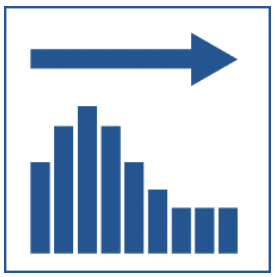


SIPmath™ 3.0 Standard

FOR MAKING UNCERTAINTY ACTIONABLE

Sam Savage
January 24, 2022

Probability Management

CONTENTS

SIPMATH™ 3.0 STANDARD

FOR MAKING UNCERTAINTY ACTIONABLE

1 EXECUTIVE SUMMARY

The discipline of probability management represents uncertainties as data that obey both the laws of arithmetic and the laws of probability. This has been accomplished for decades in the fields of financial engineering, insurance, and stochastic optimization with vectors of realizations of Monte Carlo simulations. The SIPmath 2.0 Standard, introduced in 2014, added metadata and created an open standard for this approach. The resulting data elements are called SIPs (Stochastic Information Packets). The simplicity of the approach allows for implementation in a wide variety of platforms including native Excel.

However, there are also disadvantages in terms of storage and data management that have been overcome by incorporating two independent breakthroughs in simulation technology: the portable HDR Pseudo Random Number Generator from Doug Hubbard, and the Metalog Distribution from Tom Keelin. The result is the SIPmath 3.0 Standard, which represents Virtual SIPs as JSON data.

2 OVERVIEW OF THE STANDARD

The SIPmath 2.0 Standard represents uncertainties as a vector (of dimension N trials) of specific realizations of a probability distribution along with metadata. The data structure is known as a SIP (Stochastic Information Packet). A statistically coherent set of SIPs, that is, one that preserves the statistical relationships between its constituents, is a Stochastic Library Unit with Relationships Preserved (SLURP). Virtually any sort of relationship may be preserved in this manner, as opposed to simpler forms of correlation as produced by methods such as the Gaussian Copula.

2.1 ADVANTAGES OF ARRAY SIPS

There are four primary advantages of the SIP representation using vectors of realizations.

1. SIPs are ***actionable*** in that the output SIPs on one computer simulation may be used directly as input into other simulations.
2. SIPs are ***arithmetical***, in that if the SIPs are coherent, that is, statistical relationships or lack thereof between SIPs is preserved and they have the same dimension (number of trials), then they have ***group theoretic*** properties that are similar to numbers. They may be added, multiplied, or divided element by element to allow stochastic calculation. This allows diverse simulations running on various platforms to be aggregated through arithmetical operations across the enterprise or industry.
3. SIPs are ***auditable***. Since they consist of numerical data, they may be audited at any level of a simulation or network of simulations and contain metadata that includes provenance.
4. SIPs are ***agnostic***, in that as data they may be implemented in any software platform that processes arrays.

2.2 DISADVANTAGES OF ARRAY SIPS

1. Large quantities of data must be stored. Even multiple instantiations of independent, identically distributed (IID) variables require a separate SIP for each, to ensure that no two variables are correlated. For example, imagine the distribution of time to failure of a part on an aircraft, and imagine a SIP of 10,000 trials representing this distribution. If there were 100 such parts on the plane, the original SIP would need to be permuted 99 times to ensure that all the parts did not fail simultaneously. Now suppose there were 1,000 planes in the fleet. This would require total storage of 10,000 x 100 x 1,000 or 1 billion numbers.
2. The number of trials may not be usefully increased beyond the dimension of the original sample.

2.3 VIRTUAL SIPS

Virtual SIPs maintain the four advantages listed above while overcoming the two disadvantages. Joint distributions may easily be created through classical Copula methods such as the Gaussian Copula. And when there are highly non-linear statistical relationships between variables, they may be extended through conditional virtual SIPs or driven by a Copula Layer of array SIPs of specifically rank ordered uniform U(0,1) variables. Thus, a combination of both array and virtual SIPs provides adequate flexibility for most Monte Carlo applications.

