ROUNDING ERROR USING LOW PRECISION APPROXIMATE RANDOM VARIABLES

MICHAEL GILES* AND OLIVER SHERIDAN-METHVEN†

Abstract. For numerical approximations to stochastic differential equations using the Euler-Maruyama scheme, we propose incorporating approximate random variables computed using low precisions, such as single and half precision. We propose and justify a model for the rounding error incurred, and produce an average case error bound for two and four way differences, appropriate for regular and nested multilevel Monte Carlo estimations. Our rounding error model recovers and extends the statistical model by Arciniega and Allen [1], while bounding the size systematic and biased rounding errors are permitted to be. By considering the variance structure of multilevel Monte Carlo correction terms in various precisions with and without a Kahan compensated summation, we compute the potential speed ups offered from the various precisions. We find single precision offers the potential for approximate speed improvements by a factor of 7 across a wide span of discretisation levels. Half precision offers comparable improvements for several levels of coarse simulations, and even offers improvements by a factor of 10–12 for the very coarsest few levels, which are likely to dominate higher order methods such as the Milstein scheme.

Key words. approximations, random variables, inverse cumulative distribution functions, random number generation, finite precision, half precision, floating point, rounding error, multilevel Monte Carlo, the Euler-Maruyama scheme, the Milstein scheme, Kahan compensated summation, and high performance computing.

AMS subject classifications. 65G50, 65C10, 41A10, 65C05, 65Y20, 60H35, 65B10, 65L70, 34M30, 97N20, and 65C30.

1. Introduction.

Rounding error has long been a source of scientific interest and frustration. For a large fraction of the scientific community, the effects of rounding error are usually negligible and of little or no consequence. However, for an appreciable portion of the community, especially those pushing computer hardwares and numerical algorithms to the fastest speeds achievable, rounding error can present a considerable hurdle to the achievable fidelity and speed.

The example *par excellence* of rounding error in scientific computing is in summation operations, frequent in linear algebra applications and to a lesser extent some statistical applications. Calculating the scalar product between two vectors, and thus also calculating vector and matrix multiplications, involves (among other things) summing a list of numbers. Being able to accurately sum a list of numbers has long been under the attention of mathematicians and computer scientists [21, 29, 33, 50], and its importance in scientific computing cannot be understated. When the numbers being summed are ill conditioned, the impact of rounding error grows with the problem's size, and can quickly nullify even the simplest of calculations, and thus high accuracy summation algorithms are frequently required. Outside of linear algebra, the accuracy of gradients and sensitivity estimates from finite difference methods and numerical differentiation is capped by the maximum available precision due to rounding error. Lastly, for the numerical solution of differential equations by simulation methods (deterministic or stochastic), iterative methods such as the Euler scheme incur rounding errors which get worse as the simulation's discretisation becomes finer.

To combat the effects of rounding error, there are two particularly common approaches. The first is to try and bypass the issue by simply working in a higher precision, typically at the cost of computational speed. Historically this has motivated

*mike.giles@maths.ox.ac.uk
†oliver.sheridan-methven@hotmail.co.uk

the introduction of double precision, extended double precision, and even quadruple precision data types. Similarly, several software libraries offer arbitrary levels of precision, such as: the mpmath [28] Python library, the GNU multiple precision (GMP) arithmetic C/C++ library [20], and the GNU multiple precision binary floating point with correct rounding (MPFR) C library [12]. In a similar vein, hardware providers have also focussed efforts on improved floating point accuracy and reproducibility [6].

The second approach to reduce the influence of rounding error is to try and compensate and correct against it. For summations, the best known approach is the Kahan compensated summation [29], although other compensation procedures have also been introduced and well explored [2, 10, 31, 33, 34, 38, 40, 41]. These proceed by inferring an estimate for the rounding error introduced at each stage of the summation, and then discount for this in the subsequent summations, thus compensating for the rounding error.

There is a third means of circumventing rounding error, which is more mathematical in its nature, which looks to extrapolate accurate answers from less accurate approximations. The best example of this is Richardson extrapolation [35, 44]. Without this technique, approximation schemes would need to go to such fine discretisations that rounding error would be significant, whereas using Richardson extrapolation is one possible technique to avoid encountering rounding error. However, the use of Richardson extrapolation is very problem specific, and whilst it is a very powerful mathematical technique, it does not readily present itself as a general purpose computational tool for avoiding rounding error in most circumstances.

Both high precision libraries and rounding error compensation schemes have historically been reserved for specialised applications seeking extraordinarily high accuracy, and closer to the edges of most scientific computing applications. However, in more recent years there has been an increased demand for ever lower precisions and data types, such as the IEEE half precision float [26], and the more recent "brain float" [7, 30]. The large driving force behind these is the increasingly popular demand in machine learning applications, where the underlying data are very noisy and imprecise, and smaller data types are preferable for faster access, storage, and computation on the latest CPU, GPU, and TPU hardwares. Furthermore, with the greater desire for increased parallelisation on vector hardware and reduced precision calculations, lower precision data types are gaining considerable momentum and traction.

To understand the nature of the nett rounding error arising in calculations, there have been two fronts of development. The first has been defining the precise rounding modes and data types used in scientific calculations. To ensure floating point calculations were standardised, the famous IEEE 754 standard for floating point arithmetic was introduced [25], and is now the *de facto* industry standard. This entails addition, subtraction, multiplication, division, and square roots all producing exact results with respect to the appropriate rounding mode [51, page 15]. Similarly, the rounding modes a computer uses to round floating point values are standardised, with "round to nearest even" typically being the default mode. Of course, while floating point arithmetic may be well defined, there are several difficulties and nuances, as discussed by Goldberg [19].

The second front has been with the mathematical modelling of rounding errors. While the IEEE 754 standard specifies the hardware's behaviour, this does not readily give insight into the behaviour of the emergent nett rounding error. Describing the nett effect that results during calculations has received much mathematical attention [22, 24, 53, 54, 55], and one of the best overviews is by Higham [22], who analyses the standard model for deterministic rounding error [22, 2.2, (2.4)]. Furthermore,

in recent years there has been a piqued interest in stochastic rounding modes and associated error models, with prominent recent work by Higham and Mary [23] and Ipsen and Zhou [27] performing probabilistic error analyses, which frequently give tighter and more realistic error bounds than the worst case deterministic scenarios. In the setting of analysing partial differential equations using stochastic rounding there is the recent work by Croci and Giles [9].

With this wealth of attention from academia and industry, the work we present produces a rigorous model for the rounding error incurred during the numerical simulation of stochastic differential equations. Typical treatments of numerical methods for such stochastic differential equations assume no rounding error occurs or is otherwise negligible [32, 9.3, page 316] [18]. The earliest work to compensate for rounding error in simulations for ordinary differential equations appears to be by Vitasek [52]. In the setting of stochastic differential equations, the most relevant works are by Arciniega and Allen [1] and Omland [42]. Arciniega and Allen [1] present an *ad hoc* statistically motivated model for the rounding error which occurs in the Euler-Maruyama scheme, giving an average case bound for the overall rounding error. Omland [42] takes a more rigorous approach, closer to a first principles model, starting with the floating point rounding modes and standard error model by Higham [22], and produces a worst case bound for the error in the Euler-Maruyama scheme [42, theorem 4.8]. Related to modelling rounding error within stochastic simulations is work by Delattre and Jacod [11] and Rosenbaum [45], statistically modelling the effects of simple rounding models on statistical inference problems driven by underlying stochastic processes. They analyse the convergence of empirical statistical estimators for functionals of certain stochastic processes typical in finance, estimating parameters occurring in diffusion coefficients using empirical samples.

The contribution of this work will be to produce a heuristic model for the rounding error in a similar manner to Arciniega and Allen [1]. However, our model will be much more rigorously justified by a detailed inspection of the dominant rounding errors arising in the Euler-Maruyama scheme. Furthermore, we will show that there are two primary sources of error which contribute to the nett error. The first is a zero mean process, similar to that described by Arciniega and Allen [1], and any deviation from being zero mean is comprehensively analysed, quantified, and shown to be negligible compared to the other error processes occurring. The second is a possibly non zero mean systematic error term omitted by Arciniega and Allen [1]. This second process is of a much smaller size than the first, but due to its non zero mean nature, we show that its nett contribution will grow at the same rate as that arising from the zero mean process.

The significance of this new model is that it both rigorously grounds the leading order error process as being zero mean and now additionally quantifies the permissible size of lower order systematic rounding errors in the Euler-Maruyama scheme. Furthermore, our model is amenable to incorporation within the nested multilevel Monte Carlo scheme utilising approximate random variables developed by Giles and Sheridan-Methven [15, 46, 47, 49]. Although work has been done by Brugger et al. [5] and Omland et al. [43] on constructing multilevel Monte Carlo schemes in the presence of rounding error, our model directly facilitates a treatment jointly allowing for approximate random variables and low precision calculations, only correctly handled by a nested multilevel Monte Carlo scheme.

A secondary contribution of this work will also be to demonstrate the applicability of a Kahan compensated summation within the Euler-Maruyama scheme, an extension of the similar idea by Vitasek [52] in the setting of ordinary differential equations. The

inclusion of Kahan compensated summation we show considerably increases the range of applicability of half precision to many more discretisation levels beyond just the coarsest few.

The combination of these two contributions should provide considerable interest to both theoreticians, but crucially also practitioners working in high performance scientific computing. Presenting concrete mathematical underpinnings for modelling the effect of rounding errors, our work newly describes the impact of both leading order zero mean and second order systematic error processes on the nett rounding error that arises. Being able to bound the permissible size of systematic processes is a novel result now available to those wishing to model or construct further numerical schemes building off this work. Additionally, integrating these results in the nested multilevel Monte Carlo framework and demonstrating the utility of half precision and Kahan compensated summation opens up our work to practitioners. The empirical results demonstrate the substantial benefits to be had by adopting our framework and heavily engaging with low precision floating point formats such as half precision, whilst our analytic result give the necessary reassurance and confidence that accuracy can be all the while maintained, and need not be sacrificed.

Section 2 overviews the numerical solution of stochastic differential equations, providing the primary context and setting of our work, presenting our model for the leading order error process arising in the Euler-Maruyama scheme. Section 3 will showcase how our model can be incorporated into a multilevel Monte Carlo framework, demonstrating practical applications of the model and highlighting the temporal savings that can be expected using low precisions. Section 4 presents the conclusions from this work.

2. Numerical solutions to stochastic differential equations. There are various settings appropriate for analysing the effects of rounding error, and the numerical solutions of stochastic differential equations is one particularly important setting. Frequently the terminal solution X_T of the stochastic differential equation $\mathrm{d}X_t = a(t, X_t)\,\mathrm{d}t + b(t, X_t)\,\mathrm{d}W_t$ needs to be approximated for given drift and diffusion processes a and b. To achieve this, whole path approximations $\widehat{X}_t \approx X_t$ for $t \in [0, T]$ are produced, where the most popular methods are the Euler-Maruyama and Milstein schemes. For a thorough detailing see Kloeden and Platen [32] and Glasserman [18]. The approximations simulate the process over N time steps of size $\Delta t \equiv \delta \coloneqq \frac{T}{N}$, where the update at the n-th iteration at time $t_n \coloneqq n\delta$ requires a Wiener process increment ΔW. The usual numerical schemes use a standard Gaussian random variable Z_n to simulate from this process, where $\Delta W_n \coloneqq \sqrt{\delta} Z_n$, and these Z_n are independently and identically distributed.

