Abstract—Here we present stochastic differential equations (SDEs) on a memristor crossbar, where the source of gaussian noise is derived from the random conductance due to ion drift in the memristor during programming. We examine the effects of line resistance on the generation of normal random vectors, showing the skew and kurtosis are within acceptable bounds. We then show the implementation of a stochastic differential equation solver for the Black-Scholes SDE, and compare the distribution with the analytic solution. We determine that the random number generation works as intended, and calculate the energy cost of the simulation.

Index Terms—Memristor, Crossbar, Stocasticity, Stochastic Differential Equation, Vector-Matrix Multiplication, Black-Scholes.

I. INTRODUCTION

MEMRISTOR crossbar arrays are an emerging CMOS alternative for acceleration of problems requiring compute-intensive matrix multiplication. Vector-matrix multiplication (VMM) operations can be done in $O(1)$ time complexity using Ohm’s law to sum analog currents. Despite their promise as VMM accelerators, current devices suffer from nonidealities stemming from immature fabrication techniques [1], preventing perfectly accurate VMM operations. Because of this, several previous works have cleverly taken advantage of these nonidealities [2] [3] to create analog computing devices beyond VMM accelerators.

Stochastic differential equations (SDEs) are used in many areas of applied mathematics, modeling systems with inherent stochastic terms or systems in the presence of noise [4]. Their wide use is despite the fact that there are no analytic solutions for most models of interest, meaning that numerical methods are necessary. Monte Carlo methods are one method of tackling the problem, where many realizations of the process are simulated, and then quantities of interest are computed from the resulting distributions.

However, Monte Carlo simulation suffers from two bottleneck operations: solution of nonlinear equations for certain types of solvers and generation of gaussian random variables [5]. Therefore, memristor crossbars can be used to solve the latter problem, where gaussian random numbers can be generated directly in memory, reducing time and energy cost compared to conventional algorithms implemented on CMOS such as the Zigurat algorithm [6] [7] or the Box-Muller transform.

Previous works on hardware acceleration of Monte Carlo methods for SDEs have been focused on FPGAs [8] or fully analog computation [9] [10], but to date no work has been written evaluating the potential in this space for memristor crossbars. There have been tangentially related works on MCMC algorithms on in-memory computing platforms, such as [11] (on SRAM) and [2] (on memristor crossbars), which are not directly applicable but provide motivation for this work.

In this paper we examine generation of random numbers on an in memory computing device, allowing the acceleration of Monte Carlo simulation of stochastic differential equations. The inherent noisiness of crossbar operations is shown to be not a barrier to their use as accelerators, but an advantage which allows even more efficient calculation of Monte Carlo simulations.

II. BACKGROUND

In this section we describe the characteristics of memristor crossbars as well as basic mathematical results about stochastic differential equations and numerical integration techniques.

A. Stochasticity in Memristor Crossbars

Memristor crossbars are a type of resistive RAM that use programmable resistance grids to perform highly efficient vector-matrix multiplication. Because fabrication techniques are still in development for such devices, the resolution of analog memristive devices is quite low compared to conventional computers. The low “resolution” is due to limited programming tuning accuracy, with 99% of memristors within 4% of the target conductance in the most advanced crossbars [12].

In OxRAM devices, the stochasticity in the cycle-to-cycle variability can be explained by variations in the conductive filaments. When the device is set and reset, the size and location of the filament changes, causing a change in the conductance state. Because of this random property, the high conductance state (HCS) displays a gaussian distribution [13], with standard deviation equal to anywhere from 10% to 40% of the mean, which can be controlled via the write current [2].

Although read operations can cause drifts in neighboring memristors in the same rows and columns, we neglect this effect because it is tolerable for a single read operation [14] even at larger (>1V) voltages that use up the random sample drawn from the memristor conductance distribution.
The crossbar model used in this paper was written in PyTorch [15]. It supports variable input, output, and line resistance. The programming is modeled as linear with finite precision enforced by perturbing the write conductance by some percentage error, in this work a pessimistic 5%. It should be noted that this variation is distinct from the cycle to cycle variability described above, which occurs in a region where the IV curve is not linear and accurate programming is very difficult.