2.3.1 THE F-INVERSE METHOD

The most common technique for generating a random variate is to drive the inverse of its cumulative distribution (F-Inverse, also known as a Quantile Function) with a uniform random number (a U(0,1). In Excel, for example, the following formula generates a normal random variate with mean = 10, sigma = 2.

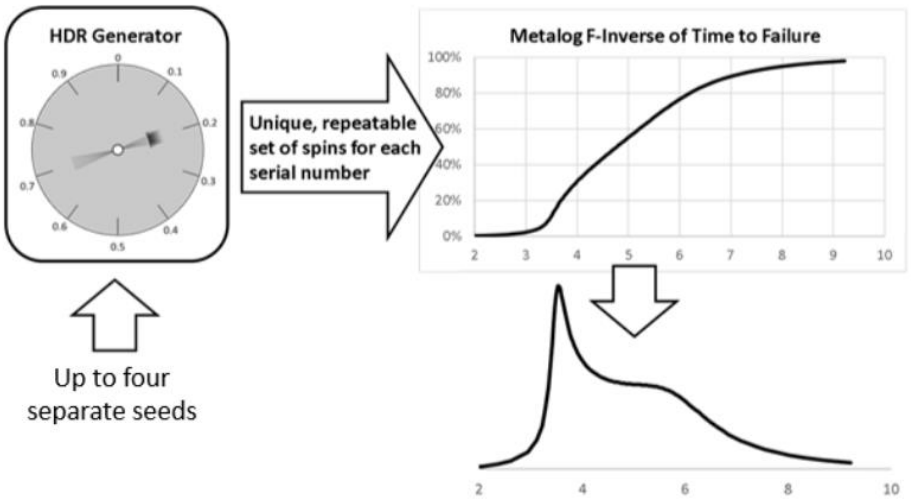
=NORM.INV(RAND(),10,2)

2.3.1.1 HDR Random Number Generator

The HDR cross-platform generator removes the need to distribute actual trials and plays the role of a seed-able RAND formula in the F-Inverse method. Designed by Doug Hubbard, this "counter" type of pseudo random number generator allows simulations running on diverse computer platforms to generate either identical or independent streams of random numbers as required though a multi-dimensional seed. This allows simulation results to be networked together while maintaining statistical coherence. The generator performs well under the respected Dieharder random number tests and has formulas that fit into a single cell of Excel to perform interactive, as opposed to recursive, simulations.

2.3.1.2 Metalog Distributions

The Metalog Distribution transforms uniform random numbers, such as those generated by the HDR or other algorithms, into nearly any continuous random variate. Invented by Tom Keelin, this system replaces hundreds of arcane mathematical equations for different types of uncertainties with a single elegant family of formulas. The system not only replicates traditional statistical results but goes beyond them by detecting multiple populations within a single data set and providing analytical expressions for multi-modal distributions. It has many uses beyond simulation, and in particular has been used to implement the previously unsolved problem of an analytical expression for the sum of lognormal distributions.



2.4 THE JSON SIPMATH 3.0 STANDARD

The purpose of the JSON SIPmath 3.0 Standard is to provide a cross platform format for the specifications of Monte Carlo simulation variables which will yield the same sequences of random variates in all environments. It supports SIPs in three formats: Virtual SIPs using the F-Inverse Method; array SIPs composed of realizations as in the SIPmath 2.0 Standard; and compressed array SIPs in DIST format.

2.5 SOFTWARE ROUTINE REQUIREMENTS

The power of the SIPmath 3.0 Standard derives from the fact that once a few routines are implemented in the client environment, then identical simulation streams of nearly any continuous random variate may be expressed with at most 22 numerical parameters. Furthermore, metadata may include a few specified Monte Carlo trials for calibration purposes. The following routines are assumed to be present on the client for interpreting the parameters stored in the library in JSON format.

1. The HDR random number generator.
2. The Metalog M Function.
3. A table lookup.
4. A matrix multiplication.
5. A Gaussian copula calculation.

