Stochastic Formal Methods for Rare Failure Events due to the Accumulation of Errors

Marc Daumas, David Lester

Stochastic Formal Methods for Rare Failure Events due to the Accumulation of Errors

Marc Daumas
CNRS-LIRMM visiting LP2A
University of Perpignan Via Domitia
Perpignan, France 66860
Email: Marc.Daumas@Univ-Perp.Fr

David Lester
School of Computer Science
University of Manchester
Manchester, United Kingdom M13 9PL
Email: David.R.Lester@Manchester.Ac.UK

***Abstract*— This paper provides an accurate bound on the number of numeric operations (fixed or floating point) that can safely be performed before accuracy is lost based on the assumption that accumulated errors are uniformly distributed in $\pm\frac{1}{2}$ unit in the last place. This work has important implications for control systems with safety-critical software, as these systems are now running fast enough and long enough for their errors to impact on their functionality. Furthermore, worst-case analysis would blindly advise the replacement of existing systems that have been successfully running for years and that will continue running before software development practices evolve. We present here new theorems that we are currently validating with the PVS proof assistant. This theory will allow code analyzing tools to produce formal certificates of accurate behavior. FAA regulations for aircraft require that the probability of an error be below 10^{-9} for a 10 hour flight [1]. Such a low failure rate is stretching the limits of generic calculations solely based on the standard deviation of random variables for the intermediate sums. We need many individual errors for the Central Limit Theorem approximation to be sufficiently accurate (distance well below 10^{-9}). The precise bound presented here enhances the number of bits of the result that can safely be regarded as correct.**

I. Introduction

Formal proof assistants are used in areas where errors can cause loss of life or significant financial damage as well as in areas where common misunderstandings can falsify key assumptions. For this reason, formal proof assistants have been much used in floating point arithmetic [2], [3], [4], [5], [6]. Previous references just link to a few projects using proof assistants such as ACL2, HOL [7], Coq [8] and PVS [9].

All these projects deal with worst case behavior. Recent work has shown that worst case analysis may be meaningless for applications that run for a long time. For example, a process adds numbers in ± 1 with a measure error of $\pm 2^{-25}$. If this process adds 2^{25} items, then the accumulated error is ± 1, and note that 10 hours of flight time at operating frequency of 1 kHz is approximately 2^{25} operations. Yet we easily agree that provided the measure errors are not correlated, the actual accumulated error will be much smaller.

We recall in Section II the model that we are using. Section III recalls our formal developments in probability and presents our new theorems. The Doobs-Kolmogorov inequality was used in [10] but precise definitions of measure and round-off errors used in Section IV provide a much more precise way to compute the probability that a piece of software will successfully run within an acceptable error bound. We use the result of Section IV for intermediate sums before approximations based on the Central Limit Theorem are sufficiently accurate. As this new bound is long to obtain, it might not be required for ordinary pieces of software. Yet, failure rate as defined in aeronautics is very low and this new bound is necessary whatever the computational price.

We cannot use the Central Limit Theorem before the approximation to normal distribution is sufficiently close so that the sum including the additional approximation errors is bounded by 10^{-9} (there are potentially 2^{25} additional errors). Our suggestion is to carry on precise computing as long as possible and revert to results linked to Doobs-Kolmogorov inequality [10] and the Central Limit Theorem when the approximation error becomes negligible compared to 10^{-9}

In the rest of this text, we assume that the created round-off and measure errors are unbiased independent random variables or that their expectation conditional to the previous errors is zero.

II. Stochastic model and state of the art

A. Individual round-off and measure errors

We are dealing with fixed or floating point numbers. A floating point number represents $v = m \times 2^e$ where e is an integer and m is a fixed point number [11]. IEEE 754 standard [12] uses sign-magnitude notation for the mantissa and the first bit b_0 of the mantissa is implicit in most cases ($b_0 = 1$) leading to the following definition where s and all the b_i are either 0 or 1 (bits).

$$v = (-1)^s \times b_0.b_1 \cdots b_{p-1} \times 2^e$$

Some circuits such as the TMS320 use two's complement notation for m leading to the following definition [13].

$$v = (b_0.b_1 \cdots b_{p-1} - 2 \times s) \times 2^e$$

In fixed point notation e is a constant provided by the data type and the first bit is no longer implicitly equal to 1.

For all the previous notations, we define for any representable number v, the unit in the last place function where e is the exponent of v as above.

$$\mathrm{ulp}(v) = 2^{e-p+1}$$

A variable v is set either by an external sensor or by an operation. Trailing digits of numbers randomly chosen from a logarithmic distribution [14, p. 254-264] are approximately uniformly distributed [15]. So we can assume that if v is a data obtained by an accurate sensor, the difference between v and the actual value $\overline{v}$ is uniformly distributed in the range $\pm\text{ulp}(v)/2$. We can model the error $v-\overline{v}$ by a random variable X with expectation $\mathbb{E}(X) = 0$ and variance $Var(X) = \text{ulp}(v)^2/12$. The sensor may be less accurate leading to a larger variance but we assume that it is not biased.

