Flexible GMRES with Analog Accelerators

Anshul Gupta[†], Vassilis Kalantzis[†], Mark S. Squillante[†], Chai Wah Wu[†], Haim Avron[§], Shashanka Ubaru[†], Tayfun Gokmen[†], Malte Rasch[†], Tomasz Nowicki[†], Lior Horesh[†]
[vkal,ubaru,malte.rasch]@ibm.com,[anshul,mss,cwwu,tgokmen,tnowicki,lhoresh]@us.ibm.com,haimav@tauex.tau.ac.il
[†]IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA
[§]Department of Applied Mathematics, Tel Aviv University, Tel Aviv 6997801, Israel

1 Introduction

The solution of sparse linear systems is a computationally expensive kernel that lies at the heart of numerous applications in science, engineering, business, and optimization. As conventional CMOS-based digital microprocessors reach their scaling and power dissipation limits, it becomes imperative to explore alternative hardware architectures to assure the continued scaling of linear system solvers.

One such emerging hardware alternative concerns analog crossbar array architectures [4]. These architectures can achieve high degrees of parallelism with low energy consumption by mapping matrices onto arrays of non-volatile memristive elements [5] capable of storing information and executing basic arithmetic operations. Performing Matrix-Vector Multiplication (MVM) and outer-product updates is then possible in a time that is independent of the number of nonzero entries in the operand matrices. In fact, considerable computational speedups have been achieved with analog crossbar arrays [9] for machine learning applications that can tolerate the noise and lower precision of these devices, e.g., training of neural networks [1, 13].

When it comes to the solution of linear systems of algebraic equations, the use of analog crossbar arrays has been relatively unexplored. Nonetheless, recent years have witnessed efforts based on inner-outer iterations to solve dense linear systems iterations [11], as well as problems originating from Partial Differential Equations [10, 17]. Analog designs with enhanced precision have also been proposed [7], by taking advantage of an autonomous linear system solver [16], but the improvement in precision comes at the cost of increased hardware complexity and energy consumption.

In this work, we propose the combined use of analog computing technology and approximate inverses [8] paired with Richardson iterations [14] as preconditioners in the flexible GMRES (FGMRES) Krylov subspace solver [15]. Our simulation experiments on a realistic problem show nearly an order of magnitude speedup over GMRES preconditioned with ILU(0), both with respect to similar solvers on a conventional microprocessor for attaining the same level of accuracy. These speedups are expressed in terms of complexity per digit of accuracy obtained in the solution, and could widen even further if they included the memory inefficiencies stemming from substitutions with sparse triangular factors. The proposed approach also has exciting applications in the context of the upcoming exascale platforms [2]. Extreme-scale solvers are likely to be highly composable [3] and hierarchical [12] with multiple levels of nested algorithmic components. A solver like the one proposed here, running on analog accelerator-equipped nodes of a massively parallel platform, is an ideal high-speed and low-power candidate for a local subdomain or coarse-grid solver to tackle the local components of a much larger sparse linear system.

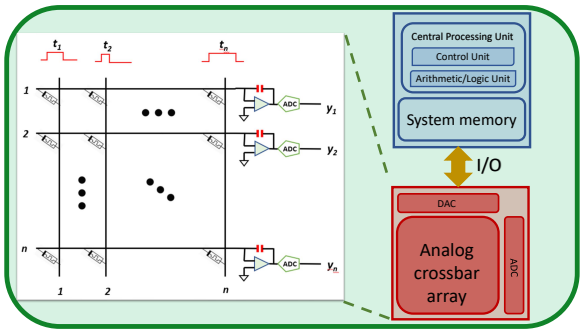

Figure 1. *Hybrid digital-analog architecture.*

2 Analog Accelarator Characteristics

A hybrid digital-analog architecture consisting of a CPU and a memristive crossbar array is illustrated in Figure 1. An $n \times n$ matrix M can be mapped onto an $n \times n$ crossbar array in $O(n)$ time such that the analog conductance of the memristor at the i, j-th cross point represents the digital value of the corresponding entry $m_{i,j}$ of M. Once written to the analog device, M can be multiplied by an $n \times 1$ input vector r to yield an approximation $\widehat{y}$ to the product $y = Mr$ as output. The time to perform this MVM is $O(n)$, including the time for fetching data from digital system memory, digital to analog conversion (DAC) of r and retrieving the results $\widehat{y}$ after analog-to-digital (ADC) conversion. Due to the stochastic and systematic noises associated with writing the matrix, imperfections in analog arithmetic, and the DAC and ADC conversions, the computed result $\widehat{y}$ is represented by the following equation [10]:

$$\widehat{y} = (M + E)r + \widehat{N}^a.$$

In the above, $\widehat{N}^a$ is the overall effective additive noise from various sources and E is a nondeterministic error matrix that captures the various multiplicative noises.

3 Analog-accelerated Flexible GMRES

FGMRES is a flexible variant of the GMRES algorithm that allows the preconditioner to vary between iterations [15]. While FGMRES requires the storage of an additional set of m vectors, where m is the number of inner iterations in each cycle, the flexibility of this method provides a perfect setting for analog hardware since the time-varying inaccuracies of these devices are handled directly in the algorithm.