To ensure the stochastic process has a unique strong solution and the Euler-Maruyama scheme converges, we assume the standard assumptions from Kloeden and Platen [32, 4.5, 10.2], which are that: a and b are jointly Lebesgue measurable, spatially Lipschitz continuous, have linear spatial growth, $\frac{1}{2}$-Hölder temporal continuity with linear spatial growth, and that X has a measurable initial condition.

The rounding error model we will later introduce will be applicable to the original Euler-Maruyama scheme, but also to a modified version utilising high speed and low fidelity "approximate random variables". Consequently, we will now review approximate random variables and detail the associated requirements needed for our error model.

2.1. Approximate random variables. Unfortunately, sampling from the Gaussian distribution is expensive, and so there have been several approaches to bypass this

cost. Most of these look to substitute the exact Gaussian increment Z_n with another random variable $\widetilde{Z}_n$ with similar statistics. For clarity and consistency with Giles and Sheridan-Methven [15, 16, 46], we call these substitutes *approximate random variables*, and the originals as *exact random variables*. The most well known is to use Rademacher random variables, producing what's known as the weak Euler-Maruyama scheme [32, page XXXII], where the Rademacher random variables have the desired mean. More advanced methods include more generalised moment matching procedures, as discussed by Muller [39], and piecewise polynomial approximations and generalised approximate random variables by Giles and Sheridan-Methven [15, 46].

We briefly pause to remark that the aforementioned methods for generating Gaussian random variables can all be viewed as utilising approximations to the Gaussian distribution's inverse cumulative distribution function in the inverse transform method [15, 18]. However, there are alternative methods for generating Gaussian random variables, including the Ziggurat [37], Box-Muller [4], and Marsaglia [36] methods, each being extensively used by practitioners. For our purposes, we will be assuming the exact random variables are produced using the inverse transform method, and that the approximate random variables are generated using the piecewise polynomial approximations by Giles and Sheridan-Methven [16]. There are two main reasons for this. The first is that we will be able to utilise the body of analytic results derived by Giles and Sheridan-Methven [15, 16] concerning both the approximate random variables, and their subsequent use in nested multilevel Monte Carlo schemes. The second is that using the approximate random variables produced using [16] have been demonstrated to be substantially faster than their exact random variable counterparts on modern vectorised hardware [16]. This speed improvement is by design, and scales to smaller data types and wider vector lengths, whereas the alternatives methods do not, and their limitations are discussed extensively in [16, section 3.1.1], which we don't repeat here.

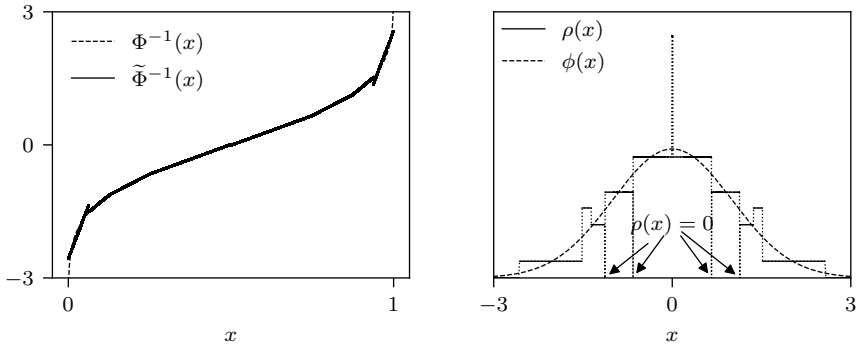
The Euler-Maruyama schemes using the exact Gaussian random variables Z_n give rise to the approximation $\widehat{X}$, and the approximate random variables $\widetilde{Z}_n$ produce $\widetilde{X}$, where the schemes are respectively

$$\widehat{X}_{n+1} = \widehat{X}_n + a(t_n, \widehat{X}_n)\delta + b(t_n, \widehat{X}_n)\sqrt{\delta} Z_n$$

and

$$\widetilde{X}_{n+1} = \widetilde{X}_n + a(t_n, \widetilde{X}_n)\delta + b(t_n, \widetilde{X}_n)\sqrt{\delta} \widetilde{Z}_n.$$

One of the approximations we will utilise later is the piecewise linear approximation by Giles and Sheridan-Methven [16]. This generates approximate Gaussian random variables by the inverse transform method [18] using a piecewise linear approximation $\widetilde{\Phi}^{-1}$ to the Gaussian distribution's inverse cumulative distribution function Φ^{-1}. The exact construction is detailed by Giles and Sheridan-Methven [16], although an example approximation is demonstrated in Figure 2.1. A piecewise linear approximation using 8 intervals is shown in Figure 2.1(a), and the resultant probability density function of the approximation ρ is shown in Figure 2.1(b). The piecewise linear approximation is non-monotonic with small discontinuities at the interval boundaries. Consequently, the resulting probability density function has compact support, although there are a few resulting tiny non-obvious inaccessible domain intervals with zero measure, as indicated. The rounding error model we will later propose will hold for both exact Gaussian random variables, and also certain classes of approximate Gaussian random variables, for which we require that they satisfy Assumption 2.1.



(a) A piecewise linear approximation using 8 intervals.

(b) The resultant probability density function with non-obvious intervals of zero measure indicated.

FIG. 2.1. *A piecewise linear approximation of the Gaussian distribution's inverse cumulative distribution function, and the resultant probability density function.*

ASSUMPTION 2.1. *Let any approximate Gaussian random variables $\widetilde{Z}$ be zero mean, uniformly bounded, and have finite variance $\mathbb{V}(\widetilde{Z}) = O(1)$, and also have all higher order moments be finite. Furthermore, let there exist a corresponding probability density function ρ such that $\mathbb{P}(\widetilde{Z} \in [z, z + \mathrm{d}z]) = \rho(z)\,\mathrm{d}z$. Let ρ be bounded by K such that $\rho \leq K < \infty$ and be smooth almost everywhere such that $\rho \in C^\infty(\mathbb{R}\backslash\mathcal{M})$, where $\mathcal{M}$ is a finite set of M points where ρ is discontinuous. Lastly, let ρ decay sufficiently fast such that for any finite constant α we have $\sum_{k=-\infty}^{\infty} 2^{2k} \max_{y' \in [2^k, 2^{k+1}]} \rho(y' - \alpha) < \infty$.*

LEMMA 2.1. *The exact Gaussian distribution satisfies Assumption 2.1.*

Proof. We immediately have that the Gaussian distribution is zero mean and has unit variance, and is uniformly bounded [3, appendix C.2]. The probability density function for the Gaussian distribution is $\rho \equiv \phi$ where $\phi(z) \coloneqq \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}z^2)$, which is $C^\infty(\mathbb{R})$ and maximal at zero where $\rho \leq \phi(0) = \frac{1}{\sqrt{2\pi}}$. To show $\sum_{k=-\infty}^{\infty} 2^{2k} \max_{y' \in [2^k, 2^{k+1}]} \rho(y' - \alpha) < \infty$ we consider separately the cases when $\alpha > 0$ and $\alpha \leq 0$. When $\alpha > 0$, we note that the summand is increasing as k increases from $-\infty$, and ρ will be maximal for the one index k^* where $\alpha \in [2^{k^*}, 2^{k^*+1}]$. The maximal value will therein be obtained for $y' = \alpha$, and thereafter for indices $k > k^*$ the ρ term will be decreasing. Thus we can approximate the possibly divergent part of the summation by the integral

$$\begin{aligned}
\sum_{k=k^*+1}^{\infty} 2^{2k} \max_{y' \in [2^k, 2^{k+1}]} \rho(y' - \alpha) &= \sum_{k=k^*+1}^{\infty} 2^{2k} \rho(2^k - \alpha) \\
&\approx \int_{2^{k^*+1}}^{\infty} x^2 \rho(x - \alpha)\,\mathrm{d}x \\
&\leq \int_{-\infty}^{\infty} x^2 \rho(x - \alpha)\,\mathrm{d}x \\
&< \infty.
\end{aligned}$$

The summation can be strictly bounded from above by using similar integrals, and

thus is not divergent. The partial summation from $k = -\infty$ up to $k = k^*$ is immediately finite as ρ is bounded.

For the case $\alpha \leq 0$, in each $[2^k, 2^{k+1}]$ interval the ρ is maximal at $y' = 2^k$, and decreases as k increases. The relevant summation in this case can also be bounded identically and shown to be non divergent. ∎

LEMMA 2.2. *The approximate Gaussian distribution resulting from the piecewise linear approximation by Giles and Sheridan-Methven [16] satisfies Assumption 2.1.*

Proof. For a finite number of approximation intervals, the probability density function is symmetric, uniformly bounded, and has compact support and thus finite variance. The number of discontinuities is finite, and as ρ has compact support it immediately satisfies the summation bound from Assumption 2.1. ∎

LEMMA 2.3. *The approximate Gaussian distribution resulting from the piecewise cubic approximation by Giles and Sheridan-Methven [16] satisfies Assumption 2.1.*

Proof. The proof follows identically to the proof of Lemma 2.2. ∎

The motivation for introducing these approximate random variables was to increase simulation speed. However, for the ultimate speed improvements, it is desirable to both switch to approximate random variables, and simultaneously decrease the arithmetic precision used, giving a twofold speed improvement. Reducing the precision alone has been explored with applications to field programmable gate arrays [5, 8, 42, 43], as have multilevel Monte Carlo schemes using varying fidelities of approximate random variables [39]. However, performing both simultaneously is touched upon by Giles et al. [17], although using a quite restrictive truncated uniform random bit Monte Carlo algorithm, and extensions to more general approximation schemes without varying the precision is done by Giles and Sheridan-Methven [15]. However, the work by Giles et al. [17] is primarily a cost most, and does not model the effect of rounding error. Thus, while our contribution is an extension of these works, it is an important and novel demonstration and vindication of the utility of low precisions with approximate random variables.

In order to describe the effect of rounding error resulting from the Euler-Maruyama scheme, the two most prominent works are by Arciniega and Allen [1] and Omland [42], which are models for the average and worst case errors respectively. Arciniega and Allen [1] provide an *ad hoc* statistical model and analysis for the rounding error arising from finite precision floating point calculations within the Euler-Maruyama scheme. Denoting the estimate produced when working in finite precision as $\overline{X}$, they propose that at the n-th iteration, all of the composite floating point arithmetic in the Euler-Maruyama update culminates in an additive error ε_n where $\overline{X}_{n+1} = \overline{X}_n + a(t_n, \overline{X}_n)\delta + b(t_n, \overline{X}_n)\sqrt{\delta}Z_n + \varepsilon_n$. This error is assumed to follow a Gaussian distribution, be zero mean, and have a variance $\mathbb{V}(\varepsilon) \leq C\varrho^2$, where ϱ is the unit roundoff and C is some constant. The main result from their analysis [1, theorem 2.2] is $\mathbb{E}(|\widehat{X}_N - \overline{X}_N|^2) \leq CN\varrho^2$. The model from Omland [42] uses a more rigorous finite precision framework. For brevity, letting $\oplus$ and $\otimes$ represent floating point addition and multiplication respectively, then the model by Omland [42] in effect considers $\overline{X}_{n+1} = \overline{X}_n \oplus ((a(t_n, \overline{X}_n) \otimes \delta) \oplus ((b(t_n, \overline{X}_n) \otimes \sqrt{\delta}) \otimes Z_n))$ and produces the worst case bound $\mathbb{E}(|\widehat{X}_N - \overline{X}_N|^2) \leq CN^2\varrho^2$ [42, theorem 4.8]. The model we will present will take the model from Omland [42] as its starting point, but under assumptions appropriate for the Euler-Maruyama scheme, will ultimately reduce to a model closer resembling that by Arciniega and Allen [1]. Our model will also allow for systematic rounding effects, and bounds how large these are permitted to be and

have the model still be applicable.

