A Case Study in Analog Co-Processing for Solving Stochastic Differential Equations

Yipeng Huang, Ning Guo, Simha Sethumadhavan, Mingoo Seok, Yannis Tsividis
Department of Computer Science and Department of Electrical Engineering
Columbia University
yipeng@cs.columbia.edu

***Abstract*—Stochastic differential equations (SDEs) are an important class of mathematical models for areas such as physics and finance. Usually the model outputs are in the form of statistics of the dependent variables, generated from many solutions of the SDE using different samples of the random variables. Challenges in using existing conventional digital computer architectures for solving SDEs include: rapidly generating the random input variables for the SDE solutions, and having to use numerical integration to solve the differential equations. Recent work by our group has explored using hybrid analog-digital computing to solve differential equations. In the hybrid computing model, we solve differential equations by encoding variables as continuous values, which evolve in continuous time. In this paper we review the prior work, and study using the architecture, in conjunction with analog noise, to solve a canonical SDE, the Black-Scholes SDE.**

***Index Terms*—Analog computers; Differential equations; Stochastic processes**

I. Introduction

In this paper we review our research group's findings from building prototype analog accelerator chips, and using them to solve a variety of scientific computation problems. In particular, we focus this paper on a case study in solving a stochastic differential equation (SDE). Our prior work focused on deterministically solving equations, so we treated analog noise as a nuisance to be minimized where possible. In contrast, in this paper we embrace and harness analog noise for useful computation. In the course of the paper, we discuss challenges in using noise for computation, the bringup and validation procedure for the case study, calibrating the accelerator for accuracy, and we compare the approach against a digital baseline.

II. Review of the Columbia University prototype analog accelerator for differential equations

The key feature of analog accelerators such as the Columbia University prototype is they encode variables using analog current and voltage, and the variables evolve in continuous time. That contrasts with conventional digital architectures which encode data in digital binary numbers and operate step-by-step. The analog, continuous-time model of computation is uniquely suited for solving equations for physical models because the models are also analog and continuous.

The Columbia University prototype analog accelerator chip is a reconfigurable analog fabric, consisting of integrators, multipliers, and mixed-signal components for converting analog variables into digital chip inputs and outputs. The chip encodes data as analog current, so the chip can sum variables by simply joining circuit branches; fanout current mirrors in the chip copy variables to different branches. A reconfigurable crossbar of switches allows us to chain analog operations end-to-end as needed for a given equation. Circuit design details of the prototypes, implemented at the 65nm technology node, are in papers by Guo et al. [1], [2]. The prototypes are successors to an earlier design built by Cowan et al. [3], [4]; the newer prototypes have features that permit calibration for more accurate results and easier interfacing with conventional digital architectures.

To use the chip, we would set up the analog accelerator chip so that the analog variables in the chip evolve according to an ordinary differential equation (ODE) we like to solve. The key component enabling solving ODEs using the analog chip is the integrator, which is physically a capacitor. An integrator takes an analog input representing the time derivative of a variable, and outputs its time integral. The chip routes those variables through analog operators to find a nonlinear function of the variables, which the chip feeds back as integrator inputs, such that the circuit behavior follows an ODE. We obtain the solution to the ODE by observing the analog variables directly, or by sampling them with analog-to-digital converters (ADCs) for further processing in a conventional digital computer. Thorough documentation of how to use the prototypes is in the latest version of our user's guide [5].

In prior work, we have used the chip to solve a variety of problem classes, starting with the basic idea of solving nonlinear ODEs useful in control systems [1], [2]. Then, in effort to make the approach more broadly applicable, in prior work we used the analog accelerator to solve systems of algebraic equations, which are the most important mathematical primitive in modern scientific computing. The analog approach for doing so is analogous to the iterative numerical methods for solving algebraic equations, which start with an initial guess for the solution vector for an algebraic equation, and iterate the guess toward a correct solution step-by-step. Notable iterative numerical methods include steepest descent for linear equations and Newton's method for nonlinear equations. The key trick for solving algebraic equations in the analog model of computation is to use continuous versions of numerical methods, phrased as ODEs we can solve in the

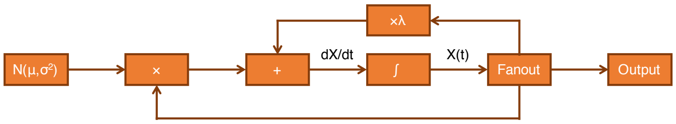

Fig. 1. Analog accelerator setup for solving the Black-Scholes equation. From left to right, the Gaussian white noise source, a multiplier providing gain to the noise, a summation point for all components comprising the Black-Scholes stochastic ODE, an integrator for finding the time integral, a fanout current mirror for copying the integrator output, and finally the output into an ADC. A variable gain amplifier provides gain in the feedback path.

analog chip. The findings are in papers by Huang et al., for problems in linear algebra [6], [7], and nonlinear systems of equations [8]. In the nonlinear equations example, we used analog approximate solutions to help a precise digital solver running on a GPU, achieving a performance improvement of 5.7× and energy savings of 11.6×, for certain problem parameters.