2.5.1 THE HDR RANDOM NUMBER GENERATOR

This counter type pseudo random generator takes four seeds, uses only integer arithmetic, and is supported by a single formula in Excel. It generates the same stream of U(0,1) variates on any platform to within about 15 decimal places of accuracy. See:

- https://www.informs-sim.org/wsc19papers/339.pdf
- https://www.probabilitymanagement.org/hdr
- https://www.probabilitymanagement.org/s/HDR_2_0-Seed-Standard.xlsx

2.5.2 THE METALOG M FUNCTION

The Metalog distribution creates an F-Inverse function, known as the Metalog M Function, for nearly any continuous probability distribution based on data. The parameters of the M Function are:

1. An optional upper and lower bound.
2. A set of a-coefficients. In theory there could be any number of these, but in practise more than 16 coefficients, or terms, would rarely be used.

There are four forms of the Metalog which transform the M function in different ways depending on whether the distribution is unbounded, bounded from below, bounded from above, or both. The unbounded F-Inverse is denoted as $M_k(y)$ for a Metalog with k terms. The argument y is a probability, and in a virtual SIP is the output of a U(0,1) generator such as the HDR. The basic formula, Mk(y), is further transformed for the bounded cases. For example, a Metalog with a lower bound of LB is of the form LB+EXP(Mk(y)), much as a lognormal distribution is defined as EXP(Normal).

See https://en.wikipedia.org/wiki/Metalog_distribution

2.5.2.1 Determining the Parameters of the M Function

The parameters are found through a least squares process, which has been implemented in multiple environments as described in Appendix A. There may be trade-offs between goodness of fit and the potential of overfitting, so human judgement and subject matter expertise may be involved at this stage.

2.5.2.2 Interpreting the Parameters of the Metalog

This merely requires the Metalog M Function, appropriately transformed to accommodate the specified bounds. Implementations in multiple environments are described in Appendix A.

2.5.3 A TABLE LOOKUP

A Table Lookup is used in conjunction with the HDR for discrete random variables.

2.5.4 A MATRIX MULTIPLICATION

Matrix Multiplication is necessary for interpreting Cholesky factors if provided, and also may be used in implementing the Metalog M Function.

2.5.5 A GAUSSIAN COPULA ROUTINE

A Gaussian Copula routine is required to interpret a correlation matrix if present. This available in most programming environments but should be verified using the Calibration Trial values in the library Metadata.

2.6 RESERVED WORDS AND CANONICAL LIBRARIES

A good way to understand the standard is through canonical library examples and reserved words list.

2.6.1 RESERVED WORD LIST

Reserved Word	Definition
aCoefficients	The "a" Coefficients of a Metalog Distribution. Represented as "aCoefficients":[2.30,-0.20 ...
arguments	Arguments for a function
calibrationTrials	Arrays containing the first few simulated trials for cross platform calibration
choleskyMatrix	Cholesky factors resulting from a Gaussian copula
column	The column of the copula layer to be used as input to generate a SIP
copula	Denotes the Copula section of the library
correlationMatrix	Correlation matrix for use in a Gaussian copula
dateCreated	Date library was created
density	Density values stored as metadata for display on the client.
DIST	Denotes 1.1 Distribution String base 64 compressed SIP array
down	Denotes rounding to lower integer
function	A function or procedure for generating either Uniform(0,1) variables or SIP variables
GaussianCopula	Refers to the standard GaussianCopula algorithm as implemented on the client available in nearly all platforms.
GeneralizedMetalog	A Metalog distribution whose "a" Coefficients are determined by a parameter driven lookup into a data structure
genMetalogTable	The data structure for a Generalized Metalog
globalVariables	Variables defined for use in other parts of the library, for example "globalVariables": [{ "name": "correlationMatrixValue", "value": { ...

