lambda(x-\varepsilon)} \frac{dx-\varepsilon}{dx} = \lambda e^{-\lambda(x-\varepsilon)}(1)$ and there is no change in the scale of the shifted pdf). An example distribution follows with λ = 0.45 and ε = 4.



Note that the exponential pdf always has maximum value (λ) at its lower boundary, since

$$x = \varepsilon \implies e^{-\lambda(x-\varepsilon)} = 1 \Longrightarrow f(x) = \lambda(1) = \lambda$$

Note, also, the similarity in form to the discrete geometric counterpart (the geometric pdf for probability p is $f(x) = p(1-p)^{x-1}$ and is defined for integers x >0):



Moments

The exponential moment generating function (mgf) is derived as follows:

$$E\left(e^{\theta x}\right) = \int_0^\infty e^{\theta x}\lambda e^{-\lambda x}\,dx = \int_0^\infty \lambda e^{-(\lambda-\theta)x}\,dx$$

$$= -\frac{\lambda}{\lambda-\theta}e^{-(\lambda-\theta)x}\Big|_{x=0}^{x=\infty} = \frac{\lambda}{\lambda-\theta}$$

which gives

$$E(x) = \frac{d}{d\theta}\left(\frac{\lambda}{\lambda-\theta}\right)\Big|_{\theta=0} = \frac{\lambda}{(\lambda-\theta)^2}\Big|_{\theta=0} = \frac{1}{\lambda}$$

$$E(x^2) = \frac{d^2}{d\theta^2}\left(\frac{\lambda}{\lambda-\theta}\right)\Big|_{\theta=0} = \frac{2\lambda}{(\lambda-\theta)^3}\Big|_{\theta=0} = \frac{2}{\lambda^2}$$

$$Var(x) = E(x^2) - [E(x)]^2 = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}$$

Percentiles

$$p_0 = \int_0^X \lambda e^{-\lambda x}\,dx = -e^{-\lambda x}\Big|_0^X = 1 - e^{-\lambda X} \Longrightarrow X = -\frac{\ln(1-p_0)}{\lambda}$$

When a minimum value is known, a location parameter,ε, can be specified. This effectively shifts the distribution and models an exponential distribution beginning at a given point or having a lower bound other than 0 (absence of a location parameter implies a lower bound of 0). Let $y-\varepsilon=x$, where x is exponentially distributed with parameter λ. Then

$$E(y-\varepsilon)=E(y)-\varepsilon=E(x)=\frac{1}{\lambda}\Longrightarrow\lambda=\frac{1}{E(y)-\varepsilon}$$

Percentiles are now found in the y domain by

$$p_0=\int_0^X\lambda e^{-\lambda y}\,dy=\int_\varepsilon^Y\lambda e^{-\lambda(y-\varepsilon)}\,dy=-e^{-\lambda(y-\varepsilon)}\Big|_\varepsilon^Y=1-e^{-\lambda(Y-\varepsilon)}\Longrightarrow Y=-\frac{\ln(1-p_0)}{\lambda}+\varepsilon$$

Typically λ and ε are unknown, but can be can be estimated from empirical data by methods such as moment or percentile matching.

Moment Matching

By matching moments it is meant that a theoretical probability distribution (exponential in this case) is being fit to an observed data set such that a given number of theoretical moments (typically the first k) are identical to the same k observed moments. This is intended to derive parameters (λ and ε for the exponential) that give the theoretical distribution with closest "fit" to the distributional "shape" of the observed data. Generally, the number of moments matched equals the number of parameters in the theoretical distribution so that, when solving for λ and ε of an exponential distribution, we solve for two unknowns in a system of two equations. Also, it is the standardized moments (mean, variance, skewness, etc.) that are matched. The first two observed standardized moments are defined as

$m_1=\frac{\sum_{i=1}^n y_i}{n}$ and $m_2=\frac{\sum_{i=1}^n(y_i-m_1)^2}{n}$, the mean and variance, and matching these to $E(y-\varepsilon)$ and $Var(y-\varepsilon)$, we have

$$E(y-\varepsilon)=E(y)-\varepsilon=\frac{1}{\lambda}\ \Longrightarrow\ \lambda=\frac{1}{m_1-\varepsilon}\ \Longrightarrow$$

$$Var(y-\varepsilon)=Var(y)=\frac{1}{\lambda^2}=m_2=m_1^2-2\varepsilon m_1+\varepsilon^2$$

which is quadratic in ε, giving

$$\varepsilon=\frac{2m_1\pm\sqrt{4m_1^2-4(m_1^2-m_2)}}{2}=m_1\pm m_2$$

and

$$\lambda=\frac{1}{m_1-\varepsilon}$$

where ε is chosen such that it is a lower bound to the observed distribution (typically m_1-m_2).