2.2. A leading order error model. Starting with the more fundamentally rooted model from Omland [42], we expect to recover, under appropriate assumptions, the more statistically motivated model from Arciniega and Allen [1]. To this end we look to expand the model from Omland [42] and see the effects of arithmetic roundoff within the Euler-Maruyama update. Before presenting the analysis, we will briefly introduce a small amount of floating point notation.

Numbers used in calculations must be stored in finite precision, where we denote the set of representable numbers as $\overline{\mathbb{R}} \subset \mathbb{R}$, where we introduce the rounding operator $R\colon \mathbb{R} \to \overline{\mathbb{R}}$ which implements the desired rounding mode, which we assume is round to nearest even. For finite precision binary arithmetic operations $\circledast$ (such as $\oplus$, $\otimes$, etc.), then for two floating point numbers $x, y \in \overline{\mathbb{R}}$, we assume the standard rounding model by Higham [22, 2.2, (2.4)] [50, page 99, (13.7)] that $x \circledast y = R(x * y) = (x * y)(1 + \varepsilon)$ where $|\varepsilon| \leq \varrho$.

For further notational convenience, when we talk about a random variable or process g being of a typical size h, we usually mean in the sense of absolute size and strong expectation, where we are really discussing the size of $\mathbb{E}(|g|)$. However, such explicit notation is overly verbose, so instead we will make regular use of O-notation. For brevity, we will write statements such as $\mathbb{E}(|g|) = O(h)$ instead as $|g| = O(h)$ or even $g = O(h)$, where the underlying meaning of such statements should be clear from context.

We begin by expressing the finite precision Euler-Maruyama update as $\overline{X}_{n+1} = \overline{X}_n \oplus (A_n \oplus B_n)$, where $A_n \coloneqq a(t_n, \overline{X}_n) \otimes \delta$ and $B_n \coloneqq (b(t_n, \overline{X}_n) \otimes \sqrt{\delta}) \otimes \widetilde{Z}_n$. For an appropriately non dimensionalised stochastic process, such that $T = 1$, $X_0 = O(1)$, $a = O(1)$, $b = O(1)$, then we anticipate $\overline{X}_n = O(1)$, $\widetilde{Z}_n = O(1)$, $A_n = O(\delta)$, and $B_n = O(\sqrt{\delta})$. Similarly, for the precision levels and discretisations we have $\varrho \ll 1$ and $\delta \ll \sqrt{\delta} \ll 1$. This then gives us the size ordering $\mathbb{E}(|A_n|) \ll \mathbb{E}(|B_n|) \ll \mathbb{E}(|\overline{X}_n|)$.

The first addition $A_n \oplus B_n$ will produce an absolute error η'_n where $A_n \oplus B_n = A_n + B_n + \eta'_n$, and η'_n will be of a size comparable with the unit roundoff and the larger of A_n and B_n, which is B_n. As b is assumed to have linear growth we obtain $|\eta'_n| \sim |B_n \varrho| = O(\sqrt{\delta}\varrho(1 + |\overline{X}_n|))$, and after performing this first floating point addition we will be left with $\overline{X}_{n+1} = \overline{X}_n \oplus (B_n + A_n + \eta'_n)$, where we have written $B_n + A_n + \eta'_n$ in order of decreasing magnitudes.

For the remaining addition operation $\overline{X}_n \oplus \cdots$, as we have $X_n = O(1)$ and $\mathbb{E}(|\overline{X}_n|) \gg \mathbb{E}(|B_n|)$, the nett result from the remaining floating point addition then is that this will produce a second absolute arithmetic error η_n where $|\eta_n| \sim |\overline{X}_n \varrho| = O(\varrho(1 + |\overline{X}_n|))$, and the Euler-Maruyama update will become $\overline{X}_{n+1} \approx \overline{X}_n + B_n + A_n + \eta_n + \eta'_n$, where again we have written the contributions in order of decreasing magnitudes. We identify two dominant sources of error. The first is η'_n, arising from the addition of the drift term to the diffusion term. The second is η_n, arising from the addition of this sum to the underlying process. We expect $|\eta'_n| = O(\varrho\sqrt{\delta}(1 + |\overline{X}_n|))$ and $|\eta_n| = O(\varrho(1 + |\overline{X}_n|))$, and thus $\mathbb{E}(|\eta_n|) \gg \mathbb{E}(|\eta'_n|)$. If we then allow for the inclusion of other higher order contributions η'', we can then see that we expect to obtain $\overline{X}_{n+1} = \overline{X}_n + B_n + A_n + \eta_n + \eta'_n + \eta''_n$.

The Arciniega and Allen [1] model assumes that only η_n is significant, and makes the assertion that this can be modelled as a zero mean Gaussian random variable with a variance only proportional ϱ^2. In our model, we will more rigorously justify the zero mean nature, drop the requirement that this exactly follows a Gaussian distribution, and show that the smaller second order contributions from η' are not

necessarily negligible, but can contribute to the nett rounding error at the same rate as the leading order η process. We propose Model 2.1 as an appropriate model for the rounding errors arising in the Euler-Maruyama scheme.

MODEL 2.1. *Let the Euler-Maruyama scheme use random variables $\widetilde{Z}$ which satisfy Assumption 2.1. The composite effects of rounding error introduce two dominant sources of error, η and η', where at each step we have*

$$\overline{X}_{n+1} = \overline{X}_n + a(t_n, \overline{X}_n)\delta + b(t_n, \overline{X}_n)\sqrt{\delta}\widetilde{Z}_n + \eta_n + \eta'_n.$$

The larger of these is $\eta_n = O(\varrho(1 + |\overline{X}_n|))$, which is a martingale increment, and the smaller of these is $\eta'_n = O(\varrho\sqrt{\delta}(1 + |\overline{X}_n|))$, which is a possibly non martingale increment.

Inspecting Model 2.1, the key modelling assumption requiring justification is the martingale[1] nature of η_n. We already justified in our discussions that $\eta'_n = O(\varrho\sqrt{\delta}(1+|\overline{X}_n|))$, and so claiming this is a non martingale increment is no further restriction. It is straightforward to reason that $\mathbb{E}(|\eta_n|) = O(\varrho(1 + \mathbb{E}(|\overline{X}_n|)))$, which if we take $\mathbb{E}(|\overline{X}_n|) = O(1)$ simplifies to $\mathbb{E}(|\eta_n|) = O(\varrho)$. Thus, to justify η_n being a martingale increment it is sufficient to reason that $|\mathbb{E}(\eta_n)| \ll \mathbb{E}(|\eta_n|)$, which we achieve through Lemma 2.4.

LEMMA 2.4. *Assuming $\mathbb{E}(|\overline{X}_n|) = O(1)$ and the random variables $\widetilde{Z}$ satisfy Assumption 2.1, then under the round to nearest even rounding mode, the leading order rounding error η_n has $\mathbb{E}(\eta_n) = O(\varrho^2)$.*

Proof. The operation producing η_n is the floating point addition between $\overline{X}_n$ and $(A_n \oplus B_n)$. Dropping the subscripts for brevity, we denote this by $\alpha \oplus R(\beta)$, where $\alpha \coloneqq \overline{X}_n = O(1)$ and $\beta \coloneqq A_n + B_n = O(\sqrt{\delta})$. The absolute rounding error η is then given by

$$\begin{aligned}\eta \coloneqq&(\alpha \oplus R(\beta)) - (\alpha + R(\beta))\\ \equiv&(R(\alpha+\beta) - (\alpha+\beta)) - (R(\beta) - \beta) + (R(\alpha + R(\beta)) - R(\alpha+\beta)),\end{aligned}$$

where we will bound the final three parenthesised differences in turn.

Inspecting the first term, we can see this is the absolute error resulting from rounding the quantity $\alpha + \beta$. Without much loss of generality, as we have assumed $\alpha = O(1)$, let us suppose $\alpha \in (1, 2)$. Using IEEE floating point representation, the set of representable numbers $(1, 2) \cap \mathbb{R}$ will all be equally spaced. Defining the quantity $z \coloneqq \alpha + \beta$, this will fall inside some interval $I_y \coloneqq [y - \varsigma, y + \varsigma]$, where $y \pm \varsigma$ are adjacent floating point numbers. Without loss of generality, we assume $y - \varsigma$ is odd and $y + \varsigma$ is even, where we either have $z < y$ and we round down, or $z \geq y$ and we round up, as depicted in Figure 2.2(a).

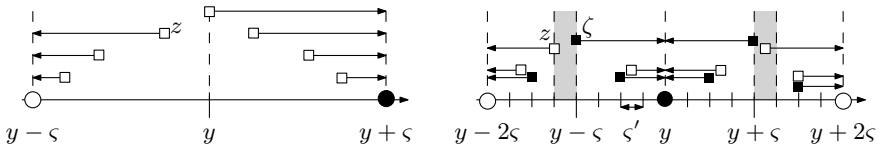
We can then evaluate the expectation $\mathbb{E}((R(z) - z)\mathbb{1}_{\{z \in I_y\}})$ where we have

$$\mathbb{E}((R(z) - z)\mathbb{1}_{\{z\in I_y\}}) = \mathbb{E}((R(z) - z)\mathbb{1}_{\{z\in[y-\varsigma,y)\}}) + \mathbb{E}((R(z) - z)\mathbb{1}_{\{z\in[y,y+\varsigma]\}}).$$

These expectations can be written as integrals, giving

$$\mathbb{E}((R(z) - z)\mathbb{1}_{\{z\in I_y\}}) = \int_{y-\varsigma}^{y} ((y-\varsigma) - z)\mathbb{P}(\mathrm{d}z) + \int_{y}^{y+\varsigma} ((y+\varsigma) - z)\mathbb{P}(\mathrm{d}z).$$

[1] For readers unfamiliar with martingales (and stochastic processes), cf. [32, 2.3].



(a) Possible values for z in I_y, denoting z with "□".

(b) Possible values for ζ in I'_y, demonstrating when the round to nearest even tie break causes rounding error by shaded regions. We denote ζ with "■", and z with "□".