The read step is modeled using a 1-bit DAC, i.e. floating point vector inputs are converted into n-bit fixed precision binary inputs, and the n outputs are scaled and added together off the crossbar [16]. All simulations in this paper used a line resistance of 5 Ω, an input and output resistance of 1 KΩ [17], unless stated otherwise.

B. Stochastic Differential Equations

Stochastic differential equations are a kind of differential equation which involves a random term, used to model stochastic processes such as ion drift, financial markets, and neuronal activity. The Itô stochastic differential equation is

\[ dX = r(t, X)dt + \sigma(t, X)^T dW, \]  

where \( r \) and \( \sigma \) are vector valued functions and \( dW \) is the derivative of Brownian motion. The time evolution of the probability distribution \( p \) of \( X \) follows the 2nd-order Fokker-Planck partial differential equation

\[ \frac{\partial p}{\partial t} = - \frac{\partial}{\partial x} (rp) + \frac{1}{2} \frac{\partial^2}{\partial x^2} (\sigma^2 p) \]  

which can be used to find analytic solutions to certain stochastic differential equations.

C. SDE Integrators

Only a small class of SDEs can be solved analytically. Just as with their deterministic counterparts, numerical methods must be used for all but the simplest SDEs. The one step approximation to the differential equation (1) yields the Euler-Maruyama method:

\[ X_n - X_{n+1} = r(t_n, X_n)\Delta t + \sigma(t_n, X_n)\Delta W \]  

which is the simplest numerical stochastic integration method. \( \Delta W \) is a normal random vector with zero mean and standard deviation \( \Delta t \). The method has an order of convergence of \( O(\Delta t^{1/2}) \) where \( \Delta t \) is the time step size.

III. Solution

In this section we describe the implementation of a stochastic integrator over the crossbar, including generation and characterization of noise, as well as the solver itself. A memistor crossbar simulator is used, including input/output and line resistance, tuning variability, quantized DAC inputs, and stuck on and stuck off nonidealities. The crossbar is modeled as a semi-passive array, with 8 × 8 tiles of memristors.

A. Random Sampling

The integrator described above needs a source of gaussian random vectors. The crossbar can provide this through the uncertainty in the low conductance state of a memristor, which follows a gaussian distribution centered around the target conductance with variability proportional to the conductance. Although line resistance can cause voltage degradation in larger passive arrays, we do
not observe this to have an effect on the distributions of the outputs, shown in Figure 2. The pair of memristors at the end of the word-line produce the same output variance, regardless of the line resistance.

Fig. 2. A 32x32 simulated crossbar was used to generate 500 16-length random gaussian vectors using the first word line. (a) Shows the change in the distribution due to line resistance in the first pair of memristors on the top word line. (b) Shows the same for the last pair of memristor. The mean and standard deviation are not changed by the increase in line resistance. (c) Shows the skew and kurtosis of the data in the top two plots. The kurtosis has 3 subtracted since the kurtosis of the unit normal distribution is 3. These higher order moments show no correlation with the position on the word line, indicating that line resistance does not negatively affect the properties of the gaussian sources. For reference, the standard deviation of the sample distribution of the kurtosis for 500 normally distributed variables is approximately 0.2, although the distribution is not symmetric. Therefore, most of the samples fell within one standard deviation of the sample mean of their test statistic.

For generation of gaussian random vectors with arbitrary covariance, the unit normal random vector can be multiplied by the lower triangular matrix of the Cholesky decomposition of the covariance matrix. This matrix multiplication has limited precision, resulting in a gaussian distribution close to but not exactly equal to the desired gaussian spectra. This is useful for modeling correlated processes.

B. Function Evaluation

The crossbar is not capable of computing general nonlinear functions such as \( r(t, X) \) and \( g(t, X) \). Therefore, they must be evaluated on a normal computer, using a computer-in-the-loop architecture. Linear functions can be evaluated as usual on the crossbar, and the allocation of computational tasks to the crossbar should be decided based on the problem. Recent work on neural networks in the framework of stochastic differential equations are promising for memristor crossbar acceleration because they involve many linear weight-stationary operations [18].

