

Calculation Cover Sheet

Complete only applicable items.

1. QA: L
Page: 1 of 25

2. Calculation Title Calculation of Upper Subcritical Limits for Nuclear Criticality in a Repository			
3. Document Identifier (including Revision Number) B000000000-01717-0210-00027 REV 00			4. Total Pages 25
5. Total Attachments one ^{KDW} 5/30/98	6. Attachment Numbers - Number of pages in each 2 ^{KDW} 5/30/98 I: electronic attachment, II: 13		
	Print Name	Signature	Date
7. Originator	James W. Pegram, Jr	<i>James W Pegram Jr</i>	July 23, 1998
8. Checker	M.K. Au-Yang	<i>mkauryang</i>	July 23, 1998
9. Lead Design Engineer	D.A. THOMAS	<i>DA Thomas</i>	07/30/98

10. Remarks

Revision History

11. Revision No.	12. Date Approved	13. Description of Revision
00		Initial Issuance

Table of Contents

<u>Item</u>	<u>Page</u>
1. PURPOSE.....	3
2. METHOD.....	3
2.1 NON-TRENDING NORMAL DISTRIBUTION TOLERANCE LIMIT (NDTL).....	7
2.2 NON-TRENDING DISTRIBUTION-FREE TOLERANCE LIMIT (DFTL).....	8
2.3 TRENDING LOWER UNIFORM TOLERANCE BAND (LUTB).....	9
2.4 DETAILS OF LUTB METHOD.....	10
2.5 ASSUMPTIONS REQUIRED FOR STATISTICAL ANALYSES.....	11
3. ASSUMPTIONS.....	11
3.1 ASSUMPTION OF NORMALITY IN LUTB CALCULATIONS.....	11
3.2 ASSUMPTION OF ADEQUACY OF CONFIDENCE LEVELS, POPULATION COVERAGE.....	11
3.3 ASSUMPTION OF MINIMAL IMPACT OF REVISED DATA.....	11
4. USE OF COMPUTER SOFTWARE.....	12
4.1 SOFTWARE APPROVED FOR QA WORK.....	12
4.2 SOFTWARE ROUTINES.....	12
4.2.1 MICROSOFT OFFICE 97.....	12
4.2.2 FORTRAN PROGRAM DCALC.F.....	12
5. CALCULATION.....	13
5.1 CALCULATION INPUTS.....	13
DEFINITION.....	14
COMMENT.....	14
5.2 DESCRIPTION.....	16
5.3 PROCEDURE.....	16
5.4 IDENTIFYING TRENDING VARIABLES – DOE DATA.....	17
5.5 IDENTIFYING TRENDING VARIABLES – CRC DATA.....	18
5.6 CALCULATION OF LUTBS.....	19
6. RESULTS.....	22
7. REFERENCES.....	23
8. ATTACHMENTS.....	24

1. Purpose

The purpose of this document is to present the methodology to be used for development of the Subcritical Limit (SL) for post closure conditions for the Yucca Mountain repository. The SL is a value based on a set of benchmark criticality multiplier, k_{eff} , results that are outputs of the MCNP calculation method. This SL accounts for calculational biases and associated uncertainties resulting from the use of MCNP as the method of assessing k_{eff} .

The context for an SL estimate include the range of applicability (based on the set of MCNP results) and the type of SL required for the application at hand. This document will include illustrative calculations for each of three approaches. The data sets used for the example calculations are identified in Section 5.1. These represent three waste categories, and SLs for each of these sets of experiments will be computed in this document. Future MCNP data sets will be analyzed using the methods discussed here.

The treatment of the biases evaluated on sets of k_{eff} results via MCNP is statistical in nature. This document does not address additional non-statistical contributions to the bias margin, acknowledging that regulatory requirements may impose additional administrative penalties. Potentially, there are other biases or margins that should be accounted for when assessing criticality (k_{eff}). Only aspects of the bias as determined using the stated assumptions and benchmark critical data sets will be included in the methods and sample calculations in this document.

The set of benchmark experiments used in the validation of the computational system should be representative of the composition, configuration, and nuclear characteristics for the application at hand. In this work, a range of critical experiments will be the basis of establishing the SL for three categories of waste types that will be in the repository. The ultimate purpose of this document is to present methods that will effectively characterize the MCNP computations with respect to bias, as applicable to the repository setting. Combining varied sets of critical experiments into a single source of benchmark criticals provides wider ranges of applicability and, potentially, additional variability contribution to the treatment for the uncertainty of the bias. This will allow the estimation of the bias characteristics that will be useful in establishing the SL. If extrapolation is required, there may be need for ad hoc analyses to evaluate the bias characteristics, or at a minimum to recalculate the SL, based on the new range for the trending variable. This may also require extending the data set of critical experiments.

2. Method

The methods to be employed for determining SL values are based on the concepts included in NUREG/CR-6361 (Ref. 7.1), ANSI/ANS-8.1-1983 (Ref. 7.2), and ANSI/ANS-8.17-1984 (Ref. 7.3), for the application to post closure conditions in a repository. The application of the SL to repository conditions is the driving force in the establishment of the statistical methods used. The

concept is to relate the uncertainty and bias related to assessment of criticality experiments using MCNP calculations for ranges of conditions expected at a repository in the future.

For a subcritical configuration, it is desirable to possess a confidence that the calculation of k_{eff} for a system guarantees sufficient subcriticality. This assurance of subcriticality requires the determination of an acceptable margin based on estimated biases and uncertainties associated with the MCNP computer code and the cross section data library used to calculate k_{eff} . This Section describes three methods for the determination of the SL from the bias and uncertainty terms associated with the calculation of criticality. These approaches are used to quantify uncertainty and bias terms as they relate to criticality experiments and calculations discussed here.

For a system to be considered subcritical the calculated multiplication factor, k_{cs} , must be less than or equal to an established maximum allowable multiplication factor based on benchmark calculations and estimated uncertainty terms.

If the calculational bias β is defined as $\beta = k_{\text{c}} - 1$, then the uncertainty in the biases, $\Delta\beta$, is identical to the uncertainty in k_{c} (i.e., $\Delta k_{\text{c}} = \Delta\beta$). According to this definition of bias, the bias is negative if k_{c} is less than 1 and positive if k_{c} is greater than 1.

The value of k_{c} and therefore the bias, β , are not necessarily constant over the range of a parameter of interest. If a trend exists that causes the benchmark values of k_{eff} to vary with one or more parameters (e.g., burnup, or average energy of a neutron causing fission), then the bias, β , can be determined from a best fit of the calculated k_{eff} values as a function of each of the parameters.

The set of critical experiments used as benchmarks in the computation of β should be representative of the composition, configuration, and nuclear characteristics of the systems for which the multiplication factor is to be determined.

In addition to the bias β that is based on a given computational method, data, and a suite of benchmarks, there is an uncertainty in the bias, $\Delta\beta$. This uncertainty may include uncertainties in the critical experiment, statistical and/or convergence uncertainties in the benchmark calculations, uncertainties due to extrapolation beyond the range of experimental data, and uncertainties due to limitations or weaknesses in the geometrical or nuclear modeling of the critical experiments. This uncertainty may include statistical/convergence and modeling uncertainties.

Based on the criteria set forth in Ref. 7.1 and described above, an Upper Subcritical Limit (USL) may be determined based on the analysis of a number of critical systems. The USL is determined such that there is a high degree of confidence that a calculated result is subcritical; a system is considered acceptably subcritical if a calculated k_{eff} plus calculational uncertainties lies

at or below this limit.

The USL is the magnitude of the sum of the biases, including administrative margins, uncertainties and statistical margins applied to a set of critical benchmarks. The USL is determined such that there is high confidence that this maximum allowable multiplication factor exceeds the calculated k_{eff} for waste types represented by the set of experimental benchmark experiments which produced the USL.

In the notation of Ref. 7.1:

$$k_s = k_c - \Delta k_s - \Delta k_c$$

where:

k_c = value of k_{eff} resulting from the benchmark criticality experiments using specific computational code and data

Δk_s = uncertainty in the value of k_s

Δk_c = uncertainty in the value of k_c .

Note that the term, Δk_m , addressing “additional margin to ensure subcriticality” (Ref. 7.1) is omitted in this instance, since that is, in itself, a regulatory bias that can be introduced at application and needs no statistical treatment at this time. There may be other non-statistical biases that could be included in Δk_m which are not addressed here.

By defining bias as $\beta = k_c - 1$, then the uncertainty in the bias is the uncertainty in k_c . This allows the classification of bias as negative if k_c is less than 1, and positive if k_c is greater than 1.

Note that the uncertainty in the bias may include contributions from several sources, such as experimental measurements, calculation in the computational system, nuclear modeling, geometrical modeling, as well as others. With a sufficient set of experiments over an adequate range of parameter values, these influences, be they random or biased in nature, can be assessed for application using statistical methods.