FIG. 2.2. *Rounding to the nearest even. We denote even representable values using "●" and odd values using "◯". Arrows show the values rounded to.*

At time t_n, the variable $\overline{X}_n$ is $\mathcal{F}_n$-measurable, but $\widetilde{Z}_n$ is $\mathcal{F}_{n+1}$-measurable, and thus z and β will have the same distribution as $\widetilde{Z}$. Denoting the probability density function of β as ρ, which by extension satisfies Assumption 2.1, we obtain

$$\mathbb{E}((R(z)-z)\mathbb{1}_{\{z\in I_y\}}) = \int_0^{\varsigma} (\varsigma - z)(\rho(y-\alpha+z) - \rho(y-\alpha-z))\,\mathrm{d}z\,.$$

As $\varsigma \ll 1$, if ρ is smooth everywhere in the interval I_y, then we can approximate this using a Taylor series expansion, otherwise we use the bound from Assumption 2.1, obtaining

$$|\mathbb{E}((R(z)-z)\mathbb{1}_{\{z\in I_y\}})| \le \begin{cases} C\varsigma^3|\rho'(y-\alpha)| + O(\varsigma^5|\rho'''(y-\alpha)|) & \text{if } I_y \cap \mathcal{M} = \emptyset \\ C\varsigma^2 K & \text{if } I_y \cap \mathcal{M} \neq \emptyset, \end{cases}$$

for some constant C.

Using this expectation in the law of total expectation, then in the limits $I_y \to \mathrm{d}I_y$ we obtain

$$\begin{aligned} \mathbb{E}(R(z)-z) &= \mathbb{E}(\mathbb{E}(R(z)-z \mid z \in I_y)) \\ &\approx \int_{\mathbb{P}(I_y)>0} \frac{\mathbb{E}((R(z)-z)\mathbb{1}_{\{z\in I_y\}})}{\mathbb{P}(I_y)} \mathbb{P}(\mathrm{d}I_y) \\ &\approx \int \frac{\mathbb{E}((R(z)-z)\mathbb{1}_{\{z\in I_y\}})}{2\varsigma}\,\mathrm{d}y\,, \end{aligned}$$

where in the last approximation we used $\mathbb{P}(I_y) \approx \rho(y-\alpha)2\varsigma$ and $\mathbb{P}(\mathrm{d}I_y) \approx \rho(y-\alpha)\,\mathrm{d}y$. In the limit $\delta \ll 1$ then we can approximate our integration domain as $y \in [0,\infty)$, giving

$$|\mathbb{E}(R(z)-z)| \le \left| \int_0^{\infty} C\varsigma^2 \rho'(y-\alpha)\mathbb{1}_{\{I_y\cap\mathcal{M}=\emptyset\}}\,\mathrm{d}y \right| + \int_0^{\infty} C\varsigma K \mathbb{1}_{\{I_y\cap\mathcal{M}\neq\emptyset\}}\,\mathrm{d}y\,.$$

In IEEE representation, ς is a constant between powers of two, and thus $2\varsigma = \varrho 2^k$ for $k \in \mathbb{Z}$. Consequently, the first integral is readily decomposed into sub intervals where ς is a constant. For the second integral we let I^*_m denote the interval containing the discontinuity at position $m \in \mathcal{M}$. These discontinuities occur at the corresponding k

values k_m such that $I_m^* \subset [2^{k_m}, 2^{k_m+1}]$, and so we obtain

$$|\mathbb{E}(R(z)-z)| \leq C\left|\sum_{k=-\infty}^{\infty} \varrho^2 2^{2k} \int_{2^k}^{2^{k+1}} \rho'(y-\alpha)\mathbb{1}_{\{I_y\cap\mathcal{M}=\emptyset\}}\,\mathrm{d}y\right| + CK\sum_{m\in\mathcal{M}}\varsigma\int_{I_m^*}\mathrm{d}y\,.$$

The first integral can largely be evaluated exactly. The integration domain will contain at most M intervals containing singularities, and thus $M+1$ subintervals of the form $\int_a^b \rho'(y-\alpha)\,\mathrm{d}y$ where ρ' is continuous in the domain $[a,b]$, and thus $\int_a^b \rho'(y-\alpha)\,\mathrm{d}y = [\rho(y-\alpha)|_a^b$. We then use the bound $|\int_a^b \rho'(y-\alpha)\,\mathrm{d}y| \leq |\rho(a-\alpha)| + |\rho(b-\alpha)| \leq 2\max_{y'\in[a,b]}\rho(y'-\alpha)$. For the second integral we use $\int_{I_m^*}\mathrm{d}y = \varsigma$ to give

$$|\mathbb{E}(R(z)-z)| \leq C\varrho^2\sum_{k=-\infty}^{\infty} 2^{2k}(M+1)\max_{y'\in[2^k,2^{k+1}]}\rho(y'-\alpha) + CK\varrho^2\sum_{m\in\mathcal{M}}2^{2k_m} = O(\varrho^2),$$

where in the last equality we used Assumption 2.1. This shows the desired $O(\varrho^2)$ holds for the first parenthesised error constituting η. The same line of reasoning similarly holds the for $R(\beta)-\beta$ term also constituting η (akin to taking $\alpha\to 0$), arriving at an identical limiting bound.

It remains to bound the final $R(\alpha+R(\beta))-R(\alpha+\beta)$ term constituting η. Unlike the previous two terms, we will see that this term only contributes a rounding error when the round to nearest even tie break rule is required, and in most scenarios $\alpha+R(\beta)$ and $\alpha+\beta$ will round to the the same number. To tackle this final term we introduce the slightly larger interval $I_y' \coloneqq [y-2\varsigma, y+2\varsigma]$, where $y-2\varsigma$, y, and $y+2\varsigma$ are all representable and adjacent. Without loss of generality we assume y is even and $y\pm 2\varsigma$ are odd. Given $\beta = O(\sqrt{\delta})$ and $\alpha = O(1)$, we know that $|\beta - R(\beta)| \leq \varsigma'$ where $\varsigma' \ll \varsigma$, and thus β is rounded first on a much finer granularity than $\alpha+\beta$. Keeping our definition $z \coloneqq \alpha+\beta$ and introducing $\zeta \coloneqq \alpha+R(\beta)$, then the discrete set of values ζ can take has a much finer granularity than the three representable numbers in I_y', namely $\zeta \in \{y\pm n\varsigma'\}\cap I_y'$ for integers $n\in\mathbb{N}$. We display the set of values ζ can take in I_y' in Figure 2.2(b), where we demonstrate several possible rounding scenarios.

Inspecting Figure 2.2(b), we can see that in most situations ζ and z round to the same number. These only round to different numbers when ζ lies on a tie break value and z takes a different value that is rounded to an odd number, as indicated by the shaded regions in Figure 2.2(b). Thus, for the final term we have the expectation

$$\begin{aligned}E((R(\zeta)-R(z))\mathbb{1}_{\{z\in I_y'\}}) &= E((R(\zeta)-R(z))\mathbb{1}_{\{z\in[y-\varsigma-\varsigma',y-\varsigma)\}}\mathbb{1}_{\{\zeta=y-\varsigma\}})\\ &\quad + E((R(\zeta)-R(z))\mathbb{1}_{\{z\in(y+\varsigma,y+\varsigma+\varsigma']\}}\mathbb{1}_{\{\zeta=y+\varsigma\}}).\end{aligned}$$

The first expectation will round $\zeta\to y$ and $z\to y-2\varsigma$, giving a nett rounding error of 2ς, and the second expectation will round $\zeta\to y$ and $z\to y+2\varsigma$ giving a rounding error of -2ς. Thus, expressing these expectations as integrals we obtain

$$\begin{aligned}E((R(\zeta)-R(z))\mathbb{1}_{\{z\in I_y'\}}) &= \int_{y-\varsigma-\varsigma'}^{y-\varsigma} 2\varsigma\mathbb{P}(\mathrm{d}z) + \int_{y+\varsigma}^{y+\varsigma+\varsigma'} -2\varsigma\mathbb{P}(\mathrm{d}z)\\ &= 2\varsigma\int_0^{\varsigma'}(\rho(z+y-\varsigma-\varsigma'-\overline{a}) - \rho(z+y+\varsigma-\overline{a}))\,\mathrm{d}z\,.\end{aligned}$$

For this final integral expression we can again either perform a Taylor series expansion

or use our bounds from Assumption 2.1 to obtain to leading order

$$|E((R(\zeta)-R(z))\mathbb{1}_{\{z\in I'_y\}})| \leq \begin{cases} C\varsigma(\varsigma\varsigma' + (\varsigma')^2)|\rho'(y-\alpha)| & \text{if } I'_y \cap \mathcal{M} = \emptyset \\ C\varsigma\varsigma' K & \text{if } I'_y \cap \mathcal{M} \neq \emptyset. \end{cases}$$

As $\varsigma' \ll \varsigma$, we see that this bound is equivalent to that found earlier for the other two terms constituting η. Again, by using the law of total expectation and the same steps as before we obtain the same $O(\varrho^2)$ bound. Combining the three bounds completes the proof. ∎

While Model 2.1 is justified by Lemma 2.4, we can appreciate that the proof of Lemma 2.4 makes use of a few *ad hoc* approximations and limiting cases. However, our ultimate aim is only to justify our model, rather than derive it, so such conveniences are permissible. This serves to illustrate from first principles why the leading order error term η is effectively zero mean. Overall then, Lemma 2.4 provides a much more rigorous justification of the zero mean nature of the leading order error than was simply asserted in the Arciniega and Allen [1] model.

We now have $\mathbb{E}(|\eta_n|) = O(\varrho)$, $|\mathbb{E}(\eta_n)| = O(\varrho^2)$, and $\mathbb{E}(|\eta'_n|) = O(\varrho\sqrt{\delta})$. Consequently, for any reasonable discretisation where $\sqrt{\delta} \gg \varrho$, any non-martingale behaviour exhibited by the η_n process is negligible compared to the η'_n process, and instead constitutes part of the η''_n process mentioned earlier.

To illustrate how the bound for the final nett rounding error is produced from our model for the incremental rounding error, we recall a convenient lemma from Giles and Sheridan-Methven [15, lemma 4.3] [49, lemma 5.2.3], which we present without proof as Lemma 2.5.

LEMMA 2.5. *Suppose for a process $\mathcal{E}_n$ we have $\mathcal{E}_{n+1} = \mathcal{E}_n + \delta\mathcal{A}_n + \sqrt{\delta}\widetilde{Z}_n\mathcal{B}_n + \Xi_n + \Theta_n$, using a discretisation interval δ. We assert $\mathcal{E}_0 = 0$ almost surely, $\widetilde{Z}_n$ are i.i.d. zero mean random variables with all finite moments bounded, and $\mathcal{A}_n$ and $\mathcal{B}_n$ are $\mathcal{F}_{t_n}$-adapted with $|\mathcal{A}_n| \leq L_A|\mathcal{E}_n|$ and $|\mathcal{B}_n| \leq L_B|\mathcal{E}_n|$ for some strictly positive and finite constants L_A and L_B. The process Ξ is a martingale difference process where $\mathbb{E}(\Xi_n \mid \mathcal{F}_{t_n}) = 0$, and for integers $p \geq 2$ and a constant $s \in \mathbb{R}$ there are finite and strictly positive constants c_1 and c_2 such that $\mathbb{E}(|\Xi_n|^p) \leq c_1\delta^{p(s+1/2)}$. The process Θ similarly has $\mathbb{E}(|\Theta_n|^p) \leq c_2\delta^{p(s+1)}$. Then there exists constants c_3 and c_4 which depends only on L_A, L_B, and p such that $\mathbb{E}(\sup_{n\leq N}|\mathcal{E}_n|^p) \leq c_3(c_4c_1 + c_2)\delta^{ps}$, where $c_4 = 18p^{3/2}(p-1)^{-3/2}$.*

Proof. The proof is given by Giles and Sheridan-Methven [15, lemma 4.3] [49, lemma 5.2.3], and proceeds by a combination of Jensen's inequality, the discrete Burkholder-Davis-Gundy inequality, and the discrete Grönwall inequalities. ∎

By considering the difference between the process $\overline{X}_n$ calculated in high and low precision, then the result $\mathbb{E}(|\widehat{X}_N - \overline{X}_N|^2) \leq CN\varrho^2$ from Arciniega and Allen [1, theorem 2.2] immediately follows from Lemma 2.5. Furthermore, we obtain an identical bound from Lemma 2.5 for the nett error that arises from Model 2.1.