Round-off errors created by operators are discrete and they are not necessarily distributed uniformly [16]. The distribution is very specific but as soon as we verify that the expectation is $\mathbb{E}(X) = 0$ and the error is bounded, we may safely use a replacement variable X' such that

$$\forall\epsilon \quad \mathbb{P}(|X'| > \epsilon) \geq \mathbb{P}(|X| > \epsilon).$$

B. Error of an accumulation loop

We will use only one example for very long accumulations as calculations carried on at the end of Section IV heavily rely on the fact that all the errors introduced are uniformly distributed over the same interval. In other cases such as the second example of [10], one may either revert to the exponential-time formula stated in Section IV-A or use the magnitude of the largest possible error for all the random variables.

The example given in listing 1 sums data produced by a fixed point sensor x_i with a measure error X_i.

Listing 1. Simple discrete integration from [17]

```
1 a_0 = 0
2 for (i = 0; i < n; i = i + 1)
3     a_{i+1} = a_i + x_i
```

We can safely assume that X_i are independent identical uniformly distributed random variables over $\pm\text{ulp}(x_i)/2$. Data are fixed point meaning that the sum a_i+x_i does not introduce any rounding error and the weight of one unit in the last place does not depend on x_i so we write ulp instead of $\text{ulp}(x_i)$. After n iterations, we want the probability that the accumulated measure error have always been constrained into user specified bounds ϵ. Using the Doobs-Kolmogorov inequality [10] where $S_i = \sum_{j=1}^{i} X_j$, we have that

$$\mathbb{P}\left(\max_{1\leq i\leq n}(|S_i|) \leq \epsilon\right) \leq 1 - \frac{n\text{ulp}^2}{12\epsilon^2}.$$

We will see that we can exhibit a tighter bound using [18]. This bound can be produced using a time- and space-polynomial algorithm provided all the random variables are uniformly distributed over the same interval. Complexity is a key issue as typically $n \precsim 2^{25}$ meaning that the quadratic algorithm proposed Section IV should be considered as the biggest amenable algorithm.

III. Probability distribution of being safe

A. Probability

We presented in [10] an account of probability with an informal approach while taking foundational matters seriously. The PVS system underlying these results is built on the firm foundations for probability theory (using measure theory) [19], [20]. A middle way between extreme formality and an accessible level of informality is to be found in [21].

We begin by recalling in Figure 1 definitions implemented in PVS [10]. The PVS development of probability spaces in Figure 2, takes three parameters: T, the sample space, S, a σ-algebra of permitted events, and, $\mathbb{P}$, a probability measure, which assigns to each permitted event in S, a probability between 0 and 1. Properties of probability that are independent of the particular details of T, S and $\mathbb{P}$ are then provided in this file.

If T is countable – as it is for discrete random variables – then we may take $\sigma = \wp(T)$. As we wish to discuss continuous random variables then we partially instantiate this PVS file with `T = real`, and `S = borel_set` (the Borel sets). If we go further and also specify $\mathbb{P}$, we will have described the random variable distributions as well. Of particular interest later is the fact that the sum of two random variables is itself a random variable, and consequently any finite sum of random variables will be a random variable.

Note that the product probability $\mathbb{P}_3$ has the effect of declaring that the experiments carried out in probability spaces $(T_1, \sigma_1, \mathbb{P}_1)$ and $(T_2, \sigma_2, \mathbb{P}_2)$ are independent. Obviously, the process of forming products can be extended to any finite product of finitely many probability spaces.

In Figure 3, we define the conditional probability $\mathbb{P}(A;\ B)$ (written `P(A,B)` as PVS will not permit the use of ";" as an operator). We take the opportunity to prove Bayes' Theorem along the way.

B. Continuous Uniform Random Variables

If X is a continuous random variable distributed uniformly over the interval $[a, b]$, then informally it takes any value within the interval $[a, b]$ with equal probability.

To make this more formal, we define the *characteristic function* of a set S as the function χ_S, which takes the values 1 or 0 depending on whether it is applied to a member of S.