The value of k_c and the bias β are unlikely to be constant over a range of critical experiments. If the bias was found to be a constant, a simple remedy would be to remove the bias. The trends of bias which cause the benchmark k_{eff} values to vary with neutronics parameters such as burnup or lethargy may be modeled as simple linear regression functions. The average energy of a neutron causing fission (AENCF) is defined as the ratio of the energy loss to fission divided by the weight loss of fission. Lethargy is a mathematical transformation of this value, being the natural

logarithm of 10 Mev divided by the AENCF. This regression can provide an expected value of the bias, as a function of the selected parameter. Statistical measures such as tolerance limits can be used to provide a given level of coverage of a population of values with a stated confidence level.

There may also be situations in which no neutronics parameter predictor has a statistically significant correlation with the MCNP results to establish a regression model. Without a significant trending parameter, then simpler, more direct statistical methods will suffice.

For trending situations, the quantity and range of applicability of the critical experiments should be such that the independent variable range is sufficient to bound the range for which application of the SL is desired. A sufficient quantity of data is necessary to give stability to the estimates of the regression model coefficients. A form of model validation (PRESS, the predicted sum of squares method) is discussed in Ref. 7.10.

For non-trending situations, adequate numbers of experiments over sufficient ranges of all variables are needed to assure meaningful characterization of the bias. This is necessary to meet practical needs such as testing the assumption of normality and the requirement of the desired confidence level and level of population coverage provided, e.g., the proportion of the population to be less than the SL at a specified confidence level.

The application of the subcritical limit defines the type of SL needed. For the needs of the repository, the type of limit required is global in nature. The application of the SL will be for a potential population of waste rather than for a specific element of that waste. This is in concert with the method described in Ref. 7.1 as SL Method 2.

Statistical tolerance limits meet this need. The methods in Ref. 7.5, which provide the technical basis of SL Method 2 in Ref. 7.1, provide techniques to establish constant width tolerance limits for a simple linear regression model.

A statistical tolerance limit provides a given confidence that a specified proportion of the population is included relative to the bounding value(s), which is the statistical tolerance limit. In the context of the SL, the desire is to have a high confidence that a small portion of the k_{eff} values produced by MCNP are expected to be higher than the limit. The SL is determined in a manner that most effectively accounts for the bias and its uncertainties associated with the computational system used to compute k_{eff} .

Statistical tolerance limits may be established for regression models used to trend bias as well as for situations in which there is no trending parameter. In the latter instance, the set of experimental critical experiments is effectively simply a random sample of k_{eff} values, which can be analyzed to provide a meaningful SL, given adequate numbers of observations covering the range of applicability.

The stated definition of the bias simplifies the discussion and the use of the methods, since it is given that the true value of the criticality multiplier k_{eff} is unity (1.0) for a critical system. This makes it possible to do calculations in terms of k_{eff} , rather than the difference of the system calculation k_{eff} and the true value. This eliminates the need to define the SL in terms of a margin but directly in terms of the criticality multiplication factor.

2.1 Non-Trending Normal Distribution Tolerance Limit (NDTL)

This method for determining a SL for the repository is a non-trending situation in which the values of k_{eff} are sufficient in number and scope of coverage to provide a basis for evaluation of a SL for a given class of waste form. Non-trending here specifically means that the efforts to regress the k_{eff} values for the computational system on relevant characteristics such as burnup, H/X (hydrogen to fissile atoms), lethargy, etc. have not exhibited a statistically meaningful relationship. Statistical tolerance limits based on normal distribution theory are sensitive to normality of the underlying population from which the data is obtained.

Given that the k_{eff} values produced by the computational system for the benchmark data set can be shown to be normally distributed, then the SL, denoted here as SL_N , can be calculated as:

$$SL_N = \bar{k} - k(\gamma, P, df) * s_p \quad (1)$$

where \bar{k} is the average of the k_{eff} values, unless this is greater than unity (1.0) in which instance the appropriate value for \bar{k} should be 1.0 to disallow positive bias, and where $k(\gamma, P, df)$ is a multiplier defined and tabulated in Ref. 7.6. Note that there is no relationship of \bar{k} and the $k(\gamma, P, df)$ factor used to establish a tolerance limit.

Here s_p is analogous to the square root of the pooled variance $s_p^2 = s_{k(x)}^2 + s_w^2$ shown in Ref. 7.1, Equation (4.6). In Ref. 7.1, $s_{k(x)}^2$ is equal to the variance of the set of k_{eff} values, and the pooled estimate s_p would be the square root of the sum of $s_{k(x)}^2$ and the estimate of the variance of the computational system s_w^2 . This value, s_w^2 , is a pooled estimate over the set of MCNP computational runs to determine the k_{eff} values. This quantity, s_w^2 , is the quantity defined in Ref. 7.1, Equation 4.8.

In this method, the treatment is such that positive bias can be excluded simply by substituting the value 1.0 for the computed \bar{k} estimate if this is greater than 1.0. For instance, if the data yields an estimate of $\bar{k} = 1.005$, then using the value of 1.0 for \bar{k} in Equation (1) eliminates positive bias, whereas if \bar{k} is less than or equal to 1.0, the bias is negative and no modification on this basis is warranted.

The test to be applied for reasonableness of the normality assumption is the D' test, for data sets of 50 or more observations, or the W test for less than 50 observations. These tests are described in Ref. 7.7.

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Page 8 of 25

Note that in application, the result from Equation (1) may be modified by additional regulatory penalties, and additional penalties from other uncertainties not addressed in this document.

2.2 Non-Trending Distribution-Free Tolerance Limit (DFTL)

This method is a variant of NDTL, and is applied when the hypothesis of normality of the k_{eff} data is rejected and there is no regression relationship of the criticality multipliers and some other parameter. In this non-trending case, the approach to establish the SL is through the use of distribution-free statistical methods. These methods are discussed in Ref. 7.8. The term non-parametric methods is also used to describe this approach, but for consistency and to emphasize that the underlying nature of the distribution from which the random sample is obtained is unimportant, the term distribution-free will be used.

The requirements for applying distribution-free methods to establish a statistical tolerance limit are that the data be a random sample from a continuous distribution. The methods are described in References 7.8 and 7.9.

When the resulting indices for the sample size, confidence level, and the portion of the population to be covered are included in published tables, such as Ref. 7.8, Tables A-31 and A-32, applying distribution-free methods is straight forward. One enters the table for the appropriate values for confidence, population coverage, and sample size and obtains an index value, which is applied to the ranked (sorted) values of the k_{eff} results. For instance, if the sample size is 100, and the desire is for a 95%/95% lower tolerance limit, then this index is 2, as seen in Table A-31 of Ref. 7.8. This states that the second smallest observation serves as the 95%/95% lower one-sided tolerance limit. Specific computations may be required for cases not included in published tables.

To apply the DFTL approach, the number of observations must be sufficient to accommodate the desired confidence and portion of the population to be covered. For instance, if the hypothesis of normality is rejected, and the number of observations is less than 59, one can not make a 95% confidence statement about 95% of the population being above the smallest observed value. Such a limit would be close to 95%/95% lower tolerance limit. At least one of the statement descriptors would not be strictly met, however. The level of confidence and the proportion of the population to be protected should be established in advance of any specific calculations.

The treatment of positive bias for distribution-free tolerance limits can be handled simply. The sorting of the data to establish the limit effectively treats all values greater than 1.0 (positive biases) as 1.0. This doesn't accommodate the MCNP uncertainty directly. However, this can be done by reducing each MCNP calculated k_{eff} by the quantity of three MCNP statistical uncertainty units (s_{MCNP}) before sorting. For each MCNP k_{eff} result, the value used in defining the set of k_{eff} values used for the distribution-free lower tolerance limit as the SL, denoted as SL_D , is

that $k_{\text{eff}}(j) - 3 * s_{\text{MCNP}}(j)$, where $k_{\text{eff}}(j)$ is an individual result from MCNP, $s_{\text{MCNP}}(j)$ is the MCNP error variance in the terminology of the Ref. 7.1, and $j = 1, 2, 3, \dots, n$, where n is the number of cases in the benchmark data set of interest.

If, for instance, the set of k_{eff} values to be validated consisted of 338 experiments (the laboratory critical experiment (LCE)) data, then applying this method involves sorting the k_{eff} values in ascending order such that:

$$k_{\text{eff}S_1} < k_{\text{eff}S_2} < k_{\text{eff}S_3} < \dots < k_{\text{eff}S_{338}} \quad (2)$$

and here $k_{\text{eff}S_j} = k_{\text{eff}}(j) - 3 * s_{\text{MCNP}}(j)$.