LEMMA 2.6. *Using Model 2.1 for the rounding errors, then $\mathbb{E}(|\widehat{X}_N - \overline{X}_N|^2) \leq CN\varrho^2$.*

Proof. Defining $\mathcal{E}_n := \widehat{X}_n - \overline{X}_n$ and differencing the appropriate Euler-Maruyama

schemes for $\widehat{X}_n$ and $\overline{X}_n$ we obtain

$$\mathcal{E}_{n+1} = \mathcal{E}_n + \delta \underbrace{\left(a(t_n, \widehat{X}_n) - a(t_n, \overline{X}_n)\right)}_{\mathcal{A}_n} + \sqrt{\delta}\widetilde{Z}_n \underbrace{\left(b(t_n, \widehat{X}_n) - b(t_n, \overline{X}_n)\right)}_{\mathcal{B}_n}$$
$$+ \underbrace{\sqrt{\delta} b(t_n, \widehat{X}_n)(\widehat{Z}_n - \widetilde{Z}_n) - \eta_n}_{\Xi_n} - \underbrace{\eta'_n}_{\Theta_n},$$

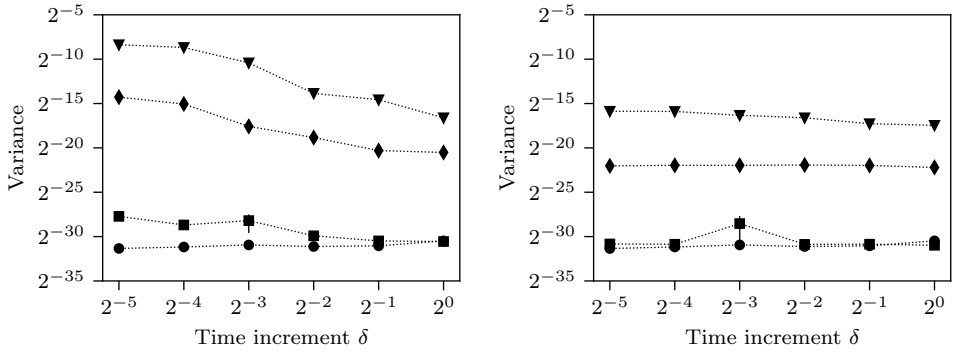
where we have indicated the equivalent terms in Lemma 2.5. The bounds on $\mathcal{A}_n$ and $\mathcal{B}_n$ follow from the standard assumptions of a and b being spatially Lipschitz continuous. Furthermore, for the Ξ_n term this is zero mean and has $\mathbb{E}(|\Xi_n|^p) \leq O(\delta^{p/2}) + O(\varrho^p) \leq O(\varrho^p)$, corresponding to $s = -\frac{1}{2}$ in Lemma 2.5 and $c_1 \propto \varrho^p$. Similarly, from Model 2.1 we have $\mathbb{E}(|\Theta_n|^p) \leq C\varrho^p\delta^{p/2}$, also corresponding to $s = -\frac{1}{2}$ and $c_2 \propto \varrho^p$. Thus from Lemma 2.5 we obtain $\mathbb{E}(|\widehat{X}_N - \overline{X}_N|^p) \leq O(\varrho^p\delta^{-p/2})$, which when we set $p = 2$ and note that $\delta \propto \frac{1}{N}$ obtains the desired bound. ∎

The significant insight provided by Lemma 2.6, which we saw in its proof when we applied Lemma 2.5, is that the nett contributions from the η and η' processes grow at the same rate. Although the η' process may be smaller than η by a factor of $\sqrt{\delta}$ in Model 2.1, because it is not zero mean, its contributions do not cancel, and thus can build up at a faster rate. Furthermore, this demonstrates that the updating process is permitted a systematic rounding error process, provided it is $O(\sqrt{\delta})$.

The variance predicted by Lemma 2.6 is shown in Figure 2.3(a). For the approximate random variables we have used the high fidelity piecewise cubic approximation from Giles and Sheridan-Methven [16]. This is to ensure that the η_n term within the Ξ_n process in the proof of Lemma 2.6 is the dominant term for moderately low precisions. Using the mpmath Python library [28] we adjust the number of bits used in the mantissa for the low precision approximate random variables. We use 7, 10, and 23 bits, corresponding to the precisions for `bfloat16`, half, and single precisions respectively. We also consider an artificial precision using 16 bits for the mantissa to represent an intermediate precision level between half and single precision. For the stochastic process we simulate a geometric Brownian motion where $a(t, X_t) \equiv \mu X_t$ and $b(t, X_t) \equiv \sigma X_t$ for strictly positive constant μ and σ. Following the setup from Giles [13] we choose $\mu = 0.05$, $\sigma = 0.2$, and $X_0 = 1$.

We can see from Figure 2.3(a) that we approximately observe the growth in the variance for half precision anticipated by Lemma 2.6. However, the brain float precision seems to exhibit a variance closer to the worst case bound from Omland [42]. Interestingly, we see that the higher precision intermediate and single precision results appear to exhibit an approximate $O(1)$ variance. The reason for this is because at such high precisions, the η term within Ξ_n is the smaller of the two terms, with the approximation error $Z_n - \widetilde{Z}_n$ being the more dominant error. As shown by Giles and Sheridan-Methven [15] this produces a resultant error independent of the discretisation δ, and so appears $O(1)$, as observed. Furthermore, the intermediate and single precision variances are approximately the same values, again indicating that this error is dominated by the approximate random variables' fidelity rather than their floating point precision.

2.3. Kahan compensated summation. The Euler-Maruyama scheme consists of performing a cumulative summation of a sequence of update terms. The problem of summing a sequence of floating point numbers and minimising the cumulative rounding error is well known, and there have been a variety of methods developed to overcome this, such as *pair wise summation* or *compensated summation*



(a) Without Kahan compensated summation. (b) With Kahan compensation summation.

FIG. 2.3. *The variance of the difference $\widehat{X}_T - \overline{X}_T$ between the exact Euler-Maruyama estimate $\widehat{X}$ and the estimate using low precision approximate random variables $\overline{X}$ from a piecewise cubic approximation with different precisions, corresponding to Lemma 2.6. The precisions use differing numbers of bits for the mantissa: (▼) 7, (♦) 10, (■) 16, and (●) 23 bits. Results are shown with and without Kahan compensated summation, which is presented in Section 2.3.*

[22, 4.1]. As the Euler-Maruyama scheme is a sequential and incremental procedure, a compensated summation is an appropriate technique. As our schemes are motivated by computational speed, then *Kahan compensated summation* [29], being the least numerically intensive, is the most suitable candidate for incorporating into the Euler-Maruyama scheme. The Kahan compensated summation adds a given increment, and then subtracts away the computed summation prior to that increment. The difference of this inferred increment from the original provides an estimate for the incurred rounding error, which is then adjusted for when adding subsequent terms in the sequence. The Kahan compensated summation procedure is outlined in Algorithm 2.1, and a C implementation appropriate for incorporating this into the Euler-Maruyama scheme in single precision is shown in Code 1. Interestingly, the related use of Kahan compensated summation in the numerical solution of ordinary differential equations was first proposed by Vitasek [52], and is demonstrated by Higham [21, pages 86–87].

Algorithm 2.1 Kahan compensated summation.

Input: A sequence $\{x_1, x_2, \ldots, x_N\}$ of N floating point numbers.
Output: A high accuracy estimate of the summation $\sum_{i=1}^{N} x_i$.

1: Initialise both an accumulator a and compensation c to zero.
2: **for all** $x_i \in \{x_1, x_2, \ldots, x_n\}$ **do**
3: Calculate a compensated increment $y \leftarrow x_i - c$.
4: Keep the original $a_{\text{original}} \leftarrow a$.
5: Add the compensated increment $a_{\text{new}} \leftarrow a + y$.
6: Update the compensation $c \leftarrow (a_{\text{new}} - a_{\text{original}}) - y$.
7: Update the accumulator $a \leftarrow a_{\text{new}}$.
8: **end for**
9: Use a to estimate $\sum_{i=1}^{N} x_i$.

An error analysis of Kahan compensated summation is provided by Higham [21, page 791, (3.11)], Knuth [33, Excercise 19, pages 229 and 571–573], and Goldberg [19].

```
float compensated_EM_scheme(float X, float dX, float * comp)
{
    float compensated_increment = dX - (*comp);
    float accumulated_sum = X + compensated_increment;
    (*comp) = (accumulated_sum - X) - compensated_increment;
    return accumulated_sum;
}
```

CODE 1

C implementation of the Euler-Maruyama scheme using the Kahan compensated summation, implementing lines 3–7 from Algorithm 2.1.

The Kahan compensated summation shown in Algorithm 2.1 has the overall absolute and relative error bounds of

$$(2\varrho + O(N\varrho^2)) \sum_{i=1}^{N} |x_i| \qquad \text{and} \qquad (2\varrho + O(N\varrho^2)) \frac{\sum_{i=1}^{N} |x_i|}{\left|\sum_{i=1}^{N} x_i\right|}$$

respectively. The ratio of $\sum_{i=1}^{N} |x_i|$ to $|\sum_{i=1}^{N} x_i|$ is known as the condition number, representing the sensitivity of the summation to rounding error [22, 50]. For a series of zero mean random variables, the condition number can be expected to be $O(\sqrt{N})$, but for several stochastic process (e.g. geometric Brownian motion) the drift term causes the increments to have a non zero mean, and hence the condition number approximately limits to a constant.

Based on the usual error analysis of Kahan compensated summation, one might expect that the leading order error from a Kahan compensated Euler-Maruyama scheme should mostly have an $O(1)$ error dependence on N, until eventually the higher order term takes effect for sufficiently large N. However, inspecting Model 2.1 and Code 1 we see that the Kahan compensated summation is designed to tackle the leading order η term. However, in computing the Euler-Maruyama update, the smaller η' error is not compensated for, and thus will persist. Thus, even with Kahan compensated summation, we see from Lemma 2.6 that we still expect a nett leading order rounding error $O(\sqrt{N}\varrho)$. Overall then, we see that as we increase N, we expect for small N an $O(\varrho)$ error, for very large N an $O(N\varrho^2)$ error, and possibly an intermediate $O(\sqrt{N}\varrho)$ error.