Definition 1:

$$\chi_S(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases}$$

Now the probability density function f of the uniform random variable over the closed interval $[a, b]$ is $\frac{1}{b-a}\chi_{(a,b]}$. From this we can calculate the distribution function:

$$F(x) = \int_{-\infty}^{x} f(x)dx,$$

from which we can calculate the probability

$$\mathbb{P}(x < X <= y) = F(y) - F(x).$$

```
probability_space[T:TYPE+,         (IMPORTING finite_measure@subset_algebra_def[T]) % sample space
                  S:sigma_algebra, (IMPORTING probability_measure[T,S])           % permitted events
                  P:probability_measure                                           % probability measure
                 ]: THEORY

BEGIN
   IMPORTING finite_measure@sigma_algebra[T,S],probability_measure[T,S],continuous_functions_aux[real]

   A,B: VAR (S)
   x,y: VAR real
   n0z: VAR nzreal
   t:   VAR T
   n:   VAR nat

   null?(A)         :bool = P(A) = 0
   non_null?(A)     :bool = NOT null?(A)
   independent?(A,B):bool = P(intersection(A,B)) = P(A) * P(B) % Note that it DOES NOT say = 0
   random_variable?(X:[T->real]):bool = FORALL x: member({t | X(t) <= x},S)
   zero: (random_variable?) = (LAMBDA t: 0)
   random_variable: TYPE+ = (random_variable?) CONTAINING zero

   X,Y: VAR random_variable
   XS:  VAR [nat->random_variable]

   <=(X,x):(S) = {t | X(t) <= x}; % Needed for syntax purposes! < > = /= >= omitted

   complement_le1: LEMMA complement(X <= x) = (x <  X)
   complement_lt1: LEMMA complement(x <  X) = (X <= x)
   complement_eq : LEMMA complement(X =  x) = (X /= x)
   complement_lt2: LEMMA complement(X <  x) = (x <= X)
   complement_le2: LEMMA complement(x <= X) = (X <  x)
   complement_ne:  LEMMA complement(X /= x) = (X =  x)

   -(X)    :random_variable = (LAMBDA t: -X(t)); % Needed for syntax purposes! + - * /  omitted

   +(X,Y)  :random_variable = (LAMBDA t: X(t) + Y(t));
   -(X,Y)  :random_variable = (LAMBDA t: X(t) - Y(t));

   partial_sum_is_random_variable:
     LEMMA random_variable?(LAMBDA t: sigma(0,n,LAMBDA n: XS(n)(t)))

   distribution_function?(F:[real->probability]):bool
                                    = EXISTS X: FORALL x: F(x) = P(X <= x)

   distribution_function: TYPE+ = (distribution_function?) CONTAINING
                                   (LAMBDA x: IF x < 0 THEN 0 ELSE 1 ENDIF)

   distribution_function(X)(x):probability = P(X <= x)

   F: VAR distribution_function

   convergence_in_distribution?(XS,X):bool
     = FORALL x: continuous(distribution_function(X),x) IMPLIES
                convergence((LAMBDA n: distribution_function(XS(n))(x)),
                                       distribution_function(X)(x))

   invert_distribution:   LEMMA LET F = distribution_function(X) IN
                                P(x < X) = 1 - F(x)                            % Lemma 2.1.11-a (G&S)
   interval_distribution: LEMMA LET F = distribution_function(X) IN
                                x <= y IMPLIES
                                P(intersection(x < X, X <= y)) = F(y) - F(x)   % Lemma 2.1.11-b (G&S)
   limit_distribution:    LEMMA LET F = distribution_function(X) IN
                                P(X = x) = F(x) - limit(LAMBDA n: F(x-1/(n+1))) % Lemma 2.1.11-c (G&S)

   distribution_0:                LEMMA convergence(F o (lambda (n:nat): -n),0) % Lemma 2.1.6-a0 (G&S)
   distribution_1:                LEMMA convergence(F,1)                        % Lemma 2.1.6-a1 (G&S)
   distribution_increasing:       LEMMA increasing?(F)                          % Lemma 2.1.6-b (G&S)
   distribution_right_continuous: LEMMA right_continuous(F)                     % Lemma 2.1.6-c (G&S)
END probability_space
```

Fig. 2. Abbreviated probability space file in PVS

```
conditional[T:TYPE+,              (IMPORTING finite_measure@subset_algebra_def[T]) % sample space
            S:sigma_algebra,      (IMPORTING probability_measure[T,S])           % permitted events
            P:probability_measure                                               % probability measure
           ]: THEORY

BEGIN

   IMPORTING probability_space[T,S,P],finite_measure@sigma_algebra[T,S]

   A,B:   VAR (S)
   n,i,j: VAR nat
   AA,BB: VAR disjoint_sequence

   P(A,B):probability = IF null?(B) THEN 0 ELSE P(intersection(A,B))/P(B) ENDIF

   conditional_complement: LEMMA
                  P(A,B) * P(B) + P(A,complement(B)) * P(complement(B)) = P(A)

   conditional_partition: LEMMA
           Union(image(BB,fullset[below[n+1]])) = fullset[T] IMPLIES
           P(A) = sigma(0,n, LAMBDA i: P(A, BB(i)) * P(BB(i)))

   bayes_theorem: THEOREM
           NOT null?(B) AND
           Union(image(AA,fullset[below[n+1]])) = fullset[T] IMPLIES
           P(AA(j),B) = P(B,AA(j))*P(AA(j))/
                        sigma(0,n, LAMBDA i: P(B, AA(i)) * P(AA(i)))

END conditional
```