	{
	
	"value": {

HDR_2_0	Denotes the 2.0 HDR Standard with four seeds, as defined at https://www.probabilitymanagement.org/s/HDRPRNG2019.pdf
histogram	Histogram values stored as metadata for display on the client.
Index	Denotes a function such as =Index in Excel, which indexes sequentially through an array based on PM_Index, the client's simulation iteration counter
url	The url of an external library
Lookup_Table	Denotes a Lookup Table, for example for use in simulating discrete random variables as implemented on the client.
lowerBound	Denotes the lower bound of a Metalog distribution
Matrix	Denotes a matrix, for example a correlation matrix or Cholesky decomposition matrix
metadata	Optional metadata to accompany a SIP, for example, the mean, max, or densities or histogram values to be displayed on a client application
Metalog_1_0	The Metalog distribution as defined in https://www.probabilitymanagement.org/s/deca20160338-1.pdf
multiDimSIP_Array	Multi-dimensional array in which the last dimension is Monte Carlo trials. The sequencing of dimensions is last first.
Name	The name of an object
Nearest	Denotes rounding to nearest integer
objectType	This name is required to be present at the top level in all SIPmath libraries. Its value must be "sipModel"
parameter	An argument to a function that must be input from the client
PM_Index	Denotes the client's simulation iteration counter
PM_Trials	The number of trials stored in a SIP array or DIST
Precision	To ensure consistent cross platform results, precision must be specified for the lowest expected level
provenance	A description of the derivation of the library, for example the name of a simulation from which it was generated
Ref	Refers to the uniform(0,1) variable whose source is either a random number generator or the output of a copula layer.
Rng	Random number generator
Rounding	A property of a SIP, may be down, nearest or up
SIP_Array	An array of Monte Carlo trials
Sips	Denotes the Sips section of the library
Type	The type of an argument
U01	Denotes the random number section of a library
Up	Denotes rounding up to an integer
upperBound	The upper bound of a Metalog
Value	The value of a key value pair
Version	Version number of the standard

2.6.2 CANONICAL LIBRARY LIST

The list of canonical libraries appears below. Below each name are links to the library itself and to an annotated PDF version.

Library Name	Description
1. Single Variable Library | Annotated	Simplest example of a single Metalog variable driven by a single HDR generator
2. Gaussian Copula Library | Annotated	Most common type of library and supports Metalog variables coupled through a Gaussian Copula
3. Indirect CSV SIP Array Library | Annotated	JSON that includes a url pointing to a CSV file with actual SIPs
4. Indirect SIPmath 2.0 Excel Range Name SIP Array Library | Annotated	JSON that includes a url pointing to a 2.0 Excel file as created by the SIPmath Tools
5. Happy Fish SIP Array Copula Library | Annotated	Combines two Metalogs of Tom Keelin's Steelhead Trout weight, coupled through a SIP Copula generated by a Happy Face. Observe the scatter plot to see the happy fish.
6. Discrete Poisson Library | Annotated	Discrete (0,1,2 etc.) calculated with an HDR and a Lookup Table instead of a Metalog
7. SIP Array in JSON Library | Annotated	This is an actual SIP as in 3 and 4 above, but the SIP is stored in the JSON itself.
8. Rounded Bounded Single HDR Library | Annotated	This includes eight Metalogs, with various forms of bounding and rounding
9. Input Param and Environment Specific Library | Annotated	Set Seed3 of Variable2 to 1 to Calibrate. 1st Three Trials are: "Variable1",2.894698, 0.865379, 1.8223, "Variable2_seed3_param",11.091734, 12.036073,7.760296, "StandardNormal",-0.07923, -1.046566, 0.081651. Also contains PsiNormal and RiskNormal
10. Generalized Metalog Library | Annotated	The Generalized Metalog provides analytical solutions to previously unsolved problems like sums of IID lognormals. These libraries require an app on the client.

2.7 THE LIBRARY SCHEMA

A schema is an organizational structure for the construction of a database or document. A JSON schema includes required parameters and constraints that allow applications to validate the JSON document.