III. The Black-Scholes stochastic ODE case study

SDEs are a combination of a deterministic model in the form of a differential equation, and a stochastic noise component in the model. Typically scientists formulate the deterministic part analytically while the stochastic part is an attempt to capture effects not accounted for in the model.

The motivation for using an analog accelerator for this class of problem is twofold:

- **The analog accelerator excels at integrating differential equations, the task that dominates the solving time for SDEs.** Numerical methods for SDEs use Monte Carlo techniques, which entails running a statistically significant ensemble of independent solutions of the differential equation, each subject to a different stochastic noise input [9]–[11]. Having a more efficient tool for solving differential equations, such as the analog accelerator, would be advantageous.
- **In an analog solution to an SDE, the analog circuit can use analog noise from natural sources for the stochastic input to the equation.** On the other hand a numerical method running on a conventional digital computer needs to dedicate some time to create pseudorandom numbers, and then reshape the distribution of those numbers to fit the distribution needed by the SDE.

Using analog noise for solving SDEs was attempted in prior work by Cowan et al. [3], [4]; this paper extends that work using the Black-Scholes equations as an example for a more generalizable comparison.

A. Problem definition: the Black-Scholes stochastic ODE

The Black-Scholes equation refers to two different SDEs: the first and simpler equation is the Black-Scholes stochastic ODE for modeling stock prices, and the second more complex equation is the Black-Scholes-Merton partial differential equation (PDE) for finding prices for options. This paper focuses on solving the Black-Scholes stochastic ODE, the solutions for which are needed in the PDE model for pricing options.

The Black-Scholes stochastic ODE has the form:

$$\begin{aligned} dX(t) &= \lambda X(t)dt + \sigma X(t)dW(t) \\ X(0) &= X_0 \end{aligned} \tag{1}$$

where $X(t)$ is the price of a stock as a function of time, X_0 is the initial price, λ is a parameter called drift, σ is a parameter called volatility, and $W(t)$ is the standard Wiener process, which we will define in Section III-C.

Intuitively, the Black-Scholes stochastic ODE captures known effects and unknown effects on the price of a stock. In the long run, we can deterministically model the price of a stock as an exponential growth or decay process. In the short run, the fluctuations in the price of a stock are unpredictable, but are proportional to the current price of the stock.

The Black-Scholes stochastic PDE is useful for setting the price of a financial derivative or option, optimized so that it minimizes the risk due to the fact we cannot predict stock prices. When we set the price of certain types of options we must use Monte Carlo simulation to generate many price trajectories for the underlying stock. This paper envisions using the analog accelerator to assist in this type of calculation.

B. Analog Black-Scholes bringup: Gaussian white noise

The Black-Scholes stochastic ODE uses Gaussian white noise as its stochastic input. An analog accelerator solving the Black-Scholes stochastic ODE can obtain this noise from several sources (leftmost side of Figure 1): one option is to use Johnson-Nyquist thermal noise from resistors as a source of white noise [12], [13]. The analog noise may need some additional processing such as amplification and bandpass filtering to be useful in an analog SDE solver. Another option is to use pseudorandom digital codes to generate analog noise.

1) Analog noise: The analog noise needs to satisfy two properties. From a time-domain perspective, the noise must be **Gaussian normally distributed** around a mean, μ, with a given variance, σ^2. The mean is set by a digital-to-analog converter, while multipliers gain up the noise to set the noise variance. From a frequency-domain perspective, the noise must be **white noise**, which means it must have constant power spectral density. One way to verify this is to check that the autocorrelation plot for a noise source is a Dirac delta function. A Dirac delta autocorrelation plot indicates that knowing the value of a signal at any moment gives us no information about the value of the noise at any other point in time.

2) Filtered analog noise: In practice, we faced significant challenges getting perfect white noise inside the analog accelerator chip for solving SDEs. The nonideal behavior came in the form of DC drift, where the mean of the noise drifts over the course of minutes. While white noise does have low-frequency components that cause the mean to drift, this drift was coming from environmental effects such as temperature. Such a large, low frequency, and uncontrollable noise component overwhelms the white noise signal at low frequencies and cannot be used for solving SDEs.