Fig. 3. Conditional probability file in PVS

In the case where X is distributed $U_{[0,1]}$, and because – for any $f(x)$ with $\int f = F$ – we have

$$\int_{-\infty}^{\infty} f(x)\chi_{(a,b]}(x)dx = (F(x) - F(a))\chi_{(a,b]}(x) + (F(b) - F(a))\chi_{(b,\infty)}(x).$$

We also observe that if X is distributed $U_{[a,b]}$, then $\mathbb{E}(X) = \frac{a+b}{2}$, and $\text{Var}(X) = \frac{(a-b)^2}{12}$. So, with $a = 0$, $b = 1$ we get: $\mu = \frac{1}{2}$, $\sigma^2 = \frac{1}{12}$.

Definition 2: If we have a sequence of continuous random variables $\{X_n\}$, then we define their partial sums as a sequence of continuous random variables $\{S_n\}$ with the property

$$S_n = \sum_{i=1}^{n} X_i.$$

Theorem 1: If continuous random variables X and Y have joint probability density functions f, then $Z = X + Y$ has probability density function:

$$f_Z(z) = \int_{-\infty}^{\infty} f(x, z - x)dx.$$

In the special case where X and Y are independent, then (because the joint probability density function $f(x, y)$ can be expressed as the product $f_X(x)f_Y(y)$) we have the *Continuous Convolution Theorem*:

Theorem 2: If continuous random variables X and Y are independent and have probability density functions f_X and f_Y respectively, then $Z = X+Y$ has probability density function:

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x)f_Y(z-x)dx = \int_{-\infty}^{\infty} f_X(z-x)f_Y(x)dx.$$

C. Continuous random vectors

We want to generalize Theorem 2 in a probability space (T, σ, P) to a large number of random variables $\{X_n\}$ to produce a bound for each intermediate sum in $\{S_n\}$. We introduce random vectors.

Definition 3: A *random vector* of dimension n is a collection of n random variables

$$X = (X_1, \cdots, X_n)$$

In this definition, the components need not to be independent variables meaning that the information on the distribution function of the components does not fully characterize the probabilistic behavior of random vector X.

Definition 4: A random vector X has *distribution function* F, if

$$\mathbb{P}(X_1 \leq x_1, \cdots, X_n \leq x_n) = F(x_1, \cdots, x_n).$$

Once we have defined the distribution function of a random vector we want to define it's probability density.

Definition 5: A random vector X is *continuous* if its distribution function can be expressed as

$$F(x_1, \cdots, x_n) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f(x_1, \cdots, x_n)dx_1 \cdots dx_n$$

for some σ-integrable function $f : \mathbb{R}^n \to [0, \infty)$. We call the function f the probability density for random vector X.

We compute simple density functions through the following theorem that is proved by induction using the definition of independent random variables.

Theorem 3: A random vector X of continuous random variables $(X_1, \cdots, X_n)$ with respective density functions

- A random variable X has *distribution function* F, if $\mathbb{P}(X \leq x) = F(x)$
- A random variable X is *continuous* if its distribution function can be expressed as

$$F(x) = \int_{-\infty}^{x} f(x)dx$$

for some integrable function $f : \mathbb{R} \to [0, \infty)$. We call the function f the probability density function for the random variable X.
- We define the probability of "A given B" (written $\mathbb{P}(A;\ B)$) as:

$$\mathbb{P}(A;\ B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

whenever $\mathbb{P}(B) > 0$.
- A *σ-algebra* over a type T, is a subset of the power-set of T, which includes the empty set $\{\}$, and is closed under the operations of complement, countable union and countable intersection.
- A *Measurable Space* (T, σ) is a set (or in PVS a type) T, and a *σ-algebra* over T.
- A function $\mu : \sigma \to \mathbb{R}_{\geq 0}$ is a *Measure* over the σ-algebra σ, when $\mu(\{\}) = 0$, and for a sequence of disjoint elements $\{E_n\}$ of σ:

$$\mu\left(\bigcup_{n=0}^{\infty} E_n\right) = \sum_{n=0}^{\infty} \mu(E_n).$$

- A *Measure Space* (T, σ, μ) is a measurable space (T, σ) equipped with a measure μ.
- A *Probability Space* $(T, \sigma, \mathbb{P})$ is a measure space $(T, \sigma, \mathbb{P})$ in which the measure $\mathbb{P}$ is finite for any set in σ, and in which:

