Stochastic Formal Methods: An application to accuracy of numeric software

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 a bound on the number of numeric operations (fixed or floating point) that can safely be performed before accuracy is lost. 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. We present here a set of formal theorems validated by the PVS proof assistant. These theorems will allow code analyzing tools to produce formal certificates of accurate behavior. For example, FAA regulations for aircraft require that the probability of an error be below 10^{-9} for a 10 hour flight [1].**

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 is meaningless for applications that run for a long time. For example, a process adds numbers in ± 1 to single precision, and therefore has a round-off 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 round-off errors are not correlated, the actual accumulated error will be much smaller.

Developments in probability share many features with developments in floating point arithmetic:

1) Each result usually relies on a long list of hypotheses. No hypothesis can be removed, but slight variations induce a large number of results that look almost identical.
2) Most people that use the results are not specialists in the specific field. They want a trustworthy result but they are not proficient enough to either select the best scheme or detect minor faults that can quickly lead to huge problems.

For these reasons, we are strongly of the opinion that validation of a safety-critical numeric software using probability should be done using an automatic proof checker. We present in Section II the model that we are using. Section III presents our formal developments in probability. The Doobs-Kolmogorov inequality provides an effective way to compute the probability that a piece of software will successfully run within an acceptable error bound.

This work is connected to continuous space Markov random walks or renewal-reward processes though these applications focus on asymptotic behavior [10], [11]. We want to precisely bound the probability of remaining within bounds for a given number of steps. This is connected to ruin probabilities [12] and the Doobs-Kolmogorov inequality for martingales [13]. 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

A. Individual round-off errors of fixed and floating point operations

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 [14]. IEEE 754 standard [15] uses sign-magnitude notation for the mantissa and the first bit of the mantissa is implicit in most cases leading to the following definition where s and all the b_i are either 0 or 1 (bits).

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

Some circuits such as TMS320 uses two's complement notation for m leading to the following definition [16].

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

For both notations, we define for any representable number x, the unit in the last place function where e is the exponent of x as above. In fixed point notation, e is a constant provided by the data type.

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

A variable v is set either by an external sensor or by an operation. Trailing digits of numbers randomly chosen from a logarithmic distribution [17, p. 254-264] are approximately uniformly distributed [18]. So we can assume that if v is a

Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07)
0-7695-2755-8/07 $20.00 © 2007 IEEE

1530-1605/07 $20.00 © 2007 IEEE

data obtained by an accurate sensor, the difference between v and the actual value $\overline{v}$ is uniformly distributed in the range $\pm$ulp$(v)/2$. We can model the representation error $v - \overline{v}$ by a random variable X with expectation $\mathbb{E}(X) = 0$ and variance $\mathbb{E}(X^2) = \text{ulp}(v)^2/12$. The sensor may be less accurate leading to a larger variance but it should not be biased.

Round-off errors created by operators are discrete and they are not necessarily distributed uniformly [19]. For each operator $\circledast$ implementing the real operation $*$, we define

$$X = V \circledast W - V * W$$

where V and W are number distributed logarithmically over specified ranges. The distribution of X is very specific but we verify that the expectation is $\mathbb{E}(X) = 0$ and we bound its variance $\mathbb{E}(X^2)$.

Fixed point additions do not create any additional round-off error provided its output is in the same format as its inputs. Reducing the format of a fixed point number creates a uniformly distributed round off error provided the input was logarithmically distributed [18].

B. Round off errors of an accumulation loop

We will use two examples. The first one is given in listing 1. It sums data produced by a fixed point sensor x_i with a measure error X_i.

Listing 1. Simple discrete integration from [20]

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

We can safely assume that X_i are independent identical uniformly distributed random variables over $\pm$ulp$(x_i)/2$. Data are fixed point meaning that the sum $a_i + x_i$ does not introduce any rounding error and the weigth of one unit in the last place does not depend on x_i so we write ulp instead of 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 of Theorem 3 where $S_i = \sum_{j=1}^{i} X_j$, we have that

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

The second example is given in listing 2. It solves initial value problem (IVP) ordinary differential equations (ODE) by computing an incremental slope $\Phi(t_i, h_i, x_i, f)$ based on the current time t_i, the current step size h_i, the current value of the function x_i and the differential equation $x'(t) = f(t, x(t))$. The function Φ may be very simple using Euler's explicit method or more complex using any Runge-Kutta method or any implicit method. We focus here on scalar ODEs although our analysis may apply to vectors. Line 4 assume for the sake of simplicity that h_i is a constant although this is not neccessary.