2.7.1 THE SCHEMA

The schema is available at https://www.probabilitymanagement.org/s/multi-lib-schema-2021-08-19.json

2.7.2 VALIDATING A LIBRARY WITH THE SCHEMA

Two things are necessary to validate a .SIPmath library: 1) the validation schema; and 2) a JSON schema validator, such as https://www.jsonschemavalidator.net/.

To validate the library, follow the steps below.

1. Visit the schema validation website above.
    a. There will be two large panes where you can paste text. The schema goes on the left, the .SIPmath file on the right.
2. Copy the schema text.
    a. Open the validation schema file linked above in a text editor such as Notepad. (Microsoft Word is not recommended, as it might put invisible formatting in the file which would interfere with the validation.)
    b. Select all the text of the schema (Ctrl-A).
    c. Copy the text (Ctrl-C).
3. Paste the schema into the left pane of the schema validator web site.
    a. Click on the left pane of the schema validation site to activate it.
    b. Type Ctrl-A to select everything in the pane.
    c. Type Ctrl-V to paste the previously copied schema text into the pane.
4. Copy the .SIPmath library text.
    a. Open the .SIPmath file in a text editor.
    b. Select all the text. (Ctrl-A).
    c. Copy it. (Ctrl-C).
5. Paste the library text to the right pane of the schema validator.
    a. Click the right pane.
    b. Type Ctrl-A.

c. Type Ctrl-V

6. The validation website will automatically compare the library with the schema and report any problems it finds. Note that the validator may not find all problems.

APPENDIX A: IMPLEMENTATION OF THE METALOG M FUNCTION

The Metalog distribution is an important advance in applied statistics with many uses beyond the SIPmath 3.0 Standard. The most authoritative source on the subject is Tom Keelin's metalogs.org, which contains numerous publications and Excel Template implementations. There are also several implantations in R, Python, and other software as outlined below.

The role of the Metalog distribution in the SIPmath 3.0 Standard is to create an F-Inverse or Quantile function, known as the M function, for conveying nearly any continuous distribution with no more than 18 parameters. When the M function is accompanied by the four seeds of the HDR random number generator, the total of 22 parameters will drive an M Function and HDR generator on the client machine to produce the same set of random variates in the same order.

There are three stages of Metalog implementation in the SIPmath 3.0 Standard:

1. Fitting the Metalog parameters to data.
2. Generating the JSON library containing the parameters.
3. Interpreting the Metalog on the client computer.

A1. FITTING THE PARAMETERS OF THE M FUNCTION TO DATA

The parameters of the M Function are:

1. An optional upper and lower bound specified by the user.
2. A set of a-coefficients. In theory there could be any number of these, but in practise more than 16 coefficients, or terms, would rarely be used. The parameters are found through a least squares process. There may be trade-offs between goodness of fit and the potential of overfitting, so human judgement and subject matter expertise may be involved at this stage.

After these parameters are determined, they must be embedded in the JSON library. Currently two software packages perform both the steps of determining the a-coefficients and creating the JSON Library.

1. Analytic Solver from Frontline Systems.
2. PySIPlibraryjson open-source Python code from ProbabilityManagement.org.

A2. INTERPRETING THE M FUNCTION ON THE CLIENT

There are four forms of the Metalog which transform the M function in different ways depending on whether the distribution is unbounded, bounded from below, bounded from above, or both. The unbounded F-Inverse is denoted as $M_k(y)$ for a Metalog with k terms.