The next need is to establish the value of the subscript index that will provide the stated confidence level that the desired portion of the population is covered. This is discussed, and formulas are provided, in Ref. 7.12. In this instance, for a typical SL situation in which the desired confidence level is 0.95 that 0.995 of the population is above the limit, there is an insufficient number of observations (benchmark experiments) to make this statement. However, as also shown in Ref. 7.12 there is a way to assess what portion of the population is above the smallest calculated k_{eff} at the desired confidence level. That calculation allows the statement that there is at least 0.95 confidence that 0.991 of the population is above the smallest k_{eff} value in the available set of 338. Thus if $k_{\text{eff}S_1} = 0.97928$, that would be the SL_D for "experiments" categorized as covered by this data set for that confidence and proportion of the population. In other words:

$$SL_D = k_{\text{eff}S_1} = 0.97928 \quad (95\%/99.1\% \text{ one-sided lower tolerance limit})$$

In Ref. 7.12, in the section on non-parametric methods describing the value to be used for $k_{\text{eff}}(\text{safe})$, the term Bias Uncertainty is a function of the MCNP 'uncertainty' value. This agrees with the treatment here, and for each k_{eff} this uncertainty is treated as $3 * s_{\text{MCNP}}$.

Note that in application, the result from SL_D above may be modified by additional regulatory margins, and additional margins from other uncertainties not addressed in this document.

2.3 Trending Lower Uniform Tolerance Band (LUTB)

Standard regression techniques such as those contained in the MS Office 97 Excel software are the basis for calculating the trend of MCNP bias with independent variables such as lethargy, burnup, H/X ratios, etc. These details are shown in Attachment I.

Assumptions that impact the statistical analyses can be evaluated either analytically or by judgment. Where such assumptions fail justification, the exact statistical characteristics for confidence and population coverage are technically questionable.

The failure of the assumption for normality of the residuals for LUTB approach requires that the normality be tacitly assumed. This assumption is not of vital concern in regression, however. A more important assumption in regression is the variation of the data. An assumption in regression inference is that the variation of k_{eff} is the same for each value of the independent (predictor) variable. This can be evaluated through plots of the residuals (observed value minus the regression prediction) versus the predictor variable. Where there is apparent random scatter of the residuals versus the predictor variable, rather than a strong pattern, this assumption is acceptable.

In regression situations where normality of the residuals is rejected, there are no reasonably available distribution-free techniques that are applicable to the regression tolerance limit situation. Throughout this document, it is assumed that normality prevails for the application of LUTB, regardless of the result of the test of the distribution of normality of the residuals of the regression model.

The determination of the independent variable for trending is based on the strength of correlation between the candidate independent variables and the k_{eff} values from the MCNP runs, and other statistical results that are included in regression outputs. The correlation of possible independent variables with the k_{eff} can be determined simply in a regression program.

2.4 Details of LUTB Method

In detail, then, the actual computation for the tolerance band as a function of the independent variable x , is shown in Equation 5 of Ref. 7.5 or, in slightly different notation:

$$SL_L(x) = b_0 + b_1 * x - s_p * [C^* + |Z_{(1-p)}| * \{(n-2) / X^2_{(n-2, \gamma)}\}^{0.5}] \tag{3}$$

where b_0 and b_1 are the estimates of the intercept and slope, respectively, for the regression line used in trending the bias as a function of x . For this rest of this document, $SL_L(x)$ will be shortened to SL_L . Constraints regarding the situations for which the bias effect is positive, which is considered to be non-conservative, is handled by treating the best estimate portion, $b_0 + b_1 \cdot x$, as never more than 1.0. The term $Z_{(1-p)}$ is a standard normal deviate such that a proportion P of the population is above that value. Since the desire is for a one-sided lower limit, the minus sign is taken as a multiplier of s_p , and the result is the larger distance from the expected value. For instance, if P is 0.995, then $Z_{(1-p)}$ is about -2.58 , and $Z_{(p)}$ is about 2.58.

The subscript notation in equations (2), (3), (4), and (5) in Ref. 7.5 for the chi square value is incorrect, since the need is for an upper confidence value for the standard deviation. Where the reference shows the subscript for the chi square to be $((n - 2), 1 - \gamma)$, the correct notation and the value used in the example calculation in that article is $((n - 2), \gamma)$, for a one-sided upper confidence limit for confidence coefficient $1 - \gamma$ for the standard deviation. In the example in that article, the correct value of the chi square term is used.

Note that in application, the result from Equation (4) in Ref. 7.5 may be modified by additional regulatory margins, and additional margins from other uncertainties not addressed in this document.

2.5 Assumptions Required for Statistical Analyses

Assumptions that impact the statistical analyses can be evaluated either analytically or by judgment, e.g., residual plots for homoscedasticity (equal variances). Where such assumptions fail justification, the exact confidence coefficient and proportion of the population coverage are questionable. The failure of the assumption for normality of the residuals for the LUTB (trending, using constant width tolerance bands) requires that the normality be assumed regardless, however, due to the fact that distribution-free methods for determining the LUTB SL_L for trending are not known. Throughout this document, it is assumed that normality of residuals prevails for the application of LUTB.

Statistical confidences and portion of the population covered used are adequate to provide the level of protection desired for the end result, SL_L . This is assumed throughout the work.

3. Assumptions

3.1 Assumption of Normality in LUTB Calculations

In regression situations where normality of the residuals is rejected, there are no reasonably available distribution-free techniques that are applicable to the regression tolerance limit situation. Throughout this document, it is assumed that normality prevails for the application of LUTB, regardless of the result of the test of the distribution of normality of the residuals of the regression model.

The basis of this assumption is necessity.

3.2 Assumption of Adequacy of Confidence Levels, Population Coverage

Statistical confidences and portion of the population covered used are adequate to provide the level of protection desired for the end result, SL_L . This is assumed throughout the work.

The basis of this assumption is that the confidence levels and the proportion of the population covered are at least as great as those used in Ref. 7.1.

3.3 Assumption of Minimal Impact of Revised Data

The k_{eff} values for statepoints (SP) 32 and 33 were assumed as 0.99092 and 0.98674. The basis for these values was early results. The final values, from Ref. 14, are $k_{\text{eff}}=0.99164$, for SP32 and

$k_{\text{eff}}=0.98725$, for SP33. The difference in the assumed and final values is small and the differences in the USL calculated using the assumed and final values is not significant. This assumption is used in Section 5.

4. Use of Computer Software

4.1 Software Approved for QA Work

This does not apply.

4.2 Software Routines

The description documentation for the software routine identified in this section, other than the acquired software routines Excel, described in Section 4.2.1, contains the following information:

- Descriptions and equations of mathematical algorithms
- Description of software routine including execution environment
- Description of test cases
- Description of test results
- Range of Input parameter for which results were verified
- Identification of any limitations on software routine applications or validity
- Reference list of all documentation relevant to the qualification
- Directory listing of executable file
- Listing of computer program source code

4.2.1 Microsoft Office 97

- Title: Excel
- Version/Revision Number: Microsoft® Excel 97

MS Office 97 Excel is the basic software used to apply the methods used to estimate SL values. The user-defined formulas, inputs and results were documented in Section 5 to allow an independent repetition of the various calculations.

4.2.2 Fortran Program dcalc.f

- Title: dcalc.f
- Version/Revision Number: Version 1, Revision 0

As an input to the Excel calculations for LUTB, a stand-alone Fortran program was created to compute the quantity D in Ref. 7.4, required to compute C^* used in the LUTB calculations. This program extends Table 3 of Ref. 7.4. The program is interactive, therefore, the input was not

documented. Output is to the screen only. The result must be transcribed by the user to the appropriate cell for the Excel spreadsheet.

The dcalc.f source code was written in Fortran language using the Fortran PowerStation, Version 4.0, Professional Edition. This published software package includes IMSL libraries that contain routines for doing many statistical computations.

A listing of the source program is included in Attachment II. The results can be verified by calculating entries in the tables for D in Ref. 7.4. The executable file, dcalc.exe, is also included in that attachment.

5. Calculation

Calculations for the LUTB SL_L will be shown for the Department of Energy (DOE) criticals data and the commercial reactor criticals (CRC) data sets. The NDTL SL_N , and the DFTL SL_D methods will be demonstrated using the LCE data.

5.1 Calculation Inputs

The data sets of interest are the LCE benchmark MCNP outputs (Ref. 7.13 with 338 results); the CRC outputs consisting of 45 data sets (References 7.14, 7.15, 7.16, and 7.17); and the DOE critical MCNP outputs with 270 results (Ref. 7.13).

Data used in the calculations are shown in the Excel spreadsheets included in Attachment I.

Other inputs required for determination of SL_L include the value of D, from the dcalc.f and dcalc.exe program, interpolated values of the statistical tolerance interval multiplier from Ref. 7.6, values from the inverse normal distribution, the inverse chi squared distribution, and critical values from Ref. 7.7. For the **W** test for the hypothesis of normality, the coefficients used to compute the numerator of the test statistic are taken from Ref. 7.7.