A few techniques can eliminate DC mean drift:

TABLE I

RATIONALE WHY USING BAND-LIMITED NOISE IS SUFFICIENT.

	Low frequency limitations	**High frequency limitations**
Black-Scholes stochastic ODE model limitations	Market prevailing interest rate measurable and modeled deterministically	Trading day or market clock tick is minimum meaningful time interval
Digital numerical method limitations	Long-run model parameters are deterministic	Numerical integration and models have non-zero time step size
Analog electronic circuit limitations	Environmental variables, such as temperature and RF interference, introduce DC drift	Analog components, such as integrators and ADCs, have finite bandwidth due to parasitic capacitance
White noise limitations	At lowest frequencies, flicker (1/f, pink) noise dominates	At highest frequencies, shot noise (Poisson process) dominates

- **Subtract two independent noise sources both subject to the same DC drift.** In our case however some of the DC drift was itself independent for different channels, so subtracting noise sources did not cancel the DC drift.
- **Calibrate frequently to eliminate the DC drift.** While this works, calibration time costs make this impractical.
- **Use a high-pass filter (e.g., corner frequency of 100Hz) to eliminate DC components.** The resulting filtered noise is still Gaussian normally distributed as required. From a frequency domain perspective, its autocorrelation function is a Dirac delta around the origin, implying white noise at high frequencies, while cutting out any DC drift in the noise mean.

While SDEs such as the Black-Scholes stochastic ODE specify the noise source should be white noise, in practice both the mathematical model and the numerical methods for solving SDEs do not need to use perfect white noise. Likewise, our analog circuit approach and the noise sources the analog accelerator has access to do not support perfect white noise either. We summarize in Table I the rationale for why band-limited noise is sufficient for solving SDEs.

3) Digital noise: For the purposes of the experiments in this paper, we used a combination of analog noise sources and digital noise generated by feeding a DAC with a Gaussian normally distributed pseudorandom number sequence. Using digitally generated noise avoids downsides of using purely analog noise such as low amplitude and DC drift due to environmental variables. Though, in principle, a purpose-built analog noise source can replace the digital source.

C. Analog Black-Scholes bringup: standard Wiener process

The standard Wiener processes, also known as Brownian motion, is the time integral of white noise [10], [13], [14]:

$$W(t) = \int_0^t \frac{dW(\tau)}{d\tau} d\tau$$

The analog accelerator integrates the white noise source, described in the previous section, to create signal trajectories belonging to the standard Wiener process (central left-to-right pipeline in Figure 1). It is important to establish correct analog accelerator solutions for the standard Wiener process, because the solutions are a component of the Black-Scholes stochastic ODE: the drift should be linearly proportional to time t, the standard deviation σ should be proportional to $\sqrt{t}$, and the variance σ^2 should be proportional to t.

D. Analog Black-Scholes bringup: exponential growth process

After taking care of the stochastic parts of the Black-Scholes stochastic ODE, this subsection focuses on solving in the analog accelerator the deterministic part of the equation, an exponential growth process (upper feedback path in Figure 1). Solving this deterministic part of the SDE has its own set of challenges, specifically in the calibration of the analog components. Three types of errors contribute to inaccurate solutions for the exponential trajectories.

- **Variation in the multiplier gain:** we would like to calibrate the multiplier to realize a minuscule gain factor λ, delaying the time when the integrators become saturated, in order to elongate the useful time duration for solving the Black-Scholes stochastic ODE. But in practice accurate calibration of the multipliers for tiny gain factors is difficult. A gain factor of $\lambda = \frac{1}{80}$ balances the need for a small factor and precise calibration for that factor.
- **Variation in the integrator initial condition:** we would also like to calibrate the integrators to start off with a minuscule initial condition X_0, also in order to elongate the useful time duration for solving the Black-Scholes stochastic ODE. We should avoid too small of an initial condition because of noise, which is another source of error we discuss next.
- **Noise at the integrator input and multiplier output:** once the above sources of error are minimized, stochastic noise becomes the biggest contributor of error in the deterministic exponential growth curve. The integrator input noise is most problematic at the beginning of integration, when the noise amplitude is comparable to the actual signal. Worse, we cannot rearrange the definition of the Black-Scholes stochastic ODE to make use of noise at the integrator inputs. The trick to reducing the integrator's sensitivity to this spurrious noise is to elongate the characteristic time constant of the integrators.