$$\mathbb{P}(X^c) = 1 - \mathbb{P}(X).$$

- If $(T_1, \sigma_1, \mathbb{P}_1)$ and $(T_2, \sigma_2, \mathbb{P}_2)$ are probability spaces then we can construct a *product probability space* $(T_3, \sigma_3, \mathbb{P}_3)$, where:

$$\begin{array}{lcl} T_3 & = & T_1 \times T_2 \\ \sigma_3 & = & \sigma(\sigma_1 \times \sigma_2) \\ \mathbb{P}'_3(a, b) & = & \mathbb{P}_1(a)\mathbb{P}_2(b) \end{array}$$

where $\mathbb{P}_3$ is the extension of $\mathbb{P}'_3$ that has the whole of σ_3 as its domain.

Fig. 1. Definitions implemented in PVS [10]

$f_1, \cdots, f_n$ is continuous and has a density function

$$f(x_1, \cdots, x_n) = f_1(x_1) \times \cdots \times f_n(x_n)$$

if and only if $X_1, \cdots, X_n$ are independent.

During the proof, we show by Fubini's theorem that for independent random variables

$$\mathbb{P}(X_1 \in A_1, \cdots, X_n \in A_n) = \mathbb{P}(X_1 \in A_1) \times \cdots \times \mathbb{P}(X_n \in A_n).$$

More complex density functions are handled through a smooth change of variables.

Theorem 4: Let $X = (X_1, \cdots, X_n)$ be a continuous random vector with density function f_X and g a transformation from $\mathbb{R}^n$ to $\mathbb{R}^n$ to a random vector $Y = (Y_1, \cdots, Y_n)$. Provided that

- g is continuously differentiable on an open subset U of $\mathbb{R}^n$,
- the Jacobian matrix

$$J_g = \left[\frac{\partial g_i}{\partial x_j}\right]_{1 \leq i,j \leq n}$$

is never singular,
- g admits an inverse on U with the same properties, in particular its Jacobian matrix is never singular and it is the inverse of the Jacobian matrix of g,
- and finally $P(X \in U) = 1$

then Y is a continuous random vector and with density function

$$f_Y(y) = f_X\left(g^{-1}(y)\right) \left| det J_g\left(g^{-1}(y)\right)\right|^{-1} \chi_V(y).$$

Notice that we obtain $\{S_n\}$ from $\{X_n\}$ through the following transformation:

$$g(x_1, \cdots, x_n) = \left(x_1, x_1 + x_2, \cdots, \sum_{i=1}^{n} x_i\right).$$

Its inverse g^{-1} on $U = \mathbb{R}^n$ is

$$g(y_1, \cdots, y_n) = \left(y_1, y_2 - y_1, \cdots, y_n - \sum_{i=1}^{n-1} y_i\right),$$

its Jacobian matrix is an upper triangular matrix with all components equal to 1 and its determinant is also equal to 1. Finally $V = \mathbb{R}^n$.

IV. Corners of hypercubes

What we are actually interested in is whether a series of calculations might accumulate a sufficiently large error to become meaningless. In the language we have developed, we are computing the probability that a sequence of n calculations has failed because it has exceeded the error-bound somewhere.

$$\mathbb{P}\left(\max_{1 \leq i \leq n}(|S_i|) \geq \epsilon\right)$$

A. Sums of Arbitrary Continuous Uniform Random Variables

For a little while, we assume that each random variable X_i is uniform on a given interval $[-\lambda_i, \lambda_i]$. Theorems 3 and 4 imply that the probability of failure is given by the ratio

$$\mathbb{P}\left(\max_{1\leq i\leq n}(|S_i|)\geq\epsilon\right)=\frac{V\left(P_n^{\mathcal{C}}\cap H_n\right)}{V(H_n)}=1-2^{-n}V(S_n)$$

where V is the volume operator applied to the following sets

$$P_n=\left\{(x_1,\dots x_n)\quad|\quad\forall i\quad|x_i|\leq 1\wedge\left|\sum_{j=1}^{i}\lambda_j x_j\right|\leq\epsilon\right\},$$

$$H_n=\left\{(x_1,\dots x_n)\quad|\quad\forall i\quad|x_i|\leq 1\right\},$$

and $P_n^{\mathcal{C}}$ is the complement of polyhedron P_n.

There is no direct formula to compute $V(P_n)$ due to linear dependencies between the constraints.

$$\left\{\begin{array}{rcl}|x_i|&\leq&1\\\left|\sum_{j=1}^{i}\lambda_j x_j\right|&\leq&\epsilon\end{array}\right.$$

The authors of this work and the authors of [18] agree that small or elementary perturbations are certainly the key to compute accurately $V(P_n)$ for small values of n. Yet perturbation are not amenable for large values of n.