The Excel software contains functions that can operate on the input data set to produce regression results, the values from the inverse normal distribution, and the inverse chi squared distribution.

The process for calculation of the SL begins with the evaluation of the data set for trending parameter candidates.

Given that a trending parameter (such as ALF or burnup) is identified, the regression can be done in the spreadsheet, producing several constants needed for the determination of the SL_L . Results of the regression calculation produce Int (the intercept of the regression), slope (the slope of the regression), fdf (the degrees of freedom for the standard error of the regression), and StdErrPred (the standard error of the regression model). The following table describes the cells used in the

calculation of SL values by the three methods.

Table 5-1. Cell Names for Variables used in Calculating SLs in Excel Spreadsheets

Cell Name	Definition	Comment
a	Minimum of the predictor variable values	
b	Maximum of the predictor variable values	
AdditiveTerm	$Z_{subp} * \text{Sqrt}((n-2)/\text{chisqterm})$	
Alpha2	Total confidence required for uniform tolerance band	
AveMCNPKeff	Average of k_{eff} values	
breakpoint	$(1 - \text{Int})/\text{slope}$	Independent variable value at which regression evaluates to 1.0
capA	g/h	Input needed to determine Dtol in the dcalc.exe program
chisqterm	$\text{Chiinv}(\text{gamma}, \text{fdf})$	Excel statistical function evaluated at arguments gamma, fdf
Confidence_Dcalc	$1 - \text{gamma}$	
CstarTol	$\text{Dtol} * g$	C* in Equation (3)
DeltaTol	$\text{Spooled} * \text{MultiplierTol}$	
DfIndex	Index used for establishing distribution-free statistical tolerance limits	
Dtol	Externally calculated value used to determine CstarTol	
fdf	Residual mean square degrees of freedom (from regression output)	

Cell Name	Definition	Comment
g	Intermediate calculation related to an input to dcalc.exe	
h	Intermediate calculation related to an input to dcalc.exe	
Int	Regression intercept	Equation (3)
Kowens	Value from Ref. 7.1	linear interpolation of multiplier for normal distribution statistical tolerance limit
MultiplierTol	$C^* + Z_{subp} * ((n-2)/chisqterm)$	
n	Count of data rows (number of k_{eff} values)	
rho_den1	Intermediate calculation related to an input to dcalc.exe	
rho_den2	Intermediate calculation related to an input to dcalc.exe	
rho_num	Intermediate calculation related to an input to dcalc.exe	
sigmaSq	Square of the regression standard error	
slope	Coefficient of the independent variable in the regression	Equation (3)
Spooled	Square root of the sum of the squares of the standard error of regression and the average MCNP variance	used in SL_L
Spooled3A	Square root of the variance of the k_{eff} values plus the average MCNP variance	used in SL_N
Ssquared	Average of the sum of the squared deviations about their mean for the independent variable	
StdErrPred	Standard error of prediction from the regression output	
XBAR	Average of the independent variables	
YBAR	Average of the dependent variables	

Zsubp	Norminv(0.995,0,1)	Excel function for inverse normal
-------	--------------------	-----------------------------------

5.2 Description

The items evaluated are the values SL_L for the DOE and the CRC data sets, and the values for the SL_N and SL_D for the LCE data sets.

5.3 Procedure

The procedure to establish the various SL estimates begins with a set of MCNP values for k_{eff} and the MCNP standard deviation, s_{mcnp} , with additional MCNP outputs for variables for which trending of the MCNP results may be of interest. This procedure presumes that the analyst has planned the MCNP runs to capture the needed outputs for candidate trending variables.

The process can be described generally:

First, obtain the MCNP outputs needed (k_{eff} , s_{mcnp} , and the candidate trending variables).

Second, investigate the relationship of the MCNP k_{eff} results with the trending variables to assess which single candidate, if any, has the strongest relationship.

If there is no relationship, then the resulting SL can be determined through the establishment of SL_N or SL_D , as appropriate, with the justification of the assumption of normality. The methods to determine SL_N and SL_D have been described in Section 2.

If some candidate trending variable has a significant and meaningful contribution to estimating the expected bias in the MCNP results, then the SL_L method can be used. This method has been described generally in Section 2.

The essential calculations for the LUTB SL_L are:

- regression fit of the k_{eff} results with the trending variable, which supplies Int and slope to equation (3)
- error term, the determination of gamma for the total confidence and the externally calculated value for Dtol
- the breakpoint of the regression – if all predicted values in range are not less than 1.0
- the multiplier from Ref. 7.6
- the average of the MCNP run variances (s_w^2) for obtaining the total variance.

5.4 Identifying Trending Variables – DOE Data

A correlation report showing the correlation matrix and the multiple regression results for the subset of the DOE data for which information on average energy of a neutron causing fission (AENCF, or, interchangeably in this document, AEN), average lethargy of fission (lethargy or, interchangeably in this document, ALF), and H/X ratio was captured for each MCNP k_{eff} is shown and discussed below. The relationship of AEN and ALF is that ALF is the natural logarithm of the quantity $[10/AEN]$. The point here is to illustrate the selection of the predictor that is most highly correlated to the k_{eff} , which therefore best defines the bias trend of MCNP results for k_{eff} .

The regression summary is shown below. Table 5-1 displays a correlation matrix, which shows the simple (Pearson) and the Spearman correlation coefficients (Ref. 7.10) for all the pairs of variables. Note that there is strong correlation of both AEN and H/X to lethargy, and that lethargy is slightly more correlated to k_{eff} than is AEN. The correlation coefficient can range from -1 to +1, and the sign is an indicator of the direction of the slope of the linear regression of k_{eff} . The magnitude indicates the strength of the relationship, with small values indicating a weak relation, and absolute values near 1 indicating a strong relation.

Table 5-2. Correlation Matrix, LCE Data

Pearson Correlations Section

	kmcpn	AENCF	ALF	H/X
kmcpn	1.000000	-0.412410	0.434119	0.265939
AENCF	-0.412410	1.000000	-0.847615	-0.446538
ALF	0.434119	-0.847615	1.000000	0.705121
H/X	0.265939	-0.446538	0.705121	1.000000

Spearman Correlations Section

	kmcpn	AENCF	ALF	H/X
kmcpn	1.000000	-0.435239	0.435239	0.328738
AENCF	-0.435239	1.000000	-1.000000	-0.869735
ALF	0.435239	-1.000000	1.000000	0.869735
H/X	0.328738	-0.869735	0.869735	1.000000

Table 5-3 displays an excerpt of the complete regression analysis output. The complete output is in Attachment I. This table contains the Regression Equation Section, which details the model coefficients characteristics and indicates which are statistically significant at the 0.05 (5%) level. Here it is clear that ALF is the variable to which the k_{eff} and hence, the MCNP, bias has the

strongest trend.

Table 5-3. Regression Summary, LCE Data

Correlation Matrix

	AEN	Lethargy	H/X	eff_mcnp
AEN	1.000000	-0.847611	-0.446473	-0.412409
Lethargy	-0.847611	1.000000	0.705068	0.434119
H/X	-0.446473	0.705068	1.000000	0.265904
Keff_mcnp	-0.412409	0.434119	0.265904	1.000000

Regression Equation

Independent Variable	Regression Coefficient	Standard Error	T-Value (Ho: B=0)	Prob Level	Decision (5%)
Intercept	0.991036	7.06675E-03	140.2393	0.000000	Reject Ho
AEN	-2.518399E-02	2.143905E-02	-1.1747	0.241241	Accept Ho
Lethargy	2.548476E-03	1.087046E-03	2.3444	0.019841	Reject Ho
H/X	-5.836925E-07	1.317112E-06	-0.4432	0.658033	Accept Ho
R-Squared	0.196107				

The regression for bias trend uses only ALF (lethargy), and is fitted explicitly in the Microsoft Excel spreadsheet used to illustrate the calculations for the DOE data set.

The concepts illustrated in Method 2 of the Ref. 7.1 are followed to establish a uniform width tolerance limit as discussed in Ref. 7.4 and Ref. 7.5. The technical references are followed closely, with changes as noted.

This information on the correlation of AEN, ALF, and H/X is the basis of trending the DOE data to ALF. For the CRC data, an examination of the relationship of AEN, lethargy, and burnup to k_{eff} will be discussed below.

5.5 Identifying Trending Variables – CRC Data

The CRC data includes burnup and AEN information for each state point analyzed via MCNP. Again, the AEN data was transformed to the corresponding lethargy value. As the results in Table 5-3 show, these two highly correlated variables are not the prime predictors with respect to the MCNP bias trend.