E. Convergence & time for analog and digital Black-Scholes

In this section we use the analog accelerator to solve the full Black-Scholes stochastic ODE, using the full accelerator configuration shown in Figure 1, and compare the accuracy and performance against a digital solver.

A conventional digital computer can use several techniques for solving the Black-Scholes ODE. Here we are more inter-

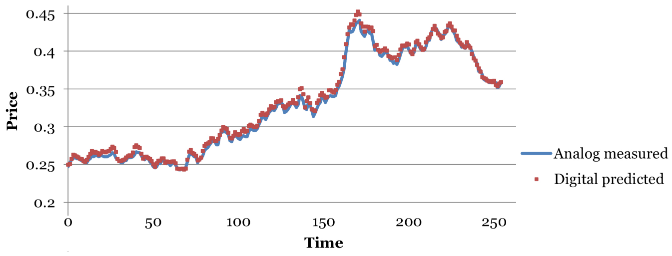

Fig. 2. Example analog and digital solutions for Black-Scholes SDE. Both solutions use the same random number sequence.

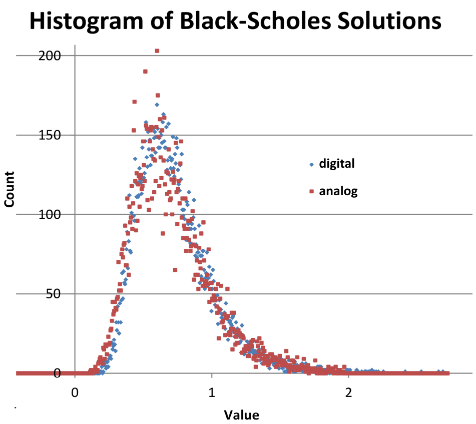

Fig. 3. The distribution of digital and analog final solutions match well. Here, the histogram is generated from an ensemble size of 16K. This plot uses the following parameters, which we have established in previous experiments to be the optimal settings to get the analog solver to work well: initial condition $X_0 = 0.25$, exponential positive feedback gain $\lambda = 0.0125$, standard deviation of noise $\sigma = 0.039563$.

ested in Monte-Carlo numerical methods that will give us the full trajectory of $X(t)$, since pricing models for many types of options need this trajectory. The basic Monte-Carlo method for SDEs include the stochastic Euler-Maruyama and higher order Milstein methods [10], [14]. Figure 2 shows one example solution for the Black-Scholes SDE using a random number sequence for both the digital and analog solvers.

In evaluating the accuracy of Monte-Carlo method solutions there is a difference between *strong convergence* and *weak convergence*. As we invest more computation steps and time in the Monte-Carlo method, the solutions improve in two ways: strong convergence is the improvement of each trajectory given by the Monte-Carlo method, measured as the decrease in the mean of the errors for each solutions; on the other hand weak convergence is the improvement of average of all the trajectories as an ensemble, measured as the decrease in the error of the mean for all the solutions.

Figure 2 for instance shows the effect of strong convergence of the digital solver in comparison to an analog solution. If the digital solver takes fewer time steps, with wider interval between the steps, the digital solver would track the analog solution less accurately. The mean of error for digital ODE solutions improves as the step sizes decrease, at the cost of taking more computation time.

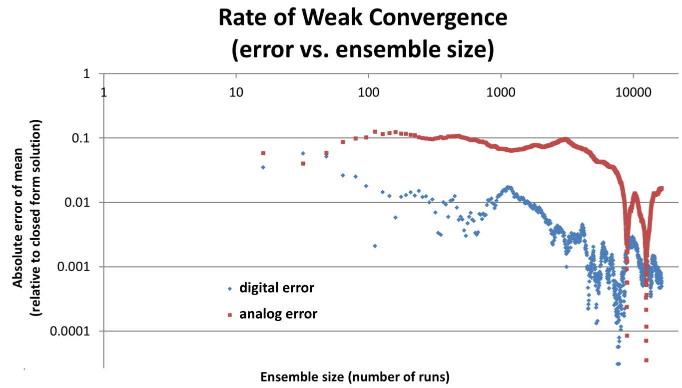

Fig. 4. The accuracy of analog and digital solutions, in terms of the weak convergence of the solution as ensemble size grows. As expected, the digital solver steadily converges to the solution. For the analog solver, the precision of the calibration seems to ultimately set a limit on how accurate the solution can be. Here, we are calibrating every 1000 runs, so for the 16K final ensemble size we had recalibrated 16 times.