Percentile Matching

Theoretical distribution parameters can also be derived by matching carefully selected theoretical percentiles to those observed in a data set. Two possible reasons for fitting percentiles instead of moments are:

- Moments are fixed. That is, for a given set of parameters a theoretical distribution has strictly one first moment, one second moment, and so forth. By choosing percentiles, one can "force" closer fit in a particular region, although this may be viewed as subjective and a form of data massaging.
- Percentile calculations are sometimes simpler than those for moments. The key here is to identify which of the two parameter functions is simpler, since they are what shape the equations to be solved. Although the exponential moment and percentile equations share equal complexity (both are simple), for some distributions, such as the Johnson S_L, S_B, and S_U, percentile calculations are much simpler than those for moments (for an explanation of these, see Shapiro and Gross, chapt. 7).

To match percentiles, two cumulative density values, p_1 and p_2, are chosen, the observations are ranked, the p_1 and p_2 observed percentiles are taken (let these be y_1 and y_2), and theoretical parameters are solved with:

$$\left.\begin{aligned} y_1 &= \frac{\ln(1-p_1)}{\lambda} + \varepsilon \\ y_2 &= \frac{\ln(1-p_2)}{\lambda} + \varepsilon \end{aligned}\right] \Rightarrow \lambda = \frac{\ln\left(\frac{1-p_2}{1-p_1}\right)}{y_2 - y_1}$$

and

$$\varepsilon = \frac{y_2 + y_2 - \ln\left[\frac{(1-p_1)(1-p_2)}{\lambda}\right]}{2}$$

Appendix 2

Program Listings

1. NLP Initial values

```
%macro configure_x_grid(xdataset, k0, k1, kdiv, tg0, tg1, tgdiv, U0, U1, Udiv,
       P0, P1, Pdiv, T0, T1, Tdiv, M0, M1, Mdiv, hr0, hr1, hrdiv);
  /* set up x initial value grid (one row for each (k, tg, U, P, T, M, hr) point) */
  /* input parameters are in triples for each x: low_value, high_value, and number_of_divs */
  /* note that ndiv+1 levels are created for each variable */
  /* save in xdataset */
  data &xdataset(keep=n k tg U P T M hr);
    /* populate grid */
    hk=(&k1-&k0)/&kdiv;
    htg=(&tg1-&tg0)/&tgdiv;
    hU=(&U1-&U0)/&Udiv;
    hP=(&P1-&P0)/&Pdiv;
    hT=(&T1-&T0)/&Tdiv;
    hM=(&M1-&M0)/&Mdiv;
    hhr=(&hr1-&hr0)/&hrdiv;
        n=0;
    do k=&k0 to &k1 by hk;
      do tg=&tg0 to &tg1 by htg;
        do U=&U0 to &U1 by hU;
          do P=&P0 to &P1 by hP;
            do T=&T0 to &T1 by hT;
              do M=&M0 to &M1 by hM;
                do hr=&hr0 to &hr1 by hhr;
                  n=n+1;
                  output;
                       end;
                end;
              end;
          end;
        end;
      end;
    end;
  run;
%mend;
```