The variances from the approximations obtained by incorporating a Kahan compensated summation into the modified Euler-Maruyama scheme are shown in Figure 2.3(b). The first thing to notice from this is that all the variances appear to be $O(1)$, in keeping with the leading order error anticipated. For the brain float and half precision variances, there is a separation between these which is approximately a factor of 2^{-6}. As we expect the leading order error to be $O(\varrho)$, then we expect a reduction in the variance of approximately $\frac{\varrho_{10}^2}{\varrho_7^2} \approx 2^{-6}$, where we have used the subscript to denote the number of mantissa bits. Hence we can see the reduction in variance is approximately as anticipated. As for the intermediate and single precisions, we see these cluster on top of each other, again indicating that the dominant error is from approximating the Gaussian distribution rather than from finite precision arithmetic.

Our incorporation of the Kahan compensated summation scheme has been qualified by heuristics and our empirical results. An exact and detailed analysis of the errors remaining after incorporating Kahan compensated summation with Model 2.1

remains open for future analysis. The key point we highlight is its high level utility and interpretation, which we further utilise and explore when discussing multilevel Monte Carlo settings in Section 3, where it will considerably improve the capabilities of working with half precision floating point numbers.

3. Multilevel Monte Carlo. We now have two possible types of simulation: an expensive but precise simulation using exact Gaussian random variables in a high floating point precision, and a cruder and cheaper simulation using approximate Gaussian random variables in low precision. The accuracy of the former can be combined with the speed of the latter by a multilevel Monte Carlo formulation [13, 14].

As a brief review of multilevel Monte Carlo, and how our work fits within this, let us suppose we wish to compute the expectation of some functional P which acts on the terminal solution X_T of the underlying stochastic process. The simulations can be performed using various levels of temporal discretisation, where we index the levels by l. We suppose there are $L+1$ levels such that $l \in \{0, 1, 2, \ldots, L\}$, where $l = 0$ corresponds to the coarsest possible discretisation, and $l = L$ the finest. The approximation coming from the usual Euler-Maruyama scheme for a particular level l we denote by $\widehat{P}_l$. Additionally, we denote those arising from the modified Euler-Maruyama scheme using high precision approximate random variables by $\widetilde{P}_l$, and low precision approximate random variables by $\overline{P}_l$. For notational simplicity we use the convention $\widehat{P}_{-1} \coloneqq \widetilde{P}_{-1} \coloneqq \overline{P}_{-1} \coloneqq 0$. Giles and Sheridan-Methven [15, 16] suggest incorporating the approximate random variables using the nested multilevel Monte Carlo framework

$$\begin{aligned}
\mathbb{E}(P) \approx \mathbb{E}(\widehat{P}_L) &= \sum_{l=0}^{L} \mathbb{E}(\widehat{P}_l - \widehat{P}_{l-1}) \\
&= \sum_{l=0}^{L} \mathbb{E}(\widetilde{P}_l - \widetilde{P}_{l-1}) + \mathbb{E}(\widehat{P}_l - \widehat{P}_{l-1} - \widetilde{P}_l + \widetilde{P}_{l-1}) \\
&= \sum_{l=0}^{L} \mathbb{E}(\overline{P}_l - \overline{P}_{l-1}) + \mathbb{E}(\widehat{P}_l - \widehat{P}_{l-1} - \overline{P}_l + \overline{P}_{l-1}),
\end{aligned}$$

where the first approximation is the regular Monte Carlo procedure [18], the first equality is the usual multilevel Monte Carlo decomposition [13], the second equality is the nested multilevel Monte Carlo framework [15, 16], and the final equality is the same nested multilevel framework utilising low precision approximate random variables.

Importantly, for a given level l, the fine path's discretisation δ^{f} and the coarse path's δ^{c} are given by $\delta^{\mathrm{f}} = 2^{-l}$ and $\delta^{\mathrm{c}} = 2\delta^{\mathrm{f}}$ respectively. The coarse path's Wiener increments are produced by the pairwise summation of the fine path's Wiener increments. Furthermore, the Wiener increments ΔW are produced using Gaussian increments Z where $\Delta W \coloneqq \sqrt{\delta} Z$, and the Gaussian increments are produced using the inverse transform method where $Z_n = \Phi^{-1}(U_n)$. Crucially, the exact random variables Z_n and approximate random variables $\widetilde{Z}_n$ are tightly coupled by ensuring they are both generated using the same underlying random variables, where $Z_n = \Phi^{-1}(U_n)$ and $\widetilde{Z}_n = \widetilde{\Phi}^{-1}(U_n)$, with U_n being the same for both.

Giles and Sheridan-Methven [16] consider the usual multilevel estimator $\hat{\theta}$ and

the nested multilevel estimator $\overline{\theta}$ where

$$\hat{\theta} \coloneqq \sum_{l=0}^{L} \frac{1}{\widehat{m}_l} \sum^{\widehat{m}_l} (\widehat{P}_l - \widehat{P}_{l-1})$$

and

$$\overline{\theta} \coloneqq \sum_{l=0}^{L} \left(\frac{1}{\overline{m}_l} \sum^{\overline{m}_l} (\overline{P}_l - \overline{P}_{l-1}) + \frac{1}{\overline{M}_l} \sum^{\overline{M}_l} (\widehat{P}_l - \widehat{P}_{l-1} - \overline{P}_l + \overline{P}_{l-1}) \right),$$

where $\widehat{m}_l$, $\overline{m}_l$, and $\overline{M}_l$ are the number of paths generated, each with a computational cost of $\hat{c}_l$, $\overline{c}_l$, and $\overline{C}_l$, and variance $\hat{v}_l$, $\overline{v}_l$, and $\overline{V}_l$ respectively. Letting $\widehat{T}$ denote the total computational time to achieve a mean squared error ε^2 using the estimator $\hat{\theta}$, and similarly $\overline{T}$ using $\overline{\theta}$, then Giles and Sheridan-Methven [16] show

$$\widehat{T} = 2\varepsilon^{-2} \left(\sum_{l=0}^{L} \sqrt{\hat{v}_l \hat{c}_l} \right)^2 \qquad \text{and} \qquad \overline{T} = 2\varepsilon^{-2} \left(\sum_{l=0}^{L} \sqrt{\overline{v}_l \overline{c}_l} + \sqrt{\overline{V}_l \overline{C}_l} \right)^2,$$

and hence an overall temporal saving of

$$\overline{T} \approx 2\varepsilon^{-2} \left(\sum_{l=0}^{L} \sqrt{\hat{v}_l \hat{c}_l} \left(\sqrt{\frac{\overline{v}_l \overline{c}_l}{\hat{v}_l \hat{c}_l}} + \sqrt{\frac{\overline{V}_l \overline{C}_l}{\hat{v}_l \hat{c}_l}} \right) \right)^2 \leq \widehat{T} \max_{l \leq L} \left\{ \frac{\overline{v}_l \overline{c}_l}{\hat{v}_l \hat{c}_l} \left(1 + \sqrt{\frac{\overline{V}_l \overline{C}_l}{\overline{v}_l \overline{c}_l}} \right)^2 \right\}.$$

When the approximation's fidelity is such that $\overline{v}_l \approx \hat{v}_l$, the term $\frac{\overline{v}_l \overline{c}_l}{\hat{v}_l \hat{c}_l} \approx \frac{\overline{c}_l}{\hat{c}_l}$ measures the potential time savings, and the term $(1 + (\overline{V}_l \overline{C}_l / \overline{v}_l \overline{c}_l)^{1/2})^2$ assesses the efficiency at realising these savings. Achieving a balance between these two is required, where the approximation should be sufficiently fast so there is the potential for large savings, but of a sufficient fidelity so the variance of the more expensive four way difference $\widehat{P}_l - \widehat{P}_{l-1} - \overline{P}_l + \overline{P}_{l-1}$ is considerably lower than the variance of the cheaper two way difference $\widehat{P}_l - \widehat{P}_{l-1}$.

Giles and Sheridan-Methven [15, 16] investigate the variance of this final four way difference when using approximate random variables assuming infinite precision arithmetic for a variety of functional types [15, lemmas 4.10 and 4.11] [49, corollaries 6.2.6.2 and 6.2.6.3]. They find that for Lipschitz continuous and differentiable functionals that

$$\|\widehat{P}_l - \widehat{P}_{l-1} - \widetilde{P}_l + \widetilde{P}_{l-1}\|_p \leq O(\delta^{1/2} \|Z - \widetilde{Z}\|_{p'}) \tag{3.1}$$

for some p' such that $2 \leq p < p' < \infty$. For simplicity we will restrict our attention to functionals which are also Lipschitz continuous and differentiable, and we consider just the underlying process and take $P(X) \equiv X$.

In an analogous manner to Lemma 2.6 we can consider the variance of this four way difference arising in the nested multilevel Monte Carlo framework when the precision is simultaneously lowered when switching to approximate random variables, giving rise to Lemma 3.1.

LEMMA 3.1. *For fine and coarse path simulations constructed using approximate random variables as described by Giles and Sheridan-Methven [15], then with rounding errors described by Model 2.1 we have*

$$\mathbb{E}(|\widehat{X}_l - \widehat{X}_{l-1} - \overline{X}_l + \overline{X}_{l-1}|^2) \approx O(\delta\, \mathbb{E}(|Z - \widetilde{Z}|^{2+\epsilon})) + O(\delta^{-1} \varrho^2)$$

as the discretisation δ parameter decreases for some $\epsilon \in (0, \infty)$.

Proof. The initial $O(\delta\,\mathbb{E}(|Z - \widetilde{Z}|^{2+\epsilon}))$ term comes from (3.1) [15, 16] taking $P(X) \equiv X$ and $p = 2$. The second $O(\delta^{-1}\varrho^2)$ term comes from the η and η' contributions from Model 2.1, and their final nett contributions arise using Lemma 2.5 in an identical manner to the proof of Lemma 2.6. ∎

We can compare the realised variances predicted by Giles and Sheridan-Methven [15, 16] when using approximate random variables in infinite precision, with those predicted by Lemma 3.1. We use the piecewise linear approximation by Giles and Sheridan-Methven [16], and use half precision capable hardware (rather than emulation using the mpmath Python library). We use C code on a Nvidia Jetson AGX Xavier machine, containing a Nvidia 12 core Volta GPU and an 8 core Arm v8.2 64 bit CPU. Both the CPU and the GPU support half precision floating point arithmetic in hardware, and so we run the code on the CPU using the `_Float16` data type for half precision, compiled with `gcc` and notably with the flags `-O0` and `-march=armv8.2-a+fp16`. The first flag ensures no compiler optimisations are issued, guaranteeing the Kahan compensated summation is not removed by the compiler, and the second ensures half precision data types and operations are accessible and used. The results for the variances of the various multilevel Monte Carlo terms are shown in Figure 3.1.