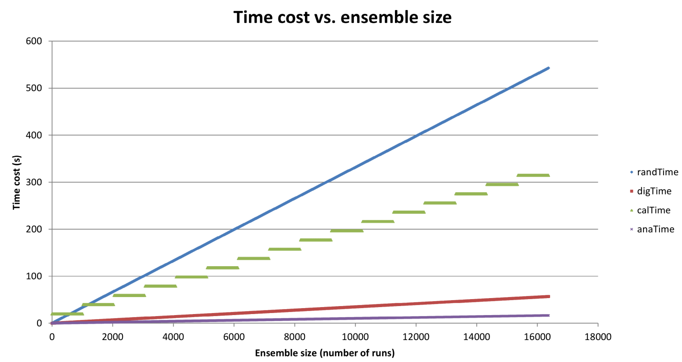

Fig. 5. The time cost of random number generation, the digital solution, the calibration routine, and the analog solution; all of these grow as the ensemble size grows. Notably, the calibration cost grows only when we recalibrate the analog solver, so it is a step function.

Figures 3 and 4 shows the effect of weak convergence of the digital and analog solvers. The mean of the distribution converges to the expected mean, as the ensemble size grows. If we keep increasing the solvers' ensemble size, the error of the mean solutions will keep decreasing.

Figure 5 compares the time cost of random number generation, the digital solution, the calibration routine, and the analog solution. Random number generation takes a lot of time because the basic method of converting from a uniform random number to a normally distributed random number is costly. A high quality analog noise source would provide Gaussian noise at zero time cost, making the analog solution much faster than the digital approach.

IV. Conclusion

In this paper we review prior work using an analog, continuous-time model of computation for solving differential equations. Using a prototype analog accelerator chip presented in prior work, we demonstrate a case study in solving the Black-Scholes stochastic differential equation.

Acknowledgment

The authors would like to thank Nicolas Clauvelin, Victor Marten, and John Milios at Sendyne Corp. for their collaboration and advice. This material is based upon work supported by the National Science Foundation under Grant No. CNS-1239134, the Defense Advanced Research Projects Agency (DARPA) under Contract No. D16PC00089, and an Alfred P. Sloan Foundation Fellowship.

References

[1] N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, "Continuous-time hybrid computation with programmable nonlinearities," in *European Solid-State Circuits Conference (ESSCIRC), ESSCIRC 2015 - 41st*, Sept 2015, pp. 279–282.

[2] ——, "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1514–1524, July 2016.

[3] G. Cowan, R. Melville, and Y. Tsividis, "A VLSI analog computer/math co-processor for a digital computer," in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, Feb 2005, pp. 82–586 Vol. 1.

[4] ——, "A VLSI analog computer/digital computer accelerator," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 1, pp. 42–53, Jan 2006.

[5] N. Guo, Y. Huang, and K. Lei, "Columbia hybrid computer user's guide," Columbia University, Tech. Rep., 2017.

[6] Y. Huang, N. Guo, M. Seok, Y. Tsividis, and S. Sethumadhavan, "Evaluation of an analog accelerator for linear algebra," in *Proceedings of the 43rd International Symposium on Computer Architecture*, ser. ISCA '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 570–582. [Online]. Available: https://doi.org/10.1109/ISCA.2016.56

[7] ——, "Analog computing in a modern context: A linear algebra accelerator case study," *IEEE Micro*, vol. 37, no. 3, pp. 30–38, 2017.

[8] Y. Huang, N. Guo, M. Seok, Y. Tsividis, K. Mandli, and S. Sethumadhavan, "Hybrid analog-digital solution of nonlinear partial differential equations," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-50 '17. New York, NY, USA: ACM, 2017, pp. 665–678. [Online]. Available: http://doi.acm.org/10.1145/3123939.3124550

[9] D. J. Higham, "An algorithmic introduction to numerical simulation of stochastic differential equations," *SIAM Review*, vol. 43, no. 3, pp. 525–546, 2001.

[10] T. Sauer, *Numerical Solution of Stochastic Differential Equations in Finance*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 529–550.

[11] D. Edwards, "Numerical and analytic methods in option pricing," 2015.

[12] R. Sarpeshkar, T. Delbruck, and C. A. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits and Devices Magazine*, vol. 9, no. 6, pp. 23–29, 1993.

[13] D. MacDonald, *Noise and Fluctuations: An Introduction*, ser. Dover books on physics. Dover Publications, 2006. [Online]. Available: https://books.google.com/books?id=_tP6eyA1RQoC

[14] J. M. Sanz-Serna, *Markov Chain Monte Carlo and Numerical Differential Equations*. Springer International Publishing, 2014, pp. 39–88.