Table 5-4. Correlation Matrix, CRC Data

Pearson Correlations Section

	k_{eff}	AENCF	ALF	Burnup
k_{eff}	1.000000	-0.230240	0.227142	-0.577491
AENCF	-0.230240	1.000000	-0.999858	0.766388
ALF	0.227142	-0.999858	1.000000	-0.760713
Burnup	-0.577491	0.766388	-0.760713	1.000000

Spearman Correlations Section

	k_{eff}	AENCF	ALF	Burnup
k_{eff}	1.000000	-0.243347	0.243347	-0.479012
AENCF	-0.243347	1.000000	-1.000000	0.840000
ALF	0.243347	-1.000000	1.000000	-0.840000
Burnup	-0.479012	0.840000	-0.840000	1.000000

The conclusions here are similar to those for the DOE data set. An important exception is that the strongest candidate for the trending variable is burnup, because it has much higher correlation with the MCNP criticality multiplier. Additionally, the range of the AEN data for the 45 state points analyzed is relatively small. This can be seen in the higher correlation of lethargy and AEN, which here is the simple correlation coefficient and that is greater in magnitude in this instance due to the small range of the AEN/lethargy values. Recall the definition of AEN, and the fact that lethargy is an inverse and logarithmic transform, and this can only exhibit such a strong linear relationship when the range is very limited.

In the spreadsheet calculations for the CRC data, the trending parameter is the burnup.

5.6 Calculation of LUTBs

Computations for the estimate of LUTB SL are currently accomplished in a Microsoft Excel spreadsheet. Most of the calculations can be made using the data (k_{eff} , the trending variable, the confidence level, and the proportion of the population to be protected) in conjunction with statistical functions which are included in Excel.

There are two inputs taken from external sources that are important for determining the LUTB SL. In terms of cell names used in the spreadsheets, one of these is the value of Dtol (Ref. 7.4), that requires evaluation of the distribution function of the bivariate Student's t distribution. This calculation is done using a Fortran code, dcalc.f, described in Attachment I I. The other input obtained from an external source is the value for the cell named Kowens in the spreadsheets. This is a value taken from tables included in Ref. 7.6. Linear interpolation is used where necessary in this table and appears to be conservative. The Excel spreadsheets for all illustrative calculations are contained in Attachment I in electronic form.

A particular difference with the presentation of Ref. 7.1 involves the apportionment of the input confidence coefficient into two components to produce the required total confidence associated with the covered portion of the population. This can be appreciated by examination of the example calculation in Ref. 7.5, whereas the total confidence for the tolerance statement is related to the confidence for the contribution of the mean term and the contribution of the variance term. The basic equation is shown in Ref. 7.5 as Equation (4), which applies to two-sided tolerance bands. The following is that equation in different notation:

$$\text{Prob}[|b_0 + b_1(x - \bar{x}) - [\beta_0 + \beta_1(x - \bar{x})]| \leq C^* s_{yx}, x \in [a, b]] = 1 - \gamma_1$$

where b_0 and b_1 are the estimates of the true regression coefficients β_0 and β_1 , and s_{yx} is the estimated standard error of the regression, and the confidence bands are for the interval $[a, b]$ for the independent variable, x .

The term $C^* = D \cdot g$ is the result of the calculations which produce D (Table 3 in Ref. 7.4), and are based on the quantities g and related quantities h and ρ needed to compute D (or to enter Table 3 in Ref. 7.4). These are simple functions of the regression calculations and are described fully in Ref. 7.4.

Next let:

$$\text{Prob}[\sigma \leq \{(n - 2)/\chi^2(\gamma_2, n - 2)\}^{1/2} s_{yx}] = 1 - \gamma_2$$

and:

$$\text{Prob}[\chi^2(n - 2) > \chi^2(\gamma_2, n - 2)] = 1 - \gamma_2 \text{ where } \chi^2 \text{ denotes the chi square distribution.}$$

For the two-sided uniform tolerance band, the equation is then:

$$b_0 + b_1(x - \bar{x}) \pm s_{yx} [C^* + z_p \{(n - 2)/\chi^2(\gamma_2, n - 2)\}^{1/2}] \text{ (Equation 4, Ref. 7.5, with modified notation)}$$

Only slight modifications are needed to apply the basic approach to one-sided tolerance bands of interest in this document. This is shown for a lower tolerance limit with confidence $1 - \gamma_1/2 - \gamma_2$:

$$b_0 + b_1(x - \bar{x}) - s_{yx} [C^* + z_{(p)} \{(n - 2)/\chi^2(\gamma_2, n - 2)\}^{1/2}]$$

Let $(1 - \gamma_1)$ be the confidence associated with the uniform width confidence bands (the C^* term, defined in Section 3 in Ref. 7.4). Let $(1 - \gamma_2)$ be the confidence for the variance term (the $(n - 2)/\chi^2(\gamma, n - 2)$), then the total confidence for a one-sided tolerance limit is $1 - \gamma_1/2 - \gamma_2$, as noted in the example calculation shown in Ref. 7.5 on page 209. Here $\chi^2(\gamma, n - 2)$ represents the chi squared probability distribution value for which the cumulative probability is γ , for $n - 2$ degrees of freedom.

It is reasonable to set $\gamma = \gamma_1 = \gamma_2$, which allows easy solution for the component γ . The resulting relationship is:

$$\text{Confidence} = 1 - \gamma/2 - \gamma$$

In particular, if the desired Confidence is 0.95, then γ is 0.033333. This means that the term related to the contribution of the confidence for the mean term, C^* , must have confidence 0.966666, as must the term related to the contribution of the variance, $(n - 2)/\chi^2(\gamma, n - 2)$.

References 7.4 and 7.5 provide the basis and calculational formulas used in LUTB computations. The critical statistical assumption of homoscedasticity (homogeneity of variances) appears to be satisfied, although the normality of the residuals is rejected.

The methods for determining the LUTB SL_L are described in particular detail in Ref. 7.4, Section 7.0, Summary, and the follow-on article, Ref. 7.5. Here the formulas for the intercept, slope of the regression, the square of the standard error of regression, and the sample variance of the independent variable require no further discussion. An excellent reference for simple linear regression methods is Ref. 7.10.

The LUTB SL_L of this document is related to the tolerance limit form of the confidence method shown in Ref. 7.4, in the section described as "Uniform width confidence bands with exact confidence coefficient $1 - \gamma$." This method is extended as discussed in Ref. 7.5 to tolerance limits, from the confidence limits established in Ref. 7.4.

Following is a summary of the relevant calculations required to establish the SL_L with the desired characteristics. Regression calculations are not addressed, since these can be done in widely available commercial software. Definitions of elements in equations will be shown below.

$$h^2 = \left(\frac{1}{h} \right) + \left(\frac{(a - \bar{x})^2}{ns^2} \right)$$

$$g^2 = \left(\frac{1}{n} \right) + \left(\frac{(b - \bar{x})^2}{ns^2} \right)$$

$$\rho = \frac{1.0 + \left(\frac{(a - \bar{x})(b - \bar{x})}{s^2} \right)}{\left(\left(1 + \frac{(a - \bar{x})^2}{s^2} \right)^{1/2} \left(1 + \frac{(b - \bar{x})^2}{s^2} \right)^{1/2} \right)}$$

$$A = g/h$$

such that $[a,b]$ is the range of interest of the independent variables, and the average of the independent variables used in the fitting is shown as \bar{x} . Here, s is the sample standard deviation of the independent variable, x .

The values ρ and A , the number of degrees of freedom of the model standard error (number of data pairs minus 2 for simple linear regression), and the desired confidence level and proportion of the population covered, are used in determining the tolerance limit values. This is described in detail in Ref. 7.4, and the result is implemented in the spreadsheet. The solution for $C^* = g \cdot D$ is shown in Ref. 7.4, and the program dcalc.exe duplicates these results for the tables shown in that source. The program dcalc.exe extends these tables to confidence coefficients and sample sizes of interest for the computations at hand. Note that the use of the solution $C^* = g \cdot D$ holds only if A is in the closed range $[0.5, 1.5]$. If A is not in this range, then A is replaced by $1/A$ for the determination of D , and then $C^* = h \cdot D$.

Equation 4 in Ref. 7.5 is the desired tolerance band in terms of two-sided tolerance bands. The conversion to a single-sided tolerance band is shown in the example. This is convenient because it fits the application of the LUTB SL_L .

The confidence coefficient associated with the tolerance band for the regression is a function of the confidence coefficient for the uniform width confidence bands for the regression line, $1 - \gamma_1$, and the confidence coefficient for the model standard error, $1 - \gamma_2$. As shown in the example calculation, the total confidence for the tolerance band is $1 - (\gamma_2/2) - \gamma_1$. The choice of equal confidence coefficients, $\gamma_2 = \gamma_1$, seems natural. In this case, for a 95% total confidence coefficient for a single-sided (lower) tolerance band, the solution yields $\gamma = 0.033333$ for both components of the tolerance band. This requires that the confidence associated with the C^* term be 0.966666, and that the confidence associated with the model error term be 0.966666.