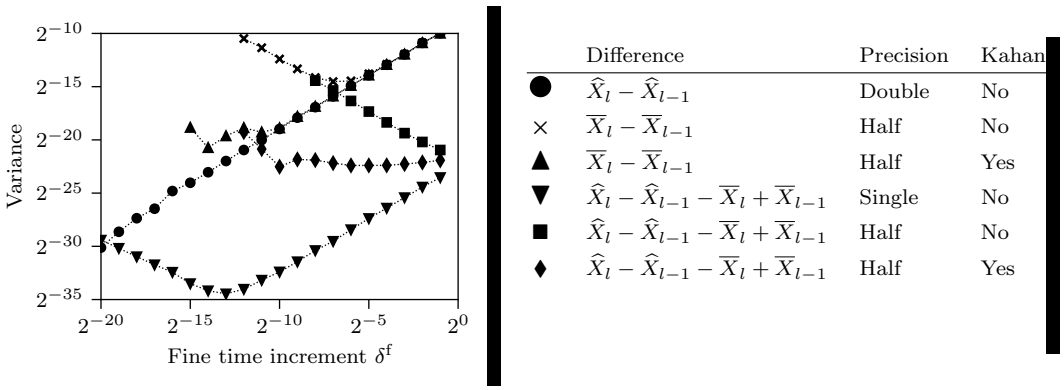


FIG. 3.1. *The variance reduction when switching to approximate random variables with a nested multilevel Monte Carlo framework, showing the variances of various two and four way differences. The key indicates the difference, precision, and whether Kahan compensated summation is used.*

Inspecting Figure 3.1 we can make several observations. The first is that the usual two way difference (computed in double precision) exhibits the usual $O(\delta)$ variance decay rate, as is to be expected and is a standard result [18, 32].

The next item of interest is the behaviour of the four way difference computed in single precision. Down to discretisations as fine as $\delta \approx 2^{-13}$ we can see the variance decays at the same $O(\delta)$ rate, as already predicted and demonstrated by Giles and Sheridan-Methven [16, 4.1]. However, the novel feature predicted by Lemma 3.1 is the emergence of the $O(\delta^{-1})$ rate for very fine discretisation, arising from Model 2.1. We can see that the onset of rounding error is not immediately catastrophic, and that for $\delta \approx 2^{-17}$ there is still a reduction in the variance between the two and four way differences which is still approximately 2^{-6}. However, eventually, for discretisations as fine as $\delta \approx 2^{-20}$ there is no reduction in variance. Ultimately, this demonstrates that double precision is typically superfluous for Monte Carlo path simulations (ignoring

sensitivity calculations for computing derivatives by finite differences), and single precision is sufficient.

The main items of particular interest are the half precision variances. For simplicity we begin by inspecting the difference without Kahan compensated summation. The variance of the two way difference term decreases in line with the double precision two way difference down to approximately $\delta \approx 2^{-7}$, and thereafter the effects of rounding error become dominant. This immediately places a lower limit on an uncompensated half precision framework, which is valid for discretisations coarser than $\delta \geq 2^{-7}$. Considering the four way difference, we see that at the very coarsest level there is an appreciable variance reduction by a factor of approximately 2^{-12}, although rounding error has already started to become dominant. We see that as the discretisations become finer the rounding error increases, and at $\delta \approx 2^{-7}$ coincides with the two way difference. However, for discretisations coarser than $\delta \geq 2^{-4}$, there is still at least a variance drop by approximately a factor of 2^{-6}. This suggests that while half precision calculations may be fast, if the rounding error is not compensated for, then they are only useful on the coarsest few levels.

If we incorporate a Kahan compensated summation to the Euler-Maruyama scheme when using half precision approximate random variables, the picture improves. The first item to note is the variance of the two way difference, which mirrors the double precision's two way variance down to approximately $\delta \approx 2^{-11}$, placing a lower limit on the minimum possible discretisation with $\delta \geq 2^{-11}$. For the four way difference, at the very coarsest level we see approximately the same variance as the uncompensated half precision four way difference, as might be expected. However, with the compensation, the error is an $O(1)$ constant as the discretisation becomes ever finer down to $\delta \approx 2^{-10}$. For discretisations $\delta < 2^{-10}$ a higher order error process appears to dominate, and for such fine discretisations the two way and four way variances coincide. This suggests that half precision simulations using Kahan compensated summation are applicable for much finer discretisation than equivalent simulations without the Kahan compensated summation. A variance reduction of approximately 2^{-6} is achieved for discretisations $\delta \geq 2^{-8}$. Thus Kahan compensated summation approximately doubles the scope of practical applicability for half precision simulations.

TABLE 3.1

Performance of various exact and approximate implementations of the inverse Gaussian cumulative distribution function, and the possible speed ups offered.

(a) The time to generate exact and approximate Gaussian random variables.

Description	Precision	Clock cycles	Source
Intel (HA)	Single	3.5 ± 0.2	[16]
Piecewise linear	Single	0.5 ± 0.1	[16]
Piecewise linear	Half	0.25	Speculated

(b) The maximum possible speed ups.

Precision	Kahan	Speed up
Single	No	7
Half	No	14
Half	Yes	10

We take the timing results on Intel AVX-512 Skylake hardware from Giles and Sheridan-Methven [16]. For the exact Gaussian distribution, we take as our baseline the single precision Intel high accuracy (HA) function. In lieu of vectorised half precision capable hardware, we speculate that for the approximate random variables, that half precision input can be processed in half the time as single precision input. Overall then we have the times shown in Table 3.1(a). Furthermore, we make the idealised assumption that the cost of the simulations is entirely based on the cost of generating random numbers, and neglect the cost of all other arithmetic operations.

These give the maximum potential speed ups shown in Table 3.1(b). For the half precision approximation using Kahan compensated summation, we again speculate and suggest an intermediate value between the single and half precision offerings.

For each possible discretisation level, the estimated speed ups predicted for each individual level from the nested multilevel Monte Carlo analysis are shown in Figure 3.2. We can see from this that for the bulk of levels the single precision approximation offers a good potential speed up by a factor of approximately 7. However, for the very coarsest few levels, the half precision approximations without Kahan compensated summation offer superior speed ups by a factor of 10–12. Dependent on the speed reduction that comes from incorporating the half precision Kahan compensated summation, there is the possibility of these offering a third intermediate regime where they are the optimal choice. For our speculated speed ups from Table 3.1(b) we see there appears to be such an intermediate region, although it is not overwhelmingly competitive compared to the two other alternatives.

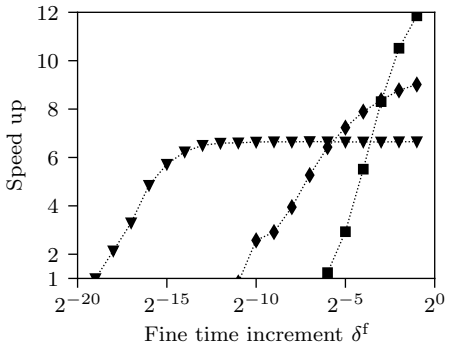


FIG. 3.2. *The potential savings from a nested multilevel Monte Carlo framework using approximate random variables from a piecewise linear approximation for various discretisation levels. (▼) Single precision. (■) Half precision without Kahan compensated summation. (♦) Half precision with Kahan compensated summation.*

It is worth remarking that when using the Euler-Maruyama scheme, as the scheme has a strong convergence order of $\frac{1}{2}$, the computational work load is approximately spread evenly over the various levels [13]. However, for numerical schemes with higher strong convergence orders, such as the Milstein scheme which has order 1 strong convergence [18, 32], the work load is predominantly concentrated on the coarsest levels [13]. The implication of this is that the potential half precision speed ups offered on the coarsest levels, even without Kahan compensated summation, may well dominate the multilevel savings. Hence, while half precision appears attractive even with multilevel Monte Carlo frameworks using the Euler-Maruyama scheme, this becomes even more so for the Milstein scheme and other higher order methods.

Currently, the use of approximate random variables and their incorporation in stochastic simulations and nested multilevel Monte Carlo frameworks only has rigorous supporting analytic results for the Euler-Maruyama scheme [15, 16]. Extensions to higher order schemes, such as the Milstein scheme, have only been discussed and demonstrated empirically [15, 16]. Consequently, there is currently insufficient supporting analysis to attempt to marry low precision approximate random variables in a nested multilevel Monte Carlo framework with higher order numerical schemes at an analytic level, and such analysis remains open for further research.

Lastly, we can speculate about the utility of brain floats compared to regular half precision floats (taken to be the IEEE specification). Brain floats and half precision floats are both 16 bits in size, but differ in their trade off between precision and range. Brain floats have a much larger range and lower precision than regular half precision. Having a lower precision will likely mean that the initial impact of rounding error will be even more severe than for half precision. This means the variance reduction will be less favourable, and the nested multilevel Monte Carlo framework less efficient on each level, and useful over even fewer discretisations. Without cause to suspect brain floats will be any faster than regular half precision, this would suggest that while half precision is attractive for multilevel Monte Carlo applications, and brain floats may similarly be attractive for the same reasons, there is no reason to speculate that brain floats will be competitive over regular half precision floats.

4. Conclusions. Performing calculations in high precisions may assuage worries about rounding errors, but makes several computations needlessly expensive. Considering the numerical simulation of stochastic differential equations, based on previous work using computationally cheap approximate random variables, we couple their incorporation into the Euler-Maruyama scheme with low precision implementations. We develop a new model for the nett rounding error incurred which allows for both systematic and unsystematic errors, analysing how the two can be balanced under appropriate assumptions. Kahan compensated summation is also discussed as a means of removing the leading order rounding error. This rounding error model is incorporated into a nested multilevel Monte Carlo scheme allowing for the speed of low precisions to be capitalised on without losing accuracy, finding that single precision is applicable for most discretisations and offers good potential speed ups, while half precision appears to offer superior speeds at only the coarsest few discretisation levels.

Introducing finite precision calculations, we discuss the appeal of working in ever lower precisions, and the attraction of using half precision in various applications. Low precision offers improved speed by increasing bandwidth and decreasing floating point calculation times, and is rapidly gaining traction in software and hardware, primarily due to applications in machine learning. However, low precision comes with appreciable finite precision rounding error, whose effects are felt in applications including: linear algebra, machine learning, stochastic simulation, and various others. To mitigate against this there are high precision libraries, compensated summation schemes, and mathematical techniques such as Richardson extrapolation, although the problem of rounding error can be severe for 16 bit floating point formats such as half precision or brain floats.

Considering the setting of stochastic simulations and approximating solutions of stochastic differential equations using the Euler-Maruyama scheme we introduce Model 2.1 as a novel description for the effects of rounding error. Similar previous models for rounding error in this setting are notably from Arciniega and Allen [1] and Omland [42], for the average and worst case scenarios respectively. The model we introduce is based on the framework by Omland [42] and recovers a similar model to the one by Arciniega and Allen [1]. However, comparing our model to that from Arciniega and Allen [1], ours has several benefits. Firstly, ours facilitates the incorporation of approximate random variables, which Giles and Sheridan-Methven [16] showed offers considerable potential speed improvements. Our model now rigorously justifies a leading order zero mean contribution, and permits a smaller second order and possibly non zero mean contribution, thus also facilitating systematic errors. Using Lemma 2.5 we show the nett contributions from these two terms grow at the same

rate for the Euler-Maruyama scheme, providing novel insight into the size permissible for systematic errors.

Reviewing Kahan compensated summation, we discuss the leading order constant error this can be expected to produce. We numerically simulated geometric Brownian motion processes, finding the errors predicted by Lemma 2.6 are observed empirically, alongside the anticipated error resulting from incorporating Kahan compensated summation. Furthermore, while Kahan compensated summation cancels the leading order rounding error, there is a balance between this and the error resulting from using approximate random variables, which becomes the dominant source of error in higher fidelity simulations.