2. NLP procedure

```
%macro get_optimal_nlp_solution(xinit, opt);
  /* use nlp procedure to locate (k, tg, U, P, T, M, hr) with optimal (minimum) hg variance */
  /* initial value for (k, tg, U, P, T, M, hr) is in data set xinit */
  /* solution results are saved in data set opt */
  /* optimal solution (x point) is in opt data set in row with _type_='parms' */
  proc nlp inest=&xinit outest=&opt noprint;
    /* objective */
    min hgvar;
    /* specify domain variables, initialize with values from xinit table */
    parms k, tg, U, P, T, M, hr;
    /* subject to */
    bounds .0043<=k<=.0054, .024<=tg<.026, .00003<=U<=.000048, 471<=P<=531, 1510<=T<=1690,
           3.9999<=M<=4.0001, .61<=hr<=1.2;
    /* hg spec is .996, but give some tolerance */
    nlincon .994<hg<.998;
    /* objective function = var estimate of hg = sum of [(square of partial_deriv) * variance] */
    /* note that all variances, except for P and hr, are static */
    /* P, being lognormal, and hr, being exponential, have variances dependent on their */
    /* current parameter level, so recalculate P and hr variances */
    /* order of terms is k, tg, U, P, T, M, hr */
    hgvar = 1/(tg+5.448*U/P*sqrt(T/M)) * (.0054-.0043)**2/20
            + (k/(tg+5.448*U/P*sqrt(T/M))**2)**2 * .00033**2
            + (k*5.448*1/P*sqrt(T/M)/(tg+5.448*U/P*sqrt(T/M))**2)**2 * (.000048-.000030)**2/12
            + (k*5.448*U*sqrt(T/M)/P/P/(tg+5.448*U/P*sqrt(T/M))**2)**2 * 100
            + (k*5.448*U/P/sqrt(M)/2/sqrt(T)/(tg+5.448*U/P*sqrt(T/M))**2)**2 * 900
```

```
            + (k*5.448*U*sqrt(T)/P/sqrt(M)**3/(tg+5.448*U/P*sqrt(T/M))**2)**2
              * (4.0001-3.9999)**2/12
            + (hr-.6)**2;
    hg=k/(tg+5.448*U/P*sqrt(T/M))+hr;
  run;
%mend;
```

3. Calculate response for given parameter levels

```
%macro calculate_hg(xds, k, tg, U, P, T, M, hr, hg, hgvar);
  /* calculate gap conductance (hg) and variance for each supplied set of parameter levels
     xds is the data set of parameter level combinations at which to calculate hg estimates
     make sure it has col n (observation number) and numeric values for each parameter
     hg and hgvar columns are updated with calculated mean and variance */

  data &xds(drop=vk vtg vU vP vT vM vhr pdk pdtg pdU pdP pdT pdM pdhr pdcomm);
    set &xds;
    /* parameter variance estimates */
    vk=(.0054-.0043)**2/20;     /* parabolic dist */
    vtg=.00033**2;              /* normal dist */
    vU=(.000048-.000030)**2/12; /* uniform dist */
    vP=100;                     /* lognormal, vP = variance of untransformed values */
    vT=900;                     /* normal dist */
    vM=(4.0001-3.9999)**2/12;   /* uniform dist */
    vhr=(hr-.6)**2;             /* exponential: 1/lambda = hr-lower bound, var=1/lambda**2 */
    /* estimate hg from physical model */
    &hg=&k/(&tg+5.448*&U/&P*sqrt(&T/&M))+&hr;
    /* estimate var(hg) as variance of linear Taylor polynomial approximation of
       physical model at given parameter levels, which is the sum of the products
       of the squares of individual partial derivatives evaluated at the given levels
       times corresponding variances */
    /* common factor in most partial derivatives */
    pdcomm=1/(&tg+5.448*&U/&P*sqrt(&T/&M))**2;
    /* partial derivatives */
    pdk=1/sqrt(pdcomm);
    pdtg=-k/pdcomm;
    pdU=-k*5.448*1/&P*sqrt(&T/&M)*pdcomm;
    pdP=k*5.448*&U*sqrt(&T/&M)/&P/&P*pdcomm;
    pdT=-k*5.448*&U/&P/sqrt(&M)/2/sqrt(&T)*pdcomm;
    pdM=k*5.448*&U*sqrt(&T)/&P/sqrt(&M)**3*pdcomm;
    pdhr=1;
    /* estimate hg variance at supplied parameter levels */
    &hgvar = pdk*pdk*vk + pdtg*pdtg*vtg + pdU*pdU*vU + pdP*pdP*vP + pdT*pdT*vT +
             pdM*pdM*vM + pdhr*pdhr*vhr;
  run;
%mend;
```