Listing 2. Solving initial value problem ordinary differential equations [21]

```
1 for ($i = 0$; $i < n$; $i = i + 1$) {
2     $x_{i+1} = x_i + h_i \times \Phi(t_i, x_i, h_i, f)$
3     $t_{i+1} = t_i + h_i$
4     $h_{i+1} = h_i$
5 }
```

Our first guess was to introduce a sequence of random variables $\{X_n\}$ that models the difference introduced by round-off errors at step i. In most cases, Φ introduces a drift due to higher order effect of random variables and a drifted correlation between the error introduced at step $i+1$ and errors on the previous steps. For example, the square of a rounded value $v + V$ where v is the stored value and V is a random variable, introduces a positive drift due to V^2 term that is always positive. So we model the effect of the round-off error by two terms X_i and Y_i. We use the Doobs-Kolomogorov inequality of Theorem 3 for the sequence $\{X_n\}$ and worst case error analysis for the sequence $\{Y_n\}$ setting the following conditional expectation

$$\mathbb{E}(X_n; X_1 \cdots X_{n-1}) = 0.$$

Random variables X_{i+1} and Y_{i+1} account for the round-off and propagated errors introduced by replacing

$$x_i + X_i + Y_i + h_i \times \Phi(t_i, x_i + X_i + Y_i, h_i, f)$$

with

$$x_i \oplus h_i \otimes \tilde{\Phi}(t_i, x_i, h_i, f)$$

where $\tilde{\Phi}$ is evalaution of Φ in computer. First order effect of round-off errors created are accounted in X_{i+1}. Higher order effect of round-off errors created and propagated effect of X_i and Y_i in Φ are accounted in Y_{i+1}.

$\{X_n\}$ is constructed to contain only independent random variables with no drift $\mathbb{E}(X_i) = 0$ and we only need to bound their variance $\mathbb{E}(X_i^2)$. We will do worst case analysis on $\{Y_n\}$ and we bound each Y_i with interval arithmetic [22]. Software such as Fluctuat [23] is already able to distinguish between first order and higher order error terms.

III. Probability distribution of being safe

A. Probability

We have two main choices in presenting an account of probability: one is to take an informal approach, the second involves taking foundational matters seriously. In this paper we will consistently try to present matters informally except for Section III-B, however the reader should be aware that the PVS system underlying these results is built on the firm foundations for probability theory (using measure theory) [24], [25]. A middle way between extreme formality and an accessible level of informality is to be found in [13].

We begin by defining the *distribution function* of a random variable.

Definition 1: A random variable X has *distribution function F*, if $\mathbb{P}(X \le x) = F(x)$

As we will be studying *continuous random variables*, these are defined as follows:

Definition 2: 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.

As an example of a continuous random variable, consider the temperature T at a certain point in an industrial process. Even if an attempt is being made to hold this temperature constant, there will be minor fluctuations, and these can be modeled mathematically by assuming that T is a continuous random variable.

The other concept we will need is that of dependent and independent random variables. Suppose we model the outcomes of the tossing of two coins C_1 and C_2 by random variables. Provided there is nothing underhand going on, we would expect the result of tossing the first coin to have no effect on the result of the second coin, and *vice versa*. If this is the case, then we say that C_1 and C_2 are *independent*. Consider an alternative scenario in which having tossed the coin C_1 and discovered that it has come up "heads", and we now define the random variable C_2 to be: the outcome: "the downward facing side of the coin C_1 is tails". In this case the random variables C_1 and C_2 are *dependent*.

The other idea we must address is that of *conditional probability*.

Definition 3: 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 $\mathbb{P}(B) > 0$.

As an example: if event A is "I am carrying an umbrella" and event B is "it is raining", then $Pr(A;\ B)$ is the probability that "I am carrying an umbrella given that it is raining". Note that although in general $\mathbb{P}(A;\ B) \neq \mathbb{P}(B;\ A)$, in this particular case, if you live in Perpignan or Manchester, then on most days: $\mathbb{P}(A;\ B) = \mathbb{P}(B;\ A)$, though for rather different reasons.