IV. APPLICATIONS

A. Solving the Black-Scholes SDE

The Black-Scholes SDE (Eq 4) for geometric brownian motion is a classic stochastic differential equation [19]. The case we consider is for a market with a fixed interest rate \( r \) and volatility \( \sigma \), since this simple case has a closed form solution.

\[
\frac{dX}{X} = rX\, dt - \sigma X\, dW
\]

subject to

\[
X(t=0) = X_0
\] (4)

Because it has a closed form solution (Eqn. 5), we can use it to test the accuracy of our solver by comparing the distribution with the distribution of the final values of the trajectories simulated on the crossbar.

\[
X = X_0 \exp(\sigma W + (r - \sigma^2/2)t)
\] (5)

The parameters \( r \) and \( \sigma \) were mapped to the crossbar, and a 3rd pair of memristors set to their high resistance state \((100 \, k\Omega)\) was used as the random source. The variability of the HRS was set at 25%. Because the problem is 1-D, no covariance matrix is needed. Figure 3(a) shows the layout of the parameters on the crossbar. All unused memristors were set to their low resistance state \((10 \, k\Omega)\), with 10% standard deviation.

The results of Monte Carlo simulation using the Euler-Maruyama method are shown in Figure 3. For calculation of \( rX \) and \( \sigma X \), the input was encoded in 16 bits. The simulation involving only generation of random numbers on the crossbar is in good agreement with the analytic solution. However, the simulation involving computation of \( r \) and \( \sigma \) displays a slight skew. This is due to the crossbar model perturbing the conductances to model imperfect writing. When this parameter is turned down the skew disappears. Although the variability is small, over 100 time steps the effect on the distribution can be significant. Because these values are so important, more care is required in programming them so that they land closer to their target conductance.
B. Static Power Consumption

Gaussian random sampling is expensive on digital devices because several calculations are needed to transform from a uniform distribution for which pseudo-random number generators exist to normal random variables. Because of this, the implicit gaussian distribution on the crossbar allows for some savings in computation despite requiring a write operation each time a new sample is needed.

The static power consumption of the memristor array was calculated following the model for the eVM synaptic array in NeuroSim [21], although their software was not used. The memristors were modeled as taking \( 100 \) of \( 1 \) pJ write pulses of \( 200 \) ns to get from low to high resistance state, as well as a 1\( \mu \)s 0.2 V read operation between each write pulse. This is a pessimistic estimate because writing to the HRS does not require so many read operations at first because it is unlikely that the resistance will overshoot the HRS. Unselected resistors were set at the low resistance state.

For the 2-memristor random variable shown here, this resulted in an average writing energy of 0.8 \( \mu \)J, which was dominated by the energy consumed by the read operations. The full simulation of the Black-Scholes equation involved roughly \( 200,000 \) write operations, resulting in an energy of consumption \( 0.16 \)J. Because all unused memristors were set at the low resistance state, the power consumption is a worst-case scenario. The energy consumed by read operations was \( 3 \mu \)J. Scaling to higher dimensional problems would increase this figure linearly.

V. Conclusion

In this paper, simulation of a simple SDE was shown over memristor hardware, where the white noise was drawn from the intrinsic uncertainty in programming. The effects of crossbar nonidealities such as line resistance were examined in order to determine their effects on the shape of the gaussian distribution. Despite nonidealities associated with inaccurate correction for variance, the solver produces a good fit to the analytic solution. This paper demonstrates that programming variability can be taken advantage of for solutions of SDEs while still being useful for vector matrix multiplication operations.

There are several directions this work can be taken in the future. Integration of a more complete crossbar model including 1T1R configurations and stochastic memristor models would improve the characterization of distributions resulting from crossbar operations. Although not discussed in this paper, the methods used here could be applied to higher dimensional SDEs where no analytical solutions exist, where the VMM acceleration of the crossbar would be significant.

References

[1] A. Amirsoleimani, F. Alibart, V. Yon, J. Xu, M. R. Pahouhandeh, S. Ecosse, Y. Beillard, R. Genov, and D. Drouin, “In-memory vector-matrix multiplication in monolithic complementary metal–oxide–semiconductor–memristor integrated circuits: Design choices, challenges, and Perspectives,” Advanced Intelligent Systems, vol. 2, no. 11, Aug. 2020.

[2] T. Dalgaty, N. Castellani, C. Turck, K.-E. Harabi, D. Querlioz, and E. Vianello, “In situ learning using intrinsic memristor variability via Markov chain Monte Carlo sampling,” Nature Electronics, vol. 4, no. 2, pp. 151–161, Jan. 2021.