6. Results

The results for the sample calculations are tabulated in Table 6-1. Results for LUTB SL_L calculations for the DOE data set (trended on ALF) and the CRC experimental set (trended on burnup) are shown. The NDTL method was applied to the LCE data set assuming normality, resulting in SL_N . The result for the DFTL SL_D is shown, with emphasis on the fact that the population proportion is not the desired 99.5%, but a slightly lower 99.1%, although the confidence level is maintained at 95%.

Table 6-1. Results of Calculations

Data Set	Subcritical Limit type, Trend variable	Value(s)
DOE	SL _L , ALF	IF(ALF<3.7759, Int+slope*ALF-DeltaTol, Int+slope*3.7759-DeltaTol), where DeltaTol = 0.02203297
CRC	SL _L , Burnup	0.99590–2.4412E-04*Burnup for all burnups.
LCE	SL _N , No trend variable (Normality)	0.97804 (95%/99.5%)
LCE	SL _D , No trend variable (Dist ⁿ free)	0.97928 (SL _D for 95%/99.1%, <i>not</i> 95%/99.5% due to sample size available)

7. References

- 7.1. *Criticality Benchmark Guide for Light-Water-Reactor Fuel in Transportation and Storage Packages*, NUREG/CR-6361, J.J. Lichtenwalter, et al., Oak Ridge National Laboratory, March 1997.
- 7.2. *American National Standard for Nuclear Criticality Safety in Operations with Fissionable Materials Outside Reactors*, ANSI/ANS-8.1-1983, American Nuclear Society, November 1988.
- 7.3. *American National Standard for Nuclear Criticality Safety Criteria for the Handling, Storage, and Transportation of LWR Fuel Outside Reactors*, ANSI/ANS-8.17-1984 (Reaffirmed August 1989).
- 7.4. *Confidence Bands of Uniform and Proportional Width for Linear Models*, Journal of the American Statistical Association, Vol. 61, pp. 182-198, D.C. Bowden and F.A. Graybill, March 1966.
- 7.5. *QUERY: Tolerance Limits in Regression*, Technometrics, Volume 10, No. 1, N.L. Johnson (Editor), February 1968, pp. 207-209.
- 7.6. *Factors for One-sided Tolerance Limits and for Variables Sampling Plans*, SCR-607, D.B. Owens, Sandia Corporation, March 1963, Table 4.2.
- 7.7. *Assessment of the Assumption of Normality (Employing Individual Observed Values)*, ANSI N15.15-1974, American National Standard, 1974.
- 7.8. *Experimental Statistics*, National Bureau of Standards Handbook 91, M.G. Natrella, August 1963, pp. 1-14, 1-15, 2-15, Tables A-31, A-32.

- 7.9. *Introduction to Mathematical Statistics, Section Edition*, R.V. Hogg and A.T. Craig, The Macmillian Company, New York, 1965, pp. 182-185.
- 7.10. *Applied Linear Statistical Models, Third Edition*, J. Neter, W. Wasserman, and M. Kutner, Richard D. Irwin Co, Inc., Homewood, Illinois, 1990, Chapters 2-4, pp. 450-468.
- 7.11. Not used.
- 7.12. *Statistical Methods for Accurately Determining Criticality Code Bias*, Proceedings of The Topical Meeting on Criticality Safety Challenges in the Next Decade, ANS Publication, K.D. Kimball and E.F. Trumble, Chelain, Washington, Sept. 7-11, 1997.
- 7.13. *Summary Report of Laboratory Critical Experiment Analyses Performed for the Disposal Criticality Analysis Methodology*. Document Identifier Number (DI#): B00000000-0717-5705-00076 REV 01, Civilian Radioactive Waste Management System (CWRMS) Management & Operations Contractor (M&O).
- 7.14. *CRC Reactivity Calculations for Crystal River Unit 3*.
DI#: B00000000-01717-0210-00002 REV 00, CRWMS M&O.
- 7.15. *CRC Reactivity Calculations for McGuire Unit 1*.
DI#: B00000000-01717-0210-00004 REV 00, CRWMS M&O.
- 7.16. *CRC Reactivity Calculations for Sequoyah Unit 2*.
DI#: B00000000-01717-0210-00006 REV 00, CRWMS M&O.
- 7.17. *CRC Reactivity Calculations for Three Mile Island Unit 1*.
DI#: B00000000-01717-0210-00008 REV 00, CRWMS M&O.

8. Attachments

The following computer files are included on a compact disk (CD-ROM) as Attachment I.

The files with extension 'XLS' are the (Microsoft Excel) spreadsheet files showing the demonstration calculations of the SLs for the analyses described in the text.

The file DCALC.F is the source file for the Fortran program used to create the executable Fortran program DCALC.EXE.

The files COMREG.RTF and CORRDOE.RTF are the correlation/regression outputs files for the commercial and DOE data subset which display the trending selection details.

Files with the extension 'CSV' are text files (comma delimited) which contain the 'raw' data

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository**Document Identifier:** B00000000-01717-0210-00027 REV 00**Page 25 of 25**

necessary to do all the analyses described herein.

Name	Ext	Bytes	Date	Time	Filename
KEFFDOE	XLS	245,760	05-26-98	10:05a	keffdoe.xls
KEFFCRC	XLS	61,952	05-28-98	10:58a	KEFFCRC.XLS
KEFF338	XLS	336,384	05-27-98	9:35a	KEFF338.XLS
DCALC	F	15,252	05-19-98	7:54a	dcalc.f
DCALC	EXE	139,776	05-19-98	7:54a	dcalc.exe
COMREG	RTF	154,592	05-29-98	11:05a	comreg.rtf
CORRDOE	RTF	148,533	06-05-98	1:43p	corrdoe.rtf
CRC45	CSV	1,953	06-19-98	3:21p	crc45.csv
DOE270	CSV	12,535	06-19-98	3:18p	doe270.csv
ALL_LCE	CSV	15,717	06-19-98	3:30p	All_Lce.csv

Note that these files are identified by name, extension, size in bytes, date and time, and concatenated file name and extension.

Attachment II contains the dcalc.f, Version 1, software routine documentation.

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-1 of II-13

dcalc.f, Version 1

Program To Provide Input Dtol To Calculate LUTB SL

**Developed by James W. Pegram, Jr.
Framatome Technologies Incorporated
Engineering Services Department
System Engineering Section**

For Framatome Cogema Fuels

under contract with the

**Management and Operating Contractor for the
Yucca Mountain High-Level Radioactive Waste Repository Project**

Table of Contents

<u>Item</u>	<u>Page</u>
1. Purpose	3
2. Objective.....	3
3. Methodology	3
4. Required Inputs.....	3
5. Calculations Performed by the dcalc.f Program	3
6. Examples of dcalc.f Results	4
7. dcalc.f Version 1 Fortran Source Code Listing.....	4

1. Purpose

The dcalc.f program was written to support subcritical limit (SL) evaluations of the Lower Uniform Tolerance Band (LUTB) approach. The SL evaluations are performed as part of the disposal criticality methodology development for the proposed Yucca Mountain High-Level Radioactive Waste Repository. Note that in this context, the term dcalc.f is used interchangeably with dcalc.exe, the stand-alone executable version of dcalc.f.

2. Objective

The objective of the dcalc.f program is to provide an extension of Table 3 of Ref. 7.4 to cover the desired confidence coefficients not included there. The program is intended to be used without supporting documentation through the implementation of run-time user defined input prompts. Output is to the terminal screen. This must be transcribed to the spreadsheet used to calculate the SL.

3. Methodology

The methodology employed by the dcalc.f program is to prompt the user for the required inputs. The program performs calculations to match the results shown in Table 3 of Ref. 7.4, for the ranges of inputs shown, and is used to extend these tables for confidence coefficients of specific interest in establishing SL estimates.

4. Required Inputs

The dcalc.f program requires that inputs be provided at run time by the user. These inputs are described in Section 5.6 of the main body of this document, and are simple algebraic functions of the basic input data used in establishing the regression of bias. Particularly, the confidence coefficient, the quantity A , the quantity ρ , and a simple function of the number of data sets included for the regression are required. The value of each of these is relative to an interval $[a,b]$ of the predictor variable and the results are limited to this interval.

5. Calculations Performed by the dcalc.f Program

The dcalc.f program performs calculations which numerically integrate the bivariate generalization of Student's t distribution, solving for the limit of integration that corresponds to the desired confidence level. The formulas are detailed in Ref. 7.4.

The methods used to implement the calculation of D_{tol} were taken directly from the program USLTATS, described in Ref. 7.1.

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-4 of II-13

6. Examples of dcalc.f Results

The following examples demonstrate the agreement of the program and the values from Table 3, Ref. 7.4.