B. A Formal Development of probability

Definition 4: 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.

If T is countable – as it is for discrete random variables – then we may take $\sigma = \wp(T)$; if the set T is the reals – as it is for continuous random variables – then we make $\sigma = B$: the Borel sets.

Definition 5: A *Measurable Space* (T, σ) is a set (or in PVS a type) T, and a *σ-algebra* over T.

Definition 6: 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:

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

Definition 7: A *Measure Space* (T, σ, μ) is a measurable space (T, σ) equipped with a measure μ.

Definition 8: 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:

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

The PVS development of probability spaces in Figure 1, 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 we wished to discuss continuous random variables then we would partially instantiate this PVS file with `T = real`, and `S = borel_set`. 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.

Definition 9: 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.

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. Currently, it is not clear whether PVS is powerful enough to capture the notion of a countably infinite sequence of random variables $\{X_n\}_{n=1}^{\infty}$; fortunately, in this work we don't currently require this result.

In Figure 2, 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.

C. 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.

```
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. 1.   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. 2. Conditional probability file in PVS

Definition 10:

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

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

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

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 $\mathrm{Var}(X) = \frac{(a-b)^2}{12}$. So, with $a = 0$, $b = 1$ we get: $\mu = \frac{1}{2}$, $\sigma^2 = \frac{1}{12}$.

D. Sums of Continuous Random Variables

Definition 11: 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:

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

E. Reliability of long calculations

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 asking what is the probability that all calculations of length upto n is correct:

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

Because they have nice convergence properties, we are especially interested in *martingales*

Definition 12: A sequence $\{S_n\}$ is a *martingale* with respect to the sequence $\{X_n\}$, if for all n:

1) $\mathbb{E}(|S_n|) < \infty$; and
2) $\mathbb{E}(S_{n+1};\ X_1,\ X_2,\ \ldots,\ X_n) = S_n$

We first observe that the sequence $S_n = \Sigma_{i=1}^{n} X_i$ (as previously defined) is a martingale with respect to the sequence $\{X_n\}$.

Lemma 1: The sequence $\{S_n\}$, where $S_n = \sum_{i=1}^{n} X_i$, and each X_n is an independent random variable with $\mathbb{E}(X_n) = 0$, is martingale with respect to the sequence $\{X_n\}$.

Alternatively as could be needed for program 2:

Lemma 2: The sequence $\{S_n\}$, where $S_n = \sum_{i=1}^{n} X_i$, and $\{X_n\}$ satisfies for all i

$$\begin{aligned} \mathbb{E}(X_i) &= 0 \\ \mathbb{E}(X_i;\ X_1 \cdots X_{i-1}) &= 0, \end{aligned}$$

the sequence $\{S_n\}$ is martingale with respect to the sequence $\{X_n\}$.

We now make use of the Doobs-Kolmogorov Inequality presented Figure 3. The statement of Theorem 3 is deceptively simple. The key as the astute reader will observe is that we have a restricted form of the Doobs-Kolmogorov Inequality in which the sample spaces of the underlying sequence of random variables are identical. This is an artifact of the PVS type system which would require us to prove multiple version of the theorem at each tuple of instantiated types.

Although the type system used in PVS is extraordinarily flexible, it is not as malleable as that used by professional mathematicians. To capture mathematics in its entirety using a theorem prover, we would need to dispense with any form of type checking[1]. For its intended use as an aide to proving programs correct, this would fatally weaken PVS as a useful tool. In addition, in many practice areas of mathematics, the full generality of categorical constructs is an unnecessary luxury, albeit one with a seductive, siren-like, appeal.

Theorem 3 (Doobs-Kolmogorov Inequality): If $\{S_n\}$ is a martingale with respect to $\{X_n\}$ then, provided that $\epsilon > 0$:

$$\mathbb{P}\left(\max_{1\le i\le n}(|S_i|) \ge \epsilon\right) \le \frac{1}{\epsilon^2}\mathbb{E}(S_n^2)$$

In our particular case where each X_i is an independent random variable with $\mathbb{E}(X_i) = 0$, and $\mathrm{Var}(X_i) = \sigma_i^2$, we observe that

$$\mathbb{P}\left(\max_{1\le i\le n}(|S_i|) \le \epsilon\right) \ge 1 - \frac{1}{\epsilon^2}\sum_{i=1}^{n}\sigma_i^2$$

The short conclusion is therefore that eventually errors will accumulate and overwhelm the accuracy of any numerical software. However, if ϵ is large enough and each of the σ_i^2 are small enough, then the number of iterations required for this to occur will be high enough to be of no practical significance. Crucially, the results hinge critically on the errors $\{X_n\}$ being independent.

IV. Future work

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[2] as proposed in [5], [6]. This work will be carried in collaboration with the developers of Fluctuat. The software will conservatively estimate the final effect of the error introduces by each individual floating point operations and compute upper bounds of their variances.

The second direction is to develop and check accurate proofs about the round-off errors of individual equations. 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 TMS320 circuit can either round or truncate the result. If truncation is used, it introduces a drift and Doobs-Kolmogorov inequality for martingales cannot be used. Should we wish to extend this work to account for drifts (non-zero means for the random variables $\{X_n\}$), then we anticipate making use of Wald Identity. Such developments will also be necessary to address higher order error terms that introduce a drift.

This library and future work will be included into NASA Langley PVS library[3] as soon as it becomes stable.

We saw with the example of listing 2 that inductions on the variances of the random variables can be crudely bounded. Yet, we may expect tighter results if we use tools that are able to infer inductions and solve them mathematically but this domain is far from the authors' research areas.

V. Conclusions

To the best of our knowledge this paper presents the first application of the Doobs-Kolmogorov Inequality to software reliability and the first generic formal developpment able to handle continuous, discrete and non-continuous non-discrete random variable in higher order logic proof assistants. In addition, we have demonstrated a slightly weaker version of this result in PVS. We claim that the utility of this weaker result is not unduly restrictive, when compared to the general result. The major restriction lies in the fact that we have to demonstrate that a sequence of overall errors is martingale with respect to the sequence of individual errors. We have been forced to make simplifications to the mathematical model of our software to ensure that this is the case. In particular, we have been forced to insist that our individual errors have no drift, and are independent.

We have been surprised that the limit on the reliability of a piece of numeric software could be expressed so succinctly. 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

[1] A weak form of type consistency is used in category theory, but this is so weak that we can introduce the Russel Paradox.

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

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