So we have to revert to the bound $V(P_n)$ using the following sets with only one constraint which denotes no failure only at iteration n

$$Pt_i(\epsilon)=\left\{(x_1,\dots x_i)\quad|\quad\forall j\quad|x_j|\leq 1\wedge\left|\sum_{j=1}^{i}\lambda_j x_j\right|\leq\epsilon\right\}.$$

We could have used

$$P_n=\bigcap_{i=1}^{n}Pt_i(\epsilon)\times H_{n-i}$$

where $Pt_i(\epsilon)\times H_{n-i}$ is the Cartesian product of sets but we obtain a better bound with

$$P_n=\bigcap_{i=1}^{n}\left(Pt_i(\epsilon)\cup Pt_{i-1}^{\mathcal{C}}(\epsilon+\lambda_i)\times H_1\right)\times H_{n-i}$$

that is easily proved since

$$Pt_{i-1}(\epsilon)\subset Pt_{i-1}(\epsilon+\lambda_i)$$

with the convention that $Pt_0=\{\}$.

Using property of the complement and volume operators, we deduce that

$$V\left(P_n^{\mathcal{C}}\cap H_n\right)\leq\sum_{i=1}^{n}2^{n-i}\left(2V(Pt_{i-1}(\epsilon+\lambda_i))-V(Pt_i(\epsilon))\right)$$

The approximation is reduced due to the fact that

$$Pt_i(\epsilon)\subset Pt_{i-1}(\epsilon+\lambda_i)\times H_1$$

and so the volume of the set difference is the difference of the volumes of the sets with the convention that $V(\{\})=1$.

In the rest of the text, we heavily use notations and results of [18] to obtain $V(Pt_n(\mu))$ for appropriate n and μ (we change summation index from i to n for this reason). For a given vector

$$A=(a_1,a_2,\dots,a_n),$$

where $a_j=\lambda_j/\mu$, we define the $n\times(n+1)$ matrix $S=(I_n A)$ and we deduce $1\times(n+1)$ matrix

$$P=(1,a_1,a_2\dots a_n).$$

We want to compute $V(Pt_n(\mu))=\nu(P)=\frac{2^n}{\pi}\sigma(S)$ and we obtain $\sigma(S)$ by a formula that require the definition of

- $m=1$
- $I=\{\kappa\}$ where $\kappa=(1)$, $S_\kappa=(1)$ and $det(S_\kappa)=1$
- $\kappa^c=(2,3,\dots(n+1))$, $det(\kappa_j^c)=a_{j-1}$ and

$$\alpha_\kappa=\prod_{j=1}^{n}a_i^{-1}$$

- for $\gamma\in\Gamma$,

$$(s_\kappa,\gamma;1)=s_\gamma=1+\sum_{j=1}^{n}\gamma_{j+1}a_j,$$

The formula becomes

$$\nu(P)=\frac{1}{n!}\prod_{j=1}^{n}a_i^{-1}\sum_{\gamma\in\Gamma}\epsilon_\gamma s_\gamma^n\mathrm{sgn}s_\gamma$$

where all the quantities have been instantiated and the cardinal of Γ is 2^n (remember that $n\precsim 2^{25}$).

B. Sums of Independent Identical Continuous Uniformly Distributed Random Variables

In the rest of this section we assume that all the a_i are equal to a (and all the λ_i are equal to λ). For a $\gamma\in\Gamma$, we define n_γ as the number of positions where $\gamma_j=-1$ for j between 2 and $n+1$.

The formula becomes

$$\nu(P)=\frac{1}{a^n n!}\sum_{\gamma\in\Gamma}(-1)^{n_\gamma}(1+(n-2n_\gamma)a)^n\mathrm{sgn}(1+(n-2n_\gamma)a),$$

or equivalently

$$\nu(P)=\frac{1}{n!}\sum_{\gamma\in\Gamma}(-1)^{n_\gamma}(a^{-1}+n-2n_\gamma)^n\mathrm{sgn}(a^{-1}+n-2n_\gamma),$$

and we change the summation index $p=n-n_\gamma$ or $n-2n_\gamma=2p-n$ to obtain

$$\frac{1}{n!}\sum_{p=0}^{n}\left(\begin{array}{c}n\\p\end{array}\right)(-1)^p(n-a^{-1}-2p)^n\mathrm{sgn}(a^{-1}-n+2p).$$

with

$$\mathrm{sgn}(a^{-1}-n+2p)=\left\{\begin{array}{ccc}1&\text{if}&2p>n-a^{-1}\\0&\text{if}&2p=n-a^{-1}\\-1&\text{if}&2p<n-a^{-1}\end{array}\right.$$

We observe that ν is trivial when $na < 1$ and so we consider only the case where $n - a^{-1} \geq 0$. When $n = a^{-1}$, we obtain the volume of the full cube

$$\frac{1}{n!} \sum_{p=1}^{n} \left(\begin{array}{c} n \\ p \end{array} \right) (-1)^p (-2p)^n = 2^n$$

deriving a known formula [22].