4. NLP main procedure (launches NLP steps)

```
%macro opt;

  /* configure grid of initial x values */
  /* xinit is a data set created with all combinations of independent variable levels */
  /* triplets following xinit are lower bound, upper bound, number of divisions */
  /* note that this generates ndiv+1 levels for each variable */
  %configure_x_grid(xinit, .0043, .0054, 3,  .024, .026, 3,  .00003, .000048, 3, 471, 531, 3,
                           1510, 1690, 3,    3.9999, 4.0001, 3,  .6, 1.2, 3 );
  %global i n;

  /* get count of initial value points - an optimal solution for each will be attempted */
  /* create optimal solution table */
  proc sql noprint;
    select max(n) into :n from xinit;
    create table opt(n num, k num, tg num, U num, P num, T num, M num, hr num,
                     hg num, hgvar num, hgvarcalc num);
  quit;

  /* get nlp solution for each initial value point in grid */
```

```
  %do i=1 %to &n;
    /* proc nlp accepts one initial set of values at a time */
    /* so copy the current set into a working ds and pass it to nlp */
    proc sql;
      create table nlpinit as
      select 'parms' as _type_, k, tg, U, P, T, M, hr from xinit where n=&i;
    quit;
    /* calculate optimal solution given current initial point */
    /* results are in data set nlpopt */
    %get_optimal_nlp_solution(nlpinit, nlpopt);
    /* save optimal solution and calculated y-variance */
    proc sql;
      insert into opt(n, k, tg, U, P, T, M, hr, hgvar)
      select &i, k, tg, U, P, T, M, hr, _rhs_
      from   nlpopt
      where  lowcase(_type_)='parms';
    quit;
  %end;

  /* calculate hg at each optimal x solution (should be within constraints if specified) */
  /* also calculate variance of hg to compare to nlp optimal value (should be identical) */
  %calculate_hg(opt, k, tg, U, P, T, M, hr, hg, hgvarcalc);

  /* please drive up to the window, your order is ready! */

%mend;
```