[3] R. Zhang, H. Jiang, Z. R. Wang, P. Lin, Y. Zhuo, D. Holcomb, D. H. Zhang, J. J. Yang, and Q. Xia, “Nanoscale diffusive memristor crossbars as physical unclonable functions,” Nanoscale, vol. 10, no. 6, pp. 2721–2726, Jan. 2018.

[4] B. Oksendal, Stochastic differential equations: An introduction with
applications. Berlin, Germany: Springer, 2013.

[5] H. Gilsing and T. Shardlow, “SDELab: A package for solving stochastic differential equations in MATLAB,” Journal of Computational and Applied Mathematics, vol. 205, no. 2, pp. 1002–1018, 2007.

[6] G. Marsaglia and W. W. Tsang, “The Ziggurat Method for Generating Random Variables”, J. Stat. Soft., vol. 5, no. 8, pp. 1–7, Oct. 2000.

[7] Jianing Su and Jun Han, “An improved Ziggurat-based hardware Gaussian random number generator,” 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2016, pp. 1606-1608, doi: 10.1109/ICSICT.2016.7998817.

[8] F. Martini, M. Piccardi, N. B. Liberati and E. Platen, "A hardware generator for multi-point distributed random variables," 2005 IEEE International Symposium on Circuits and Systems, 2005, pp. 1702-1705 Vol. 2, doi: 10.1109/ISCAS.2005.1464934.

[9] Y. Huang, N. Guo, M. Seok, Y. Tsividis, and S. Sethumadhavan, “Analog Computing in a modern context: A linear algebra accelerator case study,” IEEE Micro, vol. 37, no. 3, pp. 30–38, Jun. 2017.

[10] Y. Huang, N. Guo, S. Sethumadhavan, M. Seok and Y. Tsividis, "A Case Study in Analog Co-Processing for Solving Stochastic Differential Equations," 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631831.

[11] P. Shukla, A. Shylendra, T. Tulabandhula and A. R. Trivedi, “MC2RAM: Markov Chain Monte Carlo Sampling in SRAM for Fast Bayesian Inference,” 2020 IEEE International Symposium on Circuits and Systems (ISCAS), 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180701.

[12] H. Kim, M. R. Mahmoodi, H. Nili, and D. B. Strukov, "4K-memristor analog-grade passive crossbar circuit," Nature Communications, vol. 12, no. 1, Aug. 2021.

[13] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Analogue signal and image processing with large memristor crossbars,” Nature Electronics, vol. 1, no. 1, pp. 52–59, Dec. 2017.

[14] Y. Yilmaz and P. Mazumder, “A Drift-Tolerant Read/Write Scheme for Multilevel Memristor Memory,” in IEEE Transactions on Nanotechnology, vol. 16, no. 6, pp. 1016-1027, Nov. 2017, doi: 10.1109/TNANO.2017.2741504.

[15] A. Paszke, S. Gross, and F. Massa, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” NEURIPS2019, pp. 8024–8035, 2019.

[16] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, “Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in Crossbars,” 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Jun. 2016.

[17] K. Pham, S. Tran, T. Nguyen, and K.-S. Min, “Asymmetrical Training Scheme of Binary-Memristor-Crossbar-Based Neural Networks for Energy-Efficient Edge-Computing Nanoscale Systems,” Micromachines, vol. 10, no. 2, p. 141, Feb. 2019.

[18] B. Tzen and M. Raginsky, “Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit,” CoRR, vol. abs/1905.09883, May 2019.

[19] R. U. Seydel, Tools for Computational Finance, 5th ed. New York, NY: Springer, 2012.

[20] A. Nag et al., “Newton: Gravitating Towards the Physical Limits of Crossbar Acceleration,” in IEEE Micro, vol. 38, no. 5, pp. 41-49, Sep./Oct. 2018, doi: 10.1109/MM.2018.053631140.

[21] P. Chen, X. Peng and S. Yu, “NeuroSim: A Circuit-Level Macro Model for Benchmarking Neuro-Inspired Architectures in Online Learning,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 12, pp. 3067-3080, Dec. 2018, doi: 10.1109/TCAD.2018.2789723.