So the probability of failure is bounded by

$$B_n(\epsilon, \lambda) = \sum_{i=1}^{n} 2^{-i} \left(2V_{i-1} \left(\frac{\epsilon}{\lambda} + 1 \right) - V_i \left(\frac{\epsilon}{\lambda} \right) \right)$$

with

$$V_i(\eta) = \frac{1}{i!} \sum_{p=0}^{i} \left(\begin{array}{c} i \\ p \end{array} \right) (-1)^p (i - \eta - 2p)^i \text{sgn}(\eta - i + 2p)$$

for $\eta \geq i$ and $V_i(\eta) = 2^i$ otherwise.

We obtain the equivalent formula below starting at the first index where $i > \epsilon/\lambda$

$$\begin{aligned} B_n(\epsilon, \lambda) &= 1 - 2^{-n} V_n \left(\frac{\epsilon}{\lambda} \right) \\ &+ \sum_{i=\frac{\epsilon}{\lambda}+1}^{n-1} 2^{-i} \left(V_i \left(\frac{\epsilon}{\lambda} + 1 \right) - V_i \left(\frac{\epsilon}{\lambda} \right) \right) \end{aligned}$$

V. Future work

At the time that we are submitting this work, results of Sections III-C and IV-A are not fully certified using PVS proof assistant. But we anticipate no problem has these results are generalizations to vector of our earlier work [10] and set theory except that our results are based on a formula obtained through Fourier Analysis. As Fourier Analysis has not been implemented in any proof checking environment, we are using the formula as a assumption to our theory. We hope that Fourier Analysis will be available in the future so we can remove any assumption from the theory we have developed. Yet certifying the results of [18] will be challenging.

This work will be continued in two directions. The first direction is to modify Fluctuat to generate theorems that can be checked automatically by PVS using ProofLite[1] as proposed in [5], [6]. This work will be carried in collaboration with the developers of Fluctuat within the EVA-Flo project of the ANR. The software will conservatively estimate the final effect of the errors introduced by each individual operations and compute upper bounds of their magnitudes.

The second direction is to develop and check accurate proofs about the errors of individual operations. A uniformly distributed random variable whose variance depends only on the operation and the computed result might provide a too pessimistic bound. For example the floating point addition of a large number with a small number absorbs the small number meaning that the round-off error may be far below half an ulp of the computed result.

Two's complement operation of the TMS320 circuit can either round or truncate the result. If truncation is used, it introduces a drift and our development cannot be used. Should we wish to extend this work to account for drifts (non-zero means for the random variables $\{X_n\}$), we will also have to consider higher-order error terms that also introduce a drift.

This library and future work will be included into NASA Langley PVS library[2] as soon as it becomes stable.

VI. Conclusions

To the best of our knowledge this paper presents the first application of the volume of the *corner* of a n-dimension hypercube to software reliability with n large enough for exponential-time solution to be impractical. In addition, we are finishing certification of our results with PVS. The major restriction lies in the fact that we have been forced to insist that individual errors have no drift, and are independent. Notice that even with a high tolerance of error, and with independent errors, we will still eventually fail. Our results permit the development of safe upper limits on the number of operations that a piece of numeric software should be permitted to undertake.

It is worth pointing out that violating our assumptions (independence of errors, and zero drift) would lead to worse results, so one should treat the limits we have deduced with caution, should these assumptions not be met.

Acknowledgment

This work has been partially funded by CNRS PICS 2533 and by the EVA-Flo project of the ANR. It was initiated while one of the authors was an invited professor at the University of Perpignan Via Domitia. It benefits from links between the cole Normale Suprieure de Lyon where one author used to work and the University of Manchester started in the Mathlogaps multi-participant Early Stage Research Training network of the European Union. The authors would like to thanks Philippe Langlois, Harold Simmons and Jean-Marc Vincent for fruitful informal discussions on this work and Rina Srabonian Williams for his help in the library.

References

[1] S. C. Johnson and R. W. Butler, "Design for validation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 7, no. 1, pp. 38–43, 1992. [Online]. Available: http://dx.doi.org/10.1109/62.127129

[2] D. M. Russinoff, "A mechanically checked proof of IEEE compliance of the floating point multiplication, division and square root algorithms of the AMD-K7 processor," *LMS Journal of Computation and Mathematics*, vol. 1, pp. 148–200, 1998. [Online]. Available: http://www.onr.com/user/russ/david/k7-div-sqrt.ps

[3] J. Harrison, "Formal verification of floating point trigonometric functions," in *Proceedings of the Third International Conference on Formal Methods in Computer-Aided Design*, W. A. Hunt and S. D. Johnson, Eds., Austin, Texas, 2000, pp. 217–233. [Online]. Available: http://www.springerlink.com/link.asp?id=wxvaqu9wjrgc8l99

[4] S. Boldo and M. Daumas, "Representable correcting terms for possibly underflowing floating point operations," in *Proceedings of the 16th Symposium on Computer Arithmetic*, J.-C. Bajard and M. Schulte, Eds., Santiago de Compostela, Spain, 2003, pp. 79–86. [Online]. Available: http://perso.ens-lyon.fr/marc.daumas/SoftArith/BolDau03.pdf

[1] `http://research.nianet.org/~munoz/ProofLite/`.