5. Monte Carlo simulation

```
/* simulate gap conductance with random independent parameter values */
/* generated from individual known probability distributions */
data sim (keep=khelium gapthk viscocity temp pressure mwhelium heattrcoeff gapconductance);
  call streaminit(12345);
  /* generate random independent parameter values */
  /* ------------------------------------------------------------------------------------ */
  /* gas thermal conductivity (of helium) */
  /* this is a physical 'constant' with value .0049 (*** UNITS ***) */
  /* but is known to vary between .0043 and .0054 */
  ka=.0043; kb=.0054;
  /* its pdf is unknown, but engineers have historically observed values near the midpoint */
  /* of the above range most often and expect decreasing, accelerated, and symmetrical */
  /* likeliness away from the centerpoint */
  /* lacking any specific knowledged of actual pdf shape, a parabolic pdf of the form */
  /* (x-ka)(kb-x)*6/[(ka-kb)**3] will be used
  /* the area under (x-ka)(kb-x) is [(ka-kb)**3]/6 and scales total cumulative density to 1 */
  kscale=6/(kb-ka)**3;
  /* cumulative density on (ka,kb) is strictly increasing and found up to quantile q by */
  /* evaluating the integral of the parabolic pdf (a cubic polynomial) on (ka,q) */
  /* the strictly increasing nature and near linearity in derivative of the cdf */
  /* makes evaluation by Newton's method an efficient option */
  /* by avoiding the general cubic root methods we avoid choosing from three solutions */
  /* two of which are from outside of our known range, and may involve imaginary components */
  /* a, b, c, and d are the coefficients of the cubic polynomial (cdf of quadratic pdf) */
  /* d0, here, is the static portion of the cubic poly constant */
  /* for each cdf value solved for, the cumulative density will be subtracted from d0 */
  /* to facilitate solving cdf(x) - cumulative density = 0 */
  a=-1/3*kscale; b=(ka+kb)/2*kscale; c=-ka*kb*kscale; d0=(-ka**3/6+ka*ka*kb/2)*kscale;
  /* ------------------------------------------------------------------------------------ */
  /* gap thickness is a dimensional quantity with known normal distribution, */
  /* nominal specification of .025 (inches) and tolerance of +- .001 (inches) */
  /* gives mean and standard deviation estimates of .025 and .00033, respectively */
  gapmean=.025; gaps=.00033;
  /* ------------------------------------------------------------------------------------ */
  /* viscocity is a physical property of the clad material and is believed to be */
  /* .000039 (g/cm-sec) */
  /* however, it may vary between .000030 and .000048, its distribution is unknown, */
  /* and all values in the given range are considered to be equally likely */
  visca=.00003; viscb=.000048;
  /* ------------------------------------------------------------------------------------ */
```

```
/* temperature is expected to be symmetrically distributed around 1600 (deg K) with standard */
/* deviation of 30, values are expected to cluster at 1600 and tail off rapidly */
/* a normal distribution is used to model this */
tempmean=1600; temps=30;
/* ------------------------------------------------------------------------------------------ */
/* pressure (psi) is expected to have a distribution skewed to the right with a lower bound */
/* of 450, mean of 500, and variance of 100 */
/* it is believed that pressure values are lognormally distributed */
/* the corresponding log value mean and variance are 3.8924 and .0392, respectively */
prlbound=450; prlogmean=3.8924; prlogs=sqrt(.0392);
/* ------------------------------------------------------------------------------------------ */
/* molecular weight oh helium is a physical constant with value 4 and minimal variation */
/* its distribution is unknown here, but to account for possible variation, */
/* a uniform distribution on (3.9999,4.0001) will be used */
mwhellb=3.9999; mwhelub=4.0001;
/* ------------------------------------------------------------------------------------------ */
/* radiant heat transfer coefficient */
/* is believed to have a minimum (and most likely) value of .6, an average value of .8, */
/* and rapidly descending probabilities from the minimum value in exponential form */
/* let y=x-.6, then E(y) = E(x-.6) = E(x)-.6 = .8-.6 = 1/lambda */
/* => exponential distribution with lambda=1/(.8-.6) */
heattr_lambda=5; heattr_eps=.6;
/* ------------------------------------------------------------------------------------------ */
/* let's do it! */
do i=1 to 250000;
  /* ---------------------------------------------------------------------------------------- */
  /* generate a gas thermal conductivity value */
  /* get a random value on (0,1) */
  u=rand('uniform');
  /* subtract u from cdf so that zero solution solves cdf(x)-u=0 => cdf(x)=u */
  d=d0-u;
  /* find solution to cdf(x)-u=0 */
  /* this value is the inverse of the parabolic cdf for the given cumulative density */
  /* the cubic is strictly increasing on (ha,hb) being a cdf, */
  /* so convergence is 'guaranteed' */
  /* begin search at centerpoint of interval, most solutions are expected nearby */
  /* also, cdf points in vicinity approximate linearity (little variation in derivative) */
  x=(ka+kb)/2;
  F = a*x**3 + b*x**2 + c*x + d;
  do while(abs(F)>.0001);
    if(F>0)then
      /* polynomial (cumulative density) strictly increasing on interval so subtract */
      /* quantity of x expected to reduce F by current ordinate (based on slope */
      /* estimated by first derivative of cubic polynomial) */
      x=x-F/(3*a*x*x+2*b*x+c);
    else
      /* F<0 so add to x which (F is increasing) adds to F */
      x=x-F/(3*a*x*x+2*b*x+c);
    F = a*x**3 + b*x**2 + c*x + d;
  end;
  khelium=x;
  /* ---------------------------------------------------------------------------------------- */
  /* generate a random (normally distributed) gap thickness value */
  gapthk=rand('normal',gapmean, gaps);
  /* ---------------------------------------------------------------------------------------- */
  /* generate a random (uniformly distributed) viscocity value */
  viscocity=rand('uniform')*(viscb-visca)+visca;
  /* ---------------------------------------------------------------------------------------- */
  /* generate a random (normally distributed) temperature value */
  temp=rand('normal',tempmean, temps);
  /* ---------------------------------------------------------------------------------------- */
  /* generate a random (lognormally distributed) pressure value */
  pressure=exp(rand('normal',prlogmean,prlogs))+prlbound;
  /* ---------------------------------------------------------------------------------------- */
  /* generate a random (uniformly distributed) helium molecular weight value */
  mwhelium=rand('uniform')*(mwhelub-mwhellb)+mwhellb;
  /* ---------------------------------------------------------------------------------------- */
  /* generate a random (exponentially distributed) heat transfer coefficient value */
  u=rand('uniform');
  heattrcoeff=-log(1-u)/heattr_lambda+heattr_eps;
  /* ---------------------------------------------------------------------------------------- */
```

```
    /* finally, calculate a 'random' gap conductance value */
    gapconductance=khelium/(gapthk+5.448*(viscocity/pressure*sqrt(temp/mwhelium))) + heattrcoeff;
    output;
  end;
run;
```