If $A = 0.5$, $n - 2 = 12$, and $\rho = 0.7$, for a confidence coefficient of 0.95, then D_{tol} is calculated as 4.357, and the tabled value in Ref. 7.4 is 4.36.

If $A = 0.5$, $n - 2 = 12$, and $\rho = 0.7$, for a confidence coefficient of 0.90, then D_{tol} is calculated as 3.575, and the tabled value in Ref. 7.4 is 3.58.

7. dcalc.f Version 1 Fortran Source Code Listing

```

program Dcalc
  use msimsl
  use msflib
  Parameter(io5 = 5, io6= 6, io8 = 8)
  implicit real*8 (a-h, o-z)
  common /int/ pi, a, d, rho, y, abserr, relerr, nof, n

  pi = dacos(-1.0d0)
  call setup
  write(io6,*) " Enter the value for rho "
  read (io5,*) rho
  write(io6,*) " Enter the value for A "
  read (io5,*) capa
  write(io6,*) " Enter the sample size minus 2 "
  read (io5,*) nminus2
  write(io6,*) " Enter the confidence level, e.g., .95 "
  read (io5,*) alpha2

  n = nminus2 + 2
  a = capa
  confalpha = 1.0 - alpha2
  if (capa .ge. 0.5 .and. capa .le. 1.5) then
    !call d_calc (gamma,iout) !!5/04/98 for tolerance calc
    !jwp
    !cstartol = D * g

    call d_calc (confalpha,iout)!!5/04/98 for confidence calc
    !jwp
    cstarconf = D
  write (iout, 160) D

  else
    oneovera = 1.0 / capa
    a = oneovera

    call d_calc (confalpha,iout)!!5/04/98 for confidence calc
    !jwp

```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-5 of II-13

```
        cstarconf = D

        write (iout, 160) D

    endif

160  format (/,t4,'subroutine conf',
*      /,t4,19hSelected D value = ,1pe12.3)

    pause
    stop
    end

    subroutine setup
    implicit real*8(a-h,o-z)

    common /int/ pi,a,d,rho,y,abserr,relerr,nof,n

    iout=9
    pi=dacos(-1.0d0)
    abserr=1.d-5
    relerr=1.d-5

    return
    end

    subroutine d_calc (gamma,iout)
C
C      UNIFORM TOLERANCE BANDS IN REGRESSION
C
C      This original of this program was done by T. L. Hebble of Y-12
C
C      The program calculates entries (D) in Table 3 from Bowden, D. C. and
C      Graybill, F. G., "Confidence Band of Uniform Width", JASA, 1966 (61)
C      pp 182-198. The need for this table is also found in a Technometrics
C      Query edited by N. L. Johnson about uniform tolerance bounds in
C      regression. (See Bowden, D. C., "Tolerance Interval in Regression",
C      Query, N. L. Johnson, Editor, Technometrics, 1968 (61 No. 1) pp 207-209.
C
C      Entries D are determined for input values of A, N, GAMMA, and RHO.
C
C      Another important tolerance interval reference:
C      Lieberman, G. J., and R. G. Miller, "Simultaneous Tolerance Intervals
C      in Regression", Biometrika, 1963 (50 Pts 1&2) pp 155-168.

    implicit real*8 (a-h, o-z)
    logical NEW
    parameter (NEW=.TRUE.)
C
    common /int/ pi, a, d, rho, y, abserr, relerr, nof, n
    external f1, f2
    NEW = .TRUE.

    ans = 1.0d0 - gamma
```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-6 of II-13

c this is the assumed minimum value

```
d = 1.0d0
ad = d * a
```

```
nof = 0
call intgrt (a1)
d1 = d
```

c this is the assumed maximum value

```
d = 10.0d0
ad = d * a
```

```
nof = 0
call intgrt (a2)
d2 = d
```

```
do 10 i=1, 100
d = (d1 + d2) / 2.0d0
ad = a * d
```

```
nof = 0
call intgrt (ap)
if (ap .lt. ans) then
  d1 = d
  a1 = ap
else
  d2 = d
  a2 = ap
endif
if ((dabs (ans-ap) / ans) .lt. 0.0005) go to 20
```

10 continue

```
write (iout, '(t4, 1h ,6(f7.4,2x) )' ) d1, d, d2, a1, ans, a2
20 continue
```

```
return
201 format(/,t4,3x,'d1',7x,'d',8x,'d2',7x,
& 'a1',7x,'ans',6x,'a2',/,60('-') )
end
```

```
subroutine intgrt (result)
implicit real*8 (a-h,o-z)
common /int/ pi, a, d, rho, y, abserr, relerr, nof, n
external f1, f2
call quanc8 (f2,-d,d,abserr,relerr,result,err1,nof1,flag1)
c debug
c write (*,'(6h outer, 1pe15.6, i5, .e15.6)') err1, nof1, flag1
return
end
```

```
double precision function f2 (y1)
implicit real*8 (a-h,o-z)
common /int/ pi, a, d, rho, y, abserr, relerr, nof, n
```

```

c      external f1, f2
      external fl
      y=y1
      al=a*d
      call quanc7 (f1,-a1,a1,abserr,relerr,result,err2,nof2,flag2)
c      count total # of calls to integrand
      nof=nof+nof2
c      debug
c      write (*,'(6h inner, lpe15.6, i5, e15.6)') err2, nof2, flag2
      f2=result
      return
      end

      double precision function fl (x)
      implicit real*8 (a-h,o-z)
      common /int/ pi, a, d, rho, y, abserr, relerr, nof, n
      znum = x**2 - (2.d0*rho*x*y) + y**2
      zden = (n-2) * (1.d0-rho**2)
      zterm = ((zden+znum)/zden)**(-n/2.d0)
      uterm = 2.d0*pi*(1.d0-rho**2)**.5d0
      fl = zterm / uterm
      return
      end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      title:  quanc8
C
C      Estimate the integral of fun(x) from a to b
C      to a user provided tolerance.
C      An automatic adaptive routine based on the 8-panel
C      Newton-Cotes rule.
C
C      input ..
C
C      fun      The name of the integrand function subprogram fun(x).
C              The function name should appear in an external statement
C              in the calling program.
C      a        The lower limit of integration.
C      b        The upper limit of integration.  (b may be less than a.)
C      relerr   A relative error tolerance.  (should be non-negative)
C      abserr   An absolute error tolerance.  (should be non-negative)
C
C      output ..
C
C      result   An approximation to the integral hopefully satisfying the
C              least stringent of the two error tolerances.
C      errest   An estimate of the magnitude of the actual error.
C      nofun    The number of function values used in calculation of result.
C      flag     A reliability indicator.  If flag is zero, then result
C              probably satisfies the error tolerance.  If flag is xxx.yyy,
C              then xxx = the number of intervals which have not converged
C              and 0.yyy = the fraction of the interval left to do when the
C              limit on nofun was approached.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c      subroutine quanc8 ( fun, a, b, abserr, relerr, result, errest,
>                          nofun, flag )
c

```


Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-8 of II-13

```

        implicit real*8 (a-z)
        dimension qright (31), f (16), x (16), fsave (8,30), xsave (8,30)
        integer nofun
        integer levmin, levmax, levout, nomax, nofin, lev, nim, i, j
c
c *** stage 1 *** general initialization
c   set constants
c
        levmin = 1
        levmax = 30
        levout = 6
        nomax = 5000
        nofin = nomax - 8*(levmax-levout+2**(levout+1))
c
c   trouble when nofun reaches nofin
c
        w0 = 3956.0d0 / 14175.0d0
        w1 = 23552.0d0 / 14175.0d0
        w2 = -3712.0d0 / 14175.0d0
        w3 = 41984.0d0 / 14175.0d0
        w4 = -18160.0d0 / 14175.0d0
c
c   initialize running sums to zero
c
        flag = 0.0d0
        result = 0.0d0
        cor11 = 0.0d0
        errest = 0.0d0
        area = 0.0d0
        nofun = 0
        if (a .eq. b) return
c
c *** stage 2 *** initialization for first interval
c
        lev = 0
        nim = 1
        x0 = a
        x(16) = b
        qprev = 0.0d0
        f0 = fun (x0)
        stone = (b - a) / 16.0d0
        x(8) = (x0 + x(16)) / 2.0d0
        x(4) = (x0 + x(8)) / 2.0d0
        x(12) = (x(8) + x(16)) / 2.0d0
        x(2) = (x0 + x(4)) / 2.0d0
        x(6) = (x(4) + x(8)) / 2.0d0
        x(10) = (x(8) + x(12)) / 2.0d0
        x(14) = (x(12) + x(16)) / 2.0d0
        do 25 j = 2, 16, 2
            f (j) = fun (x(j))
25 continue
        nofun = 9
c
c *** stage 3 *** central calculation
c   requires qprev, x0, x2, x4, ..., x16, f0, f2, f4, ..., f16
c   calculates x1, x3, ..., x15, f1, f3, ..., f15, qleft, qright,
c   qnow, qdiff, area
c
30 x (1) = (x0 + x(2)) / 2.0d0