[2] `http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html`.

[5] M. Daumas, G. Melquiond, and C. Muñoz, "Guaranteed proofs using interval arithmetic," in *Proceedings of the 17th Symposium on Computer Arithmetic*, P. Montuschi and E. Schwarz, Eds., Cape Cod, Massachusetts, 2005, pp. 188–195. [Online]. Available: http://perso.ens-lyon.fr/marc.daumas/SoftArith/DauMelMun05.pdf

[6] C. Muñoz and D. Lester, "Real number calculations and theorem proving," in *18th International Conference on Theorem Proving in Higher Order Logics*, Oxford, England, 2005, pp. 239–254. [Online]. Available: http://dx.doi.org/10.1007/11541868_13

[7] M. J. C. Gordon and T. F. Melham, Eds., *Introduction to HOL: A theorem proving environment for higher order logic*. Cambridge University Press, 1993.

[8] G. Huet, G. Kahn, and C. Paulin-Mohring, *The Coq proof assistant: a tutorial: version 8.0*, 2004. [Online]. Available: ftp://ftp.inria.fr/INRIA/coq/current/doc/Tutorial.pdf.gz

[9] S. Owre, J. M. Rushby, and N. Shankar, "PVS: a prototype verification system," in *11th International Conference on Automated Deduction*, D. Kapur, Ed. Saratoga, New-York: Springer-Verlag, 1992, pp. 748–752. [Online]. Available: http://pvs.csl.sri.com/papers/cade92-pvs/cade92-pvs.ps

[10] M. Daumas and D. Lester, "Stochastic formal methods: an application to accuracy of numeric software," in *Proceedings of the IEEE 40 Annual Hawaii International Conference on System Sciences*, Waikoloa, Hawaii, 2007. [Online]. Available: http://hal.ccsd.cnrs.fr/ccsd-00081413

[11] D. Goldberg, "What every computer scientist should know about floating point arithmetic," *ACM Computing Surveys*, vol. 23, no. 1, pp. 5–47, 1991. [Online]. Available: http://doi.acm.org/10.1145/103162.103163

[12] D. Stevenson *et al.*, "An American national standard: IEEE standard for binary floating point arithmetic," *ACM SIGPLAN Notices*, vol. 22, no. 2, pp. 9–25, 1987.

[13] *TMS320C3x — User's guide*, Texas Instruments, 1997. [Online]. Available: http://www-s.ti.com/sc/psheets/spru031e/spru031e.pdf

[14] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*. Addison-Wesley, 1997, third edition.

[15] A. Feldstein and R. Goodman, "Convergence estimates for the distribution of trailing digits," *Journal of the ACM*, vol. 23, no. 2, pp. 287–297, 1976. [Online]. Available: http://doi.acm.org/10.1145/321941.321948

[16] J. Bustoz, A. Feldstein, R. Goodman, and S. Linnainmaa, "Improved trailing digits estimates applied to optimal computer arithmetic," *Journal of the ACM*, vol. 26, no. 4, pp. 716 – 730, 1979. [Online]. Available: http://doi.acm.org/10.1145/322154.322162

[17] N. Brisebarre, M. Daumas, P. Langlois, and M. Martel, "Survol et limitations des modles discrets-continus pour la sret numrique," Centre pour la Communication Scientifique et Directe, Villeurbanne, France, Tech. Rep., 2006.

[18] D. Borwein, J. M. Borwein, and B. A. Mares Jr., "Multivariable sinc integrals and volumes of polyhedra," *The Ramanujan Journal*, vol. 6, no. 2, pp. 189–208, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1015727317007

[19] P. R. Halmos, "The foundations of probability," *American Mathematical Monthly*, vol. 51, pp. 493–510, 1944.

[20] ——, *Measure Theory*. Van Nostrand Reinhold, 1950.

[21] *Probability and Random Processes*. Oxford University Press, 1982.

[22] S. M. Ruiz, "An algebraic identity leading to wilson's theorem," *The Mathematical Gazette*, vol. 80, no. 489, pp. 579–583, 1996. [Online]. Available: http://arxiv.org/abs/math.GM/0406086