Low precision implementations of approximate random variables are best utilised with a nested multilevel Monte Carlo framework, paralleling the setup by Giles and Sheridan-Methven [16]. Crucially, for practitioners working in high performance scientific computing, this integration both opens up the capabilities of realising the potential speed ups offered from low precision calculations and also gives analytic assurances that accuracy need not be lost. The nested multilevel Monte Carlo setup produces a four way difference, whose variance we predict in Lemma 3.1 and empirically observe. As expected, we find the errors resulting from introducing approximate random variables and also from low precision calculations are orthogonal effects. Using the nested multilevel Monte Carlo framework, we calculate that for a very wide range of possible discretisation levels that single precision simulations offer a speed up by a factor of approximately 7, as already shown by Giles and Sheridan-Methven [16]. However, for the very coarsest few levels, we demonstrated that half precision calculations, despite incurring significant rounding error from the very onset, can be successfully utilised within a multilevel Monte Carlo scheme. On these coarsest few levels, half precision can offer speed improvements by a factor of 10–12. For the Euler-Maruyama scheme the work load is usually evenly spread across the various levels, whereas for the higher order Milstein scheme the work load is dominated by the coarsest few levels. Thus we have been able to demonstrate the utility and applicability of half precision approximate random variables for stochastic simulation applications. Lastly, dependent on the cost model for the added arithmetic required for Kahan compensated summation, we were also able to demonstrate that this may provide a further intermediate region, extending the range of half precision to even finer discretisation levels, compounding their benefits.

All of the code to produce these figures is freely available and hosted by Sheridan-Methven [48].

5. Acknowledgements. We would like to acknowledge and thank those who have financially sponsored this work. This includes the Engineering and Physical Sciences Research Council (EPSRC) and Oxford University's centre for doctoral training in Industrially Focused Mathematical Modelling (InFoMM), with the EP/L015803/1 funding grant. Furthermore, this research stems from a PhD project [49] which was funded by Arm and NAG. Funding was also provided by the EPSRC ICONIC programme grant EP/P020720/1, the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA), and by Mansfield College, Oxford.

References.

[1] Armando Arciniega and Edward Allen. Rounding error in numerical solution of stochastic differential equations. *Stochastic analysis and applications*, 21(2): 281–300, 2003.

[2] Ivo Babuška. Numerical stability in mathematical analysis. In *IFIP congress (I)*, volume 68, pages 11–23, 1968.

[3] Stephen J. Blundell and Katherine M. Blundell. *Concepts in thermal physics*. Oxford University press, 2nd edition, 2014.

[4] George E.P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29:610–611, 1958.

[5] Christian Brugger, Christian de Schryver, Norbert Wehn, Steffen Omland, Mario Hefter, Klaus Ritter, Anton Kostiuk, and Ralf Korn. Mixed precision multilevel Monte Carlo on hybrid computing systems. In *2104 IEEE conference on computational intelligence for financial engineering & economics (CIFEr)*, pages 215–222. IEEE, 2014.

[6] Neil Burgess, Chris Goodyer, Chris Hinds, and David Lutz. High-precision anchored accumulators for reproducible floating-point summation. *IEEE transactions on computers*, 2018.

[7] Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. Bfloat16 processing for neural networks. In *2019 IEEE 26th symposium on computer arithmetic (ARITH)*, pages 88–91. IEEE, 2019.

[8] Gary Chun Tak Chow, Anson Hong Tak Tse, Qiwei Jin, Wayne Luk, Philip H.W. Leong, and David B. Thomas. A mixed precision Monte Carlo methodology for reconfigurable accelerator systems. In *Proceedings of the ACM/SIGDA international symposium on field programmable gate arrays*, pages 57–66, 2012.

[9] Matteo Croci and Michael B. Giles. Effects of round-to-nearest and stochastic rounding in the numerical solution of the heat equation in low precision. *IMA Journal of Numerical Analysis*, 43(3):1358–1390, 04 2022. ISSN 0272-4979. doi: 10.1093/imanum/drac012.

[10] Theodorus Jozef Dekker. A floating-point technique for extending the available precision. *Numerische Mathematik*, 18(3):224–242, 1971.

[11] Sylvain Delattre and Jean Jacod. A central limit theorem for normalized functions of the increments of a diffusion process, in the presence of round-off errors. *Bernoulli*, 3(1):1–28, 1997.

[12] Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann. MPFR: a multiple-precision binary floating-point library with correct rounding. *ACM transactions on mathematical software (TOMS)*, 33(2): 13–es, June 2007.

[13] Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations research*, 56(3):607–617, 2008.

[14] Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[15] Michael B. Giles and Oliver Sheridan-Methven. Analysis of nested multilevel Monte Carlo using approximate normal random variables. *SIAM/ASA Journal on Uncertainty Quantification*, 10(1):200–226, 2022.

[16] Michael B. Giles and Oliver Sheridan-Methven. Approximating inverse cumulative distribution functions to produce approximate random variables. *ACM transactions on mathematical software (TOMS)*, 49(3), September 2023.

[17] Michael B. Giles, Mario Hefter, Lukas Mayer, and Klaus Ritter. Random bit multilevel algorithms for stochastic differential equations. *Journal of complexity*, 2019.

[18] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53 of *Stochastic modelling and applied probability*. Springer science & business media, 1st edition, 2013.

[19] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM computing surveys (CSUR)*, 23(1):5–48, 1991.
[20] Torbjörn Granlund and the GMP development team. GNU MP: the GNU multiple precision arithmetic library, 2012. URL http://gmplib.org/. Version 5.0.5.
[21] Nicholas J. Higham. The accuracy of floating point summation. *SIAM journal on scientific computing*, 14(4):783–799, 1993.
[22] Nicholas J. Higham. *Accuracy and stability of numerical algorithms*, volume 80. SIAM, 2nd edition, 2002.
[23] Nicholas J. Higham and Theo Mary. A new approach to probabilistic rounding error analysis. *SIAM journal on scientific computing*, 41(5):A2815–A2835, 2019.
[24] Thomas E. Hull and J. R. Swenson. Tests of probabilistic models for propagation of roundoff errors. *Communications of the ACM*, 9(2):108–113, February 1966. ISSN 0001-0782.
[25] IEEE. IEEE standard for binary floating-point arithmetic, 1985. Computer society standards committee. Working group of the microprocessor standards subcommittee.
[26] IEEE. IEEE standard for binary floating-point arithmetic, 2008. Computer society standards committee. Working group of the microprocessor standards subcommittee.
[27] Ilse C.F. Ipsen and Hua Zhou. Probabilistic error analysis for inner products, 2019. arXiv:1906.10465.
[28] Fredrik Johansson et al. mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18), December 2013. URL http://mpmath.org/.
[29] William Morton Kahan. Further remarks on reducing truncation errors. *Communications of the association for computing machinery (ACM)*, 8:40, 1965.
[30] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. A study of BFLOAT16 for deep learning training, 2019. URL https://arxiv.org/abs/1905.12322.
[31] Andreas Klein. A generalized Kahan-Babuška summation algorithm. *Computing*, 76(3):279–293, 2006.
[32] Peter E. Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23 of *Stochastic modelling and applied probability*. Springer, 1999. Corrected 3rd printing.
[33] Donald E. Knuth. *The art of computer programming: seminumerical algorithms*, volume 2. Addison-Wesley Professional, 2014.
[34] Peter Linz. Accurate floating-point summation. *Communications of the ACM*, 13(6):361–362, June 1970.
[35] Gurii Ivanovich Marchuk and Vladimir V. Shaidurov. *Difference methods and their extrapolations*, volume 19. Springer science & business media, 2012.
[36] George Marsaglia and T.A. Bray. A convenient method for generating normal variables. *SIAM Review*, 6(3):260–264, 1964.
[37] George Marsaglia and Wai Wan Tsang. The Ziggurat method for generating random variables. *Journal of statistical software*, 5(1):1–7, 2000.
[38] Ole Møller. Quasi double-precision in floating point addition. *BIT numerical mathematics*, 5(1):37–50, 1965.
[39] Mervin E. Muller. An inverse method for the generation of random normal devi-

ates on large-scale computers. *Mathematical tables and other aids to computation*, 12(63):167–174, 1958.

[40] Arnold Neumaier. Rounding error analysis of some methods for summing finite sums (rundungsfehleranalyse einiger verfahren zur summation endlicher summen). *ZAMM - Journal of Applied Mathematics and Mechanics (zeitschrift für angewandte mathematik und mechanik)*, 54(1):39–51, 1974.

[41] Takeshi Ogita, Siegfried M Rump, and Shin'ichi Oishi. Accurate sum and dot product. *SIAM journal on scientific computing*, 26(6):1955–1988, 2005.

[42] Steffen Omland. Mixed precision multilevel Monte Carlo algorithms for reconfigurable computing systems, June 2016. PhD thesis/dissertation, D 386, Technische Universität Kaiserslautern (TUK).

[43] Steffen Omland, Mario Hefter, Klaus Ritter, Christian Brugger, Christian de Schryver, Norbert Wehn, and Anton Kostiuk. Exploiting mixed-precision arithmetics in a multilevel Monte Carlo approach on FPGAs. In *FPGA based accelerators for financial applications*, pages 191–220. Springer, 2015.

[44] Lewis Fry Richardson. VIII. The deferred approach to the limit. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, 226(636-646):299–361, 1927.

[45] Mathieu Rosenbaum. Integrated volatility and round-off error. *Bernoulli*, 15(3): 687–720, 2009.

[46] Oliver Sheridan-Methven. Approximating inverse cumulative distribution functions, 2020. URL https://github.com/oliversheridanmethven/approximating_inverse_cumulative_distribution_functions. GitHub repository.

[47] Oliver Sheridan-Methven. Getting started with approximate random variables: a brief guide for practitioners, 2020. URL https://github.com/oliversheridanmethven/approximate_random_variables. GitHub repository.

[48] Oliver Sheridan-Methven. Low precision approximate random variables, 2020. URL https://github.com/oliversheridanmethven/low_precision_approximate_random_variables. GitHub repository.

[49] Oliver Sheridan-Methven. Nested multilevel Monte Carlo methods and a modified Euler-Maruyama scheme utilising approximate Gaussian random variables suitable for vectorised hardware and low-precisions, 2021. DPhil. thesis, Mathematical Institute, University of Oxford.

[50] Lloyd N. Trefethen and David Bau. *Numerical linear algebra*, volume 50. SIAM, 3rd edition, 1997.

[51] Warwick Tucker. *Validated numerics: a short introduction to rigorous computations*. Princeton University press, 2011.

[52] Emil Vitasek. The numerical stability in solution of differential equations. In *Conference on the numerical solution of differential equations*, pages 87–111. Springer, 1969.

[53] James H. Wilkinson. Error analysis of direct methods of matrix inversion. *Journal of the ACM (JACM)*, 8(3):281–330, 1961.

[54] James H. Wilkinson. Numerical linear algebra on digital computers. *IMA bulletin*, 10(9/10):354–356, 1974.

[55] James H. Wilkinson. Error analysis revisited. *IMA bulletin*, 22(11/12):192–200, 1986.