Recalling the fact that hybrid analog-digital architectures can execute MVM at high speed, a natural choice for preconditioning FGMRES is through approximate inverses constructed in the digital space [8] and mapped to analog hardware for application during the iterative phase. The highly parallelizable and generally fast construction of an approximate inverse preconditioner can be amortized over multiple iterations/right-hand sides. Moreover, this avoids the need to perform triangular substitutions, e.g., as in ILU(0), an operation which can not be parallelized easily. On the other hand, the inaccuracies and inconsistencies of analog computations can lead to inferior convergence compared to its digital counterpart. A remedy then is to use the analog approximate inverse to precondition Richardson iteration, a light-weight stationary iterative method, and use the latter as a preconditioner in FGMRES. We note that Richardson iteration preconditioned by analog approximate inverses has been studied in [10] as part of our previous work.

4 Results

Figure 2 plots the relative residual norm as a function of the number of inner iterations (top) and the number of Floating Point Operations (FLOPs) (bottom), for the matrix `bcsstk21` [6]. In summary, combining analog approximate inverses with Richardson iteration proved to be the best choice. It is important to note that "h-ai-Ric(0)" runs at essentially the same digital cost as unpreconditioned GMRES, but its convergence rate is considerably higher. This shows that even without Richardson smoothing, simply using analog approximate inverses as preconditioners in FGMRES can lead to major improvements.

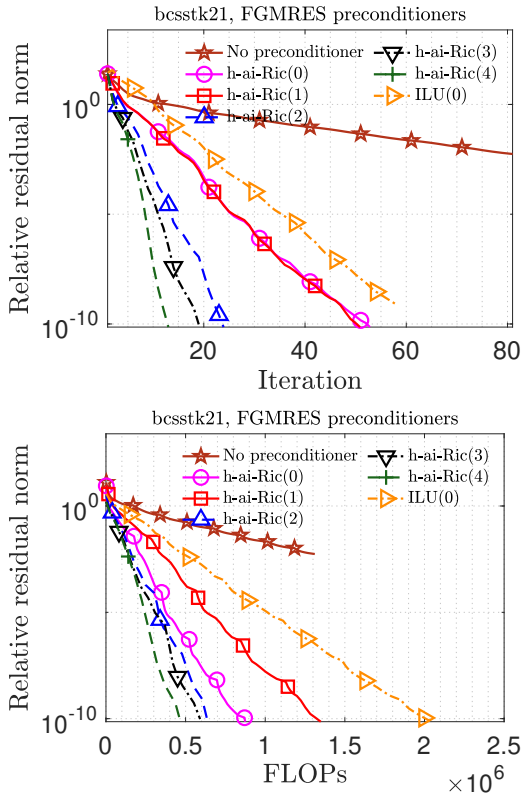

Figure 2. *Relative residual norm achieved by FGMRES preconditioned by up to 5 steps of Richardson iterations paired with an approximate inverse preconditioner. Comparisons with standard GMRES and GMRES preconditioned by ILU(0) are also provided. Top: Results as function of number of iterations; Bottom: Results as function of number of FLOPs.*

References

[1] Stefano Ambrogio et al. 2018. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558, 7708 (2018), 60–67.

[2] Hartwig Antz et al. 2020. Preparing sparse solvers for exascale computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 378, 20190053 (2020).

[3] Jed Brown et al. 2012. Composable Linear Solvers for Multiphysics. In *11th Int. Symp, on Parallel and Distributed Computing*. 55–62.

[4] Geoffrey W Burr et al. 2017. Neuromorphic computing using non-volatile memory. *Advances in Physics: X* 2, 1 (2017), 89–124.

[5] L. Chua. 1971. Memristor-The missing circuit element. *IEEE T-CT* 18, 5 (1971), 507–519.

[6] Timothy A Davis and Yifan Hu. 2011. The University of Florida sparse matrix collection. *ACM TOMS* 38, 1 (2011), 1–25.

[7] Ben Feinberg et al. 2021. An Analog Preconditioner for Solving Linear Systems. In *Proc. 2021 IEEE HPCA*. IEEE, 761–774.

[8] Marcus J Grote and Thomas Huckle. 1997. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing* 18, 3 (1997), 838–853.

[9] W. Haensch et al. 2019. The next generation of deep learning hardware: Analog computing. *Proc. IEEE* 107 (2019), 108–122.

[10] Vasileios Kalantzis et al. 2021. Solving sparse linear systems with approximate inverse preconditioners on analog devices. *arXiv preprint arXiv:2107.06973* (2021).

[11] Manuel Le Gallo et al. 2018. Mixed-precision in-memory computing. *Nature Electronics* 1, 4 (2018), 246–253.

[12] Lois Curfman McInnes et al. 2014. Hierarchical and Nested Krylov Methods for Extreme-Scale Computing. *Parallel Comput.* 40, 1 (2014), 17–31.

[13] Malte J Rasch et al. 2019. RAPA-ConvNets: modified convolutional networks for accelerated training on architectures with analog arrays. *Frontiers in neuroscience* 13 (2019), 753.

[14] Lewis Fry Richardson. 1911. IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 210, 459-470 (1911), 307–357.

[15] Youcef Saad. 1993. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. on Scientific Computing* 14, 2 (1993), 461–469.

[16] Zhong Sun et al. 2019. Solving matrix equations in one step with cross-point resistive arrays. *PNAS* 116, 10 (2019), 4123–4128.

[17] Mohammed A Zidan et al. 2018. A general memristor-based partial differential equation solver. *Nature Electronics* 1, 7 (2018), 411–420.