```
doobs[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[T,S,P],martingale,reals@bounded_reals

  epsilon: VAR posreal
  X,S:     VAR [nat -> random_variable]
  pn:      VAR posnat

  doobs_kolmogorov: THEOREM martingale?(X,S) IMPLIES
     P(max(image(abs o S,below(pn))) >= epsilon)
        <= E(sq(S(pn)))/sq(epsilon)

END doobs
```

Fig. 3. Doobs-Kolmogorov inequality in PVS

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 was partially done while one of the authors was an invited professor at the University of Perpignan Via Domitia. The authors would like to thanks Philippe Langlois and Jean-Marc Vincent for fruitful informal discussions on this work.

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

[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] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1998.

[11] C.-D. Fuh, "Uniform Markov renewal theory and ruin probabilities in Markov random walks," *Annals of Applied Probability*, vol. 14, no. 3, pp. 1202–1241, 2004. [Online]. Available: http://dx.doi.org/10.1214/105051604000000260

[12] S. Asmussen, *Ruin Probabilities*. World Scientific, 2000.

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

[14] 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

[15] 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.

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

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

[18] 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

[19] 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

[20] N. Brisebarre, M. Daumas, P. Langlois, and M. Martel, "Survol et limitations des modèles discrets-continus pour la sûreté numérique," Centre pour la Communication Scientifique et Directe, Villeurbanne, France, Tech. Rep., 2006.

[21] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*. Springer Verlag, 2002. [Online]. Available: http://www.springer.de/cgi/svcat/search_book.pl?isbn=0-387-95452-X

[22] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis*. Springer, 2001. [Online]. Available: http://www.springeronline.com/sgw/cda/frontpage/0,10735,5-40106-22-2093571-0,00.html

[23] E. Goubault, M. Martel, and S. Putot, "Some future challenges in the validation of control systems," in *European Congress on Embedded Real Time Software*, Toulouse, France, 2006. [Online]. Available: http://www.enseignement.polytechnique.fr/profs/informatique/Matthieu.Martel/erts.pdf

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

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