```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-9 of II-13

```

f (1) = fun (x(1))
do 35 j = 3, 15, 2
    x(j) = (x(j-1) + x(j+1)) / 2.0d0
    f(j) = fun (x(j))
35 continue
nofun    = nofun + 8
step     = (x(16) - x0) / 16.0d0
qlleft   = ( w0 * (f0 + f(8))      + w1 * (f(1) + f(7))  +
>         w2 * (f(2) + f(6))      + w3 * (f(3) + f(5))  +
>         w4 * f(4) ) * step

qright (lev+1) = ( w0 * (f(8) + f(16)) + w1 * (f(9) + f(15)) +
>         w2 * (f(10) + f(14)) + w3 * (f(11) + f(13)) +
>         w4 * f(12) ) * step
qnow     = qlleft + qright (lev+1)
qdiff    = qnow - qprev
area     = area + qdiff

c
c *** stage 4 *** interval convergence test
c
    esterr = dabs (qdiff) / 1023.0d0
    tolerr = dmax1 (abserr, relerr*dabs(area)) * (step/stone)
    if (lev .lt. levmin) go to 50
    if (lev .ge. levmax) go to 62
    if (nofun .gt. nofin) go to 60
    if (esterr .le. tolerr) go to 70

c
c *** stage 5 *** no convergence
c locate next interval.
c
50 nim = 2*nim
   lev = lev + 1

c
c store right hand elements for future use.
c
do 52 i = 1, 8
    fsave (i, lev) = f (i + 8)
    xsave (i, lev) = x (i + 8)
52 continue

c
c assemble left hand elements for immediate use.
c
qprev = qlleft
do 55 i = 1, 8
    j = -i
    f (2*j+18) = f (j + 9)
    x (2*j+18) = x (j + 9)
55 continue
go to 30

c
c *** stage 6 *** trouble section
c number of function values is about to exceed limit.
c
60 nofin = 2*nofin
   levmax = levout
   flag = flag + (b - x0) / (b - a)
go to 70

c
c current level is levmax.

```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-10 of II-13

```
c
62 flag = flag + 1.0d0
c
c   ***   stage 7   ***   interval converged
c       add contributions into running sums.
c
70 result = result + qnow
   errest = errest + esterr
   cor11 = cor11 + qdiff / 1023.0d0
c
c       locate next interval.
c
72 if (nim .eq. 2*(nim/2) ) go to 75
   nim = nim/2
   lev = lev-1
   go to 72
75 nim = nim + 1
   if (lev .le. 0) go to 80
c
c       assemble elements required for the next interval.
c
   qprev = qright (lev).
   x0     = x(16)
   f0     = f(16)
   do 78 i = 1, 8
       f (2*i) = fsave (i, lev)
       x (2*i) = xsave (i, lev)
78 continue
   go to 30
c
c   ***   stage 8   ***   finalize and return
c
80 result = result + cor11
c
c       make sure errest not less than roundoff level.
c
   if (errest .eq. 0.0d0) return
82 temp = dabs (result) + errest
   if (temp .ne. dabs(result)) return
   errest = 2.0d0 * errest
   go to 82
   end
   subroutine quanc7 ( fun, a, b, abserr, relerr, result, errest,
>                   nofun, flag )
c
   implicit real*8 (a-z)
   dimension qright (31), f (16), x (16), fsave (8,30), xsave (8,30)
   integer nofun
   integer levmin, levmax, levout, nomax, nofin, lev, nim, i, j
c
c   ***   stage 1   ***   general initialization
c       set constants
c
   levmin = 1
   levmax = 30
   levout = 6
   nomax  = 5000
   nofin  = nomax - 8*(levmax-levout+2** (levout+1))
c
```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-11 of II-13

```

c      trouble when nofun reaches nofin
c
w0 = 3956.0d0 / 14175.0d0
w1 = 23552.0d0 / 14175.0d0
w2 = -3712.0d0 / 14175.0d0
w3 = 41984.0d0 / 14175.0d0
w4 = -18160.0d0 / 14175.0d0
c
c      initialize running sums to zero
c
flag = 0.0d0
result = 0.0d0
cor11 = 0.0d0
errest = 0.0d0
area = 0.0d0
nofun = 0
if (a .eq. b) return
c
c      *** stage 2 *** initialization for first interval
c
lev = 0
nim = 1
x0 = a
x(16) = b
qprev = 0.0d0
f0 = fun (x0)
stone = (b - a) / 16.0d0
x(8) = (x0 + x(16)) / 2.0d0
x(4) = (x0 + x(8)) / 2.0d0
x(12) = (x(8) + x(16)) / 2.0d0
x(2) = (x0 + x(4)) / 2.0d0
x(6) = (x(4) + x(8)) / 2.0d0
x(10) = (x(8) + x(12)) / 2.0d0
x(14) = (x(12) + x(16)) / 2.0d0
do 25 j = 2, 16, 2
    f (j) = fun (x(j))
25 continue
nofun = 9
c
c      *** stage 3 *** central calculation
c      requires qprev, x0, x2, x4, ..., x16, f0, f2, f4, ..., f16
c      calculates x1, x3, ..., x15, f1, f3, ..., f15, qlleft, qright,
c      qnow, qdiff, area
c
30 x (1) = (x0 + x(2)) / 2.0d0
f (1) = fun (x(1))
do 35 j = 3, 15, 2
    x(j) = (x(j-1) + x(j+1)) / 2.0d0
    f(j) = fun (x(j))
35 continue
nofun = nofun + 8
step = (x(16) - x0) / 16.0d0
qlleft = ( w0 * (f0 + f(8)) + w1 * (f(1) + f(7)) +
> w2 * (f(2) + f(6)) + w3 * (f(3) + f(5)) +
> w4 * f(4) ) * step

qright (lev+1) = ( w0 * (f(8) + f(16)) + w1 * (f(9) + f(15)) +
> w2 * (f(10) + f(14)) + w3 * (f(11) + f(13)) +
> w4 * f(12) ) * step

```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-12 of II-13

```
      qnow      = qleft + qright (lev+1)
      qdiff     = qnow - qprev
      area      = area + qdiff
c
c ***   stage 4   ***   interval convergence test
c
      esterr = dabs (qdiff) / 1023.0d0
      tolerr = dmax1 (abserr, relerr*dabs(area)) * (step/stone)
      if (lev .lt. levmin) go to 50
      if (lev .ge. levmax) go to 62
      if (nofun .gt. nofin) go to 60
      if (esterr .le. tolerr) go to 70
c
c ***   stage 5   ***   no convergence
c      locate next interval.
c
50  nim = 2*nim
      lev = lev + 1
c
c      store right hand elements for future use.
c
      do 52 i = 1, 8
          fsave (i, lev) = f (i + 8)
          xsave (i, lev) = x (i + 8)
52  continue
c
c      assemble left hand elements for immediate use.
c
      qprev = qleft
      do 55 i = 1, 8
          j = -i
          f (2*j+18) = f (j + 9)
          x (2*j+18) = x (j + 9)
55  continue
      go to 30
c
c ***   stage 6   ***   trouble section
c      number of function values is about to exceed limit.
c
60  nofin = 2*nofin
      levmax = levout
      flag = flag + (b - x0) / (b - a)
      go to 70
c
c      current level is levmax.
c
62  flag = flag + 1.0d0
c
c ***   stage 7   ***   interval converged
c      add contributions into running sums.
c
70  result = result + qnow
      errest = errest + esterr
      cor11 = cor11 + qdiff / 1023.0d0
c
c      locate next interval.
c
72  if (nim .eq. 2*(nim/2) ) go to 75
      nim = nim/2
```

Title: Calculation Of Upper Subcritical Limits For Nuclear Criticality In A Repository

Document Identifier: B00000000-01717-0210-00027 REV 00

Attachment II, Page II-13 of II-13

```
      lev = lev-1
      go to 72
75  nim = nim + 1
      if (lev .le. 0) go to 80
c
c      assemble elements required for the next interval.
c
      qprev = qright (lev)
      x0     = x(16)
      f0     = f(16)
      do 78 i = 1, 8
          f (2*i) = fsave (i, lev)
          x (2*i) = xsave (i, lev)
78  continue
      go to 30
c
c  ***   stage 8   ***   finalize and return
c
80  result = result + cor11
c
c      make sure errest not less than roundoff level.
c
      if (errest .eq. 0.0d0) return
82  temp = dabs (result) + errest
      if (temp .ne. dabs(result)) return
      errest = 2.0d0 * errest
      go to 82
      end
```