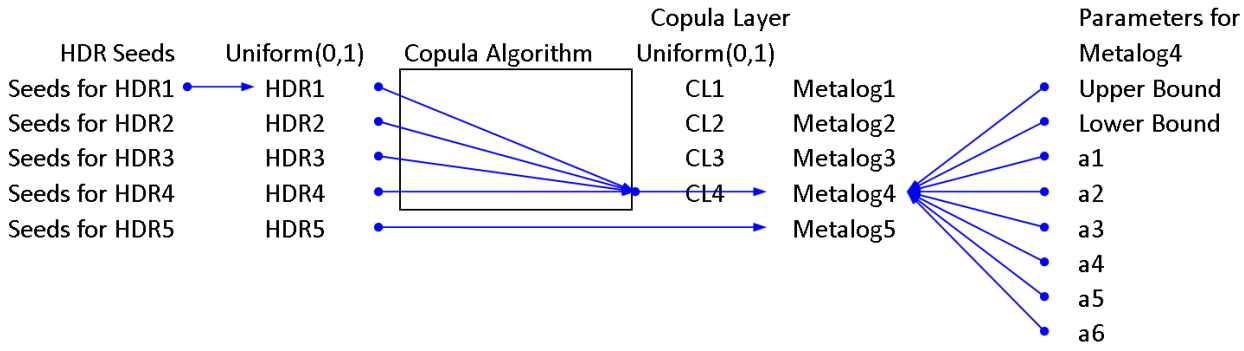
$$M_k(y) = \sum_{i=1}^{k} a_i g_i(y)$$

Where y is a probability and in a virtual SIP is the output of a U(0,1) generator such as the HDR, a_i is the i^{th} a-coefficient, and the g_is are known as the Metalog Basis Functions, not unlike Sines and Cosines forming the basis functions of Fourier Series. Although there is no limit on the number of coefficients, for most practical applications it will not exceed 16.

The basic formula, $M_k(y)$, is further transformed for the bounded cases. For example, a Metalog with a lower bound of LB is of the form LB+EXP($M_k(y)$), much as a lognormal distribution is defined as EXP(Normal).

See https://en.wikipedia.org/wiki/Metalog_distribution

A Flow diagram for interpreting the Metalog and HDR generators on the client is shown below.



Starting on the left are the seeds of five independent HDR U(0,1) variables. In this example, the first four of these enter a Copula Algorithm to create a Copula Layer of U(0,1) variable CL1 through CL4 which are not independent. Metalog1 through 4 are correlated through the Copula Layer while Metalog5 is independent.

A3. TAKEN FROM THE SOFTWARE PAGE FROM METALOGS.ORG

Software

Software implementations of Metalog distributions are that are known to us are listed below. There are likely others of which we are unaware.

Excel Workbooks. We offer the complete Metalog system (including metalog panel) within free, convenient Excel workbooks programmed by the author of this website. The worksheets in these workbooks are programmed in native Excel, without macros or named ranges, which means that they can be easily and safely copied into other Excel workbooks while retaining their full functionality. Click here for details.

Frontline Systems' Analytic Solver, RASON, and Solver SDK. Analytic Solver V2021.5 brings Metalog distributions to the fore, with a powerful new facility to automatically fit user data to the full range of possible (bounded and unbounded, multi-term) Metalog distributions and to compare Metalog distributions with classical distributions based on user-selected goodness of fit criteria. This release also includes a metalog panel to aid choosing among metalogs with different numbers of terms.

Lumina Decision Systems' Analytica. Lumina Decision Systems, which provides software to help bring clarity to difficult decisions, includes the complete Metalog system (as originally published) in the recent release of its flagship product Analytica 5.0. For documentation of Analytica's Metalog implementation, click here.

R. A CRAN approved rmetalog package is available. For questions and feedback, contact Isaac Faber.

Python. Two Python packages are available. Pymetalog closely mirrors the R package. Metalogistic is coded so as to take full advantage of the SciPy platform of operations on continuous probability distributions.

SIPmath Modeler Tools. SIPmath™ Modeler Tools is a simulation environment in Excel provided by Probability Management as a free Excel add-in. The current version of the Tools offers two implementations. One is the SPT-metalog distributions (e.g. 3-term metalogs parameterized by 10/50/90 quantiles). Here is a tutorial. For the general metalogs (including more terms), we offer a separate tutorial.

SmartOrg's Portfolio Navigator. SmartOrg, which provides a web-based software environment to enable better decision conversations in new product development and portfolio management, includes metalogs in their new release of Portfolio Navigator 7 to aid probability distribution visualization and communication.

Web browser. MakeDistribution.com allows easy experimentation with fitting metalogs